

**Aula 00 (Prof.  
Evandro)**

*TRT 10ª Região (Analista Judiciário -  
Apoio Especializado - Tecnologia da  
Informação) Arquitetura de  
Computadores - 2024 (Pós-Edital)*

Autor:

**Evandro Dalla Vecchia Pereira**

26 de Dezembro de 2024

# Índice

1) Apresentação do Curso - Evandro .....	5
2) Apresentação Flashcards .....	8
3) Linux - Conceitos Básicos - Teoria .....	10
4) Linux - Shell Script - Teoria .....	39
5) Linux - Gerenciamento de Processos - Teoria .....	52
6) Linux - Montagem de Volumes, Permissões, Superusuário - Teoria .....	59
7) Linux - Comandos Diversos - Teoria .....	67
8) Linux - Serviços de Rede e Segurança - Teoria .....	79
9) Linux - FreeBSD - Conceitos, Instalação, Configurações e Administração - Teoria .....	105
10) Linux - Conceitos Básicos - Questões Comentadas - Cebraspe .....	113
11) Linux - Conceitos Básicos - Questões Comentadas - FGV .....	116
12) Linux - Conceitos Básicos - Questões Comentadas - FCC .....	120
13) Linux - Conceitos Básicos - Questões Comentadas - VUNESP .....	122
14) Linux - Conceitos Básicos - Questões Comentadas - Cesgranrio .....	123
15) Linux - Conceitos Básicos - Questões Comentadas - Multibancas .....	125
16) Linux - Shell - Questões Comentadas - Cebraspe .....	145
17) Linux - Shell - Questões Comentadas - FGV .....	147
18) Linux - Shell - Questões Comentadas - FCC .....	150
19) Linux - Shell - Questões Comentadas - Vunesp .....	155
20) Linux - Shell - Questões Comentadas - Multibancas .....	156
21) Linux - Gerenciamento de Processos - Questões Comentadas - Cebraspe .....	166
22) Linux - Gerenciamento de Processos - Questões Comentadas - FGV .....	170
23) Linux - Gerenciamento de Processos - Questões Comentadas - VUNESP .....	176
24) Linux - Gerenciamento de Processos - Questões Comentadas - Multibancas .....	180
25) Linux - Montagem de Volumes - Questões Comentadas - Cebraspe .....	183
26) Linux - Montagem de Volumes - Questões Comentadas - FGV .....	186
27) Linux - Montagem de Volumes - Questões Comentadas - FCC .....	190
28) Linux - Montagem de Volumes - Questões Comentadas - VUNESP .....	194



# Índice

29) Linux - Montagem de Volumes - Questões Comentadas - CESGRANRIO .....	198
30) Linux - Montagem de Volumes - Questões Comentadas - Multibancas .....	199
31) Linux - Comandos Diversos - Questões Comentadas - Cebraspe .....	208
32) Linux - Comandos Diversos - Questões Comentadas - FGV .....	211
33) Linux - Comandos Diversos - Questões Comentadas - FCC .....	218
34) Linux - Comandos Diversos - Questões Comentadas - VUNESP .....	224
35) Linux - Comandos Diversos - Questões Comentadas - Multibancas .....	234
36) Linux - Serviços de Rede e Segurança - Questões Comentadas - Cebraspe .....	252
37) Linux - Serviços de Rede e Segurança - Questões Comentadas - FGV .....	257
38) Linux - Serviços de Rede e Segurança - Questões Comentadas - FCC .....	262
39) Linux - Serviços de Rede e Segurança - Questões Comentadas - VUNESP .....	268
40) Linux - Serviços de Rede e Segurança - Questões Comentadas - Multibancas .....	271
41) Linux - FreeBSD - Questões Comentadas - FGV .....	292
42) Linux - FreeBSD - Questões Comentadas - Multibancas .....	294
43) Linux - Conceitos Básicos - Lista de Questões - Cebraspe .....	296
44) Linux - Conceitos Básicos - Lista de Questões - FGV .....	298
45) Linux - Conceitos Básicos - Lista de Questões - FCC .....	301
46) Linux - Conceitos Básicos - Lista de Questões - VUNESP .....	303
47) Linux - Conceitos Básicos - Lista de Questões - Multibancas .....	304
48) Linux - Shell - Lista de Questões - Cebraspe .....	318
49) Linux - Shell - Lista de Questões - FGV .....	320
50) Linux - Shell - Lista de Questões - FCC .....	322
51) Linux - Shell - Lista de Questões - Vunesp .....	326
52) Linux - Shell - Lista de Questões - Multibancas .....	327
53) Linux - Gerenciamento de Processos - Lista de Questões - Cebraspe .....	333
54) Linux - Gerenciamento de Processos - Lista de Questões - FGV .....	335
55) Linux - Gerenciamento de Processos - Lista de Questões - VUNESP .....	339
56) Linux - Gerenciamento de Processos - Lista de Questões - Multibancas .....	341



# Índice

57) Linux - Montagem de Volumes - Lista de Questões - Cebraspe .....	343
58) Linux - Montagem de Volumes - Lista de Questões - FGV .....	345
59) Linux - Montagem de Volumes - Lista de Questões - FCC .....	348
60) Linux - Montagem de Volumes - Lista de Questões - VUNESP .....	351
61) Linux - Montagem de Volumes - Lista de Questões - Cesgranrio .....	354
62) Linux - Montagem de Volumes - Lista de Questões - Multibancas .....	355
63) Linux - Comandos Diversos - Lista de Questões - Cebraspe .....	362
64) Linux - Comandos Diversos - Lista de Questões - FGV .....	364
65) Linux - Comandos Diversos - Lista de Questões - FCC .....	369
66) Linux - Comandos Diversos - Lista de Questões - VUNESP .....	373
67) Linux - Comandos Diversos - Lista de Questões - Multibancas .....	380
68) Linux - Serviços de Rede e Segurança - Lista de Questões - Cebraspe .....	393
69) Linux - Serviços de Rede e Segurança - Lista de Questões - FGV .....	396
70) Linux - Serviços de Rede e Segurança - Lista de Questões - FCC .....	399
71) Linux - Serviços de Rede e Segurança - Lista de Questões - VUNESP .....	403
72) Linux - Serviços de Rede e Segurança - Lista de Questões - Multibancas .....	405
73) Linux - FreeBSD - Lista de Questões - FGV .....	418
74) Linux - FreeBSD - Lista de Questões - Multibancas .....	420





## APRESENTAÇÃO DO CURSO

Iniciamos nosso **Curso Regular de Sistemas Operacionais e Arquitetura de Computadores** em teoria e questões, voltado para provas **objetivas e discursivas** de concurso público. Tais assuntos são cobrados em diversos concursos em que há vagas específicas para a área de TI.

As aulas em PDF possuem por característica essencial a **didática**. Ao contrário do que encontramos em alguns livros, o curso todo se desenvolverá com uma leitura de fácil compreensão e assimilação.

Além disso, teremos videoaulas! Essas aulas destinam-se a complementar a preparação. Quando estiver cansado do estudo ativo (leitura e resolução de questões) ou até mesmo para a revisão, abordaremos alguns pontos da matéria por intermédio dos vídeos. Com outra didática, você disporá de um conteúdo complementar para a sua preparação. Ao contrário do PDF, evidentemente, **AS VIDEOAULAS NÃO ATENDEM A TODOS OS PONTOS QUE VAMOS ANALISAR NOS PDFS, NOSSOS MANUAIS ELETRÔNICOS**. Por vezes, haverá aulas com vários vídeos; outras que terão videoaulas apenas em parte do conteúdo. Nosso **foco** é sempre o **estudo ativo!**



## APRESENTAÇÃO PESSOAL

Meu nome é Evandro Dalla Vecchia Pereira, sou autor do livro "Perícia Digital - Da investigação à análise forense", Mestre em Ciência da Computação (UFRGS), Bacharel em Ciência da Computação (PUCRS), Técnico em Redes de Computadores (Ecom/UFRGS) e em Processamento de Dados (Urcamp). Perito Criminal na área de Perícia Digital desde 2004 no Instituto-Geral de Perícias/RS. Professor de pós-graduação em diversas instituições, nas áreas de Perícia Digital, Perícia Criminal e Auditoria de Sistemas. Lecionei em cursos de graduação de 2006 a 2017, nas instituições PUCRS, Unisinos, entre outras e sou professor em cursos de formação e aperfeiçoamento de Peritos Criminais, Delegados, Inspetores, Escrivães e Policiais Militares.

No Estratégia Concursos leciono desde o começo de 2018, inicialmente na área de Computação Forense e, na sequência, também assumi as áreas de Arquitetura de Computadores e Sistemas Operacionais, tanto na elaboração de materiais escritos como na gravação das videoaulas.

Deixarei abaixo meus contatos para quaisquer dúvidas ou sugestões. Terei o prazer em orientá-los da melhor forma possível nessa caminhada que estamos iniciando.

**Instagram:** @profevandrodallavecchia

**Facebook:** <https://www.facebook.com/profevandrodallavecchia>



## PARE TUDO! E PRESTE ATENÇÃO!!

Hoje eu faço parte de uma equipe **SENSACIONAL** de professores! Depois de muita luta conseguimos reunir **um time** de profissionais extremamente QUALIFICADO e sobretudo **COMPROMISSADO** em fazer o melhor pelos alunos. Para tal criamos um conjunto de ações para nos aproximarmos dos alunos, entendermos suas necessidades e evoluirmos nosso material para um patamar ainda mais diferenciado. São 3 as novidades que gostaria de convidá-lo a conhecer:

//estratégia tech



Nosso podcast alternativo ... livre, descontraído e com dicas rápidas que todo CANETA PRETA raiz deve ouvir. Já temos alguns episódios disponíveis e vários outros serão gravados nas próximas semanas ... acompanhe em:

<http://anchor.fm/estrategia-tech>



Telegram

a new era of messaging

Nosso grupo do Telegram é um local onde ouvimos os alunos e trocamos ideias com eles. Está crescendo a cada dia. A regra do grupo é: só vale falar sobre concursos. Lá divulgamos nossas aulas ao vivo e falamos sobre os concursos abertos, expectativas de novos concursos, revisões de véspera, e por aí vai...

[http://t.me/estrategia\\_ti](http://t.me/estrategia_ti)

Instagram




Criamos um perfil no Instagram ... e qual o objetivo? Fazer com que os alunos percam tempo nas redes sociais? Claro que não!! Estamos consolidando diversos posts dos professores! São dicas especiais, um patrimônio que deve ser explorado por todos os concurseiros de TI!

<http://instagram.com/estrategiaconcursosti>



# ESTRATÉGIA FLASHCARDS

 Você tem dificuldade de estudar, memorizar e revisar os conteúdos que estuda em nossas aulas? Então nós temos a ferramenta perfeita para você!

Apresentamos o **Estratégia Cards**: app de flashcards que vai revolucionar sua forma de **estudar** e **revisar** conteúdos de provas de concurso público. Com nossa tecnologia inovadora e interface amigável, você dominará os tópicos mais complexos de maneira eficiente e divertida.

## 🌟 Recursos do Estratégia Cards:

<b>Curadoria de Flashcards</b>	Flashcards criados e revisados por professores especializados em cada área, com qualidade e voltados para concursos públicos.
<b>Flashcards Personalizados</b>	Crie seus próprios flashcards, cobrindo os principais tópicos e matérias dos concursos públicos.
<b>Repetição Espaçada</b>	Técnica de aprendizagem que envolve revisar informações em intervalos crescentes para melhorar a retenção de longo prazo e combater o esquecimento.
<b>Estatísticas Personalizadas</b>	Visualize graficamente o percentual de acertos, erros ou dúvidas dos decks estudados.
<b>Modo Offline</b>	Estude em qualquer lugar, mesmo sem conexão à internet, fazendo o download dos decks.
<b>Estudo por Áudio</b>	<i>Está dirigindo ou fazendo esteira e quer continuar estudando?</i> Basta utilizar a opção “Escutar”.
<b>Decks Favoritos</b>	Você pode escolher decks específicos como favoritos e visualizá-los em uma aba separada do app.
<b>Opções de Estudo</b>	Você poderá estudar todos os cards de um deck; ou apenas os que você errou; ou apenas os que você não estudou ainda; entre outras opções.

## E como eu consigo baixar?



É muito fácil! Basta pesquisar por “Estratégia Cards” na loja oficial do seu smartphone.

Se você tiver um Android, basta acessar a **Google Play**;



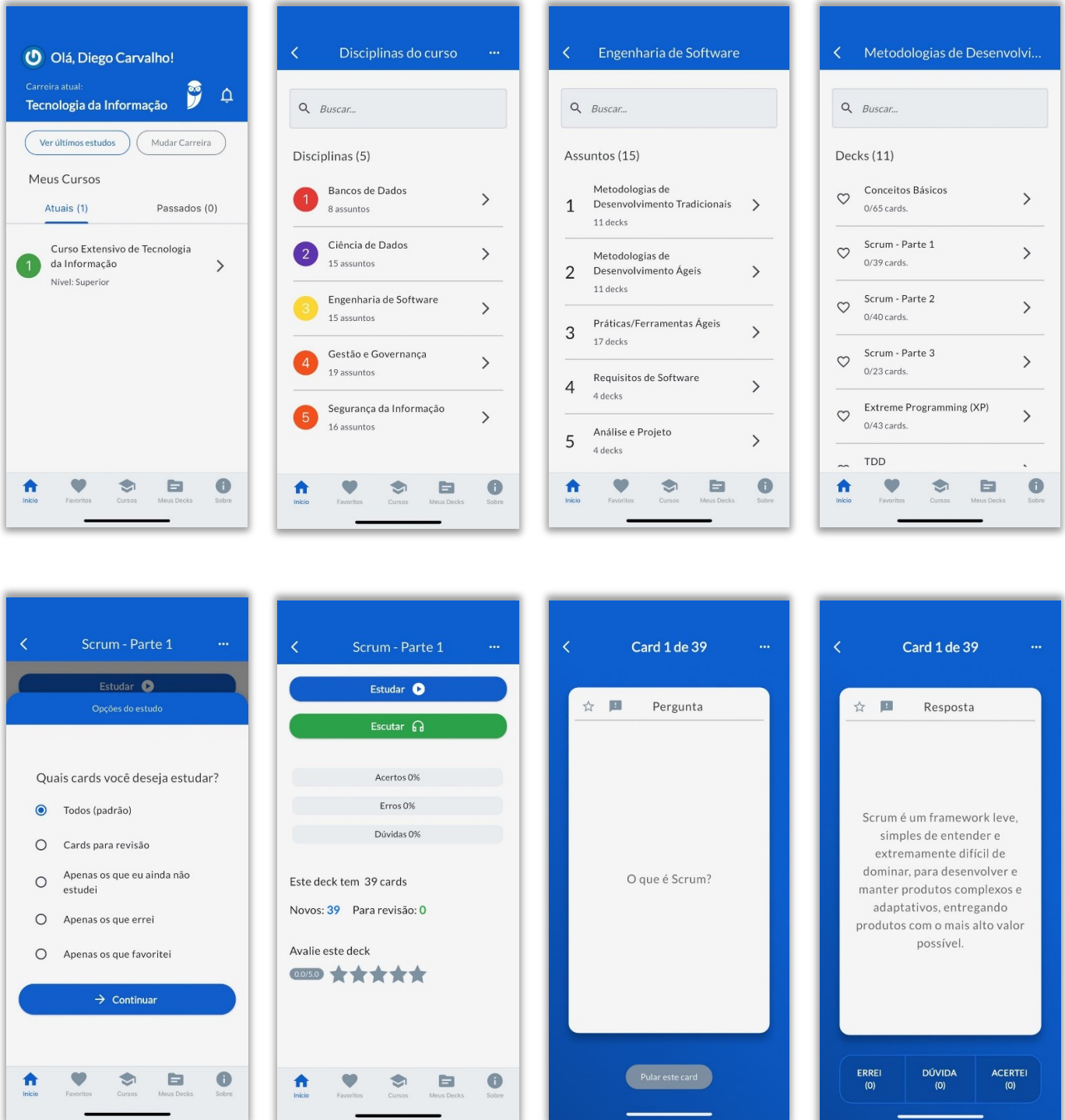
Se for tiver um iPhone, basta acessar a **App Store (iOS)**.



## É para acessar?

Para acessar, basta ter uma conta no Estratégia Concursos. Em seguida, utilize suas credenciais de login e senha para acessar o aplicativo. Por fim, acessa a carreira de Tecnologia da Informação.

## Como utilizar o app:



## LINUX - CONCEITOS BÁSICOS

Começamos dizendo que o **Linux é um sistema operacional interativo**, ou seja, diferentemente dos sistemas em lote ou os sistemas de tempo real, ele permite uma interação do usuário diretamente com o computador, mesmo durante a execução de um programa. Essa interação ocorre através de dispositivos de entrada (teclado, mouse, entre outros) e saída (monitor de vídeo, impressora, entre outros).

O usuário "administrador", que tem "poder total" sobre o sistema é o **root** e seu diretório "home" é o **"/root"**. Os usuários "comuns" têm como seu diretório "home" o **"/home/nome\_usr"**, ex.: **"/home/evandro"**.

A instalação do Linux pode ocorrer de algumas formas, sendo as mais comuns através de uma mídia (CD, DVD, pen drive) ou através da rede. No caso da mídia, obviamente ela precisa ser **"bootável"** e a BIOS deve estar configurada para ler primeiro o CD/DVD ou a USB (se for pen drive). Abaixo uma tela de instalação a partir de um CD.

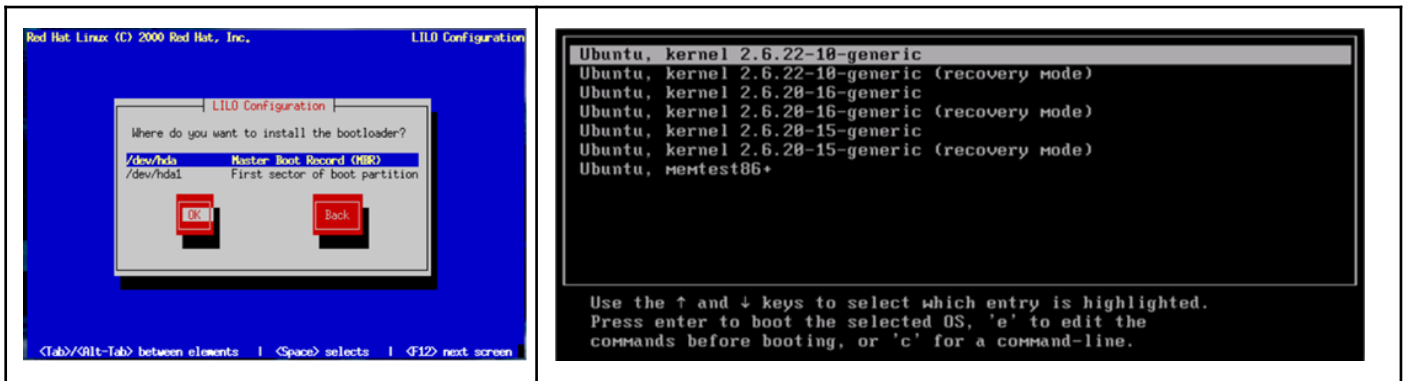


A instalação pela rede exige mais conhecimento e geralmente é feita por técnicos de TI nas empresas. É necessário configurar um servidor TFTP e um DHCP que irá fornecer a mídia de instalação para as máquinas da sua rede local. Se a BIOS da máquina cliente suportar, é possível então inicializar o sistema de instalação a partir da rede (usando PXE e TFTP) e prosseguir com a instalação a partir da rede.

Muitas pessoas utilizam o Linux e o Windows no mesmo computador, cada um em uma partição (com seus respectivos sistemas de arquivos). E como é possível escolher qual deles deve ser inicializado quando a máquina é ligada? Existem **gerenciadores de boot** que auxiliam nisso, sendo os mais conhecidos o **LILLO** (*L*inux *L*Oader) e o **GRUB** (*G*Rand *U*nifield *B*ootloader). Abaixo podemos ver os dois (à esquerda o LILLO e à direita o GRUB).







Ao escolher a partição, o *kernel* (veremos em breve o que é e para que serve) é carregado e inicializado. Na sequência há a execução dos scripts de inicialização do sistema.

Em relação à **portabilidade**, o Linux se sai muito bem! Ele não foi pensado para ser um sistema portátil, mas acabou indo nessa direção, tendo um dos núcleos de sistemas operacionais mais portáteis, sendo executado em sistemas desde o iPaq (um computador portátil) até o IBM S/390 (um mainframe). Na atualidade o Linux hoje funciona em dezenas de plataformas, desde mainframes até um relógio de pulso, passando por diversas arquiteturas: x86 (Intel, AMD), x86-64 (Intel EM64T, AMD64), ARM, PowerPC, Alpha, SPARC, entre outras, além de sistemas embarcados (*handhelds*, consoles de videogames, PVR – *Personal Video Recorder*, telefones celulares, TVs etc.).

## Sistema Multitarefa e Multiusuário

O Linux permite que vários processos sejam executados ao mesmo tempo (**multitarefa**), mesmo que haja apenas um processador na máquina. O sistema operacional faz o escalonamento dos processos no processador, dando a ilusão de um paralelismo (ocorre um pseudoparalelismo). O comando "ps", por exemplo, pode ser executado e você verá diversos processos em execução. Quando há a criação de um processo novo, ele passa a ser filho de outro e a chamada de sistema utilizada é a fork().

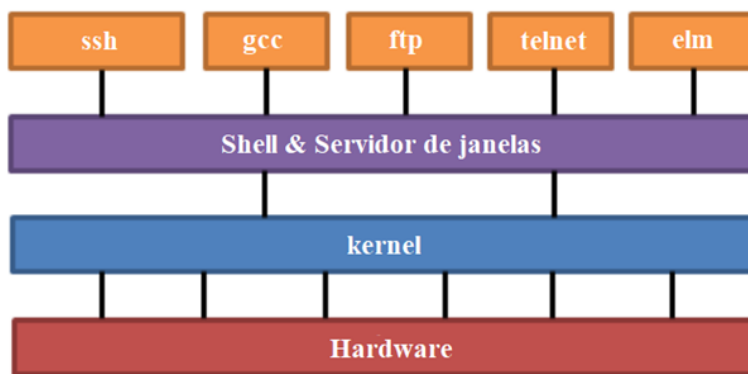
No Linux é possível que vários usuários utilizem ao mesmo tempo (**multiusuário**), então alguém pode deixar algumas aplicações em execução em sua sessão, bloquear o acesso e outra pessoa abrir uma outra sessão, executando novas aplicações. São permitidos vários usuários simultâneos, a não ser que seja definido um limite de sessões nos arquivos de configuração. O comando "who" pode ser executado para ver quem está logado no computador.

## Kernel, Shell e Servidor de Janelas

O **kernel** (núcleo) da maioria dos sistemas baseados em Unix (FreeBSD e derivados, Oracle Solaris, IBM AIX, HP-UX e todos os Linux e derivados) é **monolítico**. Mas o que é isso? Todo o conjunto de instruções de controle do hardware é executado no espaço de núcleo no modo de supervisão, ou seja, é um **único executável que possui todos os códigos de suporte necessários agregados**. Em teoria, um kernel monolítico pode travar o sistema inteiro se 1 módulo parar de funcionar adequadamente.







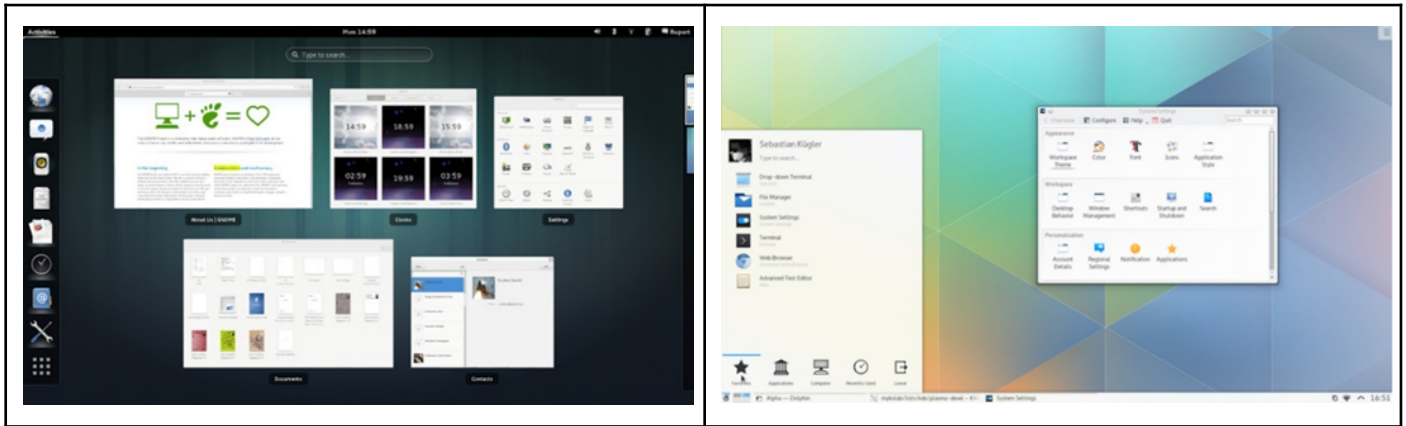
Na figura acima podemos ver cinco aplicativos que utilizam alguma interface (shell ou servidor de janelas), que por sua vez faz a "ligação" com o kernel. E somente o kernel faz a comunicação com o hardware (disco, teclado, monitor de vídeo etc.).

O shell é um interpretador de comandos (aquele terminal preto, geralmente) e sua função é ler a linha de comando, interpretar seu significado, executar o comando e devolver o resultado pelas saídas (monitor de vídeo, por exemplo). Na verdade, a interface shell é um arquivo executável, encarregado de interpretar comandos, enviá-los ao kernel e devolver resultados. Existem vários tipos de shell, sendo os mais comuns o sh (Bourne shell), o bash (Bourne again shell), o csh (C shell), o Tcsh (Tenex C shell), o ksh (Korn shell) e o zsh (Zero shell). Seus nomes normalmente correspondem ao nome do executável.

```
localhost:~# ls -las
total 40
 4 drwxr-xr-x  5 root   root   237 Jan  9  2021 .
 4 drwxrwxrwx 21 root   root   461 Jul 20 21:13 ..
 4 -rw-----  1 root   root    23 Jul 20 21:13 .ash_history
 4 drwx-----  3 root   root    61 Jul  5  2020 .cache
 4 drwx-----  5 root   root   124 Jul  5  2020 .mozilla
 4 drwxr-xr-x  4 root   root   202 Jul  6  2020 .wine
 4 -rw-r--r--  1 root   root   114 Jul  5  2020 bench.py
 4 -rw-r--r--  1 root   root    76 Jul  3  2020 hello.c
 4 -rw-r--r--  1 root   root    22 Jun 26  2020 hello.js
 4 -rw-r--r--  1 root   root   151 Jul  5  2020 readme.txt
localhost:~# █
```

Alguns dos servidores de janelas (ambientes gráficos) mais populares do mundo Linux são Gnome, KDE, Cinnamon, MATE, XFCE, Pantheon, Deepin, entre outros. Abaixo podemos ver uma tela do Gnome (à esquerda) e uma tela do KDE (à direita).





Uma característica interessante, por ser um sistema operacional de código aberto, é a possibilidade de **compilar o kernel**. Mas qual o motivo para isso? Para facilitar vamos a um exemplo: suponha que você tenha uma placa-mãe X e sua placa de rede integrada é uma da fabricante Y. No Kernel que você está usando (uma versão mais antiga) não há suporte para essa placa Y, ou seja, você não pode navegar na rede. Lendo a documentação do Kernel de uma versão mais recente, você constata que o suporte para Y foi adicionado. Então, se o kernel for atualizado será possível navegar na rede com tal placa com esse novo módulo.

Os **módulos são partes do kernel** que são carregadas somente quando solicitadas por algum aplicativo ou dispositivo e descarregadas da memória quando não são mais utilizadas. Dessa forma, evita-se a construção de um kernel grande (estático) que ocupe grande parte da memória com todos os drivers compilados e permite que partes do kernel ocupem a memória somente quando forem necessários.

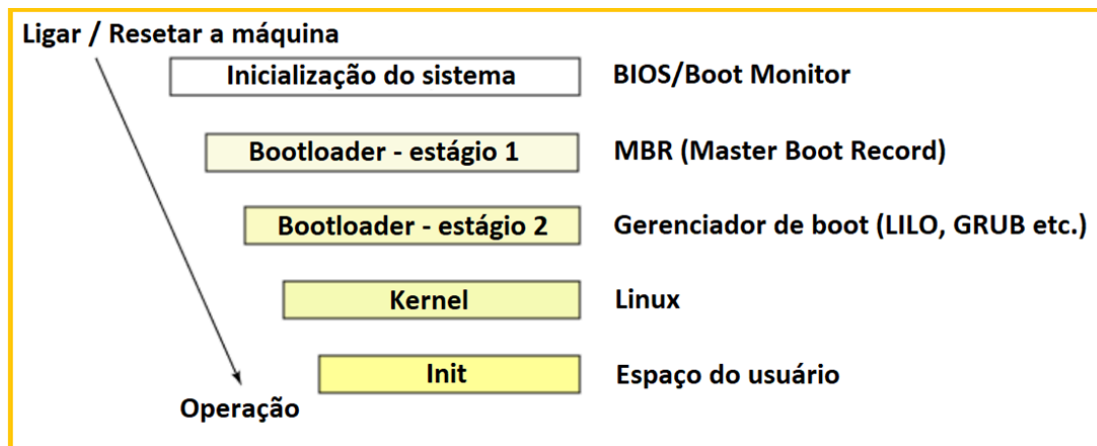
A localização padrão dos módulos do kernel é a seguinte: `"/lib/modules/versão_do_kernel/"` e um comando que permite obter informações acerca do servidor é o `"uname"`. Com o argumento `"-a"` é possível obter todas as informações, incluindo a versão do kernel. Abaixo um exemplo mostrando a versão do kernel (4.12):

```
[root@localhost lib]# uname -a  
Linux localhost 4.12.0-rc6-g48ec1f0-dirty #21 Fri Aug 4 21:02:28 CEST 2017 i586  
GNU/Linux  
[root@localhost lib]#
```

## Bootloader (Carregador de boot)

Ao inicializar o Linux, o gerenciador de inicialização é responsável por carregar a imagem do kernel e o disco RAM inicial ou sistema de arquivos (possui alguns arquivos críticos e drivers de dispositivo necessários para inicializar o sistema) na memória. Abaixo uma figura que mostra o passo a passo do boot.





Podemos observar que o boot possui dois estágios, descritos a seguir.

### Primeiro Estágio:

Em sistemas usando o método BIOS/MBR, o carregador de boot reside no primeiro setor do disco rígido, o Master Boot Record (MBR). Nesse estágio, o carregador de boot examina a tabela de partições e encontra uma partição inicializável. Depois, procura o carregador de inicialização do segundo estágio (ex.: GRUB) e o carrega na memória RAM.

Para sistemas que usam o método EFI/UEFI, o firmware UEFI lê seus dados do gerenciador de boot para determinar qual aplicativo UEFI deve ser lançado e de onde, ou seja, de qual disco e partição a partição EFI pode ser localizada. O firmware então inicia o aplicativo UEFI (ex.: GRUB), conforme definido na entrada de inicialização no gerenciador de inicialização do firmware.

### Segundo Estágio:

O carregador de inicialização do segundo estágio (ex.: GRUB) reside em /boot. Uma tela inicial é exibida, o que permite escolher qual sistema operacional será inicializado. Depois de escolhido o sistema operacional, o carregador de boot carrega o kernel do sistema operacional selecionado na memória RAM e passa o controle para ele.

### Funcionamento do boot no Linux

Os kernels geralmente são compactados, então seu primeiro trabalho é ser descompactado. Depois ele verificará e analisará o hardware do sistema e inicializará todos os drivers de dispositivo de hardware embutidos no kernel.

### Init (Systemd)

O systemd é um sistema de inicialização (init system) composto por um conjunto de programas que é executado em *background*<sup>1</sup> (*daemon*). Na atualidade a maioria das distribuições Linux utiliza o systemd para a execução do boot.

O systemd assume o controle assim que o kernel é ativado pelo gerenciador de bootloader (ex.: GRUB). A partir desse ponto são carregados todos os dispositivos e processos que são

<sup>1</sup> Veremos na sequência o conceito de background em detalhes.

inicializados com o sistema. Uma grande vantagem do systemd é a sua arquitetura e o modo de funcionamento. Nele são usadas unidades de socket (arquivos de configuração que codificam informações relacionadas à comunicação entre processos). Com isso, permite que todos os daemons requisitados no boot sejam carregados simultaneamente, bem como possibilita a transmissão coordenada entre dois sockets, o que resulta em uma rápida inicialização do sistema operacional.

## Arquitetura do systemd

Basicamente, a estrutura do systemd é composta por unidades (units). Abaixo, algumas classes de units do systemd:

- service: tais unidades são os serviços presentes no sistema operacional acessíveis ao usuário;
- timer: temporizadores usados para determinar ações para um serviço usando como base o tempo (não é o cron!);
- mount: arquivo de configuração que codifica informações sobre um diretório controlado e supervisionado pelo systemd;
- target: grupos de unidades que reúnem todas as units necessárias para iniciar um determinado serviço;
- snapshot: mecanismo usado para criar snapshots dinâmicos do estado atual do systemd manager, útil para retomar o estado após problemas;
- path: unidades especialmente utilizadas para monitorar arquivos e diretórios para eventos e, também, executar serviços;
- socket: arquivo de configuração que armazena informações acerca de um IPC ou soquete de rede, ou arquivo FIFO;
- swap: guarda informações relativas a dispositivos usados para swapping, bem como serviços que utilizam memória Swap.

Cada serviço é alocado pelo systemd em um grupo de controle dedicado (control group - cgroup). No cgroup são organizadas informações voltadas aos processos que fazem parte do grupo, como: limite, supervisão e contabilização de recursos computacionais que eles consomem.

O controle desses grupos é feito a partir de utilitários que acompanham o systemd, como por exemplo: journalctl, cgl, cgtop e systemctl (este último será descrito na sequência).

## Comandos básicos para controlar serviços no systemd

Para gerenciar o sistema e os serviços existe o aplicativo systemctl, que auxilia no controle do próprio systemd, além de atuar como gerenciador de serviços. Ou seja, ele permite monitorar, encerrar, iniciar, analisar, recarregar, checar o status dos processos, entre outras ações. Alguns exemplos de comandos são mostrados na sequência.

Listar todos os serviços disponíveis:

```
# systemctl list-unit-files --type=service
```

Ativar um serviço e habilitá-lo (ou desabilitá-lo) no boot:



```
# systemctl is-active httpd.service  
# systemctl enable httpd.service  
# systemctl disable httpd.service
```

Realizar ações básicas (iniciar, reiniciar, parar etc.) para determinado serviço:

```
# systemctl start httpd.service  
# systemctl restart httpd.service  
# systemctl stop httpd.service  
# systemctl reload httpd.service  
# systemctl status httpd.service
```

## Execução em Foreground ou Background

Os processos podem ser executados de duas formas: em primeiro plano (*foreground*) ou em segundo plano (*background*). Os processos que executam em *foreground* são os que necessitam de interação direta com o usuário, incluindo a troca de informações. Os processos em *background* não necessitam dessa interação com o usuário.

Há situações em que é necessário passar um processo que está sendo executado em *foreground* para *background* e vice-versa. Por exemplo, em uma sessão de transferência de arquivos entre computadores remotos, a baixa velocidade da linha de transmissão pode fazer com que o tempo de transferência leve muito tempo. Assim, seria interessante passar o processo para segundo plano, liberando o *shell* para outras atividades do usuário.

A passagem de um processo de *foreground* para *background* é realizada primeiro suspendendo o processo (CTRL + Z), seguido do comando "bg", que envia o processo para segundo plano (*background*). Observe que suspender a execução de um processo não significa finalizá-lo, apenas torná-lo temporariamente inativo!

A lista dos processos executados em *background* pode ser visualizada com o comando "jobs", que mostra o número da tarefa (*job*) associada. Se o usuário quiser interagir novamente com o processo, deve utilizar o comando "fg" seguido de % número\_job. Exemplo:

```
estrategia:~$ jobs  
[1] - Running          script.sh  
[2] + Suspended (tty output) teste.sh  
estrategia:~$ fg %2
```

O mais utilizado para executar em background é utilizar "&" logo após o nome do binário ou script. Por exemplo, se você deseja executar um script "aprovacao.sh", que vai realizar vários cálculos e no fim escrever em um arquivo, não é necessário que você fique com o *shell*



"travado", aguardando a conclusão da tarefa. Então é só digitar (note que o *shell* não fica esperando a conclusão do *script*, já aguarda o próximo comando):

```
estrategia:~$ aprovacao.sh &  
estrategia:~$
```

Processos que tipicamente só são executados em *background*, pois não possuem interação com o usuário, são os *daemons*. Geralmente possuem um nome que termina com a letra "d", como por exemplo, *syslogd* (*daemon* que gerencia o *log* do sistema). Em um ambiente Unix, o "processo pai" de um *daemon* é normalmente o processo *init* (PID=1). De forma geral, os sistemas operacionais iniciam *daemons* durante o processo de *boot*.

## Agendamento de Tarefas

Para o agendamento de tarefas a aplicação mais conhecida é o **Cron**, que já existe há muito tempo e passou por muitos estágios de evolução. Porém, a maioria das suas implementações atuais (*Vixie Cron*, *ISC Cron*, *BCron* etc.) ainda se baseiam no pressuposto de que o sistema está em funcionamento de forma ininterrupta. Isso é um problema quando se utiliza a virtualização, quando os sistemas que operam sob demanda, sendo ligados e desligados conforme sua necessidade.

Além disso, os usuários de *notebooks* suspendem ou desligam seus equipamentos para transportá-los, e até mesmo *desktops* e estações de trabalho são desligados para poupar energia. O resultado disso é que os *cronjobs* falham regularmente, pois o sistema se encontra desligado quando a tarefa deveria ser executada!

Para contornar essa situação, algumas distribuições contam com o *Anacron* como alternativa ao *Cron*. O *Anacron* permite a criação de listas de tarefas que serão realizadas em intervalos pré-definidos e, quando o *Anacron* é inicializado, ele verifica essas listas e executa as tarefas ainda não realizadas. No entanto, o *Anacron* possui algumas limitações. A primeira é que ele não é um *daemon*, então precisa ser executado sempre que for necessário.

Outra questão é que o *Anacron* não está preparado para lidar com períodos de tempo menores do que dias. A combinação desses problemas pode levar a situações nas quais o *Cron* e o *Anacron* funcionam ao mesmo tempo, e algumas tarefas podem ser executadas duas vezes ou nenhuma!

Uma solução é o *Fcron*, que faz o que o *Vixie Cron* e o *Anacron* fazem e ainda mais. É possível usar o *Fcron* para agendar *cronjobs* com data e hora fixas, com intervalos de tempo ou até mesmo de acordo com a disponibilidade do sistema.

Independente da implementação utilizada, a configuração é muito parecida, e vale destacar também que o *Cron* (ou uma de suas variantes) é um programa de nível multiusuário, ou seja, cada usuário pode agendar suas tarefas individualmente. Isso ocorre graças ao próprio arquivo onde são armazenados os *scripts*: o **Crontab**.





Como o próprio nome sugere, a estrutura do Crontab consiste em tabelas e nessas tabelas são preenchidas todas as informações referentes à tarefa: **minutos; horas; dias do mês; mês; dias da semana; comando**. Abaixo um exemplo do comando ("crontab -l" lista a crontab):

```
crontab -l  
55 * * * 5 /home/maria/backup.sh
```

Agora vamos analisar alguns exemplos comentados:

# executa cinco minutos depois da meia-noite, todos os dias (\* = tudo)

```
5 0 * * * $HOME/bin/script.sh
```

# executa às 14h15min no dia 1º de cada mês

```
15 14 1 * * $HOME/bin/script.sh
```

# executa às 22h nos dias de semana (1 a 5 = segunda-feira a sexta-feira)

```
0 22 * * 1-5 $HOME/bin/script.sh
```

## Sistemas de Arquivos

Quando falamos em armazenamento de informações a longo prazo temos três requisitos:

1. Deve ser possível armazenar um grande volume de informações;
2. As informações devem "sobreviver" ao término do processo (programa em execução);
3. Vários processos devem ser capazes de acessar as informações ao mesmo tempo (ex.: vários programas acessando o mesmo banco de dados).

A solução mais comum é armazenar as informações em mídias de armazenamento (HDs, SSDs, entre outras), em unidades chamadas **arquivos**. Tais informações armazenadas em arquivos devem ser persistentes, ou seja, não devem ser afetadas pela criação ou término de um processo. O arquivo geralmente só pode "sumir" se for excluído por seu criador (ou pelo administrador do sistema).

O sistema operacional faz o gerenciamento dos arquivos (estrutura, nome, como acessar, usar etc.) através do **sistema de arquivos**. Enquanto o usuário final "enxerga" os arquivos através de seus nomes, localizações, tamanhos, tipos, proprietários etc., os projetistas dos sistemas de arquivos devem se preocupar também com aspectos mais técnicos, como por exemplo se vão ser utilizadas listas encadeadas ou mapa de bits para monitorar o espaço de armazenamento livre.

Mesmo lidando com sistemas de arquivos típicos do Windows (FAT32, NTFS etc.), os sistemas de arquivos nativos do Linux são, entre outros:

**EXT** (EXTended File System): durante algum tempo o **EXT2** foi o sistema de arquivos padrão do Linux, herdando características de outros sistemas de arquivos e sendo baseado no UFS (usado





no FreeBSD e Solaris), além de possuir semelhanças com os sistemas de arquivos HFS e HFS+ (Apple).

A menor unidade de alocação do Ext é o bloco (o equivalente ao *cluster* em sistemas de arquivos FAT e NTFS), cujos tamanhos possíveis são 1024, 2048 e 4096 (escolhido no processo de formatação).

**DESPENCA NA PROVA!**



Quando falamos especificamente da família de sistemas de arquivos EXT, temos que ter em mente que a versão 2 não possuía *journaling*. Somente na versão 3 surgiu! Em seguida vamos ver mais detalhes.

A principal característica do EXT3 é o uso do recurso de *journaling*, onde o sistema de arquivos mantém um *journal* (diário) das alterações realizadas. Esse "diário" armazena uma lista das alterações realizadas, permitindo que o sistema de arquivos seja reparado de forma muito rápida após o desligamento incorreto. O Ext3 possui três modos de operação:

- *ordered* (default): o *journal* é atualizado no final de cada operação. Isso faz com que exista uma pequena perda de desempenho, já que a cabeça de leitura do HD precisa realizar duas operações de gravação, uma no arquivo que foi alterado e outra no *journal* (que é um arquivo especialmente formatado);
- ● *writeback*: o *journal* armazena apenas informações referentes à estrutura do sistema de arquivos (metadados) e não em relação aos arquivos propriamente ditos, é gravado de forma mais ocasional, aproveitando os momentos de inatividade;
- *journal*: é o mais seguro, todavia mais lento. Nesse modo, o *journal* armazena não apenas informações sobre as alterações, mas também uma cópia de segurança de todos os arquivos modificados, que ainda não foram gravados no disco. Por ser o mais lento, é o modo menos usado.

Para finalizar a família EXT, existe o EXT4, com melhorias que incluem, por exemplo, a ampliação do endereçamento para 48 bits, o que permite endereçar um volume virtualmente ilimitado de blocos e partições de até 1 exabyte (1EB = 1024 petabytes)! O limite de 2 TB para os arquivos também foi removido, abrindo espaço para o armazenamento de arquivos gigantescos (no máximo 1 EB, o mesmo tamanho do volume!).

Além do EXT2, EXT3, EXT4, existem outros sistemas de arquivos conhecidos (e cobrados em provas de concurso), a saber:

- **Reiser FS**: utilizado geralmente no Linux, foi o primeiro com suporte a *journaling* (incluído no núcleo Linux 2.4). Seu futuro é incerto depois que seu criador, Hans Reiser, foi condenado pelo assassinato de sua esposa. Há voluntários que continuam com o projeto;
- **XFS**: inicialmente desenvolvido pela Graphics para o seu sistema operacional IRIX, posteriormente teve seu código fonte liberado e foi adaptado para funcionar no



Linux. É um sistema de arquivos desenvolvido em 64 bits, compatível com sistemas de 32 bits. Em plataformas de 64 bits, possui um limite de tamanho de 8 EB para um volume e para cada arquivo. É um sistema de arquivos com journaling;

- **Btrfs** (B-tree file system) é um sistema de arquivos baseado no princípio cópia em gravação (copy-on-write - COW), inicialmente desenvolvido pela Oracle para ser usado no Linux. Foi projetado para solucionar problemas como a falta de agrupamento de discos ou volumes, snapshots, checksums e uso de múltiplos volumes simultaneamente nos sistemas de arquivos do Linux. Possui a limitação de tamanho de volume igual à limitação de tamanho de arquivo (16 EB);
- Red Hat **GFS**: é um sistema de arquivos de cluster que permite que um cluster acesse simultaneamente um disparador de obstáculo do qual é dividido entre os nós. Trata-se de um sistema de arquivos nativo que se conecta diretamente por meio da interface com a camada VFS da interface de sistema de arquivo Kernel Linux. O GFS emprega os metadados distribuídos e diários múltiplos para uma operação mais eficiente em um cluster;
- **NFS** (Network File System): protocolo de sistema de arquivos distribuído, originalmente desenvolvido pela Sun Microsystems, que permite que um usuário em um computador cliente acesse arquivos através de uma rede como se estivesse acessando na máquina local. Utiliza o sistema Open Network Computing Remote Procedure Call (ONC RPC);
- **Swap**: memória virtual do Linux que tem uma partição específica (diferente do Windows que utiliza um arquivo de paginação). Possui uma organização própria, sem utilizar um sistema de arquivos (ou pode ser entendido como tendo um sistema de arquivos próprio para isso).

## LVM (Logical Volume Manager)

O gerenciador de volumes lógicos (LVM) é um recurso que visa facilitar a vida dos administradores de sistema. Embora os usuários domésticos geralmente prefiram criar apenas uma ou duas partições ("/" ou "/" + "/home") durante a instalação do sistema, nos servidores são utilizadas muito mais partições por questões de segurança e desempenho.

O problema é que ao dividir o HD em várias partições diferentes surge o problema do aproveitamento do espaço, já que sempre algumas partições ficam cheias embora ainda exista muito espaço livre nas outras. Geralmente é possível redimensionar as partições, mas esse é sempre um processo demorado e que traz risco de perda de dados.

O LVM resolve o problema permitindo agrupar vários HDs em uma única unidade que pode ser dividida em vários volumes lógicos, vistos pelo sistema operacional como se fossem partições de disco. A grande vantagem é que novos volumes podem ser criados, excluídos ou redimensionados rapidamente e sem a necessidade de reiniciar o servidor.

Caso sejam usados HDs SCSI com suporte a hot-swap, é possível adicionar, remover ou ainda substituir HDs, fazendo as alterações necessárias nos volumes lógicos, tudo sem interrupções! O fato do sistema "visualizar" todos os HDs instalados como uma única unidade também facilita os backups.



Para utilizar o LVM é preciso compilar o Kernel ativando as opções "Multiple devices driver support (RAID and LVM)" e "Logical volume manager (LVM) Support". Em geral as distribuições já trazem esses dois componentes compilados como módulos, bastando ativá-los usando o comando "modprobe".

## Estrutura dos Diretórios

Uma característica forte do Linux é a sua estrutura de diretórios, que pode possuir pequenas modificações entre as distribuições, porém são semelhantes. Segundo o conceito FHS (*File Hierarchy Standard*), que é um padrão para que softwares e usuários possam prever a localização de arquivos e diretórios instalados, abaixo é mostrada a estrutura e o que é armazenado em cada diretório/subdiretório:

- /etc: dados de configuração (scripts de inicialização, tabela de sistemas de arquivo, configurações de login, configuração da fila de impressão, entre outros). A gente pensa que "etc" significa "etcetera", mas "etc" quer dizer "Environment Tables and Controls";
- /usr: maior parte dos aplicativos e bibliotecas do sistema;
- /usr/bin: executáveis dos programas;
- /usr/lib: bibliotecas e arquivos compartilhados;
- /usr/src: código-fonte de programas;
- /usr/doc: documentação em geral;
- /home: arquivos dos usuários (arquivos de trabalho, músicas, filmes, downloads etc.);
- /root: diretório home do root. Obs.: Esse diretório não é a mesma coisa que a raiz do sistema /;
- /var: sites hospedados (nem sempre), bases de dados do MySQL etc.;
- /var/log: logs de diversos serviços;
- /var/www: sites hospedados (nem sempre);
- /boot: imagem do Kernel e o initrd (initial ram disk), carregados no início do boot;
- /bin: comandos básicos, como "cd", "ls" e "cat";
- /lib: módulos do Kernel e bibliotecas básicas do sistema;
- /opt: arquivos de aplicativos adicionais (opcionais, ou "optionals"), que não são essenciais para o sistema. Cada aplicativo tem uma subpasta com seu respectivo nome;
- /proc: informações de depuração do Kernel, configurações que habilitam e desabilitam o suporte à algum elemento no Kernel;
- /tmp: arquivos temporários.

## Diretório /etc

Como já descrito no tópico anterior, o diretório /etc armazena dados de configuração. Vejamos alguns dos arquivos de configuração com suas descrições:

- /etc/group: arquivo de sistema que define os grupos aos quais os usuários pertencem em sistemas UNIX, como o Linux e o BSD;



- `/etc/passwd`: determina quem pode acessar legitimamente o sistema e o que eles podem fazer uma vez dentro;
- `/etc/shadow`: armazena as senhas dos usuários criptografadas, além de informações sobre datas de expiração e validade das contas;
- `/etc/syslog.conf`: usado para controlar a saída dos arquivos de log do daemon `syslogd`;
- `/etc/profile`: contém comandos que são executados para todos os usuários do sistema no momento do login. Somente o usuário `root` pode ter permissão para modificar esse arquivo;
- `/etc/network/interfaces`: usado pelos programas `ifup` e `ifdown`, respectivamente para ativar e desativar as interfaces de rede;
- `/etc/resolv.conf`: usado para configurar a resolução do host via DNS;
- `/etc/exports`: usado para especificar quais diretórios o servidor NFS (Network File System) exporta;
- `/etc/ldap.conf`: usado para a configuração padrão a ser aplicada na execução de clientes LDAP (Lightweight Directory Access Protocol);
- `/etc/ssh/sshd_config`: configuração do SSH.

## Memória Virtual

O conceito de memória virtual está presente nos sistemas operacionais modernos (incluindo o Linux, claro). Sabemos que um programa necessita passar pela memória principal para ser executado, mas e se não há memória RAM suficiente para executar todo o programa? Aí surgiu a ideia de executar o programa em partes e apenas as partes que necessitam ser executadas no momento ficam na memória RAM.

A **memória virtual** é uma área de troca de dados que serve como uma extensão da memória RAM. Por exemplo, se há 20 MB de memória RAM livre e um programa de 30 MB é executado, uma parte é carregada e o restante fica na memória virtual, para depois serem copiados para a memória principal a fim de serem executados.

Isso ocorre à medida que a execução do programa for acontecendo e o programa em questão (ou outros programas) liberarem a memória que ocupavam para o uso de outro programa. Podemos verificar que há um fluxo contínuo de dados entre o disco rígido (onde fica a memória virtual) e a memória RAM. Por isso concluímos que pouca memória RAM implica em usar muita memória virtual, que por sua vez, implica em usar muita leitura/escrita no disco rígido, deixando o funcionamento geral do sistema mais lento!

Geralmente ao instalar qualquer distribuição GNU/Linux, na opção automático da partição do disco onde será instalado o sistema, também encontramos a criação automática da **partição Swap**. Se optar por uma instalação personalizada, deve-se atentar para a criação de uma partição e marcá-la como Swap. Abaixo é mostrada uma partição Swap com cerca de 16 GB através do `Gparted` (utilitário que lista e permite alterar partições, bastante utilizado).

Quando não há partição livre ou não há espaço para a criação da memória virtual e o sistema operacional necessita de memória de troca para o melhor desempenho dos processos é possível



criar o **arquivo de memória Swap**. Nada impede que ambas as soluções sejam utilizadas simultaneamente!

## Gerenciadores de Pacotes

Segundo Morimoto, os **pacotes são as "peças" que formam todas as distribuições Linux** e podem conter programas, bibliotecas de sistema ou mesmo algumas outras "coisas" como papéis de parede e ícones. Alguns programas grandes (KDE, por exemplo) são divididos em vários pacotes para que o usuário possa instalar apenas as partes que interessam, ficando com um sistema mais enxuto.

Alguns pacotes dependem de outros (um programa pode precisar de uma biblioteca que faz parte de outro pacote, por exemplo), as chamadas **dependências**. Para evitar que algo não funcione ou com pacotes desnecessários, o instalador automaticamente verifica as dependências de cada pacote, adicionando ou removendo pacotes relacionados a ele. É por isso que às vezes ao marcar um determinado pacote alguns outros são marcados junto.

Instalar ou remover pacotes no Linux hoje em dia é muito fácil, pois existem vários gerenciadores de pacotes para facilitar essa tarefa. Vamos ver apenas quatro deles, pois são os geralmente cobrados em provas de concurso.

**APT**: resolve dependências para sistemas baseados em Debian (incluindo o Ubuntu). Possui uma sintaxe muito simples. Vamos a alguns exemplos para:

- atualizar os repositórios de software: `sudo apt-get update` ou `sudo apt update` (a lista com os repositórios devem estar no arquivo `"/etc/apt/sources.list"`);
- atualizar seus pacotes já instalados (softwares): `sudo apt-get upgrade` ou `sudo apt upgrade`;
- uma atualização mais completa (tenta atualizar pacotes para a versão mais recente e remover dependências mais antigas ou não utilizadas): `sudo apt-get dist-upgrade` ou `sudo apt full-upgrade`;
- atualização dos repositórios e atualizar também os pacotes já instalados: `sudo apt-get update && sudo apt-get upgrade` ou `sudo apt update && sudo apt upgrade`;
- instalar um software: `sudo apt-get install nome_do_pacote` ou `sudo apt install nome_do_pacote`;
- remover um pacote: `sudo apt-get remove nome_do_pacote` ou `sudo apt remove nome_do_pacote`.

Ao remover um software do sistema usando o comando `apt-get remove`, o APT realiza todo o trabalho de remoção de dependências não utilizadas. Algumas dependências podem permanecer no sistema, então é só utilizar a opção `autoremove`: `sudo apt-get autoremove` ou `sudo apt autoremove`.



Para procurar um pacote a ser instalado, basta usar: `sudo apt-cache search termo_pesquisa` ou `sudo apt search termo_pesquisa`.

O APT atualmente não oferece a possibilidade de instalar um pacote de um *site*, ou seja, o usuário deve encontrar e baixar o pacote a ser instalado por conta própria, utilizando nesse caso o DPKG.

**DPKG:** criado para instalar pacotes ".deb" na distribuição Debian. Também é utilizado por outras distribuições, como Ubuntu, KNOPPIX etc. Os principais comandos utilizados pelo gerenciador de pacotes DPKG são `dpkg` e `dpkg-reconfigure`.

O comando `dpkg` não instala automaticamente as dependências do pacote solicitado. Ou seja, se um pacote que um administrador estiver tentando instalar, tiver dependências não satisfeitas, o `dpkg` não as instalará por conta própria. Ele apenas informará que essas dependências estão ausentes, e a instalação falhará.

Para gerenciar dependências de forma automática, é recomendado utilizar ferramentas de nível mais alto, como o `apt` ou `apt-get`, que são capazes de resolver dependências e instalá-las automaticamente junto com o pacote solicitado.

A sintaxe é: `dpkg [opções] nome_pacote`

Opções:

- i : instalação simples.
- r : desinstala o pacote - exceto arquivos de configuração do pacote.
- P : desinstala o pacote - todos os arquivos do pacote.
- l : exibe os pacotes que estão instalados.
- p : exibe informações sobre o pacote instalado.
- s : exibe o status do pacote.
- l : exibe informações sobre pacotes não instalados.
- S : exibe o pacote do qual o arquivo faz parte.
- L : exibe os arquivos que fazem parte de um pacote instalado.
- c : exibe os arquivos que fazem parte de um pacote não instalado.
- help : exibe uma mensagem de ajuda.

Alguns exemplos:

- Instalação de um pacote: `# dpkg -i pacote.deb`





- Desinstalação de um pacote, mantendo os seus arquivos de configuração: `# dpkg -r pacote`
- Desinstalação de um pacote, inclusive os seus arquivos de configuração: `# dpkg -P pacote`
- Exibição de informações do pacote não instalado: `# dpkg -l pacote.deb`

Em relação ao comando `dpkg-reconfigure`: reconfigura pacotes ".deb" após terem sido instalados utilizando o `debconf` (sistema de configuração de pacotes ".deb"). Esse comando fará perguntas para reconfigurar o pacote.

Sintaxe: `dpkg-reconfigure opções nome_pacote`

Opções:

`-a` ou `--all` :: reconfigura todos os pacotes

`-h` ou `--help` :: exibe ajuda

Exemplo: Reconfigura o pacote `ssh` à `# dpkg-reconfigure ssh`

**RPM**: criado pela Red Hat para instalar pacotes ".rpm", sendo utilizado por várias distribuições, como Red Hat Enterprise/Fedora Core, Mandriva, SuSe, entre outras. O conjunto completo de modos básicos de operação do RPM é a criação, instalação, desinstalação, atualização, consulta e verificação de pacotes. O principal comando do gerenciador de pacotes RPM, é o `rpm`.

Sintaxe: `rpm [opções] nome_pacote`

Algumas opções:

`-i`: instalação simples.

`-v`: exibe detalhes da instalação.

`-h`: mostra o caractere "#", enquanto o programa é instalado.

`-U`: atualização do programa de uma versão anterior para uma mais recente.

`--nodeps`: não procura dependências.

`--force`: força a instalação.

`--root` diretório: utiliza o sistema com raiz em diretório para todas as operações.

`-e`: desinstala o pacote.

`--import`: importa a chave pública GPG do distribuidor de um pacote.

`--help`: exibe uma mensagem de ajuda.

`-q`: consulta (*query*) se um pacote já está instalado.





Exemplos:

```
#rpm -q httpd à Consulta se o pacote está instalado.
```

```
# rpm -ivh tree.i386.rpm à Instala o pacote, exibindo detalhes da instalação e mostra o caractere "#", enquanto o programa é instalado.
```

```
# rpm -e tree à Desinstala o pacote tree.
```

**YUM:** resolve dependências de pacotes para distribuições que utilizam o RPM. É um gerenciador de pacotes padrão incluído em algumas distribuições baseados no REDHAT, incluindo o Fedora e CentOS. Muito parecido com o APT, vejamos alguns comandos:

- Atualização dos repositórios do YUM: `sudo yum update`;
- Instalação de um pacote: `sudo yum install nome_do_pacote`;
- Remoção de um pacote: `sudo yum remove nome_do_pacote`;
- Busca de um pacote que será instalado: `sudo yum search nome_do_pacote`.

O YUM não inclui um comando `autoremove` para encontrar e remover dependências não utilizadas, mas incluí um recurso para a instalação de um pacote a partir de uma URL (função que não existe no APT):

```
sudo yum install $url
```

## Configuração de Serviços ou Funcionalidades

Diversos serviços ou funcionalidades no Linux são configurados através de arquivos texto e alguns deles costumam ser cobrados em provas de concurso. Vamos a eles...

**Segurança da senha do usuário:** os arquivos que armazenam os dados dos usuários do Linux são o `/etc/passwd` e o `/etc/shadow`. Cuidado! O `passwd` não armazena a senha, embora o nome sugira isso! Vamos ver o exemplo de uma linha de cada arquivo, relativo ao usuário "estrategia". Primeiro o arquivo `passwd`:

```
estrategia:x:1010:1010:Estrategia Concursos:/home/estrategia:/bin/bash
```

Separado por dois pontos, podemos identificar os seguintes campos:

- nome do usuário = "estrategia";
- "senha" do usuário (o "x" indica que a senha está armazenada no arquivo `/etc/shadow`);
- identificação do usuário (`uid = 1010`);
- identificação do grupo (`gid = 1010`);
- comentário sobre o usuário (geralmente o nome completo) = "Estrategia Concursos";
- diretório principal (home) do usuário = `/home/estrategia`;
- *shell* utilizado, ao logar, pelo usuário = `/bin/bash`.

Agora vamos ao arquivo `shadow`:



```
estrategia:$6$nAkc...CdbF:17355:0:99999:7:::
```

Separado por dois pontos, podemos identificar os seguintes campos:

- nome do usuário = "estrategia";
- *hash* da senha do usuário = "\$6\$nAkc...CdbF" (obs.: inserir os três pontos no meio porque esse campo é muito grande);
- última vez (em dias) que a senha foi modificada, desde 1 de janeiro de 1970 = 17355;
- depois de alterada a senha o usuário só poderá alterar novamente depois de x dias = 0 (0 significa que está desabilitado);
- a partir do momento que o usuário altera a senha, é obrigado a alterá-la novamente depois de x dias = 99999;
- quantos dias antes do fim do prazo para alterar a senha, deve ser mostrado um alerta = 7.

Os outros dois campos estão vazios, mas significam: inativo (quanto tempo sua conta ficou inativa/desabilitada), expirada (quantos dias depois de 01/01/1970 que sua conta está desabilitada ou expirada).

**Configuração de interfaces de rede:** os programas ifup (habilita interfaces de rede) e ifdown (desabilita) se baseiam na configuração do arquivo "/etc/network/interfaces", o qual vamos ver um exemplo abaixo, com comentários.

```
auto lo                # configuração automática para lo
iface lo inet loopback # lo é uma interface de loopback
auto eth0              # configuração automática para eth0
allow-hotplug eth0    # ativa a interface eth0 (com o ifup)
iface eth0 inet static address 192.168.100.1 netmask 255.255.255.0 network 192.168.100.0
broadcast 192.168.100.255
# a última linha faz com que ifup ative a eth0 com o endereço fixo 192.168.100.1/24
```

Para os próximos dois serviços, é muito difícil cobrar algo muito específico, basicamente as bancas cobram os nomes de arquivos, então vamos direto ao ponto.

**DNS (Domain Name System):** o serviço de resolução de nomes tem um arquivo de configuração, o "/etc/resolv.conf". Lembre que "resolv" está ligado a "resolver um nome", fica mais fácil de lembrar.

**NFS (Network File System):** o sistema de arquivos em rede utilizado pelo Linux utiliza um arquivo que controla quais diretórios o servidor NFS exporta, o "/etc/exports".

## Interoperabilidade (Samba)

Quando falamos em sistemas de arquivos FAT, NTFS, EXT, XFS, entre outros, estamos falando em sistemas de arquivos "locais", os que permitem o acesso a uma mídia interna ou conectada a um computador, por exemplo. Existem também os sistemas de arquivos em rede, que dão a ilusão



de um acesso local, porém o acesso é remoto. Os mais conhecidos são o NFS e o SMB/CIFS, os quais veremos a seguir.

O NFS (*Network File System*) é um protocolo de sistema de arquivos distribuído, originado no Unix, que permite que um usuário em um computador cliente acesse arquivos através de uma rede como se estivesse acessando na máquina local. O cliente NFS tem a finalidade de tornar o acesso remoto transparente para o usuário do computador, e esta interface cliente e servidor, executada pelo NFS através dos protocolos Cliente-Servidor, fica bem definida quando o usuário, ao chamar um arquivo/diretório no servidor, lhe parece estar acessando localmente, sendo que está trabalhando com arquivos remotamente.

Uma máquina Linux pode ser um servidor NFS e um cliente NFS, o que significa que pode exportar sistemas de arquivo para outros sistemas e montar sistemas de arquivo exportados de outras máquinas. Por exemplo, um grupo de usuários pode ter acesso aos arquivos de um determinado projeto usando um diretório compartilhado do sistema de arquivo NFS montado no diretório "/projeto". Para acessar os arquivos compartilhados, o usuário vai até o diretório "/projeto", sem senhas ou comandos especiais para lembrar. Dessa forma, os usuários trabalham como se o diretório estivesse em suas máquinas locais.

Algumas características do NFS são:

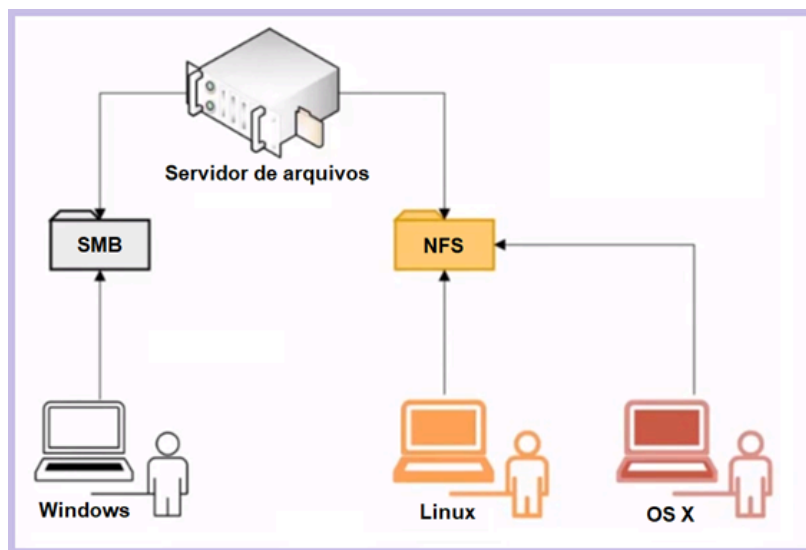
- Arquitetura cliente-servidor: o servidor recebe as requisições vindas do cliente, verifica a validade, executa e retorna ao cliente;
- *Stateless* (sem estado): considera cada requisição como uma transação independente que não está relacionada a qualquer requisição anterior, de forma que a comunicação consista em pares requisição/resposta independentes. Diante de uma falha no sistema, quando o servidor NFS volta a funcionar corretamente, o estado anterior é restaurado e o funcionamento continua como se não houvesse ocorrido nenhum problema;
- *Caching*: permite que as informações utilizadas mais recentemente sejam alocadas para posterior uso, ou pode carregar os dados em antecipação a operações futuras;
- *File Locking* (bloqueio de arquivo): permite que um processo tenha acesso exclusivo a um arquivo ou parte deste, e força outros processos que estão solicitando acesso ao mesmo arquivo, aguardarem a liberação;
- Utiliza o protocolo RPC (*Remote Procedure Call*): define um modo independente do sistema para processos se comunicarem através de uma rede de computadores, sendo que o cliente faz a chamada do procedimento remoto no servidor;
- Protocolo de transporte padrão é o UDP, no entanto, é possível utilizar o TCP;
- Compartilhar arquivos de um servidor NFS é conhecido como exportar os diretórios.

Para que os clientes possam acessar o servidor NFS é necessário que os seguintes *daemons* estejam em execução:



- `nfsd`: *daemon* NFS, atende requisições dos clientes NFS;
- `mountd`: *daemon* de montagem NFS, executa as solicitações passadas pelo `nfsd`;
- `portmap`: *daemon portmapper*, permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

Na figura abaixo podemos ver o acesso a um servidor de arquivos através da rede. Note que máquinas Unix-like (ex.: Linux e OS X) utilizam o NFS como protocolo e as máquinas Windows utilizam o SMB/CIFS, que veremos o funcionamento na sequência.



Antes de começar a falar do protocolo em si, deixo bem claro que as questões de concurso já cobraram alguns detalhes do NFS, mas em relação ao SMB/CIFS a cobrança é superficial. Vamos lá...

O **SMB/CIFS** (*Server Message Block/Common Internet File System*) é um protocolo de redes cujo uso mais comum como é o compartilhamento de arquivos em uma rede local. Ele permite que o cliente manipule arquivos como se estes estivessem em sua máquina local (mesma coisa que o NFS, heim?). Algumas das operações suportadas são: leitura, escrita, criação, exclusão e renomeação, sendo que os arquivos/diretórios manipulados estão em um servidor remoto.

O protocolo SMB/CIFS funciona através do envio de pacotes do cliente para o servidor. O servidor recebe o pacote, verifica se a requisição é válida (se o cliente possui as permissões apropriadas), executa a requisição e retorna um pacote de resposta ao cliente. O cliente verifica o pacote de resposta para determinar se a requisição inicial obteve êxito. O protocolo mais utilizado para transporte confiável é o NetBIOS sobre TCP (NBT). Outros protocolos foram utilizados na camada de transporte, mas o NBT se tornou o mais utilizado.

O SMB/CIFS é muito utilizado pelos sistemas operacionais Windows, podendo funcionar como um servidor ou cliente. Isso não quer dizer que sistemas Unix-like não possuam também esse protocolo! Na verdade, a maioria dos sistemas Unix-Like (**Linux**, por exemplo) possuem uma implementação de cliente/servidor do SMB/CIFS via **Samba**, que é um "software servidor" para Linux que permite o gerenciamento e compartilhamento de recursos em redes formadas por computadores com o Windows. Dessa forma é possível utilizar o Linux como servidor de

arquivos, servidor de impressão, entre outros, como se a rede utilizasse servidores Windows (Server 200x, 201x, entre outros).

Resumindo: o Samba é um componente que realiza a comunicação entre servidores Linux e Windows, permitindo que eles compartilhem recursos de disco e de impressão. O *daemon* `smbd` possui dois modos de compartilhamento. O primeiro modo (*sharemode*) é mais simples, gerando uma senha para cada recurso que é compartilhado. O segundo modo (*user*) permite o compartilhamento de vários recursos com um único *login* e senha. Os privilégios de acesso aos recursos devem ser sempre mediados pelos administradores do sistema.

Um outro componente interessante do pacote Samba é o *nmbd*, um servidor de nomes NetBios (mais conhecido por **WINS - Windows Internet Name Server**) responsável por entender e responder solicitações de resolução de nomes NetBios sobre IP. É ele o responsável pelo "aparecimento" do ícone do servidor Samba no ambiente de rede do Windows! Obs.: Se você não sabe o que é essa resolução de nomes, pense que é algo como o DNS, mas não de forma hierárquica, é uma resolução feita pelo Windows (sabendo isso já está bom para a prova 😊).

## Serviço de Diretório

Um serviço de diretório é um sistema de software que armazena, organiza e fornece acesso a informações em um diretório do sistema operacional. Diretórios podem ser muito limitados em escopo, suportando apenas um pequeno conjunto de tipos de nós e tipos de dados, ou eles podem ser muito amplos, suportando um conjunto de tipos arbitrário ou extensível. No DNS os nós são nomes de domínio e os itens de dados são endereços IP (e apelidos, nomes de servidor de e-mail etc.).

Em um diretório utilizado por um sistema operacional de rede, os nós representam **recursos** que são gerenciados pelo sistema operacional, incluindo usuários, computadores, impressoras, entre outros recursos compartilhados. Muitos serviços de diretório têm sido utilizados desde o aparecimento da Internet, mas o mais utilizado é um descendente do serviço de diretório X.500.

Antes de vermos uma implementação de serviço de diretório para o Linux (OpenLDAP), é importante entendermos o protocolo que fornece mecanismos de acesso aos objetos, o **LDAP (Lightweight Directory Access Protocol)**. Entidades internacionais (ITU, ISO, IETF, entre outras) trabalham na definição de padrões diversos, incluindo a padronização que dá suporte a serviços de diretórios. Um padrão de uso genérico é o **X.500** (da ISO) que possui uma grande abrangência, mas é muito complexo e não foi adotado em sua íntegra como um padrão de mercado. Um padrão mais "light" que de fato se tornou um padrão de mercado foi o LDAP.

O padrão LDAP define um sistema de nomeação hierárquico, no qual é possível referenciar qualquer objeto que esteja no AD. Um nome LDAP é composto pelo caminho completo do objeto (ex.: uma impressora, um computador etc.), partindo do domínio raiz até chegar ao objeto em si. Algumas abreviaturas (**atributos**) são utilizadas nessa nomenclatura hierárquica:

- `cn`: *common name* (nome da conta de um usuário, grupo etc.);
- `sn`: sobrenome (*surname*);
- `ou`: faz referência a uma unidade organizacional;
- `dc`: componente de domínio (normalmente o nome do domínio);
- `o`: nome da organização (geralmente o domínio raiz);



- c: *country* - país (normalmente não utilizado).

Vamos a um exemplo de um nome LDAP:

CN=evandrodv, OU=professores, DC=ti, DC=estrategiaconcursos.com.br à esse nome representa o usuário "evandrodv", cuja conta está contida na unidade organizacional "professores", no domínio "ti.estrategiaconcursos.com.br". Obs.: os dois componentes de domínio foram concatenados.

Por padrão um servidor LDAP "*escuta*" na porta 389 (TCP) e as principais características do protocolo são:

- Baseado em padrão aberto: qualquer desenvolvedor pode acessar sua especificação e realizar a implementação;
- Possui APIs bem definidas: facilita a vida dos programadores;
- Maior velocidade de consulta que um BD relacional;
- Replicável e distribuível;
- Facilita a localização de informações e recursos: pesquisa feita nome.

Algumas *operações (comandos)* que podem ser utilizados através do LDAP são:

- Bind: autentica e especifica a versão do protocolo LDAP;
- Search: procura/recupera entradas dos diretórios;
- Compare: compara uma entrada com determinado valor;
- Add: adiciona uma nova entrada;
- Delete: exclui uma entrada;
- Modify: modifica uma entrada;
- Modify DN: move ou renomeia uma entrada;
- Abandon: aborta uma requisição prévia;
- Unbind: fecha a conexão;
- Extended Operation: operação genérica para definir outras operações;
- StartTLS: protege a conexão com o TLS - implementada a partir da versão 3 do LDAP.

A representação dos dados é realizada através de uma estrutura hierárquica na forma de árvore (*Directory Information Tree - DIT*), a qual consiste em entradas de DN's (*Distinguished Names*). O LDAP utiliza a DIT como estrutura de dados fundamental:

Como podemos ver, a árvore de diretório possui uma forma hierárquica:

- Primeiro o diretório raiz;





- Na sequência: a rede corporativa, os departamentos e por fim os computadores dos usuários e os recursos de rede.

Alguns conceitos que também já foram cobrados em concursos são mostrados na sequência.

**Schema:** conjunto de objetos e atributos para o armazenamento. É modelado de acordo com o padrão X.500 da ISO.

Cada entrada (objeto) possui um identificador único (*dn - distinguished name*), o qual consiste de seu Relative Distinguished Name (RDN), construído de algum(ns) atributo(s) na entrada, seguido pelo DN da entrada pai.

**Escalabilidade:** é possível replicar servidores LDAP e incluir novos servidores à medida que aumenta a estrutura da organização. Ou seja, não é uma estrutura "engessada".

Como já mencionado anteriormente, uma solução bastante conhecida para Linux é o **OpenLDAP** (que também possui versões para Windows, Solaris, entre outros sistemas operacionais). Vamos ver como instalar o OpenLDAP:

```
sudo apt install slapd ldap-utils
```

O principal arquivo de configuração (pelo menos para concursos) é o **"/etc/ldap.conf"**, que é usado para configurar os padrões a serem aplicados quando da execução dos clientes LDAP.

## Links Simbólicos e *Hard Links*

No Linux, os links simbólicos e *hard links* são métodos para criar referências a arquivos existentes no sistema de arquivos, porém eles funcionam de maneiras diferentes e têm usos distintos. Vamos ver como funciona cada um, a seguir.

**Link Simbólico** (*Soft Link* ou *Symlink*): é um tipo especial de arquivo que aponta para outro arquivo ou diretório. Ele age como um "atalho" (como é conhecido no Windows) para o arquivo original.

- Comportamento: Se o arquivo original for alterado ou excluído, o link simbólico continuará existindo, mas ficará "quebrado" e não funcionará mais corretamente;
- Uso: Links simbólicos são úteis quando é necessário criar referências em locais diferentes no sistema de arquivos que apontem para um único arquivo ou diretório. Por exemplo, pode haver um diretório em /home/evandro/documentos e criar um link simbólico para ele em /home/evandro/desktop;
- Comando: `ln -s`, com a seguinte sintaxe:

```
ln -s /caminho/para/arquivo_original /caminho/para/link_simbólico
```

**Hard Link:** Um hard link é um outro nome para o mesmo arquivo. Diferente dos links simbólicos, um hard link aponta diretamente para os dados do arquivo no disco, não para o nome do





arquivo. Para lembrar, é só associar a palavra *hard* (rígido) com uma posição do arquivo no disco rígido (*hard disk*), fica mais fácil assim 😊.

- Comportamento: Como os *hard links* compartilham o mesmo *inode*, se um arquivo for alterado usando qualquer um dos *hard links*, o conteúdo será atualizado em ambos. Se o arquivo original for excluído, o *hard link* ainda permitirá o acesso ao conteúdo, pois ele é apenas outra referência aos mesmos dados (lembre-se que a exclusão de um arquivo não destrói fisicamente os dados, apenas a referência a ele é "destruída");
- Limitações: *Hard links* não podem ser criados para diretórios (por questões de estrutura do sistema de arquivos) e também não podem cruzar sistemas de arquivos diferentes (duas partições diferentes, por exemplo);
- Comando: `ln` (sem a opção `-s`), com a seguinte sintaxe:

```
ln /caminho/para/arquivo_original /caminho/para/hard_link
```

## Distribuições Linux

As bancas raramente cobram detalhes de determinadas distribuições, mas é importante ter uma noção daquelas mais conhecidas/cobradas em concursos, conforme veremos a seguir.

### Red Hat Enterprise Linux (RHEL)

O Red Hat Enterprise Linux (RHEL) é uma distribuição Linux de classe empresarial desenvolvida pela Red Hat. Alguns aspectos importantes sobre o RHEL são:

- Estabilidade e Segurança: O RHEL é conhecido por sua estabilidade e segurança, tendo sido projetado para oferecer um ambiente operacional confiável para servidores e sistemas críticos de negócios (ambiente empresarial);
- Compatibilidade Binária: Oferece compatibilidade binária com outras distribuições baseadas em Red Hat, como o CentOS ou o Oracle Linux. Ou seja, os aplicativos desenvolvidos e testados no RHEL normalmente podem ser executados nessas distribuições sem modificações significativas;
- Modelo de Assinatura: O RHEL é distribuído através de um modelo de assinatura (software pago). As assinaturas incluem acesso ao software, suporte técnico, atualizações de segurança, entre outros itens;
- Suporte de Longo Prazo: Existe o suporte de longo prazo para cada versão principal do RHEL. As empresas podem implantar uma versão do RHEL e receber atualizações de segurança e manutenção por um período estendido, algo em torno de 10 anos;
- Grande Conjunto de Software: O RHEL é acompanhado por um amplo conjunto de software, incluindo ferramentas de gerenciamento de sistemas, servidores Web, bancos de dados, entre outros;



- Uso em Ambientes Empresariais: O RHEL foi projetado e é amplamente utilizado em ambientes corporativos, incluindo data centers, servidores, nuvens e ambientes de missão crítica. Isso ocorre por causa de sua estabilidade, segurança e suporte de longo prazo, visto que as organizações buscam uma plataforma confiável para suas operações de TI.

Resumindo, podemos dizer que o Red Hat Enterprise Linux é uma distribuição robusta e confiável, projetada para atender às necessidades das empresas em termos de estabilidade, segurança e suporte. É uma escolha popular para ambientes de produção críticos e é amplamente utilizado em todo o mundo.

## Rocky Linux

Rocky Linux é uma distribuição Linux projetada para ser uma alternativa 100% compatível com o Red Hat Enterprise Linux (RHEL). Ela surgiu como um projeto de código aberto liderado pela comunidade, após a decisão da Red Hat de descontinuar o CentOS Linux.

As principais características sobre Rocky Linux são:

- Compatibilidade Binária com RHEL: Totalmente compatível em nível de binário, garantindo que pacotes e aplicações funcionem de forma idêntica;
- Distribuição *Enterprise*: Focada em ambientes corporativos, com suporte de longo prazo e estabilidade para produção;
- Modelo de Lançamento: A cada novo lançamento do RHEL, o Rocky Linux lança uma nova versão equivalente, assim as empresas podem realizar migrações sem mudanças de compatibilidade;
- Suporte da Comunidade e Corporativo: Embora seja gratuito, várias empresas oferecem suporte pago para o Rocky Linux, além da comunidade ativa que apoia o projeto.

## Administração e Gerenciamento do Rocky Linux

A administração do Rocky Linux é similar à do RHEL e CentOS, utilizando ferramentas comuns de gestão de sistema, como por exemplo:

- dnf: Gerenciador de pacotes, usado para instalar, atualizar e remover pacotes. Ex.:

```
sudo dnf install nome_pacote
```

- Systemd: Gerenciamento de serviços e inicialização de processos com systemd, incluindo comandos como "systemctl start", "systemctl stop", "systemctl enable" e "systemctl status" para controle de serviços;
- SELinux: Ferramenta de segurança integrada que aplica políticas de segurança em nível de sistema, com o controle de permissões de processos e serviços;
- Gerenciamento de usuários e grupos através de comandos como useradd, usermod e passwd. Também permite integração com LDAP para autenticação em redes corporativas.



## Configuração de Rede

Configurar a rede em Rocky Linux envolve ferramentas como:

- NetworkManager: A principal ferramenta para gerenciamento de conexões de rede, que permite configuração manual e automática de redes com comandos como nmcli e nmtui.
- Configuração de IP Estático e DHCP: Pode ser feita via nmcli ou editando diretamente arquivos de configuração de rede em /etc/sysconfig/network-scripts/.
- Firewall com firewalld: O firewalld é o serviço padrão para configurar regras de firewall no Rocky Linux, permitindo habilitar/desabilitar portas e serviços.

## Configuração de Serviços Essenciais

Rocky Linux é amplamente utilizado para hospedar serviços de infraestrutura. Alguns exemplos de configuração são:

- Servidor Web (Apache ou NGINX):

```
sudo dnf install httpd
```

```
sudo systemctl enable --now httpd
```

- Servidor de Banco de Dados (MySQL/MariaDB ou PostgreSQL):

```
sudo dnf install mariadb-server
```

```
sudo systemctl enable --now mariadb
```

- Ferramentas como o Samba são usadas para configurar compartilhamento de arquivos em redes mistas (Windows/Linux), enquanto o NFS geralmente é preferido para ambientes "somente Linux".

## CentOS

CentOS é uma distribuição Linux de código aberto baseada no Red Hat Enterprise Linux (RHEL). É conhecida por sua estabilidade e é amplamente utilizada em servidores e ambientes de computação em nuvem. Trata-se de uma distribuição gratuita e é mantida pela comunidade de desenvolvedores. Oferece suporte de longo prazo e é uma escolha popular para empresas que desejam uma alternativa de baixo custo em relação ao RHEL. Alguns detalhes sobre o CentOS são:

- Origem/História: Foi lançado em 2004 como um clone do Red Hat Enterprise Linux (RHEL). É desenvolvido pela comunidade e mantido pelo CentOS Project, independente da Red Hat, embora seja altamente compatível com o RHEL;
- Objetivo: Fornecer uma distribuição estável, de código aberto e de uso gratuito que seja compatível com o RHEL. Geralmente é usado em servidores, infraestrutura de rede e ambientes de computação em nuvem;



- **Compatibilidade:** Existe a compatibilidade binária com o RHEL, ou seja, os aplicativos e ferramentas desenvolvidos para o RHEL geralmente podem ser executados no CentOS sem a necessidade de modificações significativas;
- **Ciclo de Lançamento e Suporte:** Oferece versões de lançamento regular, com suporte a longo prazo (LTS). As versões principais recebem atualizações de segurança e manutenção por um período estendido de tempo, algo em torno de 10 anos;
- **Comunidade e Documentação:** Existe uma grande comunidade de usuários e desenvolvedores que oferecem suporte através de fóruns, listas de discussão e outros meios online. Também existe vasta documentação disponível, incluindo guias de instalação, tutoriais e referências de configuração;
- **Recursos e Pacotes:** O CentOS inclui uma ampla gama de pacotes de software, incluindo ferramentas de sistema, servidores Web, bancos de dados, linguagens de programação, etc. Os administradores podem usar o gerenciador de pacotes YUM (Yellowdog Updater, Modified) para instalar, atualizar e remover softwares no sistema.

Resumindo, o CentOS é uma distribuição Linux popular e confiável, adequada para diversas situações, desde servidores corporativos até infraestrutura de nuvem. Essa distribuição oferece estabilidade, compatibilidade e suporte de longo prazo, tornando-se uma escolha popular para muitas organizações e profissionais de TI.

## Oracle Linux

Oracle Linux é uma distribuição baseada no Red Hat Enterprise Linux (RHEL) e desenvolvida pela Oracle Corporation. Alguns pontos importantes sobre o Oracle Linux são:

- **Baseado no RHEL:** O Oracle Linux é construído com base no código fonte do RHEL, fazendo com que compartilhe muitas características e funcionalidades com essa distribuição. Oferece compatibilidade binária com o RHEL, permitindo que aplicativos desenvolvidos e testados no RHEL sejam executados no Oracle Linux sem modificações significativas;
- **Kernel Unbreakable Enterprise (KUE):** Versão personalizada do kernel Linux desenvolvida pela Oracle, incluindo melhorias de desempenho, segurança e suporte a tecnologias específicas da Oracle;
- **Modelo de Assinatura:** Assim como o RHEL, o Oracle Linux é distribuído usando um modelo de assinatura (paga). As assinaturas incluem acesso ao software, suporte técnico, atualizações de segurança etc.;
- **Recursos Adicionais:** Além do sistema operacional em si, o Oracle Linux é acompanhado por uma diversos recursos adicionais, como o Oracle Linux Support, que oferece suporte técnico 24/7, e o Oracle Linux Premier Support, que inclui ferramentas de gerenciamento e monitoramento avançadas;



- **Uso em Ambientes Oracle:** O Oracle Linux é frequentemente utilizado em ambientes onde as soluções da Oracle são utilizadas, como bancos de dados Oracle Database, Oracle Middleware, entre outros. Ele é otimizado para integração e desempenho em ambientes Oracle;
- **Open Source e Comunidade:** Embora seja desenvolvido e mantido pela Oracle Corporation, por incrível que pareça, o Oracle Linux é uma distribuição de código aberto. A Oracle contribui para a comunidade de software livre e permite que os usuários contribuam e modifiquem o código fonte do Oracle Linux.

Resumindo, o Oracle Linux é uma distribuição empresarial robusta e confiável, baseada no RHEL e otimizada para integração com as soluções Oracle. Oferece compatibilidade, desempenho e suporte para empresas que dependem das tecnologias da Oracle em seus ambientes de TI.

## Ubuntu

O Ubuntu é uma das distribuições mais populares e utilizadas no mundo. Alguns detalhes importantes sobre o Ubuntu são:

- **Origem/História:** O lançamento foi em 2004. É baseado no sistema operacional Debian e é desenvolvido de forma colaborativa por uma comunidade global de desenvolvedores;
- **Foco na Usabilidade e Acessibilidade:** Uma de suas principais características é o foco na usabilidade e acessibilidade, pois foi projetado para ser fácil de instalar, configurar e usar, tornando-o adequado para diversos perfis de usuários, desde iniciantes até profissionais experientes;
- **Ciclo de Lançamento Regular:** O Ubuntu segue um ciclo de lançamento regular, com novas versões sendo lançadas a cada seis meses. Cada versão recebe nove meses de suporte para atualizações de segurança e manutenção;
- **Edições e "Sabores":** O Ubuntu está disponível em diversas edições e "sabores" que atendem a diferentes necessidades e preferências dos usuários. Alguns exemplos são o Ubuntu Desktop, Ubuntu Server, Ubuntu GNOME etc.;
- **Software Center e Repositórios:** O Ubuntu Software Center é uma interface gráfica para descobrir, instalar e gerenciar aplicativos. Possui repositórios de software com milhares de aplicativos gratuitos e de código aberto;
- **Suporte Comunitário e Profissional:** Possui uma grande comunidade de usuários e desenvolvedores que oferecem suporte por meio de meios online (wikis, fóruns, listas de discussão etc.);
- **Uso em Ambientes Corporativos e de Servidor:** Embora seja mais conhecido como uma distribuição para o uso em desktops, o Ubuntu também é usado em ambientes corporativos e de servidor. É uma escolha popular para servidores Web, bancos de dados, nuvens privadas e públicas, entre outros.



Resumindo, o Ubuntu é uma distribuição popular e versátil, conhecida por sua facilidade de uso, suporte à comunidade ativa e ampla variedade de edições. É uma distribuição usada por milhões de pessoas, desde usuários domésticos até empresas e organizações governamentais.





## LINUX - SHELL SCRIPT

Linguagens de script são linguagens de programação executadas do interior de programas, a partir de outras linguagens de programação, entre outras situações. Elas servem para estender a funcionalidade de um programa e/ou controlá-lo, acessando sua API (*Application Programming Interface*) e, são frequentemente usadas como **ferramentas de configuração e instalação em sistemas operacionais**, como por exemplo, em alguns sistemas operacionais da família Linux.

O Unix Shell é ao mesmo tempo um interpretador de comandos e uma linguagem de programação. Como interpretador de comandos, ele dá acesso ao rico conjunto de utilidades do GNU (GNU's Not Unix!) e como linguagem de programação ele permite que tais utilidades sejam combinadas. Arquivos contendo comandos podem ser criados e se tornar comandos (assim como os arquivos em lote no Windows - .BAT). Esses novos comandos têm o mesmo status de comandos de sistema como os do diretório /bin. Importante destacar que Shell é uma linguagem interpretada, ou seja, os comandos são interpretados na sequência, um a um, sem haver uma compilação.

Na sequência veremos um passo a passo de como criar um shell script.

### Criando um arquivo:

Há dois modos de realizar essa ação: via modo gráfico ou via terminal. No segundo caso, pode ser utilizado o comando vi, conforme mostrado a seguir (note a "extensão" .sh que geralmente é utilizada, mas poderia ser qualquer outra, ou sem extensão!).

vi script.sh à Será criado e aberto um arquivo de leitura e escrita.

Outra opção é digitar o comando touch:

touch script.sh à cria um arquivo sem abri-lo.

### Permissão ao arquivo:

Para começar a editar o arquivo, é necessário conceder a permissão de escrita a ele. Uma opção é utilizar o comando:

chmod 777 script.sh à o comando chmod é utilizado para conceder permissões em diretórios e arquivos, enquanto o valor 777 permite que o usuário tenha total liberdade para editar o arquivo (na verdade dá liberdade total a qualquer usuário do sistema! Se o foco for dar permissão completa apenas para o dono do arquivo, seria "chmod 700 script.sh", mas nosso foco agora não é estudar o comando chmod).

Pensando mais em segurança, pode-se, por exemplo, liberar apenas a execução (x):

chmod +x script.sh à como não foi especificado para quem foi liberada a execução (usuário, grupo ou todos), todos devem ter a permissão de execução, neste caso.

### Editar e executar o arquivo:



Depois da permissão do arquivo, vamos para a edição do arquivo com o comando vi:

vi script.sh à após aberto o vi, é necessário digitar "i" para colocar o vi no modo "inserção".

Agora é necessário **definir qual Shell será utilizado (na primeira linha do arquivo)**. Vamos escolher o bash, mas poderia ser outro interpretador de comandos, como sh, ksh, csh, entre outros:

```
#!/bin/bash
```

Observe que os caracteres #! são utilizados antes do caminho onde está o shell a ser utilizado.

### Comentários

Para inserir comentários basta utilizar o caractere "#" antes do comentário, como mostrado abaixo:

```
# A linha abaixo serve para bla bla bla
```

### Definição de variáveis

As variáveis são utilizadas para armazenar dados e valores temporários ou resultados intermediários para processamento. Elas são definidas da seguinte maneira:

```
nome_da_variavel=valor
```

Onde:

- nome\_da\_variavel: nome que se deseja dar à variável. Deve começar com uma letra e não pode conter espaços ou caracteres especiais, exceto o sublinhado ("\_");
- valor: conteúdo que se deseja atribuir à variável.

Exemplos de definição de variáveis:

```
# Definindo uma variável chamada nome com o valor Maria
```

```
nome="Maria"
```

```
# Definindo uma variável chamada idade com o valor 40
```

```
idade=40
```

```
# Definindo uma variável chamada preco com o valor 50.30
```

```
preco=50.30
```

```
# Definindo uma variável chamada frutas com uma lista de valores
```

```
frutas=("melancia" "abacaxi" "pera")
```

### Comando echo



Vamos ver como é possível acessar o conteúdo das variáveis abaixo, através do comando echo.

```
echo "Oi, $nome!"
```

Ex. de exibição do comando acima: Oi, Maria!

```
# Realizando uma operação matemática com a variável idade
```

```
idade_prox_ano=$((idade + 1))
```

```
echo "No próximo ano você terá $idade_prox_ano anos!"
```

```
# Acessando elementos da variável frutas
```

```
echo "A primeira fruta é ${frutas[0]}"
```

```
echo " A segunda fruta é ${frutas[1]}"
```

Abaixo podemos ver alguns exemplos de impressões através do comando echo. Note que o caractere \ não é mostrado. Se quiser imprimir \, deve-se utilizar duas vezes (\\). Se imprimir frases sem variáveis, tanto faz utilizar aspas duplas ou não.

```
[root@localhost ~]# echo oi
oi
[root@localhost ~]# echo "oi"
oi
[root@localhost ~]# echo oi \pessoal
oi pessoal
[root@localhost ~]# echo oi \ pessoal
oi pessoal
[root@localhost ~]# echo oi \\ pessoal
oi \ pessoal
[root@localhost ~]# echo "oi \\ pessoal"
oi \ pessoal
[root@localhost ~]#
```

A opção -n serve para imprimir sem a quebra de linha. Ex.:

```
echo -n "Isso é um exemplo"
```

```
echo " de como é utilizado o echo -n."
```

A saída seria:

Isso é um exemplo de como é utilizado o echo -n.

### Entrada de dados do usuário

Se o script pedir ao usuário o fornecimento de dados para a entrada do processamento, o programador precisará digitar um comando para ler esses dados:



read nome\_variavel

### Comando de seleção

Um comando simples que permite a execução de uma tarefa baseada na ação de um usuário que esteja usando o seu sistema é o condicional if. A sintaxe é a seguinte:

```
if [ CONDIÇÃO ] ;
```

```
then
```

```
AÇÕES
```

```
fi
```

É importante lembrar que para cada if aberto, deve haver um fi que feche a sequência. A mesma regra serve para os colchetes.

### Laço while

Executa o laço enquanto uma condição for verdadeira. Sintaxe:

```
while [ CONDIÇÃO ]; do
```

```
AÇÕES
```

```
done
```

### Laço until

Executa o laço enquanto uma condição for falsa! Ou seja, trata-se de um "while invertido". Sintaxe:

```
until [ CONDIÇÃO ];
```

```
do
```

```
AÇÕES
```

```
done
```

### Operadores de comparação

Quando a comparação é entre inteiros, pode aparecer o símbolo ou um hífen e duas letras (neste caso é importante saber um pouco inglês). Vejamos abaixo com um exemplo de cada (vou colocar entre parênteses o significado, mas o operador é somente um hífen e duas letras).

-eq (is equal to - "igual a")

```
if [ "$a" -eq "$b" ]
```

-ne (is not equal to - "não igual a", ou "diferente de")



if [ "\$a" -ne "\$b" ]

-gt (is greater than - "maior que")

if [ "\$a" -gt "\$b" ]

-ge (is greater than or equal to - "maior ou igual a")

if [ "\$a" -ge "\$b" ]

-lt (is less than - "menor que")

if [ "\$a" -lt "\$b" ]

-le (is less than or equal to - "menor ou igual a")

if [ "\$a" -le "\$b" ]

< (menor que - deve-se usar com parênteses duplos)

(( "\$a" < "\$b" ))

<= (menor ou igual a - deve-se usar com parênteses duplos)

(( "\$a" <= "\$b" ))

> (maior que - deve-se usar com parênteses duplos)

(( "\$a" > "\$b" ))

>= (maior ou igual a - deve-se usar com parênteses duplos)

(( "\$a" >= "\$b" ))

## Strings

Vamos ver como se comportam as strings através de um exemplo, abaixo.

```
#!/bin/bash
frase="ESTA É APENAS UMA FRASE!!!"
subStringA=${frase:7:6}
subStringB=${frase:14:3}
StringFinal="${subStringA} {subStringB}"
echo $StringFinal
```

Podemos verificar que a variável frase recebeu uma string com tamanho 26 (posições 0 a 25, sendo 'E' localizado na posição 0 e a última exclamação na posição 25). Na sequência, as variáveis subStringA e subStringB receberam uma parte da variável frase.



A variável `subStringA` recebeu, a partir da posição 7 (oitava posição, já que começa em 0), seis caracteres, ou seja, recebeu a string "APENAS". A variável `subStringB` recebeu, a partir da posição 14 (décima quinta posição, já que começa em 0), três caracteres, ou seja, recebeu a string "UMA".

A `StringFinal` recebeu a concatenação de `subStringA` e `subStringB`, com um espaço entre o conteúdo delas, ou seja, recebeu "APENAS UMA". Por fim, o comando "echo" imprime o conteúdo de `StringFinal`.

Qual a diferença de aspas duplas para **aspas simples**? Ao usar aspas simples ao redor de uma string, cria-se um literal e impede a expansão de variáveis ou caracteres especiais dentro da string. Isso é útil quando se deseja que o conteúdo da string seja interpretado literalmente, sem expansão. Vamos a um exemplo:

```
palavra="coisa"
```

```
frase='Isso é uma $palavra'
```

Com aspas simples, a variável `$frase` não será expandida, ela será tratada como parte da string:

```
echo $frase
```

Gera como saída: Isso é uma \$palavra

Se fossem utilizadas aspas duplas em torno da string, a expansão de variáveis ocorreria:

```
frase="Isso é uma $palavra"
```

```
echo $frase
```

Gera como saída: Isso é uma coisa

Resumindo: ao usar aspas simples, é garantido que o conteúdo da string seja tratado como um literal e que o shell não interpreta variáveis ou caracteres especiais dentro dela. Isso útil em situações em que se deseja preservar exatamente o que está dentro da string.

Para realizar **substituições** de strings, uma maneira é:

```
string=${string//antigo/novo}
```

Onde: "antigo" é o caractere a ser substituído e o "novo" é o que substitui. No exemplo abaixo o "a" é substituído por "x".

```
#!/bin/bash
string="estrategia"
string=${string//a/x}
echo $string
```

Ao executar o script acima, temos:





```
[root@localhost ~]# ./estrategia.sh  
estrxtexix
```

## Funções

Para organizar, separar e estruturar um algoritmo, é necessário o uso de funções:

```
nome_funcao()
```

```
{
```

```
AÇÕES
```

```
}
```

## Inserção de argumentos

O shell script recebe dados fornecidos por outro programa ou por um usuário com a finalidade de produzir saídas. Trata-se dos **argumentos**. Alguns nomes que qualificam os argumentos são:

\$# - total de argumentos que foram passados;

\$@ - uma lista de todos os argumentos passados;

\$\* - retorno aos argumentos;

\$0 - nome do script executado;

\$1 - o primeiro argumento enviado ao script;

\$2 - o segundo argumento enviado ao script;

\$3 - o terceiro argumento, e assim por diante...

Um exemplo simples é mostrado abaixo (script.sh), sendo o script à esquerda e a saída de sua execução à direita. Note que na execução do script são passados 3 parâmetros (a, b, c) e na chamada da função imprime() são passados 4 parâmetros (um, dois, tres, quatro).



<pre>script.sh  #!/bin/sh imprime () { echo "Sou o programa \$0" echo "Recebi \$# parametros" echo "Param 1: \$1" echo "Param 2: \$2" echo "Lista de parâmetros: \$*" } imprime um dois tres quatro echo "Sou o programa \$0" echo "Recebi \$# parametros" echo \$1 \$2 \$3</pre>	<pre>Execução do script.sh  \$ ./script.sh a b c  Sou o programa script.sh Recebi 4 parametros Param 1: um Param 2: dois Lista de parâmetros: um dois tres quatro Sou o programa script.sh Recebi 3 parametros a b c</pre>
---	--

### Vetores (Arrays)

Vetores (*arrays*) são uma forma de armazenar várias informações em uma única variável, onde cada posição do vetor pode conter um valor diferente. Os vetores são indexados numericamente, começando com a posição 0, e podem ser manipulados de várias formas.

Para declarar um vetor:

```
meu_vetor=("valor1" "valor2" "valor3")
```

Ou:

```
meu_array[0]="valor1"
```

```
meu_array[1]="valor2"
```

```
meu_array[2]="valor3"
```

Para acessar um elemento específico de um vetor, utiliza-se o índice entre chaves (`{}`), como por exemplo:

```
echo ${meu_vetor[0]} # Exibe "valor1"
```

```
echo ${meu_vetor[1]} # Exibe "valor2"
```

Para acessar todos os elementos do vetor, utiliza-se `[*]` ou `[@]`:

```
echo ${meu_vetor[@]} # Exibe "valor1 valor2 valor3"
```

Para saber quantos elementos existem em um vetor:

```
echo ${#meu_vetor[@]} # Exibe o tamanho do vetor
```



Para adicionar um novo elemento ao vetor, deve-se atribuir um valor a um novo índice:

```
meu_vetor[3]="valor4"
```

Para adicionar um valor no próximo índice disponível, pode-se utilizar o comando +=:

```
meu_vetor+=("valor5")
```

Para remover um elemento específico do vetor (comando unset com o índice):

```
unset meu_vetor[2] # Remove o terceiro elemento ("valor3")
```

Para remover todos os elementos do vetor:

```
unset meu_vetor
```

Mostrando todos os elementos de um vetor, com um loop:

```
for elemento in "${meu_vetor[@]"; do
    echo "$elemento"
done
```

Mostrando todos os elementos usando os índices do vetor:

```
for i in "${!meu_vetor[@]"; do
    echo "Elemento $i: ${meu_vetor[$i]}"
done
```

## Contador

Abaixo é mostrado um exemplo de contador que imprime os valores 0 a 9 na tela. O comando echo imprime o que estiver entre aspas. Se tiver uma variável, imprime o seu valor. O comando let permite designar valores a variáveis e realizar operações aritméticas e lógicas. No caso abaixo, o let nem seria necessário, mas a banca pode colocar só para confundir!

```
#!/bin/sh
CONTADOR=0
while [ $CONTADOR -lt 10 ]; do
    echo "$CONTADOR";
    let CONTADOR=CONTADOR+1;
done
```

## Comando getopts

O getopts fornece uma maneira padrão de lidar com scripts que aceitam uma variedade de opções personalizadas. Ele permite analisar os argumentos passados para o script de forma fácil



e eficiente. Com o `getopts`, é possível definir opções de linha de comando com argumentos e processá-las de maneira estruturada. Vejamos a sintaxe:

```
getopts optstring varname
```

Sendo que:

- `optstring`: string que define as opções que o script suporta. Se uma letra é seguida por dois pontos (:), isso significa que a opção requer um argumento. Se a letra não for seguida por dois pontos, a opção não requer um argumento;
- `varname`: nome da variável que será usada para armazenar a opção que foi analisada.

Exemplo de uso:

```
#!/bin/bash
```

```
while getopts ":a:b:" opcao; do
```

```
case $opcao in
```

```
    x)
```

```
        echo "Opção -x foi fornecida com o argumento: $OPTARG"
```

```
    ;;
```

```
    y)
```

```
        echo "Opção -y foi fornecida com o argumento: $OPTARG"
```

```
    ;;
```

```
    \?)
```

```
        echo "Opção inválida: -$OPTARG"
```

```
    ;;
```

```
    :)
```

```
        echo "Opção -$OPTARG requer um argumento."
```

```
    ;;
```

```
esac
```

```
done
```

No exemplo acima, o script define duas opções: `-x` e `-y`. A opção `-x` não requer argumentos, enquanto a opção `-y` requer um argumento. O comando `getopts` irá iterar por todos os



argumentos passados para o script e identificar e processar as opções especificadas. Supondo que o script tenha o nome "script.sh", ele pode ser executado da seguinte maneira:

```
./script.sh -x                # Executa a opção -x sem argumento
./script.sh -y ARGUMENTO     # Executa a opção -y com o argumento "ARGUMENTO"
./script.sh -z                # Retorna "Opção inválida: -z"
./script.sh -y                # Retorna "Opção -y requer um argumento."
```

## Comando trap

O comando trap é usado para definir tratamentos personalizados para sinais específicos. Os sinais são eventos enviados para processos no sistema operacional e podem ser gerados por várias ações, como o pressionamento de determinadas teclas (o exemplo mais comum é Ctrl+C) ou quando um processo termina de forma inesperada. A sintaxe é:

```
trap 'comando' lista_de_sinais
```

Sendo que:

- comando: comando ou conjunto de comandos que será executado quando o(s) sinal(is) especificado(s) for(em) recebido(s) pelo script;
- lista\_de\_sinais: lista de números ou nomes simbólicos dos sinais que serão tratados. O comando trap -l pode ser utilizado para listar os nomes simbólicos de sinais disponíveis.

Exemplo:

```
#!/bin/bash
# Tratamento para o sinal SIGINT (Ctrl+C) e SIGTERM
trap 'echo "Sinal SIGINT ou SIGTERM recebido. Até logo..."; exit 1' SIGINT SIGTERM
while true; do
echo "Executando..."          # Loop infinito, imprimindo "Executando..."
sleep 1                        # Pausa de 1 segundo
done
```

No exemplo acima, o script define um tratamento personalizado para os sinais SIGINT (Ctrl+C) e SIGTERM. Quando o usuário pressiona Ctrl+C ou quando um sinal SIGTERM é enviado ao script (quando o processo pai o interrompe, por exemplo), o tratamento definido será executado, exibindo a mensagem "Sinal SIGINT ou SIGTERM recebido. Até logo...", encerrando o script com um código de saída diferente de zero ("exit 1", no caso).



Importante ressaltar que nem todos os sinais podem ser tratados, pois alguns deles possuem comportamentos específicos definidos pelo sistema operacional e podem ser usados para encerrar processos sem a possibilidade de serem tratados ou ignorados.

### Status de saída

O status de saída de um comando executado é um valor entre 0 e 255 que indica:

- 0: execução bem-sucedida!
- Diferente de 0: erro, sendo que cada número indica um erro específico.

Um exemplo de uma implementação simples de status de saída:

```
[ ! -e /bin/bash ] && { exit 2; }
```

O próximo exemplo irá checar se um comando não foi executado com sucesso. Se o retorno foi um valor diferente de 0, então mostra uma mensagem de erro:

```
ping -c1 www.estrategiaconcursos.com.br
```

```
[ $? -ne 0 ] && echo "O comando ping falhou!!!"
```

Abaixo um exemplo mais completo que aplica o comando ls em determinado caminho. Depois testa se retornou 0 (sucesso) ou erro. Em caso de erro, mostra uma mensagem de acordo com o retorno.

```
#!/bin/bash
```

```
ls /home/evandro/gabarito
```

```
if [ $? -eq 0 ]; then
```

```
    echo "O comando ls foi executado com sucesso!"
```

```
else
```

```
    echo "O comando ls falhou!"
```

```
fi
```

Vale lembrar que para um bom entendimento de um shell script, é importante conhecer o funcionamento dos comandos, e não são poucos! Dependendo da banca, vale a pena aprofundar nos estudos de alguns comandos específicos.

### Executando vários comandos em sequência

Para executar um comando de cada vez, é utilizado o caractere ponto e vírgula (;) entre os comandos que serão executados em sequência.

Sintaxe: comando1; comando2; comando3;

Exemplo: ls ; date





No exemplo acima será executado o comando ls e logo após a execução dele, será executado o comando date.

Para executar vários comandos de uma só vez, é utilizado o caractere & entre os comandos que serão executados em sequência.

Sintaxe: comando1 & comando2 & comando3 &

Exemplo: top & ls

No exemplo acima será executado o comando top e o ls em paralelo, sem que o primeiro trave o terminal para execução do segundo.

## Comando sh

É usado para executar shell scripts. A sintaxe básica é:

```
sh script.sh
```

Sendo que script.sh é o nome do script a ser executado. Importante lembrar que o script deve ter permissões de execução.

Se for usado sem especificar um script, um shell interativo é iniciado:

```
$ sh
```

A partir do shell interativo pode-se digitar comandos diretamente. Para sair do shell interativo, pode-se usar o comando exit.

Quando é usada a opção -x, é ativado o modo de depuração, ou seja, o script será executado com saídas verbosas, mostrando cada comando e sua expansão antes de serem executados. Ex.:

```
$ sh -x script.sh
```

Ou, se um comando está sendo executado diretamente no shell:

```
$ sh -x -c "comando_a_ser_executado"
```

Quando o modo de depuração está ativado, cada linha do script é exibida antes de ser executada, o que ajuda a identificar possíveis erros ou entender melhor o fluxo de execução do script.



## LINUX - GERENCIAMENTO DE PROCESSOS

As estruturas internas do kernel (núcleo do sistema operacional) registram informações sobre cada processo, incluindo, entre outras:

- Mapa de espaço de endereços do processo;
- Estado atual (espera, parado, em execução etc.);
- Prioridade de execução;
- Informações sobre recursos;
- Informações sobre arquivos e portas de rede;
- Máscara de sinalização (quais sinais estão bloqueados);
- Proprietário do processo.



O **PID** (Process Id) é o número de identificação de um processo, atribuído na ordem em que são criados. O **PPID** é o PID do processo pai. **Não há chamada de sistema que crie um processo no Linux**, o que acontece é que um **processo existente realiza seu clone - chamada de sistema fork()** - e o processo clone pode trocar o programa por um diferente. O original é o pai e a cópia gerada é o filho.

O **UID** (User Id) é a identificação do usuário que criou e o **GID** (Group Id) é a identificação do grupo. Para definir a prioridade de agendamento (quanto tempo de CPU o processo recebe), existe um valor de **nice** ("gentileza"). Esses valores vão de -20 a 19 (zero é o padrão), sendo -20 a maior prioridade e 19 a menor (Cuidado! É "invertido", assim mesmo!). No momento do clone do processo, o valor é herdado do processo pai.

O comando **nice** define a prioridade de um processo (padrão = 10) antes de executar o programa, enquanto o comando **renice** altera a prioridade de processo já em execução. O proprietário pode aumentar o valor (diminuir a prioridade), mas não pode o contrário! O root pode tudo, claro! Vamos a alguns exemplos:

\$ nice -n 5 ~/bin/teste → Reduz prioridade (eleva nice) por um fator de 5:  $10 + 5 = 15$ ;

\$ sudo renice -6 7943 → Configura o valor de nice como -6;



\$ sudo renice 8 -u evandro → Configura o valor de nice como 8 para processos do usuário evandro.

Como vimos anteriormente, a chamada de sistema `fork()` clona um processo, criando um filho:

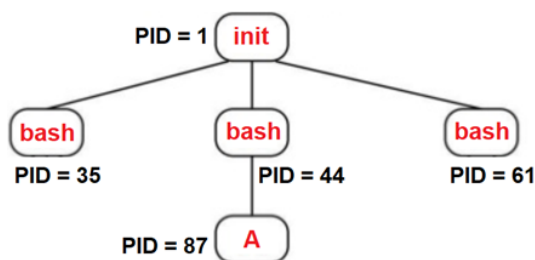
- Retorna 0 para o filho;
- O pai recebe o PID do filho.

Geralmente em seguida o filho executa um novo programa, ou seja, muda o texto do programa e os segmentos de dados e pilha são reiniciados, recebe um PID distinto e suas próprias informações contábeis. Dessa forma cada um sabe o seu papel.

Vários processos são criados na inicialização. O `init` é o mais notável, sendo responsável por:

- Executar os scripts de inicialização;
- Ser o “paizão” dos processos, pois todos são descendentes dele, exceto os criados pelo kernel;
- Chamar `exit_` quando um processo se completa (exit notifica o kernel de que ele está pronto para expirar, fornecendo um código referente ao motivo, 0 = término normal);
- Receber os processos órfãos.

Em relação à **hierarquia de processos**, existe apenas 1 pai e 0 ou mais filhos. Por exemplo, no Linux o processo `init` (PID = 1) é o primeiro a ser executado, logo após o carregamento do Kernel. A função dele é controlar todos os outros processos que são executados no computador. Digamos que a partir dele sejam abertos 3 *shells* (ex.: *bash*) e a partir de um deles seja executado um programa “A”. Abstraindo a existência de outros processos, a hierarquia descrita ficaria assim (PIDs inventados, com exceção do `init`):



**Processo órfão:** processo filho cujo processo pai encerrou antes dele. Quando um processo pai termina antes do processo filho, o processo filho é herdado pelo processo `init` (processo com PID = 1), tornando-se um processo órfão.

**Daemon:** processo em segundo plano, sem a intervenção direta do usuário. *Daemons* são frequentemente iniciados durante o boot do sistema operacional e continuam a ser executados enquanto o sistema está em funcionamento, realizando diversas tarefas e serviços em *background* (segundo plano).

## Comunicação entre Processos



A comunicação entre processos no Linux pode ser realizada de diversas maneiras. Veremos as principais abaixo:

- Pipe (tubo): Um processo escreve em um extremo do pipe e o outro processo lê do outro extremo. Ex.: comando1 | comando2
- Fila de mensagens: Processos podem enviar mensagens para uma fila de mensagens compartilhada;
- Memória compartilhada: Permite que vários processos acessem uma região de memória compartilhada;
- Semáforo: Usado para controlar o acesso a recursos compartilhados entre processos;
- Socket: Comunicação entre processos em diferentes máquinas ou no mesmo sistema. Pode ser usado com TCP ou UDP;
- Sinais: Processos podem enviar sinais uns aos outros para notificar eventos. O comando kill é o mais utilizado (veremos mais detalhes na sequência);
- Named Pipes (FIFO): Funciona como um pipe, mas tem um nome no sistema de arquivos;
- Sockets Unix: Semelhante a sockets normais, mas utiliza um namespace específico do Unix. Usado para comunicação local entre processos.

A escolha de qual maneira utilizar para a comunicação entre processos depende dos requisitos específicos do aplicativo. Vamos focar agora no tipo mais cobrado, os sinais. Os **sinais** podem ser enviados aos processos (interrupções em nível de processo) de algumas formas:

- Comunicação entre processos;
- Extinguir, interromper ou suspender processos (ex. no terminal: CTRL+C ou CTRL+Z);
- Enviados com o comando **kill** (não apenas "mata", pode enviar outros sinais também);
- Enviados pelo *kernel* (alguma infração, ex.: divisão por 0);
- Enviados pelo *kernel* (situação "interessante", ex.: morte de processo filho).

Vamos enfatizar o seguinte: o comando **kill** envia qualquer sinal (não é porque o nome do comando é "matar" que ele serve apenas para isso, porém geralmente é o uso mais frequente).  
Sintaxe:

```
kill [-SINAL] PID
```

O comando **killall** tem o mesmo propósito de kill, mas é especificado o nome do processo (se houver mais de um processo com o mesmo nome, o sinal é enviado a todos):

```
killall [-SINAL] NOME_PROCESSO
```



Alguns dos sinais são (o padrão é TERM):

#	Nome	Descrição
1	HUP	Reiniciar
2	INT	Interromper
3	QUIT	Abandonar
9	KILL	Matar
15	TERM	Término do software

O sinal pode ser especificado pelo nome (ex.: KILL) ou pelo seu código equivalente (ex.: 9). Então, para enviar o sinal KILL para um processo com PID 4321, temos o seguinte:

```
kill -SIGKILL 4321
```

ou

```
kill -9 4321
```

Para enviar o sinal TERM para um processo com nome abcd:

```
killall -SIGTERM abcd
```

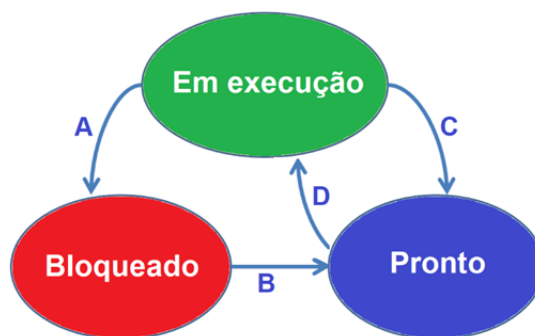
Os **estados dos processos** em sistemas Linux podem ser:

Estado	Significado
Executável	Pode ser executado, ou seja, está pronto para a execução.
Dormente	Aguardando algum recurso (ex.: E/S).



Zumbi	Tentando se destruir.
Parado	Suspensão (sem permissão para ser executado).

*Daemons* e *shells* interativos passam a maior parte do tempo "dormindo" (E/S ou conexões de rede). Vamos ver um desenho bastante utilizado em aulas de sistemas operacionais (não especificamente o Linux, mas pode ser adaptado):



## Estados dos Processos no Linux

Vamos ver os estados mais cobrados em provas de concursos:

- O estado **executável (pronto)** é quando o processo está aguardando para utilizar o processador (CPU);
- O estado **dormente (bloqueado)** é quando estava em execução e fez uma chamada de sistema que não pôde ser completada imediatamente (ex.: ler um arquivo);
- O estado **zumbi** é quando o processo termina a execução, mas ainda não teve o seu status coletado (tentando se destruir);
- O estado **parado** é quando está proibido de executar (sinais STOP ou TSTP, sinal CONT para reiniciar).

## Comandos para Monitoramento de Processos

Para o **monitoramento de processos** existem basicamente três comandos, mas o mais conhecido é o ps. Vejamos os comandos e alguns argumentos já cobrados em provas de concurso:

- **ps**: pode exibir PID, UID, prioridade, terminal de controle de processos, memória consumida, estado atual. Alguns parâmetros são:
  - a: mostra todos os processos existentes;
  - u: exibe o nome do usuário que iniciou determinado processo e a hora que isso ocorreu;





- x: exibe os processos que não estão associados a terminais;
- l: exibe mais campos no resultado;
- e: exibe as variáveis de ambiente relacionadas aos processos;
- f: exibe a árvore de execução dos processos.

→ Visão geral sobre o sistema, bastante utilizado na "vida real" e cobrado em provas: ps aux

- **top**: mostra um resumo dos processos ativos e o uso de recursos, atualizando regularmente (processos mais ativos aparecem no topo, padrão de atualização = 10 segundos);
- **pstree**: exibe uma representação gráfica da hierarquia de processos em execução no sistema. Mostra os processos e seus relacionamentos em forma de árvore, facilitando a visualização da estrutura do sistema;
- **jobs**: mostra os processos que estão parados ou rodando em segundo plano.

Em relação ao comando top, são exibidas colunas de informações relacionadas aos processos e à utilização de recursos do sistema. Algumas das colunas mais comuns são:

- PID (Process ID): Identificador único atribuído a cada processo em execução;
- USER: Nome do usuário associado ao processo;
- PR (Priority): Prioridade do processo;
- NI (Nice value): Valor "nice" atribuído ao processo (prioridade);
- VIRT (Virtual Memory): Quantidade total de memória virtual utilizada pelo processo;
- RES (Resident Memory): Quantidade de memória física (RAM) usada pelo processo;
- SHR (Shared Memory): Quantidade de memória compartilhada utilizada pelo processo;
- S (%CPU): Percentual de uso do processador (CPU) pelo processo;
- MEM (%MEM): Percentual de uso da memória pelo processo;
- TIME+: Tempo total de processador (CPU) usado pelo processo desde o início;
- COMMAND: Nome do comando ou programa associado ao processo.

Processos em segundo plano são iniciados usando o símbolo "&" no final da linha de comando ou através do comando bg:

```
$ ./script.sh &
```

```
$ bg ./script.sh
```



Quando um comando é executado em segundo plano, o terminal fica liberado para que outros comandos sejam digitados.



## MONTAGEM DE VOLUMES, PERMISSÕES, SUPERUSUÁRIO

Todos os arquivos acessíveis em um sistema UNIX são arranjados em uma grande árvore, com a raiz "/". Tais arquivos podem estar espalhados em diversos dispositivos (HDs, SSDs, pen drives, DVDs etc.). O comando `mount` serve para anexar o sistema de arquivos encontrado no dispositivo (ou uma partição) à grande árvore de arquivos. Algumas distribuições montam automaticamente e o ponto de montagem padrão é o diretório `/mnt`.

O arquivo `/etc/fstab` possui informações sobre os sistemas de arquivos que o sistema pode montar, e o arquivo `/etc/mtab` possui uma lista dos dispositivos montados (através do comando `mount`).

### Comandos para Montagem e Desmontagem

O comando `mount`, sem parâmetros, mostra quais sistemas de arquivos estão montados:

```
[root@localhost ~]# mount
root on / type 9p (rw,relatime,dirsync,access=client,trans=virtio)
devtmpfs on /dev type devtmpfs (rw,relatime,size=125948k,nr_inodes=31487,mode=755)
none on /proc type proc (rw,relatime)
none on /sys type sysfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600,ptmxmode=000)
[root@localhost ~]#
```

A forma básica do comando `mount` possui dois parâmetros:

- Dispositivo contendo o sistema de arquivos a ser montado (ex.: `/dev/sda3`);
- Ponto de montagem (ex.: `/teste`).

Assim, temos, por exemplo:

```
[root@estrategia ~]# mount /dev/sda3 /teste
```

Obs.: `/teste` deve existir antes de ser aplicado o comando!

Geralmente o tipo de sistema de arquivos é detectado automaticamente, mas pode-se especificar o tipo com o parâmetro `-t`. Por exemplo, sistema de arquivos EXT4:

```
[root@estrategia ~]# mount -t ext4 /dev/sda3 /teste
```

Alguns exemplos de sistemas de arquivos suportados são `cifs`, `ext2`, `ext3`, `ext4`, `hfs`, `iso9660`, `nfs`, `reiserfs`, `vfat`, `xf`s, entre outros.

O parâmetro `-o` possibilita definir opções, separadas por vírgulas, como por exemplo:

- `ro`: somente leitura;
- `rw`: leitura e escrita;
- `exec`: permite a execução de binários;
- `noexec`: não permite a execução de binários;



- remount: tenta remontar um sistema de arquivos já montado.

Exemplo:

```
mount -o remount,rw,noexec /dev/sda3 /teste
```

O parâmetro **-a** monta todos os sistemas de arquivos listados no arquivo `fstab` (exceto as linhas com a palavra-chave `noauto`). Importante ressaltar que os programas leem do `fstab`, porém não escrevem nele. O parâmetro `-a` pode ser utilizado para montar "tudo" sem realizar um boot.

O comando **umount** retira o sistema de arquivos da árvore hierárquica de arquivos. Não há a desmontagem caso o sistema de arquivos esteja ocupado (com arquivos abertos, swap em uso etc.):

```
umount /teste
```

## Usuários, Grupos, Permissões de Acesso

Um novo usuário pode ser adicionado utilizando-se ferramentas, como **adduser** ou **useradd**. A partir desse momento o usuário criado é adicionado nos arquivos `/etc/passwd` e `/etc/shadow`. Os arquivos recebem um número **UID referente à identificação do usuário (dono)**, **GID referente ao seu grupo**, entre outros dados.

Os números UID e GID variam de 0 a 65535 (dependendo do sistema, o valor limite pode ser maior). No caso do **usuário root, esses valores são sempre 0 (zero)**. Assim, para fazer com que um usuário tenha os mesmos privilégios que o root, é necessário que seu UID e GID sejam iguais a 0. Isso informa ao sistema que o usuário em questão é um superusuário.

Cada arquivo/diretório no Linux possui um conjunto de **permissões de acesso** e propriedades em **três níveis**: usuário dono do arquivo (**U**), grupo proprietário (**G**) e qualquer outro usuário que não se encaixe nos níveis anteriores (**O**).

Os **tipos de permissão** são:

- Read (**r**) - Leitura: permissão para visualizar o conteúdo do arquivo;
- Write (**w**) - Escrita: permissão para alterar o conteúdo do arquivo;
- eXecute (**x**) - Execução: permissão para executar o arquivo.

Dessa forma definimos a seguinte ordem:

U			G			O		
r	w	x	r	w	x	r	w	x

Existem também algumas **permissões especiais**:



- **setuid**: privilégio do dono e não de quem executou, ex.: passwd executado pelo evandro  
□ privilégio de root, mas quem executou foi o usuário evandro!
- **setgid**: tem efeito tanto em arquivos como em diretórios. Para um arquivo, executa com privilégios do grupo do usuário dono. Para um diretório, o grupo dos arquivos criados dentro do diretório será o mesmo do diretório pai. Abaixo podemos notar um "s" no lugar do "x" do grupo. Isso demonstra que o setgid está com valor 1.

```
$ ls -ld teste  
drwxrwsr-x. 2 egdoc egdoc 4096 Nov 1 17:25 teste
```

- **sticky bit**: quando usado em um diretório, todos os arquivos dentro dele poderão ser modificados somente por seus donos. Exemplo típico: "/tmp", um diretório temporário onde todos podem colocar arquivos, mas cada um pode modificar o seu. Abaixo podemos notar um "t" no lugar do "x" dos outros. Isso demonstra que o sticky bit está com valor 1.

```
$ ls -ld /tmp  
drwxrwxrwt. 14 root root 300 Nov 1 16:48 /tmp
```

Vamos rever nossa figura do "UGO" com as permissões especiais:

			U			G			O		
uid	gid	sticky	r	w	x	r	w	x	r	w	x

Analisando apenas as permissões especiais, se o setuid estiver ativo temos 100 = 4 (em decimal). Se tivermos o setgid ativo, temos 010 = 2 e se o sticky bit estiver ativo, temos 001 = 1.

Para **visualizar as permissões** é comum utilizarmos o argumento "-l" no comando "ls", pois esse argumento lista arquivos com informações adicionais, incluindo colunas com as permissões, nome do usuário e grupo do dono:

```
[root@localhost etc]# ls -l teste  
-rw-r--r-- 1 root root 6 Jul 22 18:43 teste  
[root@localhost etc]#
```

Na figura acima podemos ver que se trata de um arquivo (o primeiro caractere é "-" = arquivo; se fosse diretório, seria "d"). Depois vemos as permissões do **Usuário** (rw-), **Grupo** (r--) e **Outros** (r--). Não vemos permissões especiais.



## Modificando as Permissões

Ok, e se eu quiser **modificar as permissões** depois de ter criado um arquivo? Aí temos o comando **chmod**. Vamos ver algumas formas de utilizá-lo:

`chmod u=rwx, g=rx, o=x nome_arquivo` ☑ modifica as permissões do "nome\_arquivo" para:

- usuário = completo (*read, write, execute*);
- grupo = leitura e execução (*read, execute*);
- outros = apenas execução (*execute*).

`chmod 0751 /home/evandro/arquivo.txt` ☑ aqui temos um exemplo utilizando números em octal e temos a presença das permissões especiais (o primeiro número). Como o primeiro número é 0, fica fácil, pois em binário ficaria 000 (setuid = 0, setgid = 0, sticky bit = 0). Agora vamos ao UGO:

- Usuário = 7 = 111 = completo (*read, write, execute*);
- Grupo = 5 = 101 = leitura e execução (*read, execute*);
- Outros = 1 = 001 = apenas execução (*execute*).

Acabamos de ver que os comandos acima são equivalentes em termos das permissões concedidas!

Um último exemplo:

`chmod +x teste` ☑ dá a permissão de execução ao arquivo "teste", mas para quem? U? G? O? Como não está explícito para quem é, todos recebem o "+x". Veja abaixo antes e depois de aplicar o "`chmod +x teste`". Tem um "`ls -l teste`" antes e um depois para podermos verificar que realmente tanto o usuário, como o grupo e os outros receberam a permissão de execução (x).

```
[root@localhost etc]# ls -l teste
-rw-r--r-- 1 root root 6 Jul 22 18:43 teste
[root@localhost etc]# chmod +x teste
[root@localhost etc]# ls -l teste
-rwxr-xr-x 1 root root 6 Jul 22 18:43 teste
[root@localhost etc]#
```

## Definição de Permissões Padrão

A permissão máxima para a criação de:

- um diretório é: 777 (rwx rwx rwx);
- um arquivo é: 666 (rw- rw- rw-);

O Linux deixa a permissão de execução (x) por conta do administrador do sistema, para que seja aplicada a script, programas, ou serviços (*daemons*). Por isso a permissão máxima para arquivo é definida como "666".

**umask**: utilizado para definir as permissões padrão utilizadas para os novos arquivos e diretórios a serem criados. A máscara padrão é:



- Arquivo: 666;
- Diretório: 777.

Um exemplo é mostrado abaixo com uma máscara "u=rwx,g=rwx,o=", ou seja, permissões completas para U e para G e sem permissão nenhuma para O. Na sequência é criado o diretório "teste" e o arquivo "arquivo". Note que o arquivo, ao ser criado, não recebeu a permissão "x". Como vimos antes, a permissão de execução para arquivos deve ser dada explicitamente, depois.

```
[root@localhost diretorio]# umask u=rwx,g=rwx,o=
[root@localhost diretorio]# mkdir teste
[root@localhost diretorio]# touch arquivo
[root@localhost diretorio]# ls -l
total 4
-rw-rw---- 1 root root  0 Jul 22 20:39 arquivo
drwxrwx--- 2 root root 37 Jul 22 20:39 teste
[root@localhost diretorio]#
```

Uma outra forma de aplicação do umask é através de números octais (mais cobrado em provas de concurso). Funciona da seguinte forma: pega-se 777 e diminui pela máscara definida, no caso de diretórios, ou 666 diminui pela máscara definida, no caso de arquivos. Por exemplo, se a máscara definida for "447", então teremos diretórios criados com as permissões 330 (-wx-wx---) e arquivos com permissões 220 (-w--w----). Obs.: Se ficar negativo, considere 0, como é o caso de 666 - 447, ficaria 2 2 -1, então consideramos 2 2 0). Fica mais fácil analisar na figura abaixo:

```
[root@localhost diretorio]# umask 447
[root@localhost diretorio]# mkdir diretorio
[root@localhost diretorio]# touch arquivo
[root@localhost diretorio]# ls -l
total 4
--w--w---- 1 root root  0 Jul 22 20:43 arquivo
d-wx-wx--- 2 root root 37 Jul 22 20:43 diretorio
[root@localhost diretorio]#
```

## Permissões com ACL (Access Control List)

Existem comandos para lidar com permissões com ACL (Access Control List). São pouco cobrados em provas de concurso, mas como já apareceram, vale a pena dar uma olhada.

**setfacl:** define listas de controle de acesso (ACLs) de arquivos e diretórios. Na linha de comando, uma sequência de comandos é seguida por uma sequência de arquivos (que por sua vez pode ser seguida por outra sequência de comandos, e assim por diante). As opções -m (-modify) e -x (-remove) esperam uma ACL na linha de comandos. A diretiva "u" define o usuário, "g" define o grupo. "R" significa "recursivo". Deve haver o privilégio de root!

Mas qual a diferença de usar ACL? Com ACL é possível liberar algum tipo de permissão apenas para um determinado usuário, por exemplo, algo que não seria possível com o chmod! Vamos ver alguns exemplos a seguir.

```
$ sudo setfacl -m u:evandro:rwx aulas
```





- Concede ao usuário "evandro" acesso de leitura, escrita e execução no diretório "aulas".

```
# setfacl -R -m g:prof:rwx aulas
```

- Concede permissão de leitura, escrita e execução para o grupo prof.

**getfacl**: verifica a ACL de arquivos/diretórios. Ex.:

```
$ getfacl lista.txt
```

```
# file: lista.txt
```

```
# owner: root
```

```
# group: root
```

```
user::rwx
```

```
user:evandro:rwx
```

```
group::r-x
```

```
mask::rwx
```

```
other::r-x
```

Para verificar se um arquivo/diretório possui uma ACL:

```
$ ls -l lista.txt
```

```
-rw-rwxr--+ 1 root root 0 Jan 8 18:44 lista.txt
```

- O sinal de "+" em "-rw-rwxr--+" indica que o arquivo possui uma lista de acesso (ACL) definida.

```
$ ls -ld aulas
```

```
drwxrwxr-x+ 2 root root 4096 Jan 8 18:31 aulas
```

- Idem para diretórios!

## Modificando o Dono e/ou o Grupo

Agora vamos ver como modificar o dono ou o grupo.

Embora o comando **chown** signifique *change owner*, ele permite a alteração do dono ou do grupo do arquivo/diretório. A opção "-R" aplica a arquivos/diretórios recursivamente e "-c" mostra o resultado. Vamos ver uns exemplos a seguir (note que o prompt muda para "#", o que



indica que está executando como *root*, afinal tem que ser *root* para mudar um arquivo de dono ou de grupo).

```
# chown -R evandro /teste
```

☒ “evandro” virou dono da pasta /teste, de forma recursiva.

```
# chown evandro:inf /teste
```

☒ “/teste” passou a ter o dono “evandro” e o grupo “inf” (mudou os dois de uma vez!).

Por fim, um comando que altera apenas o grupo de um arquivo/diretório, o **chgrp**:

```
#chgrp -R inf /teste
```

☒ Recursivamente, muda o dono de grupo do diretório “/teste” para “inf”.

## Superusuário

O superusuário é aquele que pode fazer tudo no sistema! Pode executar qualquer operação válida em qualquer arquivo ou processo. Algumas chamadas de sistema (*syscalls*) só podem ser executadas pelo *root* (nome padrão para o superusuário, mas pode ter outros).

O **root** é uma conta com **UID=0**, sendo possível modificar o nome de usuário dessa conta ou criar contas com UID=0 (ações não recomendadas por motivo de segurança, porém possíveis).

Alguns exemplos de operações restritas, que somente o *root* pode realizar:

- Modificar o diretório raiz;
- Criar arquivos de dispositivo;
- Configurar o relógio;
- Aumentar limites de uso de recursos e as prioridades de processos;
- Definir o nome do host;
- Configurar interfaces de rede;
- Abrir portas de rede privilegiadas (< 1024);
- Desligar o sistema.

Por padrão, o diretório home do root é **/root** e **/sbin** é o diretório que armazena os binários de sistema importantes que são utilizados usados pelo *root*.

Quando um usuário “comum” utiliza o shell, aparece **\$** no prompt, enquanto para o **root** aparece **#**. Isso é importante para verificar em uma questão na prova, para saber se quem está executando é um usuário comum ou o root! Abaixo podemos ver que o root está utilizando o shell. Claro que está configurado para aparecer o nome do usuário à esquerda, mas mesmo que não estivesse, poderíamos ver o caractere “#”, que indica ser o root.



```
root@evandro-HP: ~/DHAV2
Arquivo Editar Ver Pesquisar Terminal Ajuda
Gerando arquivo: /media/evandro/CASOS-1TB/saida-imagem012/canal4/dir0/arc20180505175052_000000011263.h264 de tamanho: 525338
Gerando arquivo: /media/evandro/CASOS-1TB/saida-imagem012/canal7/dir0/arc20180505175044_000000011264.h264 de tamanho: 393507
Gerando arquivo: /media/evandro/CASOS-1TB/saida-imagem012/canal7/dir0/arc20180505175046_000000011265.h264 de tamanho: 92548
Processou 11266 arquivos
root@evandro-HP:~/DHAV2#
```

## Comandos

Para um usuário comum se tornar root, pode-se utilizar o comando `su` (*substitute user*). Note que o comando `su` serve para "substituir" um usuário e não precisa ser necessariamente o root! Mas na prática, geralmente se utiliza para se tornar o root, mesmo.

O comando `su` sem argumentos solicita a senha de `root` e inicia um shell de `root`. Não registra nenhum comando executado como root, mas cria log de quem e quando se tornou root, o que é útil para a auditoria de alguma coisa errada realizada no período. Para substituir por outras identidades, a sintaxe é: `su nomeUsuario` pede a senha do nomeUsuario. É útil para realizar testes em nome do nomeUsuario.

Se duas ou mais pessoas souberem a senha do root, pode acontecer uma situação em que alguém faz o login como root, "faz coisas que não deveria" e ninguém saberá quem foi. Por questões de segurança há a possibilidade de desabilitar o `login` de `root`.

O comando `sudo` permite a usuários comuns obter privilégios de outro usuário, em geral o root, para executar tarefas específicas dentro do sistema de maneira segura e controlável pelo administrador. **Ao ser executado, há uma consulta ao arquivo `/etc/sudoers`**, que possui os usuários e comandos habilitados em cada host. É solicitada a senha do próprio usuário que executou o `sudo` e, na sequência, é executado o comando que se encontra após o `sudo`. É importante salientar que é mantido um log dos comandos executados, hosts, usuários, diretório, data/hora etc. O exemplo mais comum é sem a especificação do usuário:

```
$ sudo apt-get install abc
```

No exemplo acima, como não é especificado um usuário, então o "apt-get install abc" será executado como root, após o usuário ter digitado a sua própria senha. Pode haver configuração de tempo sem ter que digitar a senha novamente (até 5 minutos). Vamos ver mais dois exemplos:

```
$ sudo -u evandro /home/evandro/prog executa o "prog" em nome de "evandro".
```

```
$ sudo -g prof /home/teste/prog seta o grupo primário para "prof" no momento da execução de "prog".
```



## COMANDOS DIVERSOS

Esta aula foi montada após um apanhado de milhares de questões sobre Linux em provas de concurso. Claro que podem surgir novas questões com outros comandos, mas é inviável colocar todos que existem aqui, ficaria um livro!

Comandos relacionados a gerenciamento de processos, superusuário, entre outros, são descritos nessas aulas específicas. O que sobrou veio para esta aula que veremos a seguir. Primeiro vamos focar em comandos utilizados para [criar/excluir](#), [copiar/mover arquivos e diretórios](#), [trocar o diretório corrente](#) e [verificar o diretório corrente](#):

- **mkdir**: significa "make directory" (cria um diretório), ex.: `mkdir /home/evandro/teste` cria o diretório "teste" dentro de /home/evandro (se não houver esse caminho, ocorre um erro);
- **cd**: significa "change directory" (muda o diretório corrente), exemplos:
  - o `cd ..` (muda para o diretório pai);
  - o `cd /` (muda para o diretório raiz);
  - o `cd ../teste` (sobe um nível e "entra" no diretório teste);
  - o `cd teste` ("entra" no diretório teste, que deve estar abaixo do diretório atual);
- **rmdir**: remove um diretório vazio, ex.: `rmdir /root/teste`;

- **dirname**: extrai o nome do diretório de um caminho de arquivo especificado

```
$ dirname /home/fulano/documentos/arquivo.txt → mostra a saída "/home/fulano/documentos";
```

- **pwd** (print working directory): imprime o caminho completo do diretório de trabalho corrente;
- **touch**: cria um arquivo vazio, e também atualiza as informações de data e hora de um arquivo existente, ex.: `touch nome_arquivo`;
- **rm**: remove um arquivo, ex.: `rm nome_arquivo`. O parâmetro `-f` é frequentemente usado para forçar a remoção sem solicitar confirmação, ignorando possíveis mensagens de aviso.
- **cp**: copia arquivos ou diretórios para outro local. Se o destino fornecido for o nome de um diretório existente, os arquivos serão copiados para dentro desse diretório, com o mesmo nome. Caso seja indicado um nome de arquivo no caminho de destino, o arquivo de origem será copiado e essa cópia renomeada também;

Sintaxe: `$ cp [opções] arquivos_origem local_destino`

- **mv**: pode ser usado para mover ou renomear arquivos. Funciona assim: renomeia ORIGEM para DESTINO, ou move ORIGEM(NS) para o DIRETÓRIO DESTINO;



Sintaxe: \$ mv [opções] arquivos\_origem local\_destino

- **echo**: mostra um texto na tela. Ex.:

\$ echo oi → mostra "oi" na saída padrão, geralmente a tela do computador. O parâmetro -n é usado para imprimir uma linha de texto sem adicionar uma nova linha ao final, ou seja, é para suprimir a quebra de linha automática que o comando echo geralmente adiciona.

Pode criar um arquivo de texto, se utilizado o redirecionamento de saída ">" ou ">>", exemplos:

- o echo oi > arquivo.txt □ cria um arquivo com o conteúdo "oi";
- o echo oi >> arquivo.txt □ adiciona no fim do arquivo.txt o conteúdo "oi" (se não existir, cria o arquivo);
- o echo oi 2> erro.txt □ "2>" utilizado para redirecionar a saída de erro padrão para um arquivo ou outro local específico. Dificilmente um comando echo gera erro, mas se gerar, o erro será escrito no arquivo erro.txt (exemplo). Neste caso, "oi" será mostrado na tela e o eventual erro será gravado em erro.txt.

Vamos ver uma sequência desses comandos a seguir, para um melhor entendimento.

```
[root@localhost ~]# mkdir teste
[root@localhost ~]# cd teste
[root@localhost teste]# pwd
/root/teste
[root@localhost teste]# mkdir opa
[root@localhost teste]# ls
opa
[root@localhost teste]# rmdir opa
[root@localhost teste]# ls
[root@localhost teste]# touch arquivo_novo
[root@localhost teste]# ls -las
total 8
 4 drwxr-xr-x  2 root  root    66 Nov 19  2019 .
 4 drwx----- 4 root  root   176 Oct 24  2017 ..
 0 -rw-r--r--  1 root  root     0 Nov 19  2019 arquivo_novo
[root@localhost teste]# rm arquivo_novo
[root@localhost teste]# ls -las
total 8
 4 drwxr-xr-x  2 root  root    37 Nov 19  2019 .
 4 drwx----- 4 root  root   176 Oct 24  2017 ..
[root@localhost teste]# cd /
[root@localhost ~]# pwd
/
[root@localhost ~]#
```

Monitoramento/Diagnóstico



Existe uma infinidade de comandos, ferramentas ou aplicativos para o monitoramento/diagnóstico em sistemas Linux. Vamos focar em alguns, de acordo com a cobrança em provas de concurso. Como existem diversos argumentos para cada comando, vamos focar naqueles que já foram alvo de questões.

**ls:** lista o conteúdo de um diretório. Os parâmetros mais cobrados são:

- -l (long): utiliza um formato de listagem longa;
- -a (all): todos os arquivos, não ignora as entradas que começam com ".";
- -s (size): imprime o tamanho de cada arquivo.

A combinação mais utilizada é o uso dos três parâmetros juntos! Assim:

```
$ ls -las
```

**cat:** concatena arquivos e mostra na saída padrão. Sintaxe: cat ARQ1 ARQ2. Mas se for utilizado o direcionador "simples" (>), coloca o conteúdo do primeiro arquivo sobre o conteúdo do segundo e não mostra na tela. Se for utilizado o direcionador "duplo" (>>), concatena o primeiro ao segundo e não mostra na tela (se não existir o segundo, este é criado). Quando usado com apenas um arquivo, mostra seu conteúdo. Abaixo um exemplo com o direcionador duplo e depois somente com um arquivo (importante saber que o arquivo "oi.txt" tem o conteúdo "oi" e que "tchau.txt" tem o conteúdo "tchau").

```
[root@localhost teste]# cat oi.txt >> tchau.txt
[root@localhost teste]# cat tchau.txt
tchau
oi
[root@localhost teste]#
```

Se o usuário quiser ver caracteres "não imprimíveis", pode utilizar alguns parâmetros:

- -v, --show-nonprinting: utiliza "^" e "M-" para mostrar caracteres não imprimíveis, exceto para Line Feed (quebra de linha) e TAB;
- -E, --show-ends: mostra "\$" no fim de cada linha;
- -T, --show-tabs: mostra TAB como "^I"
- -A, --show-all: equivalente aos três anteriores juntos (-vET).

**tail:** a tradução literal é "cauda" (cauda de um animal, mesmo), então pense no rabo de um cachorro, é o que fica por último, certo? Esse é o propósito do comando, mostrar a última parte de um arquivo. Então, se um arquivo é pequeno, tanto faz usar o "cat" ou o "tail", mas se for grande, depende o que o usuário quer ver! Obs.: por padrão, o comando tail mostra as 10 últimas linhas do arquivo. A opção "-n NUMERO" define uma quantidade das N últimas linhas a serem mostradas.

Abaixo podemos ver uma sequência desses comandos. Primeiro, o pwd mostra que o diretório corrente é o /root/teste. O comando ls com os parâmetros -las (os 3 parâmetros combinados: l, a, s) mostra três arquivos .txt, com tamanhos 4, 6 e 150 bytes. Nos arquivos menores, com certeza é



possível ver todo o conteúdo com o comando cat. O de 150 bytes seria possível se não houvesse muitas quebras de linha (mas eu criei um arquivo com os números 01 a 50 com quebra de linha, então o melhor seria usar o tail, caso se deseje ver apenas o final do arquivo). Então, podemos ver todo o conteúdo de tchau.txt e apenas o final do arquivo numeros.txt.

```
[root@localhost teste]# pwd
/root/teste
[root@localhost teste]# ls -las
total 20
 4 drwxr-xr-x  2 root  root    115 Jan  8 19:21 .
 4 drwx----- 4 root  root    157 Oct 24  2017 ..
 4 -rw-r--r--  1 root  root    150 Jan  8 19:29 numeros.txt
 4 -rw-r--r--  1 root  root     4 Jan  8 19:21 ola.txt
 4 -rw-r--r--  1 root  root     6 Jan  8 19:21 tchau.txt
[root@localhost teste]# cat tchau.txt
tchau
[root@localhost teste]# tail numeros.txt
41
42
43
44
45
46
47
48
49
50
[root@localhost teste]#
```

**head:** mostra as primeiras 10 linhas de um arquivo na saída padrão (o oposto do comando tail). A opção -n, ou --lines=NUM mostra as primeiras NUM linhas, ao invés das primeiras 10.

**find:** procura por arquivos em uma hierarquia de diretórios. Vamos ver algumas maneiras de utilizar o find, com os devidos parâmetros, a seguir.

- Listar todos os arquivos localizados em um determinado diretório, incluindo os arquivos armazenados nos subdiretórios (utilizar um "."):

```
$ find .
```

- Definir o nível de "profundidade" na navegação dos subdiretórios (maxdepth). Se quiser localizar apenas os arquivos em um diretório, sem os subdiretórios, utiliza-se profundidade 1, por exemplo:

\$ find ./teste -maxdepth 1 -name \*.php □ traz como resultado os arquivos PHP dentro do diretório ./teste (obs.: "-name" é *case sensitive* e "-iname" é *case insensitive*);

- Tirar dos resultados alguma coisa, por exemplo, localizar arquivos denominados "comando\*", de forma que o "comando.php" não apareça (utilizar a negação, com o símbolo de exclamação):

```
$ find ./teste -name 'comando*' !-name '*.php'
```

- Utilizar o OR para estabelecer mais critérios (parâmetro -o), como por exemplo, arquivos JPG e TXT:

```
$ find -name '*.jpg' -o -name '*.txt'
```





- Buscar somente arquivos (-type f):

\$ find . -type f □ busca apenas os arquivos no diretório corrente (.), conforme mostrado abaixo:

```
[root@localhost teste]# ls
numeros.txt  oi.txt      opa          tchau.txt
[root@localhost teste]# find . -type f
./oi.txt
./tchau.txt
./numeros.txt
[root@localhost teste]# █
```

- Buscar somente diretórios (-type d):

```
$ find ./teste -type d -name "exemplo*"
```

- Buscar arquivos com determinadas permissões:

\$ find . -type f -perm 0740 □ busca os arquivos com permissão 7 (rwx), 4 (r--), 0 (---).

- Buscar arquivos a partir de um determinado proprietário (owner) ou grupo:

```
$ find . -user evandro
```

ou

```
# find /var/www -group adm
```

- Buscar arquivos e diretórios com base em períodos de modificação:

- o # find / -mtime 5 □ arquivos modificados há 5 dias

- o # find / -atime 5 □ arquivos acessados há 5 dias

- o # find / -mtime +10 -mtime -60 □ arquivos modificados entre 10 e 60 dias atrás

- Buscar arquivos baseado no tamanho (-size):

```
# find / -size +10M □ procura arquivos maiores que 10 MB.
```



**grep**: mostra as linhas de um arquivo que “batem” (*match*) com um determinado padrão. Um exemplo comum é utilizar o **cat** para “visualizar” o arquivo e a saída dele passar como entrada para o **grep**, através do pipe (**|**). Ao lado podemos ver um exemplo em que é aplicado o **cat** no arquivo **numeros.txt** e em cima dessa saída (visualização do conteúdo de **numeros.txt**) é aplicado o **grep 3**, ou seja, mostra todas as linhas do arquivo que possuem o número 3.

```
[root@localhost teste]# cat numeros.txt | grep 3
03
13
23
30
31
32
33
34
35
36
37
38
39
43
[root@localhost teste]#
```

Uma opção que pode causar confusão é a “**-v**”, que inverte a correspondência, ou seja, ela seleciona todas as linhas que não contêm o padrão especificado. Cuidado!!!

```
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
--More-- (88% of 150 bytes) [root@localhost teste]#
```

**more**: faz com que um conteúdo de texto seja mostrado uma tela por vez. Se aplicarmos “**more numeros.txt**”, teremos as duas telas a seguir, sendo necessário pressionar o espaço para pular para a tela seguinte, ou o **ENTER** para “andar” de linha em linha:

**df** (“disk free”) mostra a quantidade de espaço em disco disponível no sistema de arquivos contendo cada argumento de nome de arquivo. Se nenhum nome de arquivo for fornecido, o espaço disponível em todos os sistemas de arquivos montados é mostrado.

**du** (“disk usage”): mostra o espaço ocupado por diretórios e arquivos no sistema de arquivos. Por exemplo, para verificar o espaço ocupado por um diretório:

```
du -h /caminho_do_diretorio
```

O argumento **-h** torna a saída mais legível para humanos, mostrando os tamanhos em formato legível, como KB, MB ou GB.

**free**: mostra a quantidade de memória utilizada e livre no sistema.



**which** ("qual"): mostra o caminho completo de determinado comando. Abaixo vemos, por exemplo, o caminho completo dos comandos mount, ls e cat.

```
[root@localhost ~]# which mount
/bin/mount
[root@localhost ~]# which ls
/bin/ls
[root@localhost ~]# which cat
/bin/cat
[root@localhost ~]# █
```

**wc**: significa "word count", mas além de "contar palavras", conta também as linhas e os caracteres em um arquivo. As opções mais frequentes são:

- -c: conta o número de caracteres de um ou mais arquivos;
- -l: conta o número de linhas de um ou mais arquivos;
- -L: conta o número de caracteres da maior linha do arquivo;
- -w: conta as palavras de um ou mais arquivos.

**!!**: reexecução do último comando. Não precisa nem saber qual foi o último comando executado, se digitar "!!" executa o último!

**sed**: ferramenta que ajuda a analisar e transformar textos. Existem diversas opções, mas vamos ver apenas alguns exemplos de uso abaixo.

- s/regex/substituicao/: procura expressões REGEX e as substitui:

\$ sed 's/bonito/feio/' entrada.txt > saida.txt → substitui toda palavra "bonito" para "feio" no arquivo entrada.txt, gerando o resultado no arquivo saida.txt.

- d (delete): exclui linha(s) de um arquivo. Exemplo de exclusão da 1ª linha:

```
$ sed '1d' texto.txt
```

- w (write): escreve linha(s) de um arquivo. Exemplo de leitura da 2ª linha e escrita em texto2.txt:

```
$ sed '2~2 w texto2.txt' texto.txt
```

**awk**: linguagem interpretada que permite a manipulação de textos a partir de uma sequência de padrões. Alguns parâmetros:

- -f : especifica o nome do arquivo com os comandos a serem executados;
- -F : define quem é o separador de campos (o padrão é o espaço);
- \$0 é a linha inteira, \$1 campo 1, \$2 campo 2, ...

\$ ps aux | awk '\$8=="Z" { print \$0 }' → pega a saída do ps e "joga" para o awk. Se o campo 8 (estado do processo) for Z (zumbi), imprime a linha inteira.



**cut:** é uma ferramenta de linha de comando utilizada para extrair ("cortar") seções de texto de cada linha de um arquivo ou entrada padrão. É frequentemente usada para manipular dados delimitados, como arquivos CSV, logs, ou outras saídas de texto organizadas em colunas. A sintaxe básica é:

```
cut [opções] [arquivo]
```

As opções mais comuns são:

- -b: Seleciona um intervalo de bytes específicos de cada linha;  
cut -b 1-8 arquivo.txt # Extrai os bytes de 1 a 8 de cada linha
- -c: Seleciona um intervalo de caracteres específicos de cada linha;  
cut -c 1-5 arquivo.txt # Extrai os primeiros 5 caracteres de cada linha
- -d: Define o delimitador de campo, usado em conjunto com -f (padrão: tabulação);  
cut -d ',' -f 1 arquivo.csv # Extrai o primeiro campo de cada linha, usando vírgula como delimitador
- -f: Seleciona campos específicos em cada linha. Funciona em conjunto com -d.  
cut -d ':' -f 2,4 /etc/passwd # Extrai o segundo e quarto campos do arquivo /etc/passwd

**tar:** utilizado para compactação de arquivos. Gera um arquivo ".tar". Para maior compactação é utilizado o gzip, gerando um arquivo .tar.gz. Vamos a alguns parâmetros:

- c: cria um novo arquivo .tar;
- v: mostra uma descrição do progresso de compactação;
- f: nome do arquivo;
- z: compressão gzip;
- x: descompacta um arquivo;
- C: realiza a extração para um diretório diferente do atual.

```
$ tar -cvzf arquivo.tgz /home/evandro → Compacta usando o gzip, mostrando o processo.
```

```
$ tar -xvf compactado.tar -C /home/evandro/ → descompactação para um diretório diferente do atual, mostrando o processo.
```

```
$ tar -xvf compactado.tar.gz → descompactação, mostrando o processo.
```

**dd:** ferramenta poderosa usada para copiar e converter dados. Frequentemente utilizada para operações de cópia de blocos brutos de dados (raw copy), como criar imagens de disco, copiar partições ou zerar dispositivos. A cópia realizada é bit-a-bit, portanto a cópia gerada possuirá o mesmo tamanho da origem.

```
$ dd if=/dev/sdb of=/mnt/destino → faz uma cópia bruta (raw) de um dispositivo (mídia) para o arquivo "destino", em /mnt.
```



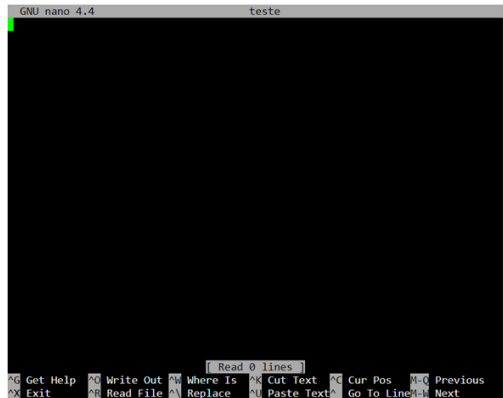
**systemctl**: ferramenta de controle para o Systemd, um sistema de inicialização e gerenciamento de serviços amplamente utilizado em sistemas operacionais Linux modernos. Com o Systemd é possível iniciar, parar e gerenciar serviços e processos no sistema. Exemplos:

- \$ sudo systemctl start NOME\_DO\_SERVICO → inicia um serviço;
- \$ sudo systemctl stop NOME\_DO\_SERVICO → para um serviço;
- \$ sudo systemctl restart NOME\_DO\_SERVICO → reinicia um serviço.

**vi**: editor de texto simples, bastante tradicional (e antigo). Para quem não conhece, temos uma tela abaixo.



**nano**: editor de texto (mais elaborado que o vi). Abaixo uma tela do nano:



**fsck**: significa “file system check”. Usado para verificar e, opcionalmente, reparar um ou mais sistemas de arquivos.

**man**: fornece um manual completo para comandos ou utilitários que podem ser executados no terminal. É só digitar man NOME\_COMANDO que aparece o manual na tela.

**apropos**: procura por uma determinada expressão nas páginas de documentação.

**useradd**: cria contas de usuário no sistema. Único parâmetro obrigatório é login do usuário. Alguns outros são:



- -c "nome do usuário": grava no arquivo passwd o nome do proprietário da conta ou qualquer outra observação e comentário importantes;
- -d diretório\_home: fornece o caminho completo do diretório home do usuário;
- -m: cria o diretório home fornecido na opção "-d";
- -g número\_do\_grupo: fornece o grupo padrão da conta do usuário;
- -s shell: fornece o caminho completo do shell utilizado pela conta (ex.: /bin/bash);
- -e, --expiredate DATA\_EXPIRAÇÃO: define a data de expiração da conta.

**ifconfig:** configura as interfaces de rede e seus parâmetros. Parâmetros:

- interface: nome da interface de rede. Ex.: eth0 para o primeiro dispositivo de rede ethernet;
- up: habilita a interface de rede para uso;
- down: desabilita a interface de rede;
- address: configura um endereço IP para a interface;
- netmask máscara: configura uma determinada máscara de sub-rede para a interface.

Para configurar o IP, máscara e broadcast de uma placa de rede:

```
# ifconfig eth0 10.0.0.1 netmask 255.255.0.0 broadcast 10.0.0.255
```

Para levantar uma interface de rede:

```
# ifconfig eth0 up
```

Para visualizar uma determinada interface de rede:

```
# ifconfig eth0
```

Para configurar um alias para uma interface de rede (alias numérico):

```
# ifconfig eth0:1 192.168.1.2 netmask 255.255.255.0 up
```

**netstat:** exibe conexões de rede, tabelas de roteamento, interfaces de rede e estatísticas de protocolo de rede.

**shutdown:** desliga ou reinicia o sistema com horários determinados. Parâmetros:

- -r: reinicia (reboot) o sistema depois de terminada a sequência de desligamento;
- -h: paralisa (halt) o sistema depois de terminada a sequência de desligamento. Não reinicia a máquina;
- -k: manda a mensagem de desligamento, mas não inicia a sequência de desligamento;
- -f: faz a carga de sistema rápida sem a checagem de discos;
- -F: força uma checagem dos discos quando for reiniciar o sistema.

**ln:** cria ligações (*links*) entre arquivos. Pode ser:



- -d: cria um hard link (é o padrão). Ligação direta (hard link): define mais de um nome para um arquivo. O arquivo será removido do disco quando o último nome for removido;
- -L ou --logical: cria uma ligação para um link simbólico. Ligação simbólica (dymlink): define um caminho para um arquivo. Pode apontar para arquivos em diferentes sistemas de arquivo. Não necessita apontar para arquivos existentes;
- -s: cria uma ligação simbólica (soft link).

\$ In aula\* teste/ → cria no diretório teste, hark link para os arquivos do diretório atual cujos nomes comecem por "aula", cada arquivo terá dois nomes: um fica no diretório atual e o outro no diretório teste.

\$ In -s /etc/passwd senha → cria no diretório atual um link simbólico "senha" para o arquivo /etc/passwd.

\$ ls -l → podemos ver a indicação de link (l) e para onde o link aponta:

```
lrwxrwxrwx 1 aluno aluno 8 Dez 19 18:10 senha -> /etc/passwd
```

**alias:** cria atalhos ou apelidos para outros comandos. Pode ser útil para simplificar comandos longos, adicionar opções padrão a comandos frequentemente utilizados ou até mesmo criar atalhos mais fáceis de lembrar. Exemplo de um alias simples:

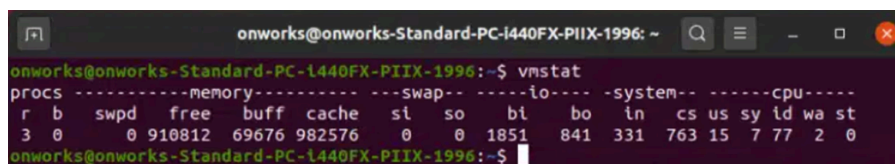
```
$ alias l='ls -l'
```

**tty:** imprime o nome do terminal associado ao processo que o está chamando. Ele exibe o caminho do dispositivo do terminal (ex.: /dev/tty1 ou pts/0).

**last:** exibe informações sobre os últimos logins no sistema. Ele mostra uma lista dos usuários que fizeram login, quando o fizeram e de onde (endereço IP ou terminal).

**lastlog:** mostra o login mais recente de um usuário específico ou todos do sistema. Formata e imprime o conteúdo do último arquivo de log /var/log/lastlog.

**vmstat:** mostra estatísticas sobre processos, memória, swap, blocos de E/S, traps, discos e atividades do processador.



```
onworks@onworks-Standard-PC-i440FX-PIIX-1996: ~  
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ vmstat  
procs -----memory----- --swap-- -----io----- -system-- -----cpu-----  
r b swpd free buff cache si so bi bo in cs us sy id wa st  
3 0 0 910812 69676 982576 0 0 1851 841 331 763 15 7 77 2 0  
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

**chroot:** utilizado para alterar o diretório raiz do processo corrente e dos seus filhos para o diretório especificado. "chroot" significa "change root" (mudar raiz) e geralmente é usado para criar um ambiente isolado, conhecido como "chroot jail", onde um processo acredita que o diretório especificado é o seu diretório raiz. A sintaxe básica é:

```
$ sudo chroot novo_diretorio comando
```





- novo\_diretorio: O diretório para o qual se deseja alterar a raiz;
- comando: O comando a ser executado no novo ambiente.

Por exemplo, para iniciar um shell dentro de um diretório específico:

```
$ sudo chroot /caminho/para/chroot /bin/bash
```

O comando acima inicia um shell dentro do diretório especificado como o novo diretório raiz.

**mkfs:** cria um sistema de arquivos em um dispositivo de armazenamento. A estrutura básica do comando mkfs é:

```
# mkfs -t tipo_do_sistema_de_arquivos dispositivo
```

Ex. com EXT4:

```
# mkfs -t ext4 /dev/sdb1
```

**date:** utilizado para exibir a data e hora atuais do sistema. Além disso, ele também pode ser usado para configurar a data e hora do sistema, mas geralmente isso requer permissões de root.

**finger:** exibe informações sobre usuários do sistema. Normalmente é utilizado para fornecer detalhes sobre um usuário específico, como nome, diretório home, informações sobre a última vez que o usuário fez login etc. O formato básico do comando finger é:

```
# finger nome_do_usuario
```

**who:** exibe informações sobre os usuários que estão atualmente conectados ao sistema. Ele mostra detalhes como nome de usuário, terminal de onde estão logados, endereço IP, hora de login e outros detalhes relacionados à sessão de login.



## LINUX - SERVIÇOS DE REDE E SEGURANÇA

### Diagnóstico/Configuração

No Linux existem algumas ferramentas para a configuração de dispositivos de rede e também para o diagnóstico, para verificar se está funcionando de acordo com o esperado. Vamos ver as ferramentas cobradas em provas de concurso na sequência.

**ping:** ferramenta usada para testar a conectividade de rede e verificar a disponibilidade de um host ou dispositivo. São enviados pacotes ICMP (Internet Control Message Protocol) para o host de destino e as respostas são aguardadas. Pode-se “pingar” através do nome (o DNS é utilizado para a tradução) ou o endereço IP diretamente:

```
$ ping www.estrategiaconcursos.com.br
```

```
$ ping 192.168.1.150
```

Pode-se definir a quantidade de pacotes, utiliza-se o parâmetro `-c` (o padrão é 4, caso não seja informada a quantidade):

```
$ ping -c 10 www.estrategiaconcursos.com.br
```

`$ ping -c` → se não for definido um valor, os pacotes são enviados até que se interrompa manualmente.

Pode-se definir o intervalo entre os pacotes (em segundos), com o parâmetro `-i`:

```
$ ping -i 2 www.estrategiaconcursos.com.br
```

**traceroute:** usado para rastrear a rota que os pacotes seguem para alcançar um destino. Mostra uma lista de saltos (hops) que os pacotes fazem, além dos tempos de resposta de cada salto. É útil para diagnosticar problemas de roteamento e identificar os nós intermediários (roteadores) pelos quais os pacotes de dados passam ao viajar da origem para o destino. A sintaxe é idêntica ao ping:

```
$ traceroute www.estrategiaconcursos.com.br
```

```
$ traceroute 192.168.1.150
```

Importante lembrar que no Windows a ferramenta que tem o mesmo propósito é o `tracert`. Cuidado com as pegadinhas!!! Nomes semelhantes, mas são diferentes!

Para definir um limite máximo de saltos, utiliza-se o parâmetro `-m`:

```
$ traceroute -m 30 www.estrategiaconcursos.com.br
```

Para definir um intervalo de tempo entre as sondas (em segundos), utiliza-se o parâmetro `-i`:

```
$ traceroute -i 2 www.estrategiaconcursos.com.br
```



Para exibir os nomes de hosts em vez de endereços IP, utiliza-se o parâmetro -n:

```
$ traceroute -n www.estrategiaconcursos.com.br
```

**netstat:** usado para exibir informações sobre as conexões de rede, tabelas de roteamento, estatísticas de interface de rede etc. Útil para diagnosticar problemas de rede, monitorar o tráfego e obter informações sobre as portas e protocolos em uso.

Para exibir informações detalhadas sobre todas as conexões e sockets:

```
$ netstat -a
```

Para listar todas as conexões de rede ativas (endereços IP e portas):

```
$ netstat -tuln
```

- -t: mostra as conexões TCP;
- -u: mostra as conexões UDP;
- -l: mostra as portas em escuta (listen);
- -n: exibe endereços IP e números de porta em formato numérico (evita a resolução de nomes).

Para exibir estatísticas de interface de rede:

```
$ netstat -i
```

Para exibir tabelas de roteamento IP:

```
$ netstat -r
```

**nmap:** ferramenta usada para descobrir e mapear redes, além de verificar a segurança de sistemas e serviços de rede. Amplamente utilizada para a varredura de portas, detecção de sistemas operacionais e análise de serviços em uma rede.

Para escanear um host específico para descobrir as portas abertas:

```
$ nmap 192.168.1.150
```

Para escanear um intervalo de endereços IP:

```
$ nmap 192.168.1.100-150
```

Para escanear todas as portas em um host (1 a 65535):

```
$ nmap -p1-65535 192.168.1.150
```

**tcpdump:** sniffer (permite a captura e a análise do tráfego de rede em tempo real – exige privilégios de root). Útil para solucionar problemas de rede, monitorar o tráfego, analisar protocolos etc.

Para capturar pacotes em uma interface de rede específica:

```
# tcpdump -i eth0
```

Para salvar a saída do tcpdump em um arquivo:



```
# tcpdump -i eth0 -w arquivo_de_saida.pcap
```

**ip**: ferramenta para configurar e exibir informações de rede relacionadas a interfaces, endereços IP, tabelas de roteamento, túneis, QoS (Qualidade de Serviço) etc. É a opção mais moderna em relação aos comandos `ifconfig` e `route`.

Para exibir informações sobre todas as interfaces de rede:

```
$ ip a
```

ou

```
$ ip addr show
```

Para exibir informações detalhadas sobre determinada interface:

```
$ ip a show eth0
```

Para ativar uma interface de rede:

```
$ sudo ip link set dev eth0 up
```

Para desativar uma interface de rede:

```
$ sudo ip link set dev eth0 down
```

Para configurar um endereço IP em uma interface:

```
$ sudo ip addr add 192.168.1.150/255.255.255.0 dev eth0
```

Para exibir informações sobre a tabela de roteamento:

```
$ ip route
```

**ifconfig**: usado para exibir e configurar informações de interfaces de rede. Gradualmente está sendo substituído pelo comando `ip`, que oferece mais recursos.

Para exibir informações sobre todas as interfaces de rede ativas:

```
$ ifconfig
```

Para exibir informações detalhadas sobre determinada interface:

```
$ ifconfig eth0
```

Para ativar uma interface de rede:

```
$ sudo ifconfig eth0 up
```

Para desativar uma interface de rede:

```
$ sudo ifconfig eth0 down
```

Para configurar um endereço IP em uma interface:

```
$ sudo ifconfig eth0 192.168.1.150 netmask 255.255.255.0
```



**ifup**: usado para ativar (subir) uma interface de rede de acordo com o que estiver definido nos arquivos de configuração. Por exemplo, se houver uma interface chamada eth0 e as configurações estiverem definidas em /etc/network/interfaces, pode-se usar:

```
$ sudo ifup eth0
```

**ifdown**: usado para desativar (baixar) uma interface de rede

```
$ sudo ifdown eth0
```

## Servidor Web

O servidor Web mais conhecido para Linux é o Apache, mas o ngnix é um servidor que está "roubando a cena". Para fins de concurso, o Apache ainda é o mais citado, então vamos focar nele nesta aula (existe uma aula específica de servidores Web com mais detalhes).

**Apache**: servidor Web livre (código-fonte aberto) que funciona em ambiente multiplataforma (Windows, Novell, OS/2, Unix, Linux, FreeBSD etc.). No Linux o servidor utiliza o daemon [httpd](#).

Existem dois tipos de páginas que podem ser adicionadas ao Apache: a página raiz e as subpáginas:

- Página raiz: especificada através da diretiva DocumentRoot, mostrada quando o usuário entra no domínio principal, ex.: <http://www.estrategiaconcursos.com.br>;
- Configuração padrão: DocumentRoot aponta para o diretório /var/www. Ex.: home/estrategia aponta para /var/www/home/estrategia.

Por padrão os arquivos de configuração ficam localizados no diretório [/etc/apache/](#) e o principal arquivo de configuração é o [httpd.conf](#).

No Apache há a possibilidade de realizar mudanças nas configurações por diretório, através dos arquivos .htaccess ("arquivos de configuração distribuída").

**Apachectl**: front end para o servidor Apache. Ajuda o administrador a controlar o funcionamento do daemon httpd. Para inicializar o Apache:

```
# apachectl start
```

Inicialização no Red Hat:

```
# service httpd start
```

Inicialização no Ubuntu:

```
# /etc/init.d/apache2 start
```

## Proxy

Um proxy funciona como intermediário entre os clientes da rede e os servidores da internet. Vamos focar no servidor proxy mais conhecido a seguir.



**Squid:** servidor proxy de código aberto. Armazena em cache o conteúdo da Web, o que acelera o acesso a recursos frequentemente acessados. Pode ser usado para implementar políticas de filtragem e controle de acesso à Internet. O arquivo de configuração é `/etc/squid/squid.conf`.

## Servidor FTP

Como o próprio nome deixa claro, o FTP (File Transfer Protocol) é um protocolo de transferência de arquivos. Trata-se de um protocolo padrão, independente de hardware. O FTP é baseado no protocolo de transporte TCP, o que garante a entrega dos pacotes. O servidor FTP utiliza as portas 20 para os dados e a 21 para o controle. Vamos ver como ele funciona:

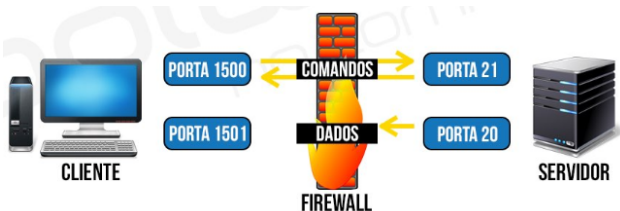
Um cliente realiza uma conexão TCP para a porta 21 do servidor. Essa conexão (conexão de controle) permanece aberta ao longo da sessão enquanto uma segunda conexão (conexão de dados), é estabelecida na porta 20 do servidor e em alguma porta do cliente (comunicada no diálogo entre cliente e servidor). A conexão de controle (porta 21) é utilizada para administração da sessão (comandos, identificação) entre cliente e servidor.

O servidor responde na conexão de controle com três dígitos de código de estado em ASCII com uma mensagem de texto opcional. Por exemplo, "200" ou "200 OK" significa que o último comando obteve êxito. Os números representam o número do código e o texto opcional representa as explicações ou parâmetros necessários. Uma transferência de arquivo em progresso, sobre uma conexão de dados, pode ser abortada utilizando uma mensagem de interrupção enviada sobre a conexão de controle.

O FTP pode ser executado em modo ativo ou passivo, os quais determinam como a conexão de dados é estabelecida. No **modo ativo**, o cliente envia para o servidor o endereço IP e o número da porta na qual ele irá ouvir e então o servidor inicia a conexão TCP. No exemplo abaixo vemos que o cliente informou ao servidor que a porta 1501 aguardará uma conexão vinda do servidor para receber os dados.

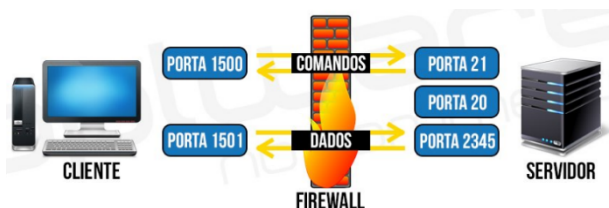


Mas e se houver um firewall que não tenha liberado a porta 1501? Olhe o que acontece:



Em situações onde o cliente está atrás de um firewall e inapto para aceitar entradas de conexões TCP, o **modo passivo** pode ser utilizado. O cliente envia um comando PASV para o servidor e recebe um endereço IP e um número de porta como resposta, os quais o cliente utiliza para abrir

a conexão de dados com o servidor. Na figura abaixo podemos ver que o cliente solicitou uma conexão no modo passivo e o servidor "disse": "Vou abrir a porta 2345 e você pode se conectar nela para receber os dados". Então, nesse caso, a porta 20 não será utilizada.



A transferência de dados pode ser feita em qualquer um dos três modos a seguir:

- Modo fluxo: dado é enviado como um fluxo contínuo, liberando o FTP de fazer algum processamento. Todo processamento é deixado para o TCP. Nenhum indicador de fim de arquivo é necessário, a menos que o dado esteja dividido dentro de registros;
- Modo de bloco: o FTP quebra o dado dentro de vários blocos (bloco de cabeçalho, contagem de byte e campo de dado) e então passa-o para o TCP;
- Modo comprimido: dado é comprimido utilizando um algoritmo simples.

O [acesso a servidores FTP](#) pode ocorrer de dois modos: através de uma interface ou através da linha de comando. Tanto usuários Linux como usuários Windows podem acessar através dos dois modos. O modo linha de comando está presente em qualquer distribuição Linux-like e Windows.

A partir de qualquer navegador credenciado (Internet Explorer, Chrome, Firefox, entre outros), também é possível acessar um servidor FTP digitando na barra de endereço:

```
ftp://[username]:[password]@[servidor]
ou
ftp://[username]:[password]@[servidor]:[porta]
```

Os [comandos](#) abaixo podem ser executados no FTP [através da linha de comando](#). Os comandos do FTP podem ser abreviados, desde que não formem expressões ambíguas.

- !: executa o comando na máquina local;
- ?: semelhante a help;
- append: adiciona dados a um arquivo existente;
- ascii: configura o tipo de transferência de arquivos para ASCII;
- bell: emite um bip quando um comando é executado;
- binary: configura o tipo de transferência de arquivos para binário;
- bye: encerra a sessão FTP;
- cd: seguido de caminho/diretório muda para o diretório informado;
- delete: apaga um arquivo. Para mais de um arquivo utiliza-se mdelete;
- debug: estabelece a modalidade de depuração;
- dir: mostra o conteúdo do diretório servidor atual;
- disconnect: semelhante a bye;
- get: obtém um arquivo do servidor. Para mais de um arquivo utiliza-se mget;
- glob: seleciona a expansão para nomes de arquivo;





- hash: demonstra cada bloco do arquivo durante a transferência. Cada bloco compõe-se de 1024 bytes;
- help: lista sumariamente todos comandos disponíveis;
- literal: permite enviar comandos arbitrários;
- ls: mostra uma lista abreviada do conteúdo do diretório servidor. Para mais de uma pasta utiliza-se mls;
- mkdir: cria um diretório ou subdiretório no servidor;
- prompt: ativa/desativa o modo interativo;
- put: envia um arquivo ao servidor. Para enviar mais de um arquivo utiliza-se mput;
- pwd: mostra o diretório de trabalho;
- quit: finaliza a sessão FTP;
- quote: envia subcomandos do servidor FTP, como se encontram no servidor;
- recv: similar a get;
- remotehelp: solicita ajuda do servidor FTP remoto;
- rename: renomeia um arquivo;
- send: semelhante a put;
- status: obtém informações de estado do servidor;
- trace: demonstra o caminho percorrido pelo arquivo na transferência;
- type: especifica o tipo de representação;
- user: inicia a sessão no servidor;
- verbose: ativa/desativa a modalidade literal.

Seguindo a lógica adotada para servidores no Linux, o servidor FTP é executado através de um *daemon*, o *ftpd*.

**ftpd**: servidor FTP (File Transfer Protocol) no Linux (sistemas Unix-like). Permite que os usuários transfiram arquivos entre um cliente FTP e um servidor FTP.

**vsftpd** (Very Secure FTP Daemon): um dos servidores FTP populares no Linux.

Instalação:

```
$ sudo apt-get install vsftpd
```

Arquivo de configuração principal: /etc/vsftpd.conf.

Inicialização:

```
$ sudo service vsftpd restart
```

## Servidor de E-mail

Antes de compreender como funciona o serviço (servidor) de e-mail no Linux, é importante entender a diferença dos 3 protocolos mais utilizados para esse serviço. Esqueça por um instante a utilização de HTTP ou HTTPS, no caso dos Webmails, vamos focar nos protocolos exclusivos para e-mail, mesmo. Vamos ver na sequência.

**SMTP** (*Simple Mail Transfer Protocol*): utilizado quando o e-mail é entregue a partir de um cliente de e-mail (Outlook, Thunderbird, entre outros) a um servidor de e-mail ou quando o e-mail é entregue a partir de um servidor de e-mail para outro servidor. Ou seja, se você enviar um e-mail

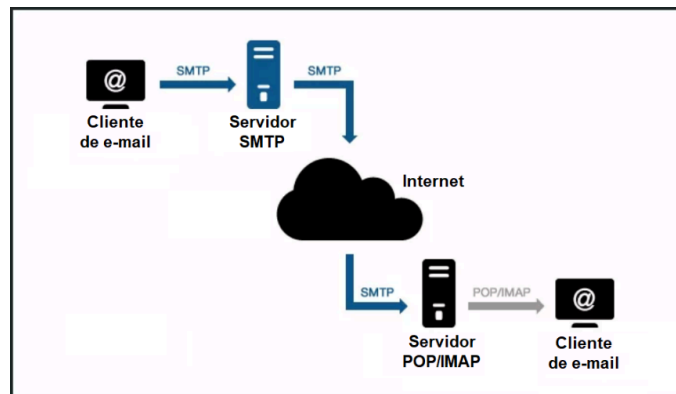


a partir de um software cliente, ele utilizará o SMTP até o seu servidor e depois o seu servidor utilizará também o SMTP para entregar ao servidor destino, quando este ficará na caixa de e-mails do usuário destinatário. **A porta padrão de um servidor SMTP é a 25.**

**IMAP** (*Internet Message Access Protocol*): também chamado de IMAP4, possui acesso a todas as pastas da conta de e-mail e deixa o *status* das mensagens igual tanto no servidor como no software e realiza a sincronia das mensagens, mantendo a conexão para que as alterações e mensagens novas recebidas sejam atualizadas quase que em tempo real. **A porta padrão de um servidor IMAP é a 143.**

**POP** (Post Office Protocol): mais conhecido como POP3, esse protocolo baixa as mensagens do servidor e as armazena localmente no computador, não deixando uma cópia das mensagens no servidor (a menos que seja marcada a opção "deixar uma cópia no servidor") nas configurações do programa de e-mail. Esse protocolo tem acesso apenas à Caixa de Entrada, não conseguindo baixar nenhuma outra pasta da conta de e-mail. **A porta padrão de um servidor POP3 é a 110.**

Abaixo podemos ver o relacionamento entre os 3 protocolos.



Podemos ver que a diferença principal é que o POP baixa as mensagens para o cliente local (permite que elas sejam armazenadas no servidor, mas não existe sincronia nesse processo).

O IMAP4 é a evolução do POP3, sincronizando a caixa de e-mails com o conteúdo do servidor, permitindo que um e-mail seja sincronizado entre vários locais sem perda de mensagens entre locais diferentes (salvo diretórios locais, todas as mensagens dentro do diretório do e-mail são sincronizadas).

No Linux, os servidores de e-mail mais comuns são o **sendmail** e o **postfix**. O postfix é considerado um MTA (Mail Transfer Agent) que se apresenta como alternativa ao sendmail, tendo como objetivo ser mais rápido, seguro e fácil de configurar que o sendmail, além de tentar manter a compatibilidade com ele. Vamos ver um pouco dos arquivos de configuração e funcionamento do postfix a seguir.

O postfix contém um arquivo denominado **main.cf**, no qual estão as configurações gerais do daemon, localizado no diretório **/etc/postfix**, pela instalação padrão. Abaixo serão demonstrados os principais parâmetros de configuração para ter-se um servidor de e-mail funcional. Nos exemplos abaixo será utilizado o nome do servidor como "mail.dominio.com.br" e o domínio "dominio.com.br", apenas para ilustração.



```
myhostname = mail.dominio.com.br
```

```
mydomain = dominio.com.br
```

O parâmetro "mydestination" deve receber os domínios que o servidor de e-mail é responsável. Por exemplo, mail.dominio.com.br, ftp.dominio.com.br, www.dominio.com.br, dominio2.com.br etc.:

```
mydestination = $myhostname, localhost.$mydomain, $mydomain, mail.$mydomain, www.$mydomain, dominio2.com.br
```

O parâmetro "mynetworks" serve para determinar as redes do provedor:

```
mynetworks = 192.168.0.0/16, 127.0.0.0/8
```

Também podem ser informados os domínios em um arquivo:

```
mynetworks = /etc/postfix/mynetworks
```

Pela configuração padrão do postfix, ele permite relay de e-mails das suas redes e de seus domínios. Se houver clientes que não fazem parte da rede e que necessitam utilizar o seu servidor de e-mail para o envio de mensagens, é possível criar um arquivo com a lista de clientes através do parâmetro "smtpd\_recipient\_restrictions" e habilitar o envio de mensagens, como mostrado abaixo:

```
smtpd_recipient_restrictions = permit_mynetworks  
                                check_client_access hash:/etc/postfix/client_access  
                                check_relay_domains
```

Exemplo do arquivo /etc/postfix/client\_access:

```
dial.amigos.com.br OK
```

```
10.0.0 OK
```

```
falcatrua.com.br REJECT
```

Esse parâmetro não consta na configuração padrão do postfix e pode ser incluído no final do arquivo de configuração main.cf.

A [entrega de e-mails no postfix](#) pode ser feita de diversas formas: enviando os e-mails para "/var/spool/mail/user" (padrão), no formato "Maildir/" (utilizado pelo gmail) entre outros.

**Iniciando o servidor:** # postfix start

Caso seja feita alguma alteração na configuração do servidor de e-mail, para atualizá-lo, basta digitar o comando: # postfix reload



Para desativar o servidor de e-mail: # postfix stop

Uma característica importante no Linux são os logs, que, na sua maioria, são armazenados em /var/log. Esse é o caso também dos logs relacionados aos servidores de e-mail, útil quando são necessárias informações sobre postfix, smtpd ou qualquer serviço relacionado ao e-mail que esteja sendo executado no servidor. Por padrão, esses logs ficam em /var/log/maillog ou /var/log/mail.log.

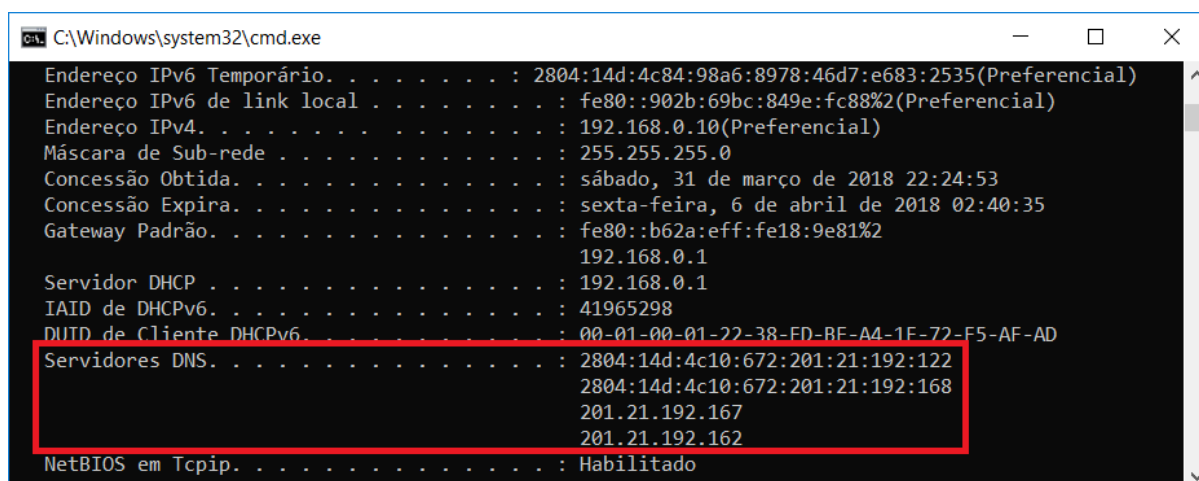
## DNS (Domain Name System)

Os equipamentos conectados à Internet necessitam de um endereço IP e através dele conseguimos identificar "quem" acessou determinado conteúdo ou realizou determinada ação, por exemplo. Mas, quem "gosta" de números é a máquina! Nós, meros mortais, temos dificuldade em memorizar muitos números (endereços IP, por exemplo). E os servidores (Web, de e-mail, de arquivos etc.) são acessados através do endereço IP que foi alocado a eles.

Uma solução para isso foi a criação de um serviço que "faz o meio de campo", ou seja, traduz nomes (bem mais fáceis de memorizar) para os endereços IP equivalentes. É o famoso *Domain Name System* (DNS). Assim, é possível acessar uma página Web através de um nome, sendo que de forma transparente ao usuário, esse nome é traduzido ao endereço IP onde se encontra o servidor Web que contém a página e a requisição é realizada a esse servidor.

Um conceito mais formal (Tanenbaum) é o seguinte: o DNS é definido como um esquema hierárquico de atribuição de nomes baseado no domínio e de um sistema de banco de dados distribuído. O DNS atua na camada de aplicação e utiliza como protocolo de transporte o UDP para as consultas/respostas e o TCP para transferências de zonas (entre servidores DNS). Tanto com o UDP como com o TCP, a porta utilizada é a 53.

Abaixo é mostrada uma tela com a resposta para o comando `ipconfig /all` (no Linux o comando equivalente é o `ifconfig`) mostrando, entre outras informações, os servidores DNS locais.



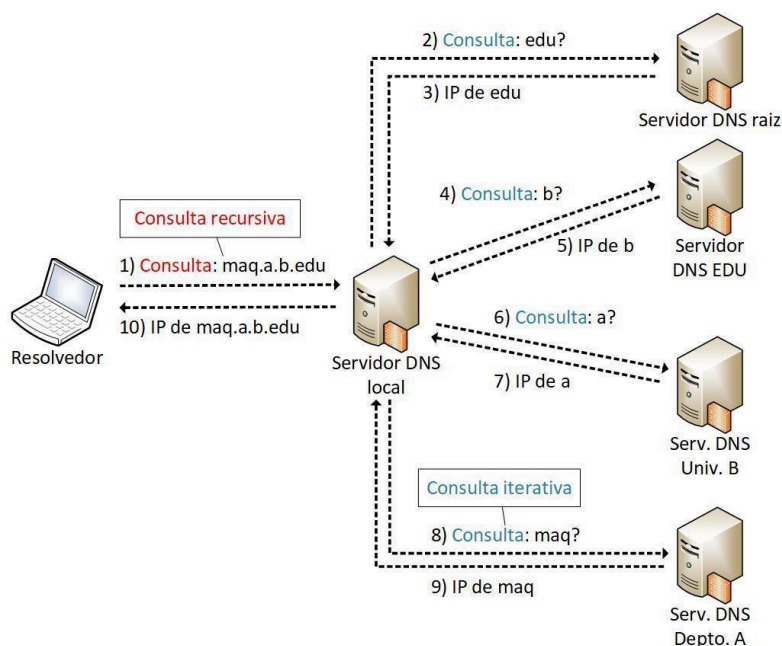
```
C:\Windows\system32\cmd.exe
Endereço IPv6 Temporário. . . . . : 2804:14d:4c84:98a6:8978:46d7:e683:2535(Preferencial)
Endereço IPv6 de link local . . . . : fe80::902b:69bc:849e:fc88%2(Preferencial)
Endereço IPv4. . . . . : 192.168.0.10(Preferencial)
Máscara de Sub-rede . . . . . : 255.255.255.0
Concessão Obtida. . . . . : sábado, 31 de março de 2018 22:24:53
Concessão Expira. . . . . : sexta-feira, 6 de abril de 2018 02:40:35
Gateway Padrão. . . . . : fe80::b62a:eff:fe18:9e81%2
                            192.168.0.1
Servidor DHCP . . . . . : 192.168.0.1
IAID de DHCPv6. . . . . : 41965298
DUID de Cliente DHCPv6. . . . . : 00-01-00-01-22-38-FD-BE-A4-1F-72-F5-AF-AD
Servidores DNS. . . . . : 2804:14d:4c10:672:201:21:192:122
                            2804:14d:4c10:672:201:21:192:168
                            201.21.192.167
                            201.21.192.162
NetBIOS em Tcpip. . . . . : Habilitado
```

Um passo a passo de uma solicitação de um cliente DNS (seu computador, por exemplo) a um servidor DNS local é mostrado a seguir.



- 1) O aplicativo (ex.: navegador) chama o **resolvedor**, passando o nome que se deseja a tradução para endereço IP;
- 2) O resolvedor realiza uma consulta ao servidor DNS local;
- 3) O servidor DNS local responde ao resolvedor;
- 4) O resolvedor informa o endereço IP ao aplicativo.

Ok, mas e se for um nome que o servidor DNS local não conhece? Seja porque nunca foi solicitado, ou por que tal informação já não se encontra mais em sua *cache*? Bom, aí é melhor olhar a figura abaixo.




A consulta realizada ao servidor DNS local é chamada **consulta recursiva**, pois o resolvedor envia a consulta e recebe a resposta final, sem precisar enviar uma consulta a cada servidor DNS de nível superior. Já em **consultas iterativas**, a resposta à requisição DNS pode ser parcial, obrigando o solicitante a encaminhar novas requisições DNS a outros servidores até obter a resposta final desejada.

A delegação de domínios de mais alto nível (*top-level domain* - TLD), tais como .com, .edu, .br, .mx, entre outros, é de responsabilidade da ICANN (*Internet Corporation for Assigned Names and Numbers*). Para o Brasil (TLD .br), o responsável é o CGI.br<sup>1</sup>, conforme podemos ver abaixo.

<sup>1</sup> Base de dados de domínios TLD disponível em <<http://www.iana.org/domains/root/db>>.

.booking	generic	Booking.com B.V.
.boots	generic	THE BOOTS COMPANY PLC
.bosch	generic	Robert Bosch GMBH
.bostik	generic	Bostik SA
.boston	generic	Boston TLD Management, LLC
.bot	generic	Amazon Registry Services, Inc.
.boutique	generic	Binky Moon, LLC
.box	generic	NS1 Limited
.bq	country-code	Not assigned
.br	country-code	Comite Gestor da Internet no Brasil
.bradesco	generic	Banco Bradesco S.A.
.bridgestone	generic	Bridgestone Corporation
.broadway	generic	Celebrate Broadway, Inc.
.broker	generic	DOTBROKER REGISTRY LTD



Note que existe um TLD ".bradesco", relacionado ao Banco Bradesco. Faça um teste em seu navegador: digite "bradesco.com.br" e "bradesco.bradesco". Qual o resultado? No momento em que testei, ambos direcionam para uma nova URL: "https://banco.bradesco/html/classic/index.shtm", pertencente ao domínio ".bradesco".

Então, se alguém quiser registrar um domínio com o sufixo .br, pode verificar se há disponibilidade desse domínio, através da URL <http://registro.br>. Se houver, pode realizar a solicitação, efetuar o pagamento e informar as configurações solicitadas pelo CGI.br sobre o provedor onde a página será hospedada (servidores DNS).

Na medida em que novos domínios são cadastrados, eles são propagados pela Internet e em poucas horas todos os servidores DNS do mundo são capazes de traduzir o domínio para o endereço IP equivalente onde está hospedado o serviço. A figura abaixo mostra a estrutura DNS, desde a raiz, os TLDs, domínios de segundo e terceiro níveis e o computador lá na ponta.





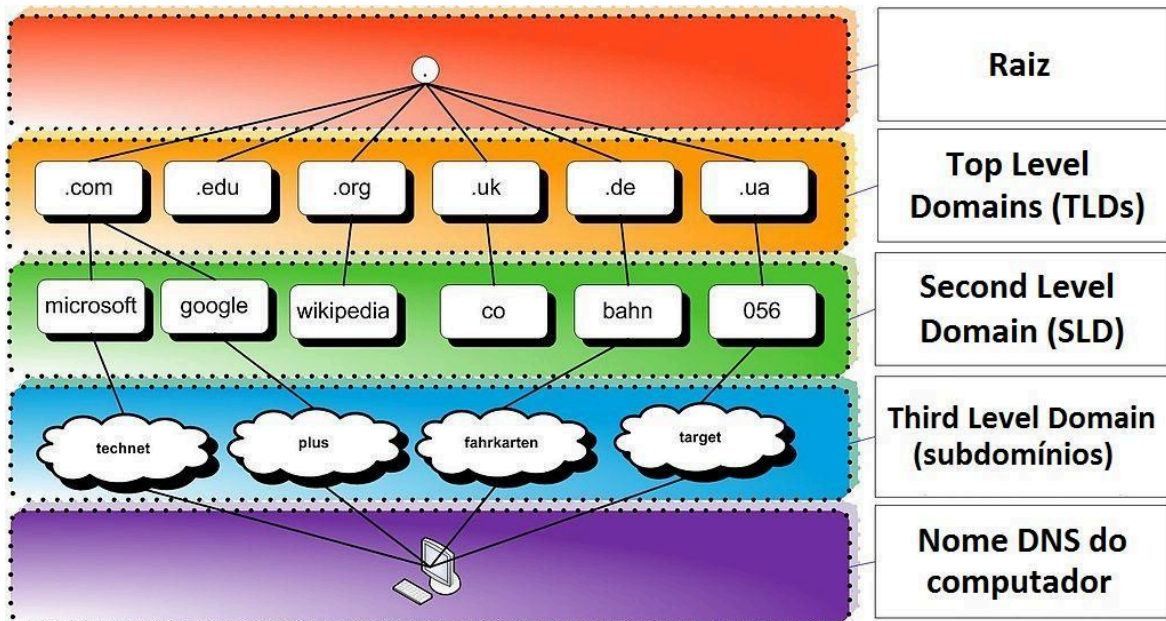
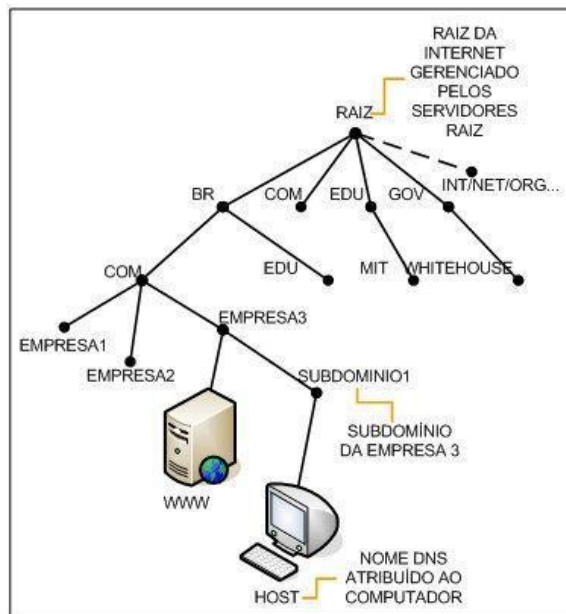


Figura adaptada de <<https://hugoemiliano.info/2017/07/05/servicos-e-protocolos-dns/>>.

Por exemplo, a URL <www.microsoft.com> pode ser compreendida da seguinte forma:

- .com: Top Level Domain (TLD);
- microsoft: Second Level Domain (SLD);
- não há terceiro nível (subdomínio) para essa URL;
- www: Nome do computador (www é um nome padrão para servidores Web).

Abaixo uma outra figura, mostrando um exemplo com subdomínio. Nesse caso a URL completa para acessar o HOST seria <HOST.SUBDOMINIO1.EMPRESA3.COM.BR>.

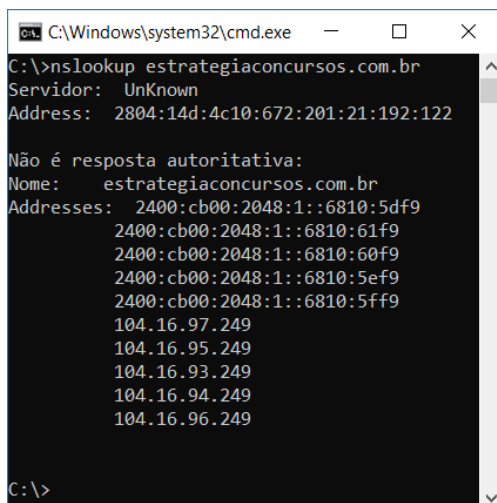


Fonte: <[http://www.abusar.org/dns\\_como.html](http://www.abusar.org/dns_como.html)>.



Para não haver consultas constantes a servidores DNS de mais alto nível (mais próximos da raiz, ou a própria raiz), os servidores DNS possuem uma memória cache<sup>2</sup>, permitindo a resposta imediata ao solicitante (quando tiver a informação). Quando não tiver a informação, deve-se buscar nos níveis superiores.

É possível também, em sistemas operacionais como Windows e Linux, configurar em traduções fixas, de domínio para endereço IP (arquivo hosts, como já vimos). Uma ferramenta comum ao Windows e Linux para obter informações sobre registros de DNS de um determinado domínio, host ou IP é o *nslookup* (vale a pena utilizá-la, pois há questões que cobram o seu conhecimento):



```
C:\Windows\system32\cmd.exe - □ ×
C:\>nslookup estrategiaconcursos.com.br
Servidor: UnKnown
Address: 2804:14d:4c10:672:201:21:192:122

Não é resposta autoritativa:
Nome: estrategiaconcursos.com.br
Addresses: 2400:cb00:2048:1::6810:5df9
           2400:cb00:2048:1::6810:61f9
           2400:cb00:2048:1::6810:60f9
           2400:cb00:2048:1::6810:5ef9
           2400:cb00:2048:1::6810:5ff9
           104.16.97.249
           104.16.95.249
           104.16.93.249
           104.16.94.249
           104.16.96.249

C:\>
```

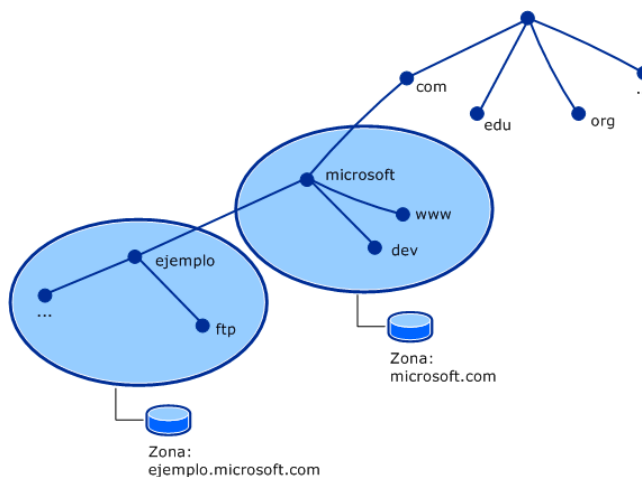
Em relação ao padrão Unix, o servidor de nomes mais conhecido é o BIND, que é conjunto de softwares DNS, que contém um *daemon* servidor de nomes (*named*), uma biblioteca *resolver* (nosso "resolvedor") e outros programas. O Bind é mantido pela ISC (*Internet Software Consortium* - <<https://www.isc.org/downloads/bind/>>).

O arquivo de configuração para o resolvedor é o [resolv.conf](#) e fica localizado em /etc. Trata-se de um arquivo em texto plano usualmente criado pelo administrador ou por aplicações que gerenciam tarefas de configuração.

O espaço de nomes do DNS é dividido em zonas não superpostas. Cada zona está associada a um ou mais servidores de nomes, que mantêm o banco de dados para a zona. A figura abaixo mostra tal conceito:

<sup>2</sup> Responsável por armazenar consultas recentes, respondendo ao solicitante diretamente.





Fonte: <<http://un-newbie.blogspot.com.br/2014/03/introduccion-transferencia-de-zona-y.html>>.

Os **registros de recursos (RRs)** são o banco de dados do DNS. São compostos por tuplas de cinco campos: <nome\_domínio, tempo\_vida, classe, tipo, valor>, descritos abaixo:

- Nome: chave de pesquisa primária para atender as consultas;
- Tempo\_vida (TTL): tempo que deve permanecer em *cache* (em segundos);
- Classe: geralmente IN (Internet);
- Tipo: SOA, A, AAAA etc. (tabela a seguir);
- Valor: número, nome de domínio ou *string* ASCII.

Tipo	Significado	Valor
SOA	Início de autoridade ( <i>Start of Authority</i> ).	Parâmetros para essa zona.
A	Endereço IPv4.	Inteiro de 32 bits.
AAAA	Endereço IPv6.	Inteiro de 128 bits.
MX	Troca de mensagens de e-mail.	Prioridade, domínio disposto a aceitar e-mails.
NS	Servidor de nomes.	Nome de um servidor para este domínio.
CNAME	Nome canônico ( <i>alias</i> = apelido).	Nome de domínio.
PTR	Ponteiro (usado para o DNS reverso <sup>3</sup> )	Nome alternativo de um end. IP.

<sup>3</sup> Envia um endereço IP como consulta e recebe o nome como resposta.

SPF	Estrutura de política do transmissor.	Codificação de texto da política de envio de mensagens de e-mail.
SRV	Identifica computadores que hospedam serviços específicos.	Host que o oferece.
TXT	Informações sobre um servidor, rede, <i>datacenter</i> etc.	Texto ASCII com descrições.

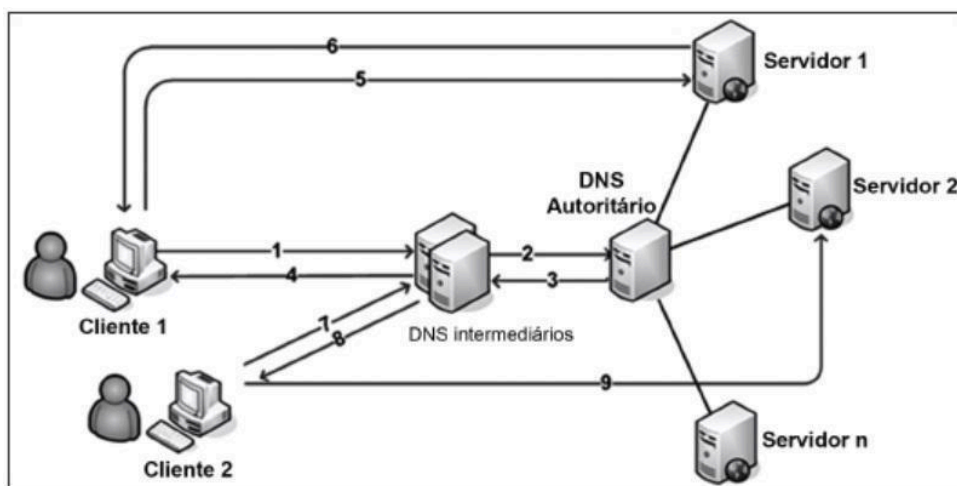
Outro conceito cobrado em provas de concurso é a resposta autoritativa ou não-autoritativa. Vejamos suas definições:

Um servidor DNS autoritativo possui autoridade sobre um nome de domínio. O DNS autoritativo dita qual será o apontamento da tabela de DNS do seu sítio. Uma **resposta autoritativa**<sup>4</sup> de um servidor é a garantia de estar atualizada, enquanto uma **resposta não-autoritativa** pode estar desatualizada (*cache* com informações antigas, por exemplo). Existe um percentual elevado de respostas não autoritativas que estão perfeitamente corretas, casos em que mudanças de endereçamento são raros.

Servidores primários e secundários são autoritativos para os seus domínios, porém não o são sobre informações a respeito de outros domínios mantidas em *cache*. **Servidores  *caching-only* nunca são autoritativos**, mas possuem a vantagem de reduzir a quantidade de tráfego DNS na rede.

Uma política que pode ser adotada para equilibrar as vantagens de cada técnica é colocar um servidor secundário ou  *caching-only* em cada segmento de rede ou subrede. É admissível uma máquina ser servidora primária para um domínio e servidora secundária para outros domínios.

Para não sobrecarregar servidores, existe uma abordagem popular e simples: o **balanceamento de carga por DNS**. Considere a figura abaixo e o passo a passo mostrado na sequência.



Fonte: Prova FCC/ELETROSUL - 2016.

1. Cliente 1 tenta acessar o *site*, é realizada uma pesquisa no DNS local para determinar qual é o endereço IP correspondente;
2. O pedido de endereço chega ao servidor de DNS autoritário do domínio;
3. A primeira vez que esta consulta é feita, o servidor DNS remoto pode retornar todos os registros de endereços que ele tem para o site;
4. O servidor DNS local, em seguida, determina o endereço de registro para retornar ao cliente;
5. Se todos os registros são retornados, o cliente utilizará o primeiro que lhe é atribuído;
6. O servidor responde ao cliente e atende ao pedido;
7. A cada pedido, o algoritmo Round Robin roda os endereços e retorna pela ordem em que eles estão;
8. Cada consulta DNS irá resultar em um cliente usando um endereço diferente;
9. Esta rotação de endereços irá distribuir pedidos para os servidores.

No Linux o servidor DNS mais utilizado e cobrado em concursos é o BIND (*Berkeley Internet Name Domain*), que se encontra na versão 9 (por isso o nome do pacote é `bind9`). Para instalar o `bind9` e ferramentas relacionadas pode-se utilizar o gerenciador de pacotes APT (ou o `apt-get`), da seguinte forma:

```
$ sudo apt-get install bind9 bind9utils bind9-doc dnsutils
```

É importante saber quais são e para que servem os arquivos de configuração. Um dos mais cobrados em concursos é o `/etc/resolv.conf`, que identifica os locais dos computadores de servidor DNS. Nesse arquivo deve ser indicado o nome de domínio DNS da rede, e qual o servidor DNS irá resolver as consultas de nomes. Um exemplo é mostrado a seguir.

```
domain ti.teste.com.br  
search ti.teste.com.br  
teste.com.br  
nameserver 192.168.1.1  
nameserver 192.168.2.1
```

Cada parâmetro é colocado em uma linha. As três palavras-chave normalmente usadas são:

- *domain*: especifica o nome do domínio local;
- *search*: lista de pesquisa para a procura do nome de servidor, normalmente determinado pelo domínio do servidor local;
- *nameserver*: Endereço IP do servidor de nomes que o resolvedor deve pesquisar, utilizados na ordem em que estão listados;

O arquivo `resolv.conf` também serve para especificar o tipo (nível) de serviço a ser executado pelo software, a saber:

- *resolver-only system*: não requer que o sistema local tenha um servidor DNS em execução, requer apenas o resolver;
- *caching-only server*: servidor que não possui uma cópia da tabela de zonas, ou seja é não-autoritativo, possuindo uma grande quantidade de registros em cache;
- *primary server (master server)*: onde todas atualizações manuais devem ser realizadas, é autoritativo;
- *secondary server*: busca informações de zona em um *master server*, é autoritativo.



O principal arquivo de configuração do Bind é o `/etc/bind/named.conf` (em versões antigas: `/etc/named.conf`). Por padrão o Bind já vem configurado para trabalhar como um servidor DNS de *cache*, que pode ser usado tanto localmente quanto por outros dispositivos da rede local. Dentro do arquivo de configuração é possível encontrar entradas tais como:

<pre>zone "." {     type hint;     file "/etc/bind/db.root"; };  zone "localhost" {     type master;     file "/etc/bind/db.local"; }; zone "127.in-addr.arpa" {     type master;     file "/etc/bind/db.127"; };</pre>	<pre>zone "0.in-addr.arpa" {     type master;     file "/etc/bind/db.0"; };  zone "255.in-addr.arpa" {     type master;     file "/etc/bind/db.255"; };</pre>
---	---

Pode-se ver que cada uma das seções indica a localização de um arquivo, onde vai a configuração referente a ela. Por exemplo, na primeira seção ("zone ".") é indicado o arquivo `/etc/bind/db.root`, que contém os endereços dos 14 root servers, que o Bind contactará na hora de resolver os domínios.

Esta configuração vem incluída por padrão e não deve ser alterada, a menos que seja um usuário experiente. Ao configurar o servidor DNS algumas novas zonas são incluídas (novas seções de configuração), contendo os domínios que se deseja configurar.

O serviço referente ao Bind pode se chamar "bind" ou "named", de acordo com a distribuição. Nos derivados do Debian o controle do serviço é realizado através do comando `/etc/init.d/bind9` (ou `/etc/init.d/bind` para a versão 8), enquanto nas distribuições derivadas do Red Hat utiliza-se o comando `service named`:

```
# /etc/init.d/bind9
restart
ou
# service named restartc
```

Servidores DNS públicos (*forwarders*), como por exemplo o Google Public DNS, podem ser configurados no arquivo `named.conf.options`. Nesse arquivo também pode ser definida a lista de controle de acessos (ACL). Um exemplo é mostrado abaixo.

<pre>options {     directory "/var/cache/bind";     // Troca a porta entre servidores DNS     query-source address * port *;     forward only;     forwarders { 192.168.1.1; };     auth-nxdomain no;     interface-interval 0;</pre>	<pre>// Não transfere info de zona a DNS secundário     allow-transfer { none; };     allow-query { internals; };     allow-recursion { internals; };     // Não torna versão pública do BIND</pre>
---	---



```
// Escuta apenas em interfaces locais
listen-on-v6 { none; };
listen-on { 127.0.0.1; 192.168.0.1;
};

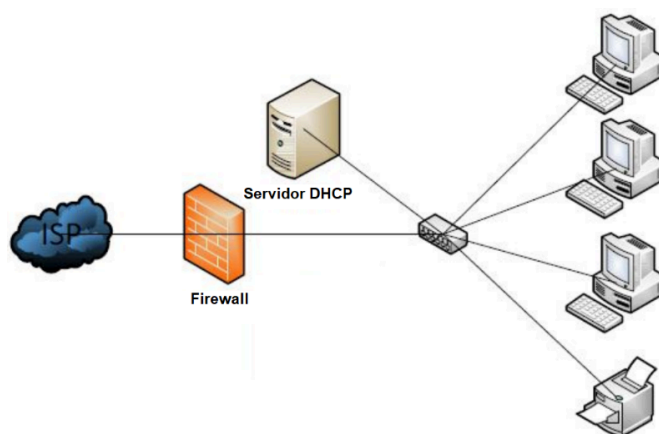
version none;
};
```

Para finalizar é importante destacar um arquivo útil para máquinas que são acessadas frequentemente: o `/etc/hosts`. A inclusão de um computador nesse arquivo dispensa a consulta a um servidor DNS para obter um endereço IP. Ou seja, a tradução ocorre automaticamente ao ler tal arquivo, sem ter que consultar o serviço de DNS. Um exemplo é mostrado abaixo.

```
# Aqui são colocados comentários
# End. IP      Nome do host      Alias
(opcional)
127.0.0.1     localhost         localmente
192.168.0.1   www.teste.com.br teste
```

## DHCP (Dynamic Host Configuration Protocol)

O DHCP (*Dynamic Host Configuration Protocol*) é um protocolo que tem a função de configurar os endereços IP dos computadores de uma rede de forma dinâmica. Ou seja, deve haver pelo menos um servidor DHCP pré-configurado para receber solicitações de computadores que não possuem endereço IP (clientes), o servidor verificar qual endereço IP disponível e envia ao solicitante. Abaixo podemos ver um exemplo:



No cenário mostrado, os três PCs e a impressora estão configurados como clientes DHCP. Cada um deles deve enviar uma mensagem *broadcast* (mensagem a todos, pois não sabe quem é o servidor DHCP), o servidor DHCP recebe a mensagem, verifica qual endereço IP está disponível (ou até mesmo reservado para o solicitante), oferece esse endereço ao cliente e, por fim, o cliente aceita (comunicando o servidor que vai utilizar tal endereço IP).

Algumas informações que devem ser configuradas no servidor:

- Escopo: intervalo de endereços IP que estarão disponíveis para atribuição automática. Também pode se referir ao intervalo que não será distribuído. Ex.: endereços IP disponíveis = 192.168.1.100 a 192.168.1.150;



- Máscara de rede: usada para fazer a divisão da rede de computadores. Uma rede classe C possui a máscara 255.255.255.0;
- Gateway: dispositivo que serve para interligar a rede local com a Internet, ex. na figura: o elemento central (pode ser um modem/roteador), que liga os cinco dispositivos da rede local (servidor + 3 PCs + impressora) à Internet (passando por um firewall), ex.: 192.168.1.1;
- DNS: endereço do servidor DNS a ser consultado, ex.: 8.8.8.8 (esse é o servidor DNS do Google).

**Importante:** o servidor DHCP precisa de pelo menos uma interface de rede configurada com endereço IP fixo!

O `dhcpd` é o servidor DHCP no Linux. Ele é útil, por exemplo, em uma máquina agindo como um roteador em uma rede local.

Obs.: `dhcpd` (*daemon* do servidor DHCP) não é o mesmo que `dhcpcd` (*daemon* do cliente DHCP).

### Instalação:

Deve-se instalar o `dhcp`, disponível nos repositórios oficiais.

### Uso:

O `dhcpd` inclui dois arquivos: `dhcpd4.service` e `dhcpd6.service`, que podem ser usados para controlar o *daemon*. Eles iniciam o *daemon* em todas as interfaces de rede para IPv4 e IPv6, respectivamente.

### Configuração:

Deve-se atribuir um endereço IPv4 estático para a interface que se deseja usar (ex.: `eth0`).

O arquivo `/etc/dhcpd.conf` contém as configurações. Esse arquivo pode se parecer com:

```
option domain-name-servers 8.8.8.8, 8.8.4.4;  
option subnet-mask 255.255.255.0;  
option routers 139.96.30.100;  
subnet 139.96.30.0 netmask 255.255.255.0 {  
  range 139.96.30.150 139.96.30.250;  
}
```

O arquivo `dhcpd.leases` é utilizado pelo servidor DHCP para armazenar informações sobre os endereços IP atribuídos a clientes DHCP, o tempo de concessão, entre outras informações. Este arquivo geralmente é mantido pelo servidor DHCP para rastrear os detalhes das concessões de endereços IP.





## Servidor NFS

O NFS (Network File System) é um protocolo de compartilhamento de arquivos que permite que sistemas operacionais Unix-like compartilhem recursos de sistema de arquivos em uma rede. O servidor NFS exporta (compartilha) seus sistemas de arquivos locais para que outros sistemas possam montá-los remotamente.

**Exportação NFS:** deve-se configurar quais diretórios ou sistemas de arquivos locais serão exportados para a rede (`/etc/exports`).

**exports:** ferramenta usada para gerenciar e listar os sistemas de arquivos compartilhados para acesso via NFS. Os parâmetros mais conhecidos são:

- `-r`: usado para reexportar (reexportar recursivamente) todos os compartilhamentos definidos no arquivo `/etc/exports`;
- `-ua`: usado para desativar (`unexport`) todos (all) os compartilhamentos NFS exportados pelo servidor. Isso quer dizer que ele revoga temporariamente todas as configurações de exportação definidas no arquivo `/etc/exports`, tornando os compartilhamentos indisponíveis até que sejam reexportados.

O NFS é mais comumente associado a sistemas operacionais baseados em Unix-like. No entanto, para integração com sistemas Windows, existem implementações do NFS para Windows.

## Servidor SAMBA

O SAMBA é um software de código aberto que implementa o protocolo SMB/CIFS (Server Message Block/Common Internet File System). O SMB/CIFS é usado para compartilhar arquivos e recursos em redes locais, permitindo a interoperabilidade entre sistemas Windows, Linux/Unix e outros. Também é realizado o compartilhamento de impressão.

**smbd:** um dos principais processos do Samba.

**etc/samba/smb.conf:** arquivo de configuração principal.

## iptables

O iptables é usado para configurar regras de firewall e controlar o tráfego de rede. Ele permite a definição de políticas de segurança, regras de encaminhamento de pacotes, o mapeamento de portas (NAT) etc.

Cadeias (chains):

- **INPUT:** pacotes destinados ao próprio sistema (chegando);
- **OUTPUT:** pacotes gerados pelo próprio sistema (saindo);
- **FORWARD:** pacotes que são roteados pelo sistema (não destinados nem gerados pelo sistema).

Regras (executa o primeiro "match", ou seja, se deu "match", não executa o que vem depois):

- `-A`: adiciona uma regra;
- `-D`: "deleta" (apaga) uma regra;
- `-j`: ação a ser tomada (ACCEPT, DROP, REJECT etc.);
- `-s`: endereço IP origem (source);



- -d: endereço IP destino (destination);
- -i: interface de entrada (input);
- -o: interface de saída (output);
- -p: protocolo;
- --dport: porta destino;
- --sport: porta origem;
- etc.

Vamos ver alguns exemplos:

\$ iptables -A INPUT -i eth0 -j ACCEPT → Adiciona a regra para aceitar pacotes que chegam pela interface eth0.

\$ iptables -A INPUT -p tcp --dport 80 -j DROP → Adiciona a regra que descarta pacotes TCP que chegam, tendo como porta destino a 80.

\$ iptables -D OUTPUT -p udp --dport 53 -j REJECT → Exclui a regra que rejeita pacotes UDP (com aviso à origem) originados no dispositivo, com porta destino 53.

\$ iptables -L → Lista todas as regras atuais configuradas.

\$ iptables -F → Limpa todas as regras atuais configuradas.

## Netfilter

O Netfilter permite interceptar, modificar, encaminhar e descartar pacotes de dados, atuando diretamente no nível do kernel para oferecer um sistema de firewall e NAT (*Network Address Translation*). O Netfilter é a base de ferramentas de firewall no Linux, como iptables e nftables (que é o sucessor moderno do iptables). As principais funções do Netfilter são:

- Filtragem de pacotes: Aceitar ou bloquear pacotes com base em critérios (endereço IP, porta etc.);
- NAT: Modificar o endereço IP e portas de pacotes de rede. É útil em redes locais que compartilham uma única conexão com a Internet;
- Redirecionamento de portas: Redirecionar pacotes destinados a um endereço ou porta para outro destino.

## UFW (*Uncomplicated Firewall*)

UFW é uma ferramenta de firewall voltada para usuários que buscam uma interface simplificada para gerenciar o firewall no Linux. Ela foi desenvolvida especialmente para Ubuntu e suas distribuições derivadas, mas também pode ser instalada em outras distribuições. A UFW funciona como uma interface para o iptables, facilitando o uso do Netfilter sem a necessidade de entender a complexidade do iptables.

As vantagens do UFW são:



- Simplicidade: Interface mais fácil para configurar regras de firewall;
- Intuitivo: Comandos básicos para gerenciar portas e serviços comuns;
- Automação: Configurações simples e rápidas para usuários que não precisam de todos os detalhes oferecidos pelo iptables.

## CIS (Center for Internet Security)

O CIS (*Center for Internet Security*) Red Hat Enterprise Linux Benchmark é um conjunto de recomendações e melhores práticas de segurança para a configuração e proteção de sistemas que utilizam o Red Hat Enterprise Linux (RHEL). Tais diretrizes foram desenvolvidas pela comunidade de segurança da informação e possuem como objetivo a redução da superfície de ataque e o fortalecimento da segurança dos sistemas.

Os principais pontos abordados no CIS Benchmark para o RHEL são:

### 1. Configuração de Segurança Básica

Configuração do GRUB: Proteção com senha e restrições de inicialização, para evitar o acesso não autorizado ao menu de boot.

Particionamento do Disco: Existe a recomendação para o particionamento lógico, incluindo a separação de partições críticas, tais como /var, /tmp, /home, entre outras.

Configuração do SELinux (*Security-Enhanced Linux*): Garantir que o SELinux esteja habilitado e configurado corretamente para proteger o sistema. O SELinux é um módulo de segurança para o kernel que implementa controles de acesso obrigatórios.

### 2. Controle de Acesso

Configuração do PAM (*Pluggable Authentication Modules*): Definição de regras de autenticação, como por exemplo complexidade de senha, bloqueio de contas após N tentativas de login falhas e tempo de inatividade.

Gerenciamento de Contas e Senhas: Diretrizes sobre políticas de senha, controle de acesso do usuário root e permissões de usuários e grupos.

### 3. Segurança de Redes

Configuração do Firewall: Implementação de regras de firewall usando firewalld ou o iptables para controlar o tráfego de rede.

Segurança do SSH: Restrições de login via SSH. Exs.: desativar o login de root, exigir autenticação por chave, limitar o acesso a partir de determinados endereços IP, entre outras.

### 4. Segurança de Sistema e Serviço

Gerenciamento de Serviços: Desativação de serviços e *daemons* desnecessários. Isso ajuda a minimizar a superfície de ataque.

Configuração de Logs: Garantir que os registros (*logs*) sejam armazenados adequadamente, com as permissões corretas e que haja monitoramento.



## 5. Auditoria e Monitoramento

Auditoria do Sistema: Configuração de regras de auditoria usando `auditd` para registrar eventos críticos no sistema.

Monitoramento e Resposta a Incidentes: Configuração de sistemas para monitorar, detectar e responder a atividades suspeitas ou maliciosas.

## 6. Segurança de Aplicações

Atualizações e *Patching*: Garantir que o sistema esteja atualizado com os últimos *patches* (correções ou modificações) de segurança.

Configuração Segura de Serviços Comuns: Recomendações específicas para configurar os serviços mais comuns, como por exemplo DNS, HTTP, e-mail, entre outros.

## Agrupamento de Servidores (*Clustering*)

*Clustering* é a conexão de vários servidores (nós) para trabalhar juntos como se fossem um único sistema, aumentando a disponibilidade, escalabilidade e desempenho de aplicações. Esse tipo de configuração é amplamente utilizado em data centers, ambientes de alto desempenho e sistemas críticos para negócios, oferecendo redundância e distribuição da carga de trabalho.

Existem vários tipos de *clustering* que podem ser implementados no Linux, dependendo do objetivo. Os principais tipos de *clustering* e as tecnologias comuns associadas são:

1. Cluster de Alta Disponibilidade (HA - *High Availability*): Tem como objetivo garantir que as aplicações ou serviços permaneçam disponíveis, mesmo quando um ou mais nós falham. Isso é realizado através da replicação de serviços e comutação automática para nós redundantes quando ocorrem falhas. Ferramentas comuns para clusters HA em Linux são:
  - Pacemaker: Um dos gerenciadores de clusters HA mais utilizados no Linux. Ele gerencia recursos e serviços, garantindo que, em caso de falha, os serviços sejam transferidos para outros nós;
  - Corosync: Um framework utilizado junto com o Pacemaker para fornecer comunicação e quorum entre os nós do cluster;
  - DRBD (*Distributed Replicated Block Device*): Uma solução de replicação de discos que permite a replicação síncrona de dados entre os nós, garantindo que dados críticos estejam disponíveis em múltiplos nós;
  - Heartbeat: Utilizado para monitorar a saúde dos nós no *cluster* e, em caso de falha, migrar os serviços para outro nó.
2. Cluster de Balanceamento de Carga: Distribui as requisições de uma aplicação entre vários nós do cluster para melhorar o desempenho e garantir que nenhum servidor fique sobrecarregado. Esse tipo de cluster é comum em servidores Web e sistemas de banco de dados. Algumas ferramentas de balanceamento de carga em Linux são:



- HAProxy: Um dos balanceadores de carga mais usados no Linux, especialmente em sistemas Web. Ele distribui requisições entre múltiplos servidores *backend*;
  - NGINX: Além de servidor Web, pode atuar como balanceador de carga HTTP e TCP/UDP;
  - Keepalived: Ferramenta usada em combinação com balanceadores de carga para garantir *failover* automático de IPs e alta disponibilidade de serviços.
3. Cluster de Alta Performance (HPC - High-Performance Computing): Projetado para agregar o poder computacional de vários nós para executar tarefas intensivas em processamento, como simulações científicas, análises de grandes volumes de dados etc. Algumas ferramentas para HPC em Linux são:
- MPI (*Message Passing Interface*): Um padrão usado em clusters HPC para comunicação entre nós que executam tarefas paralelas. Implementações populares incluem OpenMPI e MPICH;
  - SLURM (*Simple Linux Utility for Resource Management*): Um sistema de gerenciamento de workload para HPC, que gerencia o agendamento de *jobs* e a alocação de recursos no cluster;
  - OpenMP: Um modelo de programação para aplicações paralelas em sistemas com múltiplos processadores.
4. Cluster de Armazenamento (*Storage Cluster*): Permite que várias máquinas trabalhem juntas para fornecer um sistema de armazenamento distribuído. Isso aumenta a capacidade de armazenamento e melhora a redundância dos dados. Algumas ferramentas comuns para clusters de armazenamento em Linux são:
- Ceph: Um sistema de armazenamento distribuído altamente escalável que pode ser utilizado para armazenar blocos, objetos ou arquivos, oferecendo resiliência e balanceamento automático de dados;
  - GlusterFS: Outro sistema de armazenamento distribuído que permite o uso eficiente de várias máquinas para criar um *pool* de armazenamento unificado;
  - Lustre: Utilizado principalmente em ambientes HPC, fornecendo armazenamento em larga escala com alta taxa de transferência para sistemas com processamento intensivo.
5. Cluster de Virtualização (*Cloud*): Permite a execução de máquinas virtuais distribuídas e a gestão de recursos de infraestrutura de maneira centralizada e escalável. Algumas ferramentas comuns para clusters de virtualização são:
- OpenStack: Uma plataforma de nuvem *opensource* que permite a criação de infraestruturas como serviço (IaaS) utilizando clusters de servidores físicos;



- Kubernetes: Principalmente utilizado para a orquestração de contêineres, mas também para virtualização de serviços distribuídos;
- Proxmox VE: Uma solução de virtualização que oferece *clustering* para gerenciar e distribuir máquinas virtuais e contêineres.



## Conceitos do FreeBSD

O FreeBSD é um sistema operacional de código aberto baseado no BSD, uma versão do Unix desenvolvida na Universidade da Califórnia, Berkeley. Trata-se de um sistema operacional conhecido por sua robustez, segurança e desempenho. É amplamente utilizado em servidores, firewalls e outros sistemas críticos.

Algumas características do FreeBSD são:

- Sistema de Arquivos: Suporta o ZFS (*Zettabyte File System*), um sistema de arquivos avançado que oferece proteção contra corrupção de dados, suporte a grandes volumes de dados, *snapshots* etc.;
- Portabilidade: Roda em diversas arquiteturas de hardware, sendo uma escolha flexível para diferentes tipos de aplicações;
- *Ports Collection* (ou simplesmente, *ports*): Uma grande coleção de aplicativos que podem ser instalados a partir do código-fonte de maneira simplificada. Também existem pacotes pré-compilados disponíveis para uma instalação mais rápida;
- Segurança: Devido ao seu desenvolvimento rigoroso e ao foco na estabilidade e na proteção contra vulnerabilidades, o FreeBSD é bastante utilizado em ambientes que requerem segurança elevada;
- *Jails*: São uma forma de virtualização leve que permite a criação de ambientes isolados dentro de um único sistema operacional.

Os principais elementos que caracterizam uma *jail* são:

- Sistema de arquivos Isolado: cada *jail* tem seu próprio sistema de arquivos, que é separado do sistema host. O diretório raiz (*/*) da *jail* é montado em um diretório específico do sistema host (ex.: */usr/jails/nome\_da\_jail*);
- Isolamento de processos: processos dentro de uma *jail* são isolados e não podem ver ou interagir diretamente com processos fora dela ou em outras *jails*;
- Configuração de rede isolada: uma *jail* pode ter seu próprio endereço IP, que é configurado separadamente do sistema host;
- Isolamento de usuários e permissões: *jails* possuem seu próprio conjunto de usuários e grupos, que são independentes dos usuários e grupos no sistema host;





- Configuração e limitações de recursos: é possível configurar limites de recursos para *jails*, como limites de uso de processador, memória, e entrada/saída de disco, para controlar o impacto de cada jail no sistema host;
- Segurança e controle de acesso: são os principais motivos para usar *jails*, pois elas oferecem um ambiente isolado onde vulnerabilidades de segurança são confinadas à jail, sem afetar o sistema host ou outras *jails*;
- Facilidade de gerenciamento: *jails* são relativamente fáceis de criar, configurar e gerenciar usando ferramentas nativas do FreeBSD (ex.: ezjail, iocage, ou comandos, como jail e jexec);
- Isolamento de sistema: cada *jail* pode executar uma versão específica do FreeBSD, independentemente da versão do sistema host, desde que seja compatível com a ABI (*Application Binary Interface*);
- Integração com o sistema host: mesmo sendo isoladas, as *jails* ainda compartilham o kernel do sistema host. Ou seja, não há overhead significativo como ocorre em máquinas virtuais, resultando em uma virtualização muito leve;
- Flexibilidade de configuração: o administrador pode configurar uma *jail* para ser extremamente restritiva, onde apenas um conjunto muito limitado de comandos e acessos são permitidos, ou mais permissiva, com acesso a mais recursos do sistema host.

## Sistemas de Arquivos no FreeBSD

Na sequência veremos algumas características dos principais sistemas de arquivos utilizados no FreeBSD.

### ZFS (*Zettabyte File System*)

O ZFS é um sistema de arquivos avançado e um gerenciador de volumes lógico originalmente criado pela Sun Microsystems. Ele foi projetado para superar as limitações dos sistemas de arquivos tradicionais e oferece uma combinação de gerenciamento de armazenamento, proteção de dados e desempenho superior. Trata-se de uma escolha popular para sistemas FreeBSD que exigem alta disponibilidade, integridade de dados, e escalabilidade. As principais características do ZFS são:

- Gerenciamento de volume integrado: combina o sistema de arquivos e o gerenciador de volumes em uma única estrutura, eliminando a necessidade de usar ferramentas separadas para gerenciar discos e partições;



- Integridade dos dados: utiliza checksums de 256 bits para verificar a integridade de todos os dados armazenados;
- *Snapshots* e clones: permite a criação de *snapshots* e clones (cópias mutáveis de *snapshots*) de forma eficiente;
- RAID-Z: é uma implementação de RAID-5, mas sem o “buraco de escrita” tradicional que pode ocorrer em outros sistemas de arquivos. RAID-Z melhora a redundância e recuperação de falhas de disco sem os problemas de inconsistência que podem ocorrer com outros sistemas de RAID;
- Compressão e deduplicação: oferece compressão de dados em tempo real, reduzindo o uso de espaço em disco sem impacto significativo no desempenho. A deduplicação de dados também é suportada, o que economiza ainda mais espaço ao armazenar apenas uma cópia de blocos de dados idênticos;
- Cópia em gravação (*Copy-on-Write*): quando os dados são modificados em ZFS, ele escreve as novas informações em um novo local e só depois atualiza os metadados para apontar para os novos dados. Isso elimina problemas de corrupção que podem ocorrer se o sistema falhar durante a escrita;
- Escalabilidade: é projetado para ser extremamente escalável, suportando sistemas de arquivos de até 256 trilhões de zettabytes e diretórios contendo até  $2^{48}$  entradas. Ele pode lidar com grandes quantidades de dados e alto número de arquivos com facilidade;
- Facilidade de gerenciamento: Com ZFS, pode-se facilmente adicionar discos a um pool de armazenamento existente sem precisar reformatar ou mover dados;
- Autocorreção de dados: Se o ZFS detectar uma corrupção nos dados, ele tenta corrigi-la automaticamente, restaurando os dados da cópia redundante (se disponível);
- Criptografia: suporta criptografia nativa de dados, permitindo que os dados sejam protegidos em repouso. A criptografia é gerenciada por conjunto de dados, e as chaves podem ser controladas de forma flexível.

## UFS2 (Unix File System version 2)

O UFS2 é uma versão aprimorada do sistema de arquivos UFS (Unix File System), amplamente utilizado no FreeBSD e outros sistemas operacionais derivados do BSD. UFS2 foi introduzido no FreeBSD 5.0 como uma evolução do UFS1, adicionando várias melhorias e capacidades para atender às necessidades de sistemas modernos. Vamos às características:



- Suporte a volumes maiores: aumenta o tamanho máximo de arquivos e sistemas de arquivos, permitindo volumes de até 16 exabytes e arquivos de até 2 exabytes, em comparação aos limites muito menores do UFS1;
- Timestamps melhorados: suporta timestamps de 64 bits, oferecendo maior precisão e eliminando o problema do "Ano 2038" para timestamps de arquivos, que afetava sistemas de arquivos mais antigos;
- Atributos estendidos: inclui suporte a atributos estendidos, permitindo armazenar metadados adicionais para arquivos, como informações de segurança e permissões mais granulares;
- *Snapshots*: suporta *snapshots*, uma funcionalidade que permite tirar "fotografias" do sistema de arquivos em um determinado momento, extremamente útil para backups e recuperação de sistemas;
- Compatibilidade retroativa: mantém compatibilidade com UFS1, permitindo que sistemas existentes sejam migrados para o novo formato sem a necessidade de reformatação completa;
- Desempenho: oferece melhorias de desempenho em relação ao UFS1, especialmente em sistemas de arquivos maiores ou com grandes quantidades de arquivos.

## Instalação do FreeBSD

Para a instalação do FreeBSD, o mais comum é baixar a imagem (arquivo ISO) através do *site* oficial ([freebsd.org](http://freebsd.org)), de acordo com a arquitetura desejada (ex.: amd64). Depois pode-se criar uma mídia de instalação, através da gravação da imagem ISO em um DVD ou a criação de um pen drive bootável com a ISO utilizando ferramentas como Rufus (Windows) ou dd (Linux).

Com a mídia de instalação inserida no computador, deve-se configurar o BIOS/UEFI para dar boot pelo dispositivo correspondente. No menu de inicialização do FreeBSD, deve-se selecionar a opção para iniciar a instalação. Na sequência, são mostradas instruções para configurar o layout do teclado, a rede, o fuso horário, e partições do disco. Depois o usuário pode selecionar os pacotes básicos que deseja instalar (ex.: utilitários base, documentação, entre outros), além de definir a senha do root.

## Configurações do FreeBSD

Após a instalação, é desejável que se atualize o sistema para garantir que as últimas correções de segurança estejam instaladas. Para a instalação e alterações de aplicativos, Pkg é o gerenciador de pacotes no FreeBSD. Os seguintes comandos podem ser executados (lembrando que # no início do prompt indica que se trata do usuário *root*):



```
# freebsd-update fetch  
# freebsd-update install  
# pkg update  
# pkg upgrade
```

Para a configuração de Rede, deve-se editar o arquivo `/etc/rc.conf`. Por exemplo, para configurar uma interface de rede com DHCP:

```
ifconfig_em0="DHCP"
```

Para a instalação de pacotes adicionais (com o gerenciador de pacotes `pkg`), é utilizado o seguinte comando (ex.: para instalar o `vim`):

```
# pkg install vim
```

A qualquer momento pode-se ativar e configurar serviços adicionais, conforme a necessidade. Por exemplo, para ativar o servidor SSH:

```
# sysrc sshd_enable="YES"  
# service sshd start
```

## Administração do FreeBSD

Vamos ver algumas atividades da administração do FreeBSD na sequência.

Para [adicionar novos usuários](#):

```
# adduser
```

Para [modificar ou remover usuários](#):

```
# pw usermod nome_do_usuario -G wheel
```

```
# pw userdel nome_do_usuario
```

Para [verificar o estado e para reiniciar serviços](#):

```
# service nome_do_servico status
```



```
# service nome_do_serviço restart
```

Para o **monitoramento do sistema**, algumas ferramentas são:

- top: monitora o uso de recursos do sistema em tempo real, exibindo uma lista dinâmica dos processos em execução, ordenados por uso de processador (CPU) por padrão, além de outras informações importantes sobre o sistema;
- htop: alternativa avançada ao comando top, com uma interface mais amigável e recursos adicionais;
- vmstat: "*virtual memory statistics*" - ferramenta útil para monitorar a atividade do sistema em termos de processos, memória, swap, entrada/saída (I/O), CPU e outras estatísticas relacionadas ao desempenho;
- iostat: ferramenta útil para monitorar a atividade de entrada/saída (I/O) do sistema em relação a dispositivos de armazenamento, como discos rígidos e SSDs. Fornece estatísticas detalhadas sobre a taxa de transferência de dados e a utilização dos dispositivos, o que ajuda a identificar gargalos de desempenho.

Para o **backup e recuperação de backups** algumas ferramentas são:

- rsync: ferramenta poderosa para a sincronização e a cópia de arquivos e diretórios entre diferentes sistemas ou locais. Possui suporte à transferência incremental e preservação de atributos de arquivos;
- tar: utilizado para criar, extrair e manipular arquivos "tarballs" (coleções de arquivos agrupados em um único arquivo para fins de distribuição ou backup). O nome "tar" vem de *tape archive*, pois originalmente era usado para armazenar dados em fitas;
- dump: ferramenta poderosa para realizar backups de sistemas de arquivos. É projetada para fazer backups incrementais ou completos de sistemas de arquivos Unix e é particularmente útil em ambientes onde a confiabilidade e a precisão dos backups são fundamentais.

## Diferenças entre o FreeBSD e o Linux

Tanto o FreeBSD como o Linux são sistemas operacionais baseados em Unix, porém existem diferenças significativas em termos de origem, licenciamento, kernel, exemplos de uso, entre outros aspectos. Algumas das principais diferenças são mostradas a seguir.

### Origem e História



O FreeBSD é derivado do BSD (*Berkeley Software Distribution*), que é uma versão do Unix desenvolvida na Universidade da Califórnia, Berkeley. O projeto FreeBSD começou em 1993 e é mantido pela FreeBSD Foundation e pela comunidade de desenvolvedores.

O Linux foi criado por Linus Torvalds em 1991 como um clone do Unix, inspirado no sistema Minix. O núcleo do sistema (*kernel*) é desenvolvido por Linus Torvalds e colaboradores ao redor do mundo, enquanto várias distribuições Linux são mantidas por diferentes organizações e comunidades.

## Licenciamento

O FreeBSD utiliza a licença BSD, que é mais permissiva e permite que o código-fonte seja usado e redistribuído com menos restrições, inclusive em software proprietário.

O Linux utiliza a Licença Pública Geral GNU (GPL), que exige que qualquer software derivado do código-fonte do Linux também seja distribuído sob a mesma licença, garantindo que o código-fonte permaneça livre.

## Kernel e Sistemas de Arquivos

O FreeBSD inclui um kernel monolítico juntamente com um conjunto completo de ferramentas de usuário, formando um sistema operacional coeso e integrado. Os sistemas de arquivos UFS (*Unix File System*) e ZFS (*Zettabyte File System*) são comuns.

O Linux é apenas o kernel, e as distribuições combinam este kernel com várias ferramentas de usuário GNU e outros softwares. Suporta vários sistemas de arquivos, incluindo Ext2, Ext3, Ext4, Btrfs, XFS, BTRFS, entre outros.

## Comunidade e Suporte

No FreeBSD a comunidade é menor, porém dedicada, com um foco forte em desenvolvimento e documentação de alta qualidade. O suporte ocorre principalmente através de documentação oficial, listas de e-mails e fóruns.

No Linux a comunidade é muito grande e diversificada, com uma vasta gama de contribuições de indivíduos, organizações e empresas. Em relação ao suporte, além da documentação oficial, há muitos fóruns, listas de discussão e empresas que oferecem suporte comercial.

## Aplicações e Uso

O FreeBSD geralmente é usado em servidores de rede, firewalls, roteadores e sistemas de armazenamento devido à sua estabilidade e desempenho de rede. Alguns exemplos incluem empresas como Netflix e Yahoo para servir conteúdos de alta demanda.



O Linux é usado em uma variedade de ambientes, desde servidores de alto desempenho até dispositivos móveis (como Android), desktops e dispositivos IoT (Internet das Coisas). Por exemplo, o Linux é utilizado por gigantes da tecnologia como Google e Amazon, além de ser o núcleo do sistema operacional Android.





## QUESTÕES COMENTADAS - CEBRASPE

1. (CEBRASPE/EMAP/2018) O kernel do sistema operacional Linux tem a função de interpretar os comandos executados em um terminal.

Comentários:

Quem tem essa função é o shell! O kernel é o núcleo do sistema operacional!

Gabarito: Errado

2. (CEBRASPE/FUB/2018) O Linux é um sistema operacional em que cada usuário consegue ter apenas um processo ativo por vez, processo esse que é iniciado automaticamente quando o sistema é carregado.

Comentários:

O Linux permite que vários processos sejam executados ao mesmo tempo (multitarefa), mesmo que haja apenas um processador na máquina. O sistema operacional faz o escalonamento dos processos no processador, dando a ilusão de um paralelismo (ocorre um pseudoparalelismo). O comando "ps", por exemplo, pode ser executado e você verá diversos processos em execução. Quando há a criação de um processo novo, ele passa a ser filho de outro e a chamada de sistema utilizada é a fork().

Gabarito: Errado

3. (CEBRASPE/SLU-DF/2019) Windows e Linux são classificados como sistemas operacionais de tempo real crítico, porque fornecem garantias absolutas de que todas as suas ações ocorrerão dentro de intervalos de tempo determinados.

Comentários:

Windows e Linux são sistemas operacionais interativos, pois recebem instruções por teclado, mouse, caneta óptica etc., realiza as tarefas, mostra os resultados via impressora, monitor de vídeo etc. Logo, há muita interação, o que é diferente de sistemas de tempo real ou ainda os sistemas de lote (são as 3 classificações relacionadas a esse assunto).

Gabarito: Errado

4. (CEBRASPE/PG-DF/2021) Acerca da administração de sistemas Windows e Unix/Linux, julgue o seguinte item.

No sistema Linux Ubuntu, o comando `sudo apt-get upgrade` apresenta as atualizações de sistema disponíveis com opção para seleção e execução individualizada de cada atualização.

Comentários:

APT: resolve dependências para sistemas baseados em Debian (incluindo o Ubuntu). Possui uma sintaxe muito simples. Vamos a alguns exemplos para:



- atualizar os repositórios de software: `sudo apt-get update` ou `sudo apt update` (a lista com os repositórios devem estar no arquivo `"/etc/apt/sources.list"`);
- atualizar seus pacotes já instalados (softwares): `sudo apt-get upgrade` ou `sudo apt upgrade`;
- uma atualização mais completa (tenta atualizar pacotes para a versão mais recente e remover dependências mais antigas ou não utilizadas): `sudo apt-get dist-upgrade` ou `sudo apt full-upgrade`;
- atualização dos repositórios e atualizar também os pacotes já instalados: `sudo apt-get update && sudo apt-get upgrade` ou `sudo apt update && sudo apt upgrade`;
- instalar um software: `sudo apt-get install nome_do_pacote` ou `sudo apt install nome_do_pacote`;
- remover um pacote: `sudo apt-get remove nome_do_pacote` ou `sudo apt remove nome_do_pacote`.

Gabarito: Errado

5. (CEBRASPE/TCE-RJ/2021) Acerca dos sistemas operacionais Windows e Linux e da interoperabilidade entre esses sistemas, julgue o item que se segue.

No processo de inicialização de um sistema Linux, a função de carregar e descompactar a imagem do kernel é responsabilidade do carregador de boot de primeiro estágio.

Comentários:

Primeiro Estágio: Em sistemas usando o método BIOS/MBR, o carregador de boot reside no primeiro setor do disco rígido, o Master Boot Record (MBR). Nesse estágio, o carregador de boot examina a tabela de partições e encontra uma partição inicializável. Depois, procura o carregador de inicialização do segundo estágio (ex.: GRUB) e o carrega na memória RAM.

Para sistemas que usam o método EFI/UEFI, o firmware UEFI lê seus dados do gerenciador de boot para determinar qual aplicativo UEFI deve ser lançado e de onde, ou seja, de qual disco e partição a partição EFI pode ser localizada. O firmware então inicia o aplicativo UEFI (ex.: GRUB), conforme definido na entrada de inicialização no gerenciador de inicialização do firmware.

Segundo Estágio: O carregador de inicialização do segundo estágio (ex.: GRUB) reside em `/boot`. Uma tela inicial é exibida, o que permite escolher qual sistema operacional será inicializado. Depois de escolhido o sistema operacional, o carregador de boot carrega o kernel do sistema operacional selecionado na memória RAM e passa o controle para ele.

Gabarito: Errado

6. (CEBRASPE/TRT8/2022) Assinale a opção que apresenta o arquivo utilizado para inicializar um ambiente de computador em um sistema operacional Linux.

a) `/etc/group`



- b) /etc/passwd
- c) /etc/syslog.conf
- d) /etc/startup
- e) /etc/profile

#### Comentários:

Vamos verificar para que serve cada arquivo:

- /etc/group: arquivo de sistema que define os grupos aos quais os usuários pertencem em sistemas UNIX, como o Linux e o BSD;
- /etc/passwd: determina quem pode acessar legitimamente o sistema e o que eles podem fazer uma vez dentro;
- /etc/syslog.conf: usado para controlar a saída dos arquivos de log do daemon syslogd;
- /etc/startup: aparentemente não existe!
- /etc/profile: contém comandos que são executados para todos os usuários do sistema no momento do login. Somente o usuário root pode ter permissão para modificar esse arquivo.

Gabarito: Letra E



## QUESTÕES COMENTADAS – FGV

1. (FGV/IBGE/2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:

- A) create\_process;
- B) new\_process;
- C) fork;
- D) spawn;
- E) duplicate.

### Comentários:

No Linux a chamada de sistema fork() faz uma "bifurcação" de um processo, sendo que o novo vira o filho do outro. Ex.: em um shell você executa algum comando, então o fork() será chamado, o shell será o processo pai do processo novo (o comando digitado).

Gabarito: Letra C

2. (FGV/MPE-AL/2018) Instalar, atualizar e remover pacotes no sistema operacional CentOS 7 é uma tarefa frequente para desenvolvedores de sistemas. Por isso, eventualmente podem ocorrer dúvidas sobre se determinado pacote está instalado ou qual é a versão que está sendo utilizada. Para dirimir essas dúvidas, sobre o pacote httpd devemos utilizar o comando

- A) rpm -q httpd
- B) yum check httpd
- C) apt-get find httpd
- D) find -iname httpd
- E) crontab -l httpd

### Comentários:

Para quem não lembra da parte da aula sobre o gerenciador utilizado no CentOS, pelo menos deve lembrar que a resposta só pode ser uma das 3 primeiras, certo? Senão, a coisa está feia. Se também lembrar que "-q" é em relação a "query" (pesquisa), aí fica mais tranquilo para marcar a alternativa A.

Gabarito: Letra A

3. (FGV/COMPESA/2018) Com relação às características e tarefas de administração do sistema operacional Linux, analise as afirmativas a seguir.



I. O esquema de partição GPT (Guid Partition Table) oferece a possibilidade de até 128 partições primárias, com tamanhos maiores do que 2TB cada.

II. Ext4, XFS e exFAT são exemplos de sistemas de arquivo com jornal, utilizados pelo Linux.

III. A área de swap no Linux pode ser uma partição dedicada de swap, um arquivo de swap ou uma combinação de arquivos e partições de swap.

Está correto o que se afirma em

- A) I, somente.
- B) II, somente.
- C) III, somente.
- D) I e III, somente.
- E) I, II e III.

Comentários:

(I) Ao contrário da MBR, que oferece 4, a GPT realmente permite até 128, está correta! (II) exFAT é um sistema de arquivos do Windows, não do Linux! (III) Embora muitos pensem que tem que ser uma partição, pode ser um arquivo de swap ou ainda uma combinação.

Gabarito: Letra D

4. (FGV/AL-RO/2018) Wallace é administrador de um servidor com sistema operacional CentOS e deseja compartilhar um diretório com os integrantes da empresa por meio do protocolo NFS.

Para que a configuração desse compartilhamento possa ser efetuada, Wallace deve editar o arquivo `/etc/`

- A) `services`
- B) `modules`
- C) `hosts`
- D) `export`
- E) `fstab`

Comentários:

O Linux geralmente é intuitivo, então digamos que você tenha esquecido da aula, mas lembra que o NFS é um sistema de arquivos de rede e os diretórios a serem "exportados" devem estar listados no arquivo "exports" (falta um "s" na alternativa, mas tudo bem).

Gabarito: Letra D



5. (FGV/AL-RO/2018) Roberto trabalha como administrador de redes em uma empresa de cosméticos e sua principal função é a configuração de estações de trabalho dos seus colaboradores. Rotineiramente, Roberto instala e remove pacotes nas estações de trabalho às quais ele dá suporte.

Assinale a opção que indica o comando utilizado por Roberto para desinstalar pacotes RPM nas estações de trabalho com sistema operacional Linux CentOS.

- A) rpm -Uvh pacote
- B) rpm -e pacote
- C) rpm -ivh pacote
- D) rpm -qi pacote
- E) rpm -qa | grep pacote

#### Comentários:

Algumas opções do rpm:

-i: instalação simples.

-v: exibe detalhes da instalação.

-h: mostra o caractere "#", enquanto o programa é instalado.

-U: atualização do programa de uma versão anterior para uma mais recente.

--nodeps: não procura dependências.

--force: força a instalação.

--root diretório: utiliza o sistema com raiz em diretório para todas as operações.

-e: desinstala o pacote.

--import: importa a chave pública GPG do distribuidor de um pacote.

--help: exibe uma mensagem de ajuda.

-q: consulta (query) se um pacote já está instalado.

Gabarito: Letra B

6. (FGV/TRT-MA/2022) O RPM Package Manager é um sistema executado em várias distribuições Linux e UNIX. O conjunto completo de modos básicos de operação do RPM é

- A) consulta, atualização e instalação de pacotes.
- B) instalação, atualização, criação e depuração de pacotes.



- C) valoração, reinstalação, consulta e desinstalação de pacotes.
- D) instalação, desinstalação, valoração, consulta, monitoramento e depuração de pacotes.
- E) criação, instalação, desinstalação, atualização, consulta e verificação de pacotes.

### Comentários:

RPM: criado pela Red Hat para instalar pacotes ".rpm", sendo utilizado por várias distribuições, como Red Hat Enterprise/Fedora Core, Mandriva, SuSe, entre outras. O conjunto completo de modos básicos de operação do RPM é a criação, instalação, desinstalação, atualização, consulta e verificação de pacotes. O principal comando do gerenciador de pacotes RPM, é o rpm.

Sintaxe: rpm [opções] nome\_pacote

Algumas opções:

- i: instalação simples.
- v: exibe detalhes da instalação.
- h: mostra o caractere "#", enquanto o programa é instalado.
- U: atualização do programa de uma versão anterior para uma mais recente.
- nodeps: não procura dependências.
- force: força a instalação.
- root diretório: utiliza o sistema com raiz em diretório para todas as operações.
- e: desinstala o pacote.
- import: importa a chave pública GPG do distribuidor de um pacote.
- help: exibe uma mensagem de ajuda.
- q: consulta (*query*) se um pacote já está instalado.

Exemplos:

- #rpm -q httpd → Consulta se o pacote está instalado.
- # rpm -ivh tree.i386.rpm → Instala o pacote, exibindo detalhes da instalação e mostra o caractere "#", enquanto o programa é instalado.
- # rpm -e tree → Desinstala o pacote tree.

Gabarito: Letra E





## QUESTÕES COMENTADAS – FCC

1. (FCC/TRF5/2017) Considere a figura abaixo que mostra a arquitetura do sistema operacional Linux



A caixa

A) I representa a camada responsável pela interface entre o hardware e as aplicações. Dentre suas funções encontram-se gerenciamento de I/O, manutenção do sistema de arquivos, gerenciamento de memória e swapping, controle da fila de processos etc.

B) II representa a camada que permite o acesso a recursos através da execução de chamadas feitas por processos. Tais chamadas são geradas por funções padrão suportadas pelo kernel. Dentre suas funções estão habilitar funções padrão como open, read, write e close e manter a comunicação entre as aplicações e o kernel.

C) I é um processo que executa funções de leitura de comandos de entrada de um terminal, interpreta-os e gera novos processos, sempre que requisitados. É conhecido também como interpretador de comandos.

D) II é um processo que realiza modificações no shell, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o shell, o usuário torna o Linux enxuto e adaptável.

E) I é um processo que realiza modificações no kernel, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o kernel, o usuário torna o Linux enxuto e adaptável.

Comentários:

A caixa I pode ser um shell que recebe comandos do teclado, interpreta e executa, gerando novos processos (filhos) através da chamada de sistema fork(). Ex.: execução do comando "ls" no shell.

Gabarito: Letra C



2. (FCC/SEFAZ-BA/2019) Um Auditor Fiscal da área de Tecnologia da Informação deseja desinstalar um pacote chamado java-1.6.0-openjdk.x86\_64 em linha de comando, como usuário root, no Red Hat Enterprise Linux 6. Para isso, terá que utilizar o comando

- A) apt-get uninstall java-1.6.0-openjdk.x86\_64
- B) apt-get remove -rf java-1.6.0-openjdk.x86\_64
- C) rm -rf java-1.6.0-openjdk.x86\_64
- D) yum remove java-1.6.0-openjdk.x86\_64
- E) apt-get -rf java-1.6.0-openjdk.x86\_64

#### Comentários:

YUM: resolve dependências de pacotes para distribuições que utilizam o RPM. É um gerenciador de pacotes padrão incluído em algumas distribuições baseados no REDHAT, incluindo o Fedora e CentOS. Muito parecido com o APT, vejamos alguns comandos:

- Atualização dos repositórios do YUM: sudo yum update;
- Instalação de um pacote: sudo yum install nome\_do\_pacote;
- Remoção de um pacote: sudo yum remove nome\_do\_pacote;
- Busca de um pacote que será instalado: sudo yum search nome\_do\_pacote.

Gabarito: Letra D

3. (FCC/Prefeitura de Manaus-AM/2019) Em computador com sistema operacional Linux e o OpenLDAP instalado, o programador editou o arquivo ldap.conf para configurar

- A) a autenticação do servidor.
- B) a execução do serviço do LDAP.
- C) o banco de dados a ser utilizado.
- D) a geração de índices do banco de dados.
- E) o acesso dos clientes ao diretório.

#### Comentários:

O principal arquivo de configuração (pelo menos para concursos) é o "/etc/ldap.conf", que é usado para configurar os padrões a serem aplicados quando da execução dos clientes LDAP.

Gabarito: Letra E



## QUESTÕES COMENTADAS – VUNESP

1. (VUNESP/Câmara de Sertãozinho-SP/2019) Considere que o usuário de um computador com sistema operacional Linux deseja executar um comando em segundo plano (background) em um terminal Shell. Para efetivar essa ação, o usuário deve executar o comando

- A) e teclar a combinação Ctrl+c
- B) e teclar a combinação Ctrl+x
- C) seguido de bg
- D) seguido de &
- E) seguido de !

### Comentários:

O mais utilizado para executar em background é utilizar "&" logo após o nome do binário ou script. Por exemplo, se você deseja executar um script "aprovacao.sh", que vai realizar vários cálculos e no fim escrever em um arquivo, não é necessário que você fique com o shell "travado", aguardando a conclusão da tarefa. Então é só digitar (note que o shell não fica esperando a conclusão do script, já aguarda o próximo comando):

```
estrategia:~$ aprovacao.sh &  
estrategia:~$
```

Gabarito: Letra D



## QUESTÕES COMENTADAS – CESGRANRIO

1. (CESGRANRIO/Petrobras/2011) Em um sistema Unix, um arquivo de script chamado teste.sh foi copiado para o diretório /tmp. No shell do sistema, o usuário submeteu dois comandos: cd /tmp e teste.sh. Após a execução do segundo comando, o shell informou uma mensagem de erro, indicando comando não encontrado. O que deve ser feito para corrigir o problema que gerou essa mensagem?

- A) Certificar que o usuário não entrou com letras maiúsculas.
- B) Certificar que o arquivo tem permissão para ser executado.
- C) Omitir a extensão .sh ao entrar com o nome do script.
- D) Incluir ./ antes do nome do script.
- E) Mover o arquivo para o diretório home e executá-lo.

### Comentários:

No Linux, quando um executável (binário ou um script com permissão de execução) não estiver no path (lista de diretórios que permitem a execução), é possível executar através de "./" antes do nome do executável.

Gabarito: Letra D

2. (CESGRANRIO/Petrobras/2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado.

Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.
- E) na linha de comando, após o nome do arquivo que contém o script.

### Comentários:

É necessário definir qual Shell será utilizado na primeira linha do arquivo (vamos escolher o bash, mas poderia ser outro interpretador de comandos, como sh, ksh ou csh):

```
#!/bin/bash
```

Gabarito: Letra A



3. (CESGRANRIO/Transpetro/2023) Muitos sistemas operacionais, como o Linux e o Windows, suportam links simbólicos em seus sistemas de arquivos, o que permite que um arquivo apareça em mais de um diretório. Esses links apontam para outro arquivo e podem ser usados, na maioria das vezes, como o arquivo original.

No Linux, o que acontece com o link simbólico se o arquivo original for apagado?

- A) Aponta para um novo arquivo criado automaticamente.
- B) É apagado automaticamente.
- C) Passa a apontar para /mnt/null.
- D) Torna-se um "link quebrado".
- E) Torna-se um hard link.

#### Comentários:

No Linux, os links simbólicos e *hard links* são métodos para criar referências a arquivos existentes no sistema de arquivos, porém eles funcionam de maneiras diferentes e têm usos distintos. Vamos ver como funciona o link simbólico a seguir.

Link Simbólico (*Soft Link* ou *Symlink*) é um tipo especial de arquivo que aponta para outro arquivo ou diretório. Ele age como um "atalho" (como é conhecido no Windows) para o arquivo original.

- Comportamento: Se o arquivo original for alterado ou excluído, o link simbólico continuará existindo, mas ficará "quebrado" e não funcionará mais corretamente;
- Uso: Links simbólicos são úteis quando é necessário criar referências em locais diferentes no sistema de arquivos que apontem para um único arquivo ou diretório. Por exemplo, pode haver um diretório em /home/evandro/documentos e criar um link simbólico para ele em /home/evandro/desktop;
- Comando: `ln -s`, com a seguinte sintaxe:

`ln -s /caminho/para/arquivo_original /caminho/para/link_simbólico`

Gabarito: Letra D



## QUESTÕES COMENTADAS – MULTIBANCAS

1. (CESGRANRIO/Petrobras/2011) Em um sistema Unix, um arquivo de script chamado teste.sh foi copiado para o diretório /tmp. No shell do sistema, o usuário submeteu dois comandos: cd /tmp e teste.sh. Após a execução do segundo comando, o shell informou uma mensagem de erro, indicando comando não encontrado. O que deve ser feito para corrigir o problema que gerou essa mensagem?

- A) Certificar que o usuário não entrou com letras maiúsculas.
- B) Certificar que o arquivo tem permissão para ser executado.
- C) Omitir a extensão .sh ao entrar com o nome do script.
- D) Incluir ./ antes do nome do script.
- E) Mover o arquivo para o diretório home e executá-lo.

### Comentários:

No Linux, quando um executável (binário ou um script com permissão de execução) não estiver no path (lista de diretórios que permitem a execução), é possível executar através de "./" antes do nome do executável.

Gabarito: Letra D

2. (CESGRANRIO/Petrobras/2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado.

Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.
- E) na linha de comando, após o nome do arquivo que contém o script.

### Comentários:

É necessário definir qual Shell será utilizado na primeira linha do arquivo (vamos escolher o bash, mas poderia ser outro interpretador de comandos, como sh, ksh ou csh):

```
#!/bin/bash
```

Gabarito: Letra A



3. (CEPERJ/Rio previdência/2014) Shell script é uma linguagem de script para Linux, nada mais do que comandos do próprio Linux que são executados em uma determinada sequência para uma determinada finalidade. Nesse contexto, duas situações são listadas a seguir.

I- No terminal ou modo gráfico, deseja-se criar um arquivo que possa ser editado para que se torne o primeiro shell script a ser criado, sendo necessário utilizar um comando CMD1.

II- Para que seja possível executar o shell script criado, é preciso atribuir a este o direito de execução; para isso é necessário usar um comando CMD2.

Exemplos de CMD1 e de CMD2 são, respectivamente:

- A) touch shell1.sh e exec +x shell1.sh
- B) touch shell1.sh e chmod +x shell1.sh
- C) create shell1.sh e chmod +x shell1.sh
- D) new shell1.sh e chmod +x shell1.sh
- E) new shell1.sh e exec +x shell1.sh

#### Comentários:

Criando um arquivo:

Há dois modos de realizar essa ação: via modo gráfico ou via terminal. No segundo caso, pode ser utilizado o comando vi, conforme mostrado a seguir (note a "extensão" .sh).

vi script.sh: Será criado e aberto um arquivo de leitura e escrita.

Outra opção é digitar o comando touch:

touch script.sh: cria um arquivo sem abri-lo.

Permissão ao arquivo:

Para começar a editar o arquivo, é necessário conceder a permissão de escrita a ele. Uma opção é utilizar o comando:

chmod 777 script.sh: o comando chmod é utilizado para conceder permissões em diretórios e arquivos, enquanto o valor 777 permite que o usuário tenha total liberdade para editar o arquivo (na verdade dá liberdade total a qualquer usuário do sistema! Mas nosso foco agora não é estudar o chmod).

Pensando mais em segurança, pode-se, por exemplo, liberar apenas a execução (x):

chmod +x script.sh: como não foi especificado para quem foi liberada a execução (usuário, grupo ou todos), todos devem ter a permissão de execução.

Gabarito: Letra B





4. (FUMARC/AL-MG/2014) No Linux, considere a existência de um arquivo "arquivo.txt" no diretório corrente, com o seguinte conteúdo:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Deseja-se produzir, a partir do arquivo, a seguinte saída:

1 - 10 2 - 20 3 - 30

O script shell usado com comandos AWK que produz a saída desejada é:

- A) `awk arquivo.txt '{ print $2" - "$4 }'`
- B) `cat arquivo.txt | awk '{ print $2" - "$4 }'`
- C) `cat arquivo.txt | awk '{ print $1" - "$3"\n" }'`
- D) `cat arquivo.txt | awk '{ printf($2" - "$4) }'`

#### Comentários:

O comando `cat` é usado para unir, criar e exibir arquivos. No caso dessa questão, serve para exibir.

O `awk` é um utilitário especializado em manipulação de texto.

O pipe ("`|`") serve para uma comunicação entre os processos, ou seja, a saída do que está na esquerda dele é passada para o que está à direita.

Então:

`cat arquivo.txt`: exibe o conteúdo do arquivo e esse conteúdo é passado para:

`awk '{ print $2" - "$4 }'`: imprime o 2º argumento, um traço ("`-`"), o 4º argumento e continua processando o texto até o fim da linha. Logo, imprime:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Lembrando os traços entre o 2º e o 4º argumentos, fica assim:

1 - 10 2 - 20 3 - 30

Gabarito: Letra B

5. (INSTITUTO AOCP/EBSERH/2016) Em sistemas operacionais Linux é possível criar scripts para automatizar tarefas rotineiras. A extensão de arquivo utilizada para Shell Script é

- A) `.bin`
- B) `.sl`
- C) `.bash`



- D) .sh
- E) .gz

Comentários:

Vimos em aula o "script.sh", então o padrão utilizado como "extensão" é .sh. Mas lembre-se que não é obrigatório utilizar qualquer extensão! É apenas uma "convenção".

Gabarito: Letra D

6. (AOCP/EBSERH/2016) O NFS (Network File System) permite compartilhar sistemas de arquivos entre computadores conectados em rede e pode ser parte fundamental da infraestrutura da tecnologia da informação. Sobre NFS, analise as assertivas a seguir e assinale a alternativa que aponta a(s) correta(s).

I. O NFS é considerado sem estado (stateless) e, portanto, quando um servidor NFS volta a funcionar, o estado anterior é restaurado e a informação não é perdida.

II. O NFS pode ser implementado do lado servidor e do lado cliente.

III. O NFS roda sobre o protocolo RPC (Remote Procedure Call), que define um modo independente do sistema para processos se comunicarem através de uma rede de computadores.

IV. O NFS suporta apenas UDP como protocolo de transporte, pois ele apresenta desempenho significativamente melhor que o TCP em redes locais.

- A) Apenas I e II.
- B) Apenas II e IV.
- C) Apenas III.
- D) Apenas III e I.
- E) Apenas I, II e III.

Comentários:

(I) Exato! O NFS é stateless, cada requisição é uma transação independente que não está relacionada a qualquer requisição anterior. Então, quando um servidor NFS volta a funcionar, o estado anterior é restaurado e a informação não é perdida. (II) Existe quem faz requisições de acesso e manipulação de arquivos e diretórios (cliente) e quem recebe as requisições, avalia as permissões, executa e retorna (servidor). (III) O NFS utiliza o RPC (Remote Procedure Call), o cliente faz a chamada do servidor remoto (o servidor até pode estar na mesma máquina, mas o cenário mais comum é remoto). (IV) O protocolo padrão de transporte é o UDP, mas é possível optar pelo TCP.

Gabarito: Letra E



7. (CS-UFG/DEMAE/2017) Na maior parte das distribuições Linux, a memória virtual é uma partição denominada

- A) EXT
- B) SWAP
- C) SETUP
- D) BIOS

Comentários:

Vimos que pode ser partição ou arquivo (ou até ambos), mas geralmente se utiliza uma partição de SWAP.

Gabarito: Letra B

8. (Quadrix/COFECI/2017) Não é possível realizar uma instalação automática, por meio de uma rede, no sistema operacional Linux.

Comentários:

Pode-se realizar uma instalação através de uma mídia (CD, DVD, pen drive) ou através da rede, desde que a placa de rede permita e a configuração esteja adequada para um boot pela rede.

Gabarito: Errado

9. (Colégio Pedro II/Colégio Pedro II/2017) Sistema operacional é um conjunto de programas básicos e utilitários que fazem seu computador funcionar. O Debian é um sistema operacional, em cujo núcleo está o Kernel, o programa mais fundamental no computador e que faz todas as operações mais básicas, permitindo que você execute os outros programas. O Debian atualmente usa o Kernel Linux. Mas um sistema operacional não funciona somente com o Kernel, são necessários utilitários e aplicativos. O Debian utiliza estas ferramentas do projeto GNU. Por esse motivo, muitos utilizadores defendem que devemos chamar o sistema de "Debian GNU/Linux". Reconheça (1) o gerenciador de pacotes (programas) usado no Debian e em distribuições derivadas do Debian (Ubuntu, Knoppix, Big Linux etc.) que utiliza uma lista de dependências para instalar tudo o mais automaticamente possível, e (2) o arquivo de configuração utilizado por este gerenciador de pacote.

- A) Gerenciador (1) - apt-get  
Arquivo (2) - source.list
- B) Gerenciador (1) - apt-get  
Arquivo (2) - package.cfg
- C) Gerenciador (1) - make install  
Arquivo (2) - package.cfg



D) Gerenciador (1) - dpkg

Arquivo (2) - source.list

#### Comentários:

Das alternativas mostradas, apenas "apt-get" é um gerenciador que utiliza uma lista de dependências e o nome dela é "sources.list". Nas alternativas aparece com um "s" a menos, mas tudo bem, relaxe...

Gabarito: Letra A

10. (CS-UFG/SANEAGO-GO/2018) Um utilitário GNU, presente em diversos sistemas operacionais linux, que lista as partições dos discos rígidos, é:

A) fsdisk.

B) mcdisk.

C) parted.

D) gedit.

#### Comentários:

Na verdade, é GParted, mas tudo bem...releve uma letrinha 😊.

Gabarito: Letra C

11. (IF-RS/IF-RS/2018) São tipos de sistemas de arquivos utilizados no Linux:

A) FAT 16, FAT 32, NTFS

B) FAT 32, HFS, ExFAT

C) Ext2, Ext3, Btrfs

D) Ext2, Ext3, HFS

E) Ext2, Ext3, NFS, SMB

#### Comentários:

Família FAT e NTFS= Windows.

HFS = Apple.

NFS e SMB = Sistemas de arquivos de rede.

Só sobrou a letra C! Lembrando: família EXT, Btrfs, XFS, JFS, ReiserFS, entre outros = Linux!

Gabarito: Letra C



12. (COPESE-UFT/UFT/2018) Embora seja possível realizar boot de um sistema Linux a partir de um pendrive, a maioria das instalações do Linux o realiza a partir do disco rígido do computador. Esse processo consiste em duas fases básicas:

1. Executar o carregador de boot a partir do dispositivo de boot;
2. Iniciar o kernel do Linux e iniciar os processos.

Assinale a alternativa que contém um gerenciador de boot para sistemas Linux:

- A) CISC (Complex Instruction Set Computer).
- B) GRUB (Grand Unified Bootloader).
- C) ORM (Object Relational Mapping).
- D) CMS (Content Management System).

Comentários:

O LILO já foi bastante utilizado (mas nem aparece como opção) e o GRUB é mais utilizado ultimamente.

Gabarito: Letra B

13. (COPESE-UFT/UFT/2018) Sistemas operacionais Linux permitem logins simultâneos de diferentes usuários. Assinale a alternativa que contém o comando a ser utilizado para visualizar os usuários logados em um determinado instante.

- A) listusers
- B) ldconfig
- C) who
- D) uname

Comentários:

O Linux é um sistema operacional multiusuário, ou seja, pode ter N usuários logados ao mesmo tempo. Para saber QUEM está logado pode ser utilizado o comando "who".

Gabarito: Letra C

14. (IF-RS/IF-RS/2018) Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?

- A) dns 8.8.8.8
- B) echo 8.8.8.8 > /etc/hosts.conf



- C) echo nameserver 8.8.8.8 > /etc/resolv.conf
- D) nslookup 8.8.8.8
- E) nameserver 8.8.8.8

#### Comentários:

Lembrando...o DNS resolve nomes, então o arquivo de configuração é o "resolv.conf" e sabemos que os arquivos de configuração por padrão ficam em "/etc". Só tem uma opção...a letra C.

Mas, continuando...o comando echo "escreve", então o comando completo "escreveu" "nameserver 8.8.8.8" no arquivo "/etc/resolv.conf", ou seja, configurou o serviço de DNS.

Gabarito: Letra C

15. (COPEVE-UFAL/UFAL/2018) O assistente de tecnologia da informação precisa atualizar o sistema operacional GNU/Linux do servidor WEB de uma empresa. Qual comando, usando o modo root, deve ser utilizado?

- A) upgrade distro-all.
- B) upgrade apt-distro.
- C) apt-get dist-upgrade.
- D) apt-get purge "nome da distro".
- E) apt-cache showpkg "nome da distro".

#### Comentários:

Sabemos que o gerenciador de pacotes é o APT, já descartamos as duas primeiras. Mesmo sem lembrar da aula, dá para ver que pela lógica a letra C indica um upgrade da distribuição Linux.

Gabarito: Letra C

16. (COPEVE-UFAL/UFAL/2018) Dadas as afirmativas quanto aos conceitos de Kernel no sistema operacional GNU/Linux,

- I. Compilar o Kernel permite ao usuário remover drivers inúteis diminuindo o tempo de arranque do sistema operacional.
- II. Os módulos do Kernel do sistema estão armazenados no diretório /boot/.
- III. O comando uname -a exibe as informações do kernel do sistema.
- IV. Módulos são as partes do kernel que são carregadas somente quando são requisitadas por um aplicativo ou dispositivo.

Verifica-se que está(ão) correta(s)



- A) II, apenas.
- B) I e II, apenas.
- C) III e IV, apenas.
- D) I, III e IV, apenas.
- E) I, II, III e IV.

#### Comentários:

(I) Compilar o kernel permite eliminar algo inútil ou acrescentar algo útil (ex.: um driver de uma placa de rede que você possui e não tinha antes). (II) Não! Ficam em "/lib/modules/versão\_do\_kernel". (III) Exato! Conforme tela abaixo, com destaque na versão do kernel. (IV) Perfeito! Assim não carrega o que não precisa para a memória!

```
[root@localhost lib]# uname -a  
Linux localhost 4.12.0-rc6-g48ec1f0-dirty #21 Fri Aug 4 21:02:28 CEST 2017 i586  
GNU/Linux  
[root@localhost lib]#
```

Gabarito: Letra D

17. (COPEVE-UFAL/UFAL/2018) Analise a configuração de rede no GNU/Linux.

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
allow-hotplug eth0
```

```
iface eth0 inet static address 192.168.100.1 netmask 255.255.255.0 network 192.168.100.0  
broadcast 192.168.100.255
```

Dadas as afirmativas quanto à configuração de rede,

- I. O "iface eth0 inet static" informa ao sistema que a placa de rede terá o endereço IP estático.
- II. O "iface lo" informa ao sistema que a placa de rede receberá um endereço IP via servidor DHCP.
- III. Uma segunda placa de rede com fio instalada nesse computador irá receber a denominação eth1.
- IV. O "allow-hotplug" permite conectar dispositivos hotplug no computador.

Verifica-se que está(ão) correta(s)





- A) II, apenas.
- B) I e III, apenas.
- C) II e IV, apenas.
- D) I, III e IV, apenas.
- E) I, II, III e IV.

#### Comentários:

(I) O "iface eth0 inet static" informa ao sistema que a placa de rede terá o endereço IP estático, ou seja, não receberá um endereço através do protocolo DHCP. (II) O "iface lo" indica a interface de loopback (aquela para testes, geralmente o endereço IP é 127.0.0.1). (III) Uma segunda placa de rede com fio instalada nesse computador irá receber a denominação eth1, pois eth indica "ethernet" e já existe a eth0. (IV) O "allow-hotplug" permite conectar dispositivos hotplug no computador ("allow" = permite, então ficou tranquila essa).

Gabarito: Letra D

18. (Quadrix/CFBio/2018) Carregamento e inicialização do kernel e execução dos scripts de inicialização do sistema são algumas das etapas do processo de inicialização de um Linux típico.

#### Comentários:

Questão redonda, pois o núcleo do S.O. (Linux) deve ser carregado, depois os scripts de inicialização definidos pelo usuário (ou com a configuração padrão mesmo).

Gabarito: Certo

19. (Quadrix/CRM-PR/2018) Um computador pessoal com sistema operacional Linux, versão desktop, pode ter no máximo cinco usuários simultâneos.

#### Comentários:

Só teria o limite de 5 se fosse configurado previamente, mas por padrão não tem limite - até estourar a memória! 😊.

Gabarito: Errado

20. (FADESP/IF-PA/2018) Sobre os pacotes do Sistema Operacional (SO) Debian 9.5, é correto afirmar que

- A) o apt é um programa que pode substituir o dpkg para o gerenciamento de pacotes do SO.
- B) os pacotes devem ser distribuídos no formato .tar.gz.
- C) o dpkg-source compila e executa código fonte de um pacote.
- D) um pacote marcado como Essencial não pode ser desinstalado do Sistema Operacional.



E) cada pacote deve especificar as dependências sobre outros pacotes para seu funcionamento.

#### Comentários:

É comum que para uma aplicação funcione ela precise de alguns pacotes, porque alguns dependem de outros. Os gerenciadores de pacotes ajudam a verificar essas dependências.

Gabarito: Letra E

21. (CONSULPAM/Câmara de Juiz de Fora-MG/2018) Em relação ao Sistema Operacional Linux, marque o item INCORRETO:

A) Sua arquitetura é composta por um núcleo monolítico cujas funções são: gerenciar a memória, operar as entradas e saídas e o acesso aos arquivos.

B) Outra característica do Linux é com relação aos drivers de dispositivos e suporte a rede, os quais podem ser compactadas e utilizadas como se fossem módulos ou bibliotecas (LKM em inglês Loadable Kernel Modules), separados pela parte principal, cujo carregamento pode ser ativado após a execução do núcleo.

C) No quesito portabilidade, o Linux funciona com eficiência em plataformas como x64 da Intel (EM64T e AMD64) PowerPC, Alpha, SPARC, porém é de difícil instalação nos sistemas embarcados como PVR, celulares, Tv's e Handhelds.

D) A partir da década de 90, ao passo que a distribuição do Linux se popularizou, foi também limitada, pois se torna uma alternativa no uso de software livre, contra os sistemas operacionais da Apple (Mac OS) e Microsoft (Windows).

#### Comentários:

Em relação à portabilidade, o Linux se sai muito bem! Ele não foi pensado para ser um sistema portátil, mas acabou indo nessa direção, tendo um dos núcleos de sistemas operacionais mais portáteis, sendo executado em sistemas desde o iPaq (um computador portátil) até o IBM S/390 (um mainframe). Na atualidade o Linux hoje funciona em dezenas de plataformas, desde mainframes até um relógio de pulso, passando por diversas arquiteturas: x86 (Intel, AMD), x86-64 (Intel EM64T, AMD64), ARM, PowerPC, Alpha, SPARC, entre outras, além de sistemas embarcados (handhelds, consoles de videogames, PVR – Personal Video Recorder, telefones celulares, TVs etc.).

Gabarito: Letra C

22. (COPESE/UFT/2018) O sistema operacional Linux é reconhecido por permitir diversos níveis de personalização, inclusive de suportar o uso de vários ambientes gráficos. Assinale a alternativa que NÃO constituiu uma interface gráfica usada no Linux

A) Pantheon.

B) XFCM.

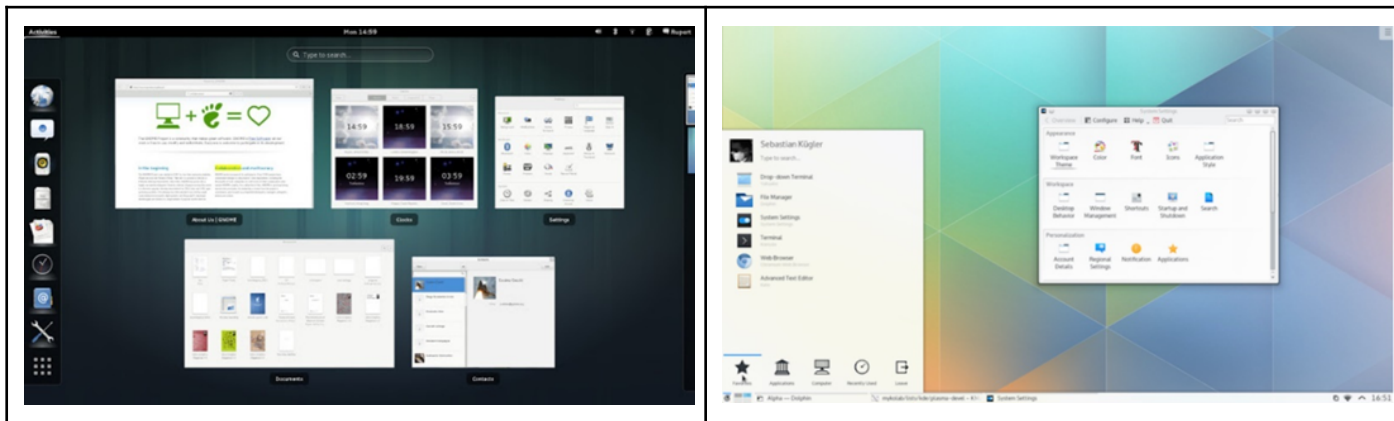
C) MATE Desktop.



D) Cinnamon Desktop.

### Comentários:

O servidor de janelas é um ambiente gráfico (que se parece bastante com o sistema operacional Windows), tornando possível que tarefas sejam executadas sem a digitação de comandos (mas o usuário pode abrir um shell dentro do ambiente gráfico, se quiser). Abaixo podemos ver dois exemplos, o Gnome (à esquerda) e o KDE (à direita).



Além desses dois, outros exemplos são Pantheon, MATE, Cinnamon, XFCE, Deepin, entre outros. Sacanagem foi colocar "XFCM" como alternativa, pois o correto seria "XFCE"!

Gabarito: Letra B

23. (CCV-UFC/UFC/2018) De acordo com a linha do arquivo '/ etc / passwd' exibida abaixo, qual das seguintes afirmações é verdadeira?

```
jferreira:x:502:1000:Joao Ferreira:/home/jferreira:/bin/bash
```

- A) O usuário Joao Ferreira possui a senha 'x'.
- B) Shadow passwords (senhas ocultas) são utilizados no sistema.
- C) O usuário Joao Ferreira pertence ao grupo com groupID 502.
- D) Membros do groupID 502 podem ler o diretório /home/jferreira.
- E) O nome de usuário (username) 'jferreira' pertence ao grupo 'jferreira'.

### Comentários:

O segundo campo seria a senha (antigamente até era), mas há um bom tempo o hash da senha é colocado no arquivo /etc/shadow e no lugar dela é colocado um "x". Ou seja, esse "x" indica que shadow passwords (senhas ocultas) são utilizados no sistema.

Gabarito: Letra B

24. (Quadrix/CRM-PR/2018) É possível instalar, em um mesmo computador desktop, os sistemas operacionais Windows 7 e Linux, de forma independente.



### Comentários:

É possível, sim! Cada um em uma partição e um gerenciador de partição é utilizado para criar um menu e mostrá-lo quando você liga a máquina. Um exemplo clássico é o GRUB.

Gabarito: Certo

25. (Quadrix/CRM-PR/2018) O usuário root do sistema Linux tem autonomia para acessar arquivos de outros usuários, com exceção dos usuários administradores.

### Comentários:

Root pode tudo! Até mesmo acessar os arquivos de outros usuários administradores!

Gabarito: Errado

26. (IDECAN/CRF-SP/2018) "O arquivo \_\_\_\_\_ só é legível pelo superusuário e serve para manter senhas criptografadas protegidas contra o acesso não autorizado. Ele também fornece informações sobre contas que não são disponíveis em \_\_\_\_\_." Assinale a alternativa que completa correta e sequencialmente a afirmativa sobre o Sistema Operacional Linux.

- A) /etc/passwd; /etc/group
- B) /etc/group; /etc/passwd
- C) /etc/passwd; etc/shadow
- D) /etc/shadow; /etc/passwd

### Comentários:

Lembre-se que no /etc/passwd é possível verificar alguns dados do usuário (menos a senha, tem um "x" em seu lugar). No /etc/shadow tem o hash da senha, além de outras informações (expiração da senha, entre outras).

Gabarito: Letra D

27. (UERR/IPERON/2018) Um administrador de uma rede baseada em Linux deseja guardar no servidor os scripts especiais para iniciar ou interromper módulos e programas diversos. Para isso, ele precisa guardar essas informações dentro do diretório:

- A) /srv
- B) /opl
- C) /etc
- D) /lib
- E) home



### Comentários:

A gente pensa que "etc" significa "etcetera", mas o diretório quer dizer "environment tables and controls", então os arquivos de configuração geralmente se encontram em /etc.

Gabarito: Letra C

28. (UERR/IPERON/2018) Em um computador com Linux, deseja-se instalar o sistema de arquivos ext3, mas com o nível de Journaling no qual se grava mudanças em arquivos de metadados, que força que a escrita do conteúdo dos arquivos seja feita após a marcação de seus metadados. Esse nível de Journaling é denominado:

- A) ordered
- B) cshell.
- C) iso9660
- D) journal.
- E) fsreiser.

### Comentários:

O Ext3 possui três modos de operação:

- *ordered (default)*: o journal é atualizado no final de cada operação. Isso faz com que exista uma pequena perda de desempenho, já que a cabeça de leitura do HD precisa realizar duas operações de gravação, uma no arquivo que foi alterado e outra no journal (que é um arquivo especialmente formatado);
- *writeback*: o journal armazena apenas informações referentes à estrutura do sistema de arquivos (metadados) e não em relação aos arquivos propriamente ditos, é gravado de forma mais ocasional, aproveitando os momentos de inatividade.
- *journal*: é o mais seguro, todavia mais lento. Nesse modo, o journal armazena não apenas informações sobre as alterações, mas também uma cópia de segurança de todos os arquivos modificados, que ainda não foram gravados no disco. Por ser o mais lento, é o modo menos usado.

Gabarito: Letra A

29. (CS-UFG/IF Goiano/2019) Os sistemas Linux buscam padronizar diretórios para a localização de arquivos. Comandos essenciais do sistema operacional, tais como cat, tar, su, rm e pwd, em geral, estão localizados nos seguintes diretórios:

- A) /lib e /var
- B) /usr/lib e /lib
- C) /bin e /usr/bin



D) /usr/bin e /var

#### Comentários:

Comandos são executáveis (binários), então, como sabemos que a estrutura de diretórios no Linux é organizada pelo tipo de informação, só nos resta a alternativa que possui "bin" duas vezes. Vejamos:

- /usr/bin: executáveis dos programas;
- /bin: comandos básicos, como cd, ls e cat.

Gabarito: Letra C

30. (IF-MS/IF-MS/2019) O sistema operacional Linux possui várias partições (áreas) em sua estrutura, cada uma com uma função definida. Assinale a alternativa que apresenta a partição que abriga a pasta raiz do sistema que contém arquivos essenciais ao seu pleno funcionamento:

A) /boot

B) /bin

C) /

D) /etc

E) /root

#### Comentários:

No Linux o diretório raiz é representado por "/", enquanto no Windows é representado por "\".

Gabarito: Letra C

31. (COSEAC/UFF/2019) No Sistema Operacional Linux o diretório local padrão do superusuário é:

A) /home.

B) /usr.

C) /root.

D) /lib.

E) /dev.

#### Comentários:

Os "usuários comuns" possuem como diretório local padrão "/home/nome\_usr" (ex.: /home/evandro), enquanto o root possui como diretório local padrão o "/root".

Gabarito: C



32. (INAZ do Pará/CORE-SP/2019) "Todos os arquivos e diretórios do sistema Linux instalado no computador partem de uma única origem: o diretório raiz. Mesmo que estejam armazenados em outros dispositivos físicos." - Disponível em: <https://canaltech.com.br/linux/entendendo-a-estrutura-de-diretorios-dolinux/>. Acesso em: 13.12.2018.

Qual pasta no sistema operacional LINUX tem a função de armazenar arquivos de dispositivos do sistema?

- A) /dev.
- B) /etc.
- C) /lib.
- D) /var.
- E) /proc.

Comentários:

Dispositivos em inglês são devices. Como Linux geralmente é intuitivo...a resposta é "/dev".

Gabarito: Letra A

33. (INAZ do Pará/CORE-SP/2019) "O agendamento de tarefas é um recurso muito interessante para a administração de sistemas operacionais. É possível programar a execução de scripts de manutenção do sistema, disparar envio de newsletters, gerar relatórios de análises de logs, entre outros" - Disponível em: [https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-\(cron-e-at\)](https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-(cron-e-at)). Acesso em: 13.12.2018.

Qual programa do sistema operacional LINUX executa tarefas de modo automático, em horários pré-determinados, e executa tarefas que não foram executadas, enquanto o sistema esteve desligado?

- A) Fcron.
- B) Cron.
- C) Anacron.
- D) Schedule.
- E) Crontab.

Comentários:

Para o agendamento de tarefas a aplicação mais conhecida é o Cron, que já existe há muito tempo e passou por muitos estágios de evolução. Porém, a maioria das suas implementações atuais (Vixie Cron, ISC Cron, BCron etc.) ainda se baseiam no pressuposto de que o sistema está em funcionamento de forma ininterrupta. Isso é um problema quando se utiliza a virtualização,





quando os sistemas que operam sob demanda, sendo ligados e desligados conforme sua necessidade.

Para contornar essa situação, algumas distribuições contam com o Anacron como alternativa ao Cron. O Anacron permite a criação de listas de tarefas que serão realizadas em intervalos pré-definidos e, quando o Anacron é inicializado, ele verifica essas listas e executa as tarefas ainda não realizadas. No entanto, o Anacron possui algumas limitações. A primeira é que ele não é um daemon, então precisa ser executado sempre que for necessário.

Outra questão é que o Anacron não está preparado para lidar com períodos de tempo menores do que dias. A combinação desses problemas pode levar a situações nas quais o Cron e o Anacron funcionem ao mesmo tempo, e algumas tarefas podem ser executadas duas vezes ou nenhuma!

Uma solução é o Fcron, que faz o que o Vixie Cron e o Anacron fazem e ainda mais. É possível usar o Fcron para agendar cronjobs com data e hora fixas, com intervalos de tempo ou até mesmo de acordo com a disponibilidade do sistema.

Gabarito: Letra A

34. (IDECAN/IF-PB/2019) O Debian, uma das distribuições do sistema operacional Linux, possui um conjunto de pastas com nome pré-estabelecidos. Essas pastas possuem significado específico para o sistema operacional, visando atender necessidades do mesmo. Assinale a alternativa que indica corretamente o nome da pasta que armazena arquivos referentes às instalações de programas não oficiais da distribuição do sistema operacional.

A) /bin

B) /lib

C) /opt

D) /var

E) /proc

Comentários:

Programas não oficiais poderiam ser chamados de "opcionais" (optionals), então vamos usar a lógica de nomes do Linux... "/opt".

Gabarito: Letra C

35. (UFRR/UFRR/2019) A respeito de diretórios no sistema operacional Linux, a estrutura é conhecida como árvore invertida, isto é, a raiz da árvore de diretórios é o topo. Qual é a representação do diretório raiz, e nome do usuário que possui privilégio para escrever neste diretório.

A) (raiz) e usuário root

B) \ (raiz) e usuário administrador



- C) / (raiz) e usuário root
- D) [ (raiz) e usuário administrador
- E) ]/ (raiz) e usuário root

#### Comentários:

Quem está acostumado com Windows sabe que o diretório raiz é "\", mas no Linux é invertida: "/". No Windows o usuário que "pode tudo" é o "Administrador", enquanto no Linux é o "root".

Gabarito: Letra C

36. (IF-ES/IF-ES/2019) O Network File System (NFS) – Sistemas de Arquivos em Rede – tem como um dos principais propósitos dar suporte a um sistema heterogêneo, no qual clientes e servidores estejam possivelmente executando sistemas operacionais e hardwares diferentes. Sobre o NFS, é CORRETO afirmar:

- A) O NFS utiliza dois protocolos cliente-servidor, em que o primeiro é responsável pela montagem e o segundo é para acesso de diretório e arquivos.
- B) O servidor tem total gerência sobre o ponto de montagem nos clientes.
- C) Os serviços NFS são implementados apenas nos servidores Linux.
- D) Como critério de segurança, os clientes não podem ter acesso aos atributos dos arquivos.
- E) O NFS faz uso de máquinas distintas para servidores e clientes, impossibilitando que a mesma máquina seja tanto cliente quanto servidor.

#### Comentários:

Para que os clientes possam acessar o servidor NFS é necessário que os seguintes daemons estejam em execução:

- nfsd: daemon NFS, atende requisições dos clientes NFS;
- mountd: daemon de montagem NFS, executa as solicitações passadas pelo nfsd;
- portmap: daemon portmapper, permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

Para essa questão, esqueça o portmap, pois só serve para descobrir a porta do servidor. Os outros dois daemons fazem o papel dos protocolos citados na alternativa A: um atende requisições dos clientes (acessos a diretórios e arquivos) e o outro faz a montagem.

Gabarito: Letra A

37. (FUNDATEC/Prefeitura de Maçambará-RS/2023) No Linux, a camada conhecida por ser o núcleo do sistema operacional, administrando o hardware e atuando, principalmente, como uma interface entre esse e qualquer programa em execução, é o:

- A) kernel



- B) system-manager
- C) shell
- D) bus
- E) socket-level

#### Comentários:

Dois conceitos bastante cobrados são:

- Kernel = núcleo do sistema operacional;
- Shell = interpretador de comandos.

O resto não tem muito a ver com a questão. "System-manager" seria gerenciador de sistema, algo genérico demais. "Bus" é barramento. "Socket-level"? Nível de socket? Nada a ver com nada.

Gabarito: Letra A

38. (CS-UFG/UFG/2023) O Sistema Operacional Linux possui diversos arquivos relacionados à configuração de redes de computadores, dos quais, podemos citar:

- A) /etc/network/dns.conf e /etc/hosts/network.
- B) /etc/hosts e /etc/interface/host.
- C) /etc/network/interfaces e /etc/resolv.conf.
- D) /etc/sysconfig/interfaces e /etc/interfaces/network.

#### Comentários:

Vamos aos dois arquivos:

- /etc/network/interfaces: arquivo usado pelos programas ifup e ifdown, respectivamente para ativar e desativar as interfaces de rede;
- /etc/resolv.conf: usado para configurar a resolução do host via DNS. Algumas distribuições têm scripts de inicialização, daemons e outras ferramentas que gravam neste arquivo.

Gabarito: Letra C

39. (FUNCERN/Câmara de Natal-RN/2023) Levando-se em consideração distribuições de sistemas operacionais Linux, o conceito FHS (File Hierarchy Standard) é um padrão para que softwares e usuários possam prever a localização de arquivos e diretórios instalados. Sendo assim, o diretório sugerido para conter arquivos de configuração do sistema, os quais são usados para controlar a operação de um programa, é o

- A) /etc
- B) /boot
- C) /tmp
- D) /root



### Comentários:

/etc: dados de configuração (scripts de inicialização, tabela de sistemas de arquivo, configurações de login, configuração da fila de impressão, entre outros);

/boot: imagem do Kernel e o initrd (initial ram disk), carregados no início do boot;

/tmp: arquivos temporários;

/root: diretório home do root. Obs.: Esse diretório não é a mesma coisa que a raiz do sistema /.

Gabarito: Letra A

40. (Quadrix/CRM-MG/2023) Assinale a alternativa que apresenta o sistema de inicialização (init system), do sistema operacional Linux, composto de um conjunto de programas executado em segundo plano.

A) Init

B) Bash

C) Cron

D) Systemd

E) Syslogd

### Comentários:

Init (Systemd): O systemd é um sistema de inicialização (init system) composto por um conjunto de programas que é executado em background (daemon). Na atualidade a maioria das distribuições Linux utiliza o systemd para a execução do boot.

O systemd assume o controle assim que o kernel é ativado pelo gerenciador de bootloader (ex.: GRUB). A partir desse ponto são carregados todos os dispositivos e processos que são inicializados com o sistema. Uma grande vantagem do systemd é a sua arquitetura e o modo de funcionamento. Nele são usadas unidades de socket (arquivos de configuração que codificam informações relacionadas à comunicação entre processos). Com isso, permite que todos os daemons requisitados no boot sejam carregados simultaneamente, bem como possibilita a transmissão coordenada entre dois sockets, o que resulta em uma rápida inicialização do sistema operacional.

Gabarito: Letra D



## QUESTÕES COMENTADAS – CEBRASPE

1. (CEBRASPE/TJ-SE/2014) No que se refere à lógica e à programação, julgue os itens que se seguem.

Em um comando Shell Script do Linux, é possível combinar diversos comandos em sequência utilizando-se apenas o comando +.

Comentários:

Na questão não fica claro se os comandos em sequência devem esperar a execução do anterior para começar a execução do seguinte, ou se são executados em paralelo. No primeiro caso o caractere usado é o ";" e no segundo caso é o "&".

Gabarito: Errado

2. (CEBRASPE/ANATEL/2014) Julgue os próximos itens, acerca do ambiente Linux.

Em um script shell, se o retorno do último comando obtido pela variável interrogação (?) for igual a 1, então o comando teve sucesso na sua execução.

Comentários:

O retorno 0 indica sucesso! Podemos rever exemplo a seguir:

```
#!/bin/bash
```

```
ls /home/evandro/gabarito
```

```
if [ $? -eq 0 ]; then
```

```
    echo "O comando ls foi executado com sucesso!"
```

```
else
```

```
    echo "O comando ls falhou!"
```

```
fi
```

Gabarito: Errado

3. (CEBRASPE/TCE-PA/2016) Acerca dos sistemas operacionais Linux e Windows Server, julgue o item que se segue.

No Linux, o tipo de permissão r possibilita ao usuário executar o comando ls e listar o conteúdo de um diretório, ao passo que o tipo w admite a gravação e exclusão de um diretório, cabendo ao tipo x facultar ao usuário a execução de um arquivo script.

Comentários:



Um script nada mais é do que um arquivo texto com comandos e, para executar esse arquivo, deve haver a permissão de execução (x) para as pessoas autorizadas (U = usuário dono, G = grupo do dono, O = outros).

Gabarito: Certo

4. (CEBRASPE/PGE-RJ/2022) Acerca da administração do sistema operacional Linux e da linguagem de script Shell, julgue o item subsequente.

Em ambiente Linux, ao se executar um arquivo que contenha a instrução a seguir.

```
echo PGE-RJ celebra o fim das obras de restauro \ do Convento do Carmo
```

Será obtido o resultado seguinte.

```
PGE-RJ celebra o fim das obras de restauro do Convento do Carmo
```

Comentários:

O comando echo serve para imprimir. Como vimos na parte teórica, o caractere \ não é mostrado. Se quiser imprimir \, deve-se utilizar duas vezes (\\). Na questão, temos:

```
echo PGE-RJ celebra o fim das obras de restauro \ do Convento do Carmo
```

Então deve ser impresso:

```
PGE-RJ celebra o fim das obras de restauro do Convento do Carmo
```

Obs.: há um espaço duplo entre "restauro" e "do", pois após a barra invertida há um espaço.

Gabarito: Certo



## QUESTÕES COMENTADAS – FGV

1. (FGV/TRT-MA/2022) Analise o script abaixo.

Assinale a opção que indica quantas linhas serão impressas na tela de um terminal Linux e quais os valores de  $\{NR\}$  e de count ao término da execução do script.

- a) 7, 273 e 6.
- b) 14, 312 e 7.
- c) 14, 263 e 7.
- d) 21, 361 e 7.
- e) 21, 312 e 6.

### Comentários:

Vamos por partes:

`NR=-31`; → NR começa com o valor 31.

`until [  $\{NR\}$  -gt 273 ]`; → Enquanto NR NÃO for maior que 273 fica no laço.

`do echo "NR= $\{NR\}$ "; NR=$((  $\{NR\}$  +49 ))`; → Imprime "NR={valor\_NR}", depois adiciona 49 a NR.

`echo " FIM= $\{NR\}$ ";` → Imprime "FIM={valor\_NR}. Note que agora imprime depois de ter adicionado 49.

`echo " Conta=$((count++))"`; → Imprime "Conta={valor\_count}" e depois adiciona 1. Note que count não foi declarado, então começa imprimindo o valor 0, depois adiciona 1. Se fosse ++count, então adicionaria antes de imprimir. Cuidado com isso!!!

`done`

Tente fazer em um papel o passo a passo, como se fosse na prova. O resultado deve ser esse (coloquei em sete colunas, mas a impressão seria uma linha abaixo da outra, totalizando 21):

NR=-31 FIM=18 Conta=0	NR=18 FIM=67 Conta=1	NR=67 FIM=116 Conta=2	NR=116 FIM=165 Conta=3	NR=165 FIM=214 Conta=4	NR=214 FIM=263 Conta=5	NR=263 FIM=312 Conta=6
-----------------------------	----------------------------	-----------------------------	------------------------------	------------------------------	------------------------------	------------------------------

Gabarito: Letra E





2. (FGV/Senado Federal/2022) Assinale a opção que indica o resultado da execução do script Shell a seguir em um prompt de comando do Linux.

- a) AREPO TENET
- b) OPERA SATOR ARE
- c) ENET OP AREP
- d) SATOR OPERA
- e) R AREP ROTAS

#### Comentários:

Vamos comentar cada linha:

palindrome="SATOR AREPO TENET OPERA ROTAS" → a variável "palindrome" recebeu uma string com cinco palavras separadas por espaço.

subStr1=\${palindrome:6:4} → a variável subStr1 recebeu, a partir da posição 6 (lembrando que começa em 0), quatro caracteres, ou seja, "AREP".

subStr2=\${palindrome:13:7} → a variável subStr2 recebeu, a partir da posição 13 (lembrando que começa em 0), sete caracteres, ou seja, "ENET OP".

Str="\${subStr2} {subStr1}" → Str recebeu a concatenação de subStr2, mais um espaço, mais subStr1, ou seja, "ENET OP AREP". Obs.: Perceba que foi concatenado subStr2 na frente de subStr1, apenas para confundir, já que estamos acostumados em ver o 1 antes do 2!!! Cuidado!!!

echo \$Str → Imprimiu "ENET OP AREP".

Gabarito: Letra C

3. (FGV/Prefeitura de Manaus-AM/2022) Dado o programa bash abaixo,

```
#!/bin/bash
```

```
a=12
```

```
b=a+15
```

```
echo ${b+20}
```

assinale a opção que mostra a saída correta do programa após sua execução.

- A) 0.
- B) 15.



- C) 20.
- D) 32.
- E) 47.

#### Comentários:

Pegadinha interessante! Veja:

A variável a recebeu o valor 12.

A variável b recebeu a string "a+15"!!! Cuidado!!!

O comando echo imprimiu na tela a soma de "a+15" com o valor 20. Como "a+15" é uma string, assumiu como valor 0, então  $0+20 = 20$ . Muito cuidado com essas "jogadas" das bancas!!!

Gabarito: Letra C

4. (FGV/TRT13/2022) Analise o código a seguir escrito em shell script:

Ao executar esse programa em um terminal será escrito na saída padrão:

- A) www.website\_trt.com.br
- B) www.website\_trt.cxm.br
- C) Website
- D) http://www.website\_trt
- E) http://www.website\_trt.com.br

#### Comentários:

Para realizar substituições de strings, uma maneira é:

```
string=${string//antigo/novo}
```

A variável var recebeu "http://www.website\_trt.cxm.br", mas a impressão (echo) foi realizada a partir da posição 7 (oitava posição, já que começa em 0), então imprimiu a partir de "www...".

Gabarito: Letra B



## QUESTÕES COMENTADAS – FCC

1. (FCC/TRE-SP/2012) No sistema Linux, para se executar um arquivo texto contendo comandos de interpretador como um script é necessário que o arquivo

- A) seja compilado.
- B) possua permissão de execução.
- C) esteja no diretório /usr/bin
- D) tenha a extensão .exe
- E) tenha a extensão .bat

### Comentários:

Um script é um arquivo texto interpretado, ou seja, não é compilado! Porém, um arquivo texto só pode ser executado se tiver permissão de execução (x). Não precisa ter extensão nenhuma, embora muitas vezes se utilize a extensão .sh. E pode estar em qualquer diretório.

Gabarito: Letra B

2. (FCC/Prefeitura de Teresina-PI/2016) O Administrador de um servidor com sistema operacional Red Hat Linux deseja escrever um script para automatizar o processo de backup do servidor. A primeira linha do arquivo de script deve ser:

- A) #/sbin/bash
- B) %#/sbin/bash
- C) #!/bin/bash
- D) \*#/bin/bash
- E) \$/sbin/bash

### Comentários:

A primeira linha deve conter #! seguido do caminho do shell a ser utilizado, que fica em "/bin". No caso da questão foi escolhido o "bash".

Gabarito: Letra C

3. (FCC/TRT23/2016) Bento, administrador de um servidor com sistema operacional Linux, escreveu um shell script para automatizar o processo de backup do sistema. Para que apenas Bento possa executar o shell script criado, as permissões devem ser alteradas utilizando o comando chmod com o parâmetro

- A) 707.
- B) 660.
- C) 700.



D) 006.

E) 077.

#### Comentários:

Para executar um shell script, o arquivo deve ter permissão de execução (x). No caso da questão, fica claro que apenas Bento (o administrador) deve executar. Vamos ver em binário como ficaria (quer dizer qualquer coisa, pois o foco é colocar 1 apenas em x, ou seja, na terceira posição):

U = \_\_ 1 (quer dizer qualquer valor).

Se for \_\_ 0, já eliminamos a alternativa. Por exemplo, 110 (6) - alternativa B, 000 (0) - alternativas D e E.

Sobraram apenas A e C. Na alternativa A para os outros (O), o valor está em 7 (111) e o enunciado diz que apenas Bento pode executar, os outros não! Portanto, só sobrou a letra C, que tem permissão 111 (rwx) para o U (usuário dono) e nada para o resto.

Gabarito: Letra C

4. (FCC/ELETRONBRAS-ELETROSUL/2016) Um profissional de TI está usando um computador com sistema operacional Linux que utiliza no shell o interpretador de comandos bash. Ele está logado como usuário teste e criou o seguinte arquivo shell script:

```
1 #!/bin/bash
```

```
2 echo 'Eletrosul- Centrais Elétricas S.A.'
```

```
3 $ variavel= 'Eu estou logado como usuário $user'
```

```
4 $ echo $variavel
```

Considerando que 1, 2, 3 e 4 indicam as linhas do arquivo e que este tenha sido salvo com o nome exemplo, é correto afirmar:

A) Para o arquivo ser executável, é necessário acionar o comando `$ chmod +x exemplo`. Depois disto o arquivo poderá ser executado com `./exemplo`.

B) A linha 1 indica que todas as outras linhas abaixo deverão ser executadas pelo compilador sh, que se localiza em `/bin/bash`.

C) Após ser executado, o arquivo imprimirá na tela apenas frase "Eletrosul – Centrais Elétricas S. A. " utilizando o comando echo.

D) Ao acionar o comando file arquivo é possível ver que a definição dele é Bourne-Again Shell Script, que se refere ao bash script.

E) As linhas 3 e 4 farão com que seja impresso na tela Eu estou logado como usuário \$teste.

#### Comentários:

Vamos analisar o script:



`#!/bin/bash`: especifica o shell bash

`echo 'Eletrosul- Centrais Elétricas S.A.'`: imprime na tela 'Eletrosul- Centrais Elétricas S.A.'

`$ variavel= 'Eu estou logado como usuário $user'`: A variável recebe a string 'Eu estou logado com o usuário teste'

`$ echo $variavel`: desconsidere \$ inicial, então imprime 'Eu estou logado com o usuário teste'

Analisando as alternativas, a única correta é a que menciona que para o arquivo (script) ser executável, é necessário acionar o comando "chmod +x exemplo" (dá a permissão de execução a todos, pois não especificou a quem seria). Depois disto o arquivo poderá ser executado com "./exemplo".

Gabarito: Letra A

5. (FCC/SANASA/2019) Considerando o uso do Linux Ubuntu, é possível criar um shell script com objetivo de encadear comandos para realizar tarefas. Considere o trecho de shell script abaixo, criado em um diretório que contém arquivos.

Para mostrar o número de arquivos encontrados após a compilação e execução a lacuna I deve ser preenchida por

A) `c=c+1`

B) `c=$((c+1))`

C) `c=#[c+1]`

D) `c=Math(soma(c+1))`

E) `c++`

Comentários:

Podemos ver que `c` é um contador, que começa com o valor 0. O laço `for i in 'ls'` vai contar quantos arquivos o 'ls' vai listar. A questão quer saber como faz para incrementar `c`. As alternativas A e E poderiam causar confusão, sendo uma pegadinha! Vejamos:

`c=c+1` : em shell script isso significa que `c` recebe a string "c+1".

`c++` : sintaxe utilizada na linguagem C, entre outras, mas não em shell script!

A sintaxe correta para o contador é: `c=$((c+1))`.

Gabarito: Letra B

6. (FCC/SANASA/2019) Em linha de comando, no Linux, para verificar os problemas e possíveis causas de erros em um script chamado `sanasa.sh`, que já possui permissão de execução, utiliza-se o comando

A) `sh -x sanasa.sh`

B) `kdsh -l sanasa.sh`



- C) `bashrun -y sanasa.sh`
- D) `chmod -x sanasa.sh`
- E) `ssh +x sanasa.sh`

#### Comentários:

Comando `sh`: usado para executar shell scripts. A sintaxe básica é:

```
sh script.sh
```

Se for usado sem especificar um script, um shell interativo é iniciado:

```
$ sh
```

A partir do shell interativo pode-se digitar comandos diretamente. Para sair do shell interativo, pode-se usar o comando `exit`.

Quando é usada a opção `-x`, é ativado o modo de depuração, ou seja, o script será executado com saídas verbosas, mostrando cada comando e sua expansão antes de serem executados. Ex.:

```
$ sh -x script.sh
```

Gabarito: Letra A

7. (FCC/TJ-SC/2021) Uma Analista de Sistemas criou o script `bash`, para ser executado em ambiente Linux em condições ideais, cujo código é apresentado abaixo.

```
#!/bin/bash
```

```
DATA=`date +%y%m%d-%H%M`
```

```
cp $1 $DATA-$2
```

Considerando que o código não apresenta erros,

- A) a Analista deve atribuir permissão de execução ao script criado com o comando `chmod a+x copiar.ba`; considerando que o nome do arquivo de script seja `copiar.ba`.
- B) o script copia um arquivo e, durante a operação, acrescenta a data e a hora local ao nome do novo arquivo.
- C) o script copia um arquivo e, durante a operação, acrescenta a data e a hora local para o novo arquivo, após seu último registro.
- D) o script usa os objetos `$1` e `-$2`. `$1` representa o primeiro parâmetro passado na linha de comando usada para executar o script e o `-$2` representa uma cópia do primeiro parâmetro com um `-` no começo do nome.
- E) o `bash` não faz diferença entre aspas simples, aspas duplas e crases, por isso o comando da 2ª linha do código poderia ser escrito como `DATA="date +%y%m%d-%H%M"` sem alterar a lógica do script.

#### Comentários:

Vamos por partes. Primeiro vamos lembrar o que significa `$0`, `$1` etc.:



- \$0 - nome do script executado;
- \$1 - o primeiro argumento enviado ao script;
- \$2 - o segundo argumento enviado ao script;
- \$3 - o segundo argumento, e assim por diante...

Depois é importante saber a sintaxe do comando cp (serve para copiar arquivos). Basicamente é:

```
cp arq_origem arq_destino
```

Em relação à variável DATA, podemos ver que ela recebe o retorno do comando date com um determinado formato (com as informações de ano, mês, dia, hora, minuto, nesta ordem).

Analisando o comando cp:

```
cp $1 $DATA-$2
```

Imagine que o script tenha recebido os argumentos teste.txt e opa.txt, então o comando cp ficaria assim:

```
cp teste.txt data_do_sistema_no_formato_especificado-opa.txt
```

Onde data\_do\_sistema\_no\_formato\_especificado é o valor retornado por date +%y%m%d-%H%M.

Com tudo isso, podemos verificar que a resposta é "o script copia um arquivo e, durante a operação, acrescenta a data e a hora local ao nome do novo arquivo".

Gabarito: Letra B





## QUESTÕES COMENTADAS – VUNESP

1. (VUNESP/CRO-SP/2015) No sistema operacional UNIX, uma aplicação pode ser desenvolvida por meio de um script que é um conjunto de comandos que podem ser executados pelo interpretador de comandos (shell). Considerando a existência do interpretador /bin/sh, a primeira linha de um arquivo de script deve conter:

- A) \*/bin/sh
- B) \$/bin/sh
- C) &/bin/sh
- D) #!/bin/sh
- E) #\*/bin/sh

### Comentários:

Para definir qual Shell será utilizado, deve-se especificar na primeira linha do arquivo (vamos escolher o sh, conforme a questão, mas poderia ser outro interpretador de comandos, como bash, ksh ou csh):

```
#!/bin/sh
```

Gabarito: Letra D

2. (VUNESP/Câmara de Sertãozinho-SP/2019) Considere que o usuário de um computador com sistema operacional Linux deseja executar um comando em segundo plano (background) em um terminal Shell. Para efetivar essa ação, o usuário deve executar o comando

- A) e teclar a combinação Ctrl+c
- B) e teclar a combinação Ctrl+x
- C) seguido de bg
- D) seguido de &
- E) seguido de !

### Comentários:

Quando deseja-se executar um comando que não tenha interação, como por exemplo, um servidor que fica atendendo requisições pela rede, executa-se em background. A maneira mais comum de fazer isso é executar o comando seguido de &. Ex.:

```
$ ./servidorX &
```

Gabarito: Letra D



## QUESTÕES COMENTADAS – MULTIBANCAS

1. (CESGRANRIO/Petrobras/2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado.

Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.
- E) na linha de comando, após o nome do arquivo que contém o script.

Comentários:

É necessário definir qual Shell será utilizado na primeira linha do arquivo (vamos escolher o bash, mas poderia ser outro interpretador de comandos, como sh, ksh ou csh):

```
#!/bin/bash
```

Gabarito: Letra A

2. (CEPERJ/Rio previdência/2014) Shell script é uma linguagem de script para Linux, nada mais do que comandos do próprio Linux que são executados em uma determinada sequência para uma determinada finalidade. Nesse contexto, duas situações são listadas a seguir.

I- No terminal ou modo gráfico, deseja-se criar um arquivo que possa ser editado para que se torne o primeiro shell script a ser criado, sendo necessário utilizar um comando CMD1.

II- Para que seja possível executar o shell script criado, é preciso atribuir a este o direito de execução; para isso é necessário usar um comando CMD2.

Exemplos de CMD1 e de CMD2 são, respectivamente:

- A) touch shell1.sh e exec +x shell1.sh
- B) touch shell1.sh e chmod +x shell1.sh
- C) create shell1.sh e chmod +x shell1.sh
- D) new shell1.sh e chmod +x shell1.sh
- E) new shell1.sh e exec +x shell1.sh

Comentários:



Criando um arquivo:

Há dois modos de realizar essa ação: via modo gráfico ou via terminal. No segundo caso, pode ser utilizado o comando `vi`, conforme mostrado a seguir (note a "extensão" `.sh`).

`vi script.sh`: Será criado e aberto um arquivo de leitura e escrita.

Outra opção é digitar o comando `touch`:

`touch script.sh`: cria um arquivo sem abri-lo.

Permissão ao arquivo:

Para começar a editar o arquivo, é necessário conceder a permissão de escrita a ele. Uma opção é utilizar o comando:

`chmod 777 script.sh`: o comando `chmod` é utilizado para conceder permissões em diretórios e arquivos, enquanto o valor `777` permite que o usuário tenha total liberdade para editar o arquivo (na verdade dá liberdade total a qualquer usuário do sistema! Mas nosso foco agora não é estudar o `chmod`).

Pensando mais em segurança, pode-se, por exemplo, liberar apenas a execução (`x`):

`chmod +x script.sh`: como não foi especificado para quem foi liberada a execução (usuário, grupo ou todos), todos devem ter a permissão de execução.

Gabarito: Letra B

3. (FUMARC/AL-MG/2014) No Linux, considere a existência de um arquivo "arquivo.txt" no diretório corrente, com o seguinte conteúdo:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Deseja-se produzir, a partir do arquivo, a seguinte saída:

1 - 10 2 - 20 3 - 30

O script shell usado com comandos AWK que produz a saída desejada é:

- A) `awk arquivo.txt '{ print $2" - "$4 }'`
- B) `cat arquivo.txt | awk '{ print $2" - "$4 }'`
- C) `cat arquivo.txt | awk '{ print $1" - "$3"\n" }'`
- D) `cat arquivo.txt | awk '{ printf($2" - "$4) }'`

Comentários:

O comando `cat` é usado para unir, criar e exibir arquivos. No caso dessa questão, serve para exibir.



O awk é um utilitário especializado em manipulação de texto.

O pipe ("|") serve para uma comunicação entre os processos, ou seja, a saída do que está na esquerda dele é passada para o que está à direita.

Então:

cat arquivo.txt: exibe o conteúdo do arquivo e esse conteúdo é passado para:

awk '{ print \$2" - "\$4 }': imprime o 2º argumento, um traço ("-"), o 4º argumento e continua processando o texto até o fim da linha. Logo, imprime:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Lembrando os traços entre o 2º e o 4º argumentos, fica assim:

1 - 10 2 - 20 3 - 30

Gabarito: Letra B

4. (INSTITUTO AOCP/EBSERH/2016) Em sistemas operacionais Linux é possível criar scripts para automatizar tarefas rotineiras. A extensão de arquivo utilizada para Shell Script é

- A) .bin
- B) .sl
- C) .bash
- D) .sh
- E) .gz

Comentários:

Vimos em aula não é obrigatório utilizar extensões! Mas, na prática, geralmente se utiliza .sh para ficar mais claro do que se trata.

Gabarito: Letra D

5. (FUNECE/UECE/2017) Considere o seguinte script Bash prog.sh em um sistema operacional Linux. A numeração das linhas à esquerda é meramente ilustrativa.

```
1.  #!/bin/bash
2.  x=-9
3.  echo -n "( "
4.  exit x
5.  echo -n x
6.  echo -n " )"
```



Assinale a opção que indica corretamente o que acontece após a execução do comando abaixo.  
`./prog.sh && echo " fim"`

- A) Será exibido "( -9 ) fim".
- B) Serão mostrados "( " e uma mensagem de erro na linha 4, afirmando que x não é um argumento numérico do comando exit.
- C) Será exibido "( " seguido do final com sucesso da execução do script.
- D) Serão mostrados "( " e uma mensagem de erro na linha 4, afirmando que a variável x não pode ser negativa.

#### Comentários:

Em uma primeira análise o script prog.sh seria executado e depois seria impresso fim. Isso ocorreria se não houvesse nenhum erro na execução do script.

Vamos analisar a execução de linha a linha do script:

`x=-9` : a variável x recebe -9;

`echo -n "( "` : imprime na tela ( com um espaço depois;

`exit x` : sairia do script com um código de erro. Se fosse `exit $x` o script interpretaria como `exit -9`, mas do jeito que está ele interpreta `exit x`. Isso provoca um erro, pois x não é um número! O erro que aparece é: `./prog.sh: line 4: exit: x: numeric argument required`, ou seja, era esperado um argumento numérico!

Gabarito: Letra B

6. (AOCP/UFOB/2018) Shellsript é uma linguagem de script utilizada em diversos sistemas operacionais. Sobre Shellsript, incluindo Script Bash, Groovy e Powershell, julgue o item a seguir.

Para que um arquivo .sh seja executado em um sistema Linux, este deve ter permissão de escrita para o usuário atual.

#### Comentários:

Para ser executado ele deve ter permissão de execução (x). Para ser editado (alterado), aí sim deve ter permissão de escrita (w).

Gabarito: Errado

7. (SUGEP-UFRPE/UFRPE/2018) Um script executável no interpretador de comandos Bash do GNU/Linux deve começar com os seguintes caracteres na primeira linha:

- A) { /bin/bash }
- B) #/bin/bash
- C) #!bin/bash
- D) <script language="bash">



E) #!/bin/bash

Comentários:

A primeira linha deve conter #! seguido do caminho do shell a ser utilizado, que fica em "/bin". No caso da questão foi escolhido o "bash".

Gabarito: Letra E

8. (CONSESP/Câm. de Castelo-ES/2018) Sob a utilização de uma distribuição Linux, leia as afirmações que se seguem.

I. Um script de shell é um grupo de comandos, funções, variáveis, ou qualquer outra coisa que pode ser usado em um shell.

II. Estes itens (grupo de comandos, funções, variáveis, ou qualquer outra coisa que pode ser usado em um shell) são digitados em um arquivo de texto simples que pode ser executado como um comando.

III. A maioria dos sistemas Linux utiliza scripts de shell de inicialização do sistema para executar comandos necessários para fazer os serviços funcionarem, durante a inicialização do sistema.

IV. Scripts de shell estão presentes na criação do kernel e não podem ser criados pelos usuários.

Assinale a alternativa correspondente.

A) Todas as afirmações estão corretas.

B) Somente três das afirmações estão corretas.

C) Somente duas das afirmações estão corretas.

D) Nenhuma das afirmações está correta.

Comentários:

I. CORRETA - Um shell script é uma sequência de comandos, além de permitir a criação de funções, variáveis etc. É considerada uma linguagem de programação.

II. CORRETA - Um shell script é um arquivo de texto e, para ser executado, deve ter a permissão de execução (x), que pode ser dada através do comando chmod.

III. CORRETA - Shell scripts são muito utilizados para a configuração inicial do sistema.

IV. ERRADA - Shell scripts podem ser criados pelos usuários, sim! Qualquer um pode criar!

Gabarito: Letra B

9. (FURB/Pref. de Massaranduba-SC/2020) Em ambiente Linux, observa-se:

1. #!/bin/bash

2. i=3

3. k=2



4. let j=k\*\*i
5. echo \$j
6. exit 0

Isso posto, ao executar o script bash acima, obter-se-á a seguinte saída:

- A) 6.
- B) NaN.
- C) 8.
- D) 0.
- E) 2.

#### Comentários:

Para começar, NaN significa "Not a Number". Vamos analisar linha a linha:

i=3 : i recebe o valor 3;

k=2 : k recebe o valor 2;

let j=k\*\*i : j recebe o resultado de k elevado a i, que é  $2^3 = 8$ ;

echo \$j : imprime o valor de j, ou seja, 8;

exit 0 : sai do script com o código 0 (sucesso).

Gabarito: Letra C

10. (FUNDATEC/PGE-RS/2021) Um usuário criou um arquivo chamado "script.sh" no seu "home directory" de uma instalação padrão do Linux Ubuntu 18 Desktop. Esse arquivo contém alguns comandos que ele costuma executar com frequência, mas, sempre que tenta executar o script, ele obtém uma mensagem de erro. Ele pediu para alguém ajudá-lo e enviou uma cópia dos comandos que executou e seus retornos, como segue abaixo:

```
$ cd
```

```
$ ls -l script.sh
```

```
-rwxr-xr-x 1 user user 11 Oct 03 18:31 script.sh
```

```
$ script.sh
```

```
script.sh: command not found
```

Com base nos comandos apresentados pelo usuário, qual seria uma forma de executar o script.sh sem erros?

- A) chmod a+x script.sh; script.sh
- B) ./script.sh





- C) `.script.sh`
- D) `./script.sh`
- E) `../script.sh`

#### Comentários:

Podemos ver que o arquivo `script.sh` está no diretório atual, mas ao tentar executar dá uma mensagem de comando não encontrado. É muito comum utilizarmos `./` antes do que queremos executar e essa seria a resposta! Mas por que geralmente é necessário utilizar `./` antes? Vamos ver o motivo:

Ao digitar `./` antes do comando, você indica ao sistema operacional que o comando deve ser executado a partir do diretório atual. Normalmente, o sistema operacional procura por comandos nas pastas listadas na variável de ambiente `PATH`. Se o comando não estiver em uma dessas pastas, você precisará especificar o caminho completo para o comando ou indicar que o sistema deve procurar no diretório atual.

Gabarito: Letra D

11. (FUNDATEC/PROCERGS/2023) Analise as assertivas abaixo sobre shell scripts no sistema operacional Linux:

- I. Uma maneira de criar um literal em um shell script e fazer com que o shell deixe a string intocada é colocar a string toda entre aspas simples.
- II. Um shell script deve sempre iniciar pela sequência de caracteres "barra" e "asterisco".
- III. Um caractere "cifrão" no início de uma linha indica que a linha é um comentário, ou seja, o shell script irá ignorar tudo o que estiver na linha após o caractere.

Quais estão corretas?

- A) Apenas I.
- B) Apenas I e II.
- C) Apenas I e III.
- D) Apenas II e III.
- E) I, II e III.

#### Comentários:

I. CORRETA - Ao usar aspas simples ao redor de uma string, cria-se um literal e impede a expansão de variáveis ou caracteres especiais dentro da string. Isso é útil quando se deseja que o conteúdo da string seja interpretado literalmente, sem expansão. Vamos a um exemplo:

```
palavra="coisa"
```

```
frase='Isso é uma $palavra'
```

Com aspas simples, a variável `$frase` não será expandida, ela será tratada como parte da string:



```
echo $frase
```

Gera como saída: Isso é uma \$palavra

Se fossem utilizadas aspas duplas em torno da string, a expansão de variáveis ocorreria:

```
frase="Isso é uma $palavra"
```

```
echo $frase
```

Gera como saída: Isso é uma coisa

II. ERRADA - Deve iniciar por #! - Ex.: #!/bin/bash

III. ERRADA - Uma linha de comentário começa por "cerquinha" (#).

Gabarito: Letra A

12. (FEPESE/Pref. de Balneário Camboriú-SC/2023) Qual o nome do comando ou argumento do loop while em programação shell script que possibilita o parse, ou o correto processamento e obtenção dos valores passados como argumentos na forma de opções, ou flags; isto é, na forma de letras precedidas por hífen (por ex. sh ./meuscript.sh -a abc -b def), ao invocar, via linha de comando, um shell script bash no Linux?

- A) arglist
- B) getopt
- C) parseopts
- D) parseargs
- E) optsparse

Comentários:

O getopt fornece uma maneira padrão de lidar com scripts que aceitam uma variedade de opções personalizadas. Ele permite analisar os argumentos passados para o script de forma fácil e eficiente. Com o getopt, é possível definir opções de linha de comando com argumentos e processá-las de maneira estruturada. Vejamos a sintaxe:

```
getopts optstring varname
```

Sendo que:

- optstring: string que define as opções que o script suporta. Se uma letra é seguida por dois pontos (:), isso significa que a opção requer um argumento. Se a letra não for seguida por dois pontos, a opção não requer um argumento;
- varname: nome da variável que será usada para armazenar a opção que foi analisada.

Gabarito: Letra B

13. (UNIFAL-MG/UNIFAL-MG/2023) Considere a seguinte tela de terminal em um servidor Linux Debian:



```
usuario@servidor:~/prova$ ls -l
total 172
-rw-r--r-- 1 usuario usuario 28672 ago  9 2022 Bibliografia.doc
-rw-r--r-- 1 usuario usuario 34463 ago  9 2022 Bibliografia.odt
-rw-r--r-- 1 usuario usuario  2402 jul 23 2022 bibliografia.txt
-rw-r--r-- 1 usuario usuario 49664 ago  9 2022 prova.doc
-rw-r--r-- 1 usuario usuario 32748 ago  7 2022 prova.odt
-rw-r--r-- 1 usuario usuario  8122 ago  6 2022 prova.txt
```

Considere ainda, o seguinte script bash criado pelo usuário (os números apenas identificam as linhas e não fazem parte do script):

```
1 #!/bin/bash
2 cd -
3 *** -czf /tmp/prova.tgz prova/*.doc prova/*.odt prova/*.txt
4 *** go-rwx /tmp/prova.tgz
5 *** -C /tmp/prova.tgz usuario@123.123.123.123:~/backup/prova.tgz
6 *** /tmp/prova.tgz
7 exit
```

São, respectivamente, comandos corretos para substituir os "\*\*\*" nas linhas 3, 4, 5 e 6 do script, de acordo com a descrição dos comandos abaixo:

- I. comando que funciona como um "arquivador" (ele junta vários arquivos em um só) mas pode ser usado em conjunto com um compactador (como o gzip ou zip);
- II. comando utilizado para mudar a permissão de acesso a um arquivo ou diretório;
- III. comando que permite a cópia de arquivos entre o cliente/servidor ssh;
- IV. comando utilizado para apagar arquivos e também pode ser usado para apagar diretórios e subdiretórios vazios ou que contenham arquivos.

- A) tac, pwd, ftp, grep
- B) cmp, tail, rsh, clear
- C) tar, chmod, scp, rm
- D) cat, chattr, sync, cut

#### Comentários:

Embora seja mostrado um shell script na questão, o foco é saber um pouco dos comandos. Vamos lá...

- I. tar geralmente é utilizado com o gzip e serve "juntar" vários arquivos em um só (o gzip compacta);
- II. chmod é usado para atribuir permissões (rwx) para o UGO (usuário dono, grupo e outros);
- III. scp (Secure Copy Protocol) é usado para copiar arquivos e diretórios entre hosts em uma rede. Ele fornece uma maneira segura de transferir dados usando o protocolo SSH (Secure Shell);



IV. rm é utilizado para remover (apagar) arquivos e também pode ser usado para apagar diretórios, desde que estejam vazios.

Gabarito: Letra C



## QUESTÕES COMENTADAS – CEBRASPE

1. (CEBRASPE/ANTT/2013) No ambiente Linux, um usuário comum pode terminar seu próprio processo por meio do comando kill, ação que não se restringe ao superusuário.

Comentários:

O comando kill envia um sinal ao processo. Pode-se utilizar o sinal TERM (terminar), KILL (matar), entre outros. Não se restringe ao root, pois cada usuário pode enviar sinais aos seus processos. O root pode enviar para o processo de cada usuário, claro. Exs.:

```
Kill -SIGKILL 12345
```

```
Kill -9 12345
```

Gabarito: Certo

2. (CEBRASPE/TJ-SE/2014) Alguns programas podem apresentar problemas que resultem no travamento do sistema operacional, o que pode ser resolvido, no Linux, por meio do comando Kill, que finaliza o processo, funcionalidade que pode ser acessada por meio de outro terminal.

Comentários:

O comando kill serve para enviar um sinal para um processo através de seu PID. Esse sinal pode ser para finalizar um processo, entre outros. A questão não fala que o kill serve apenas para finalizar, diz que é possível finalizar um processo com ele. Portanto, a questão está **correta**.

Gabarito: Certo

3. (CEBRASPE/MEC/2015) Julgue o item a seguir, relativo ao sistema operacional.

No Linux, se o processo pai alterar suas variáveis após a criação do processo filho, o filho detecta automaticamente a mudança, pois eles compartilham a mesma imagem da memória privada.

Comentários:

A partir do momento que o processo é clonado e o filho se torna independente do pai, cada um tem seus dados independentes, ou seja, a alteração em um processo não afetará o outro.

Gabarito: Errado

4. (CEBRASPE/STJ/2015) Julgue o próximo item, que se refere a sistemas operacionais.

No Linux, o comando renice -10 -u jdev -p 650 pode ser usado para reiniciar, no sistema, todos os processos do usuário jdev e o processo PID 650.

Comentários:



O comando renice serve para alterar o "niceness" de um processo (em execução), não serve para reiniciar! Vamos ver cada parâmetro do comando mostrado na questão (renice -10 -u jdev -p 650):

- 10: o novo valor de "niceness";
- u jdev: para os processos do usuário jdev;
- p 650: para o processo com PID 650.

Gabarito: Errado

5. (CEBRASPE/MPE-PI/2018) Julgue o item a seguir, acerca de sistemas operacionais.

Em sistemas operacionais Linux, os processos podem comunicar-se com outros processos por interrupções de software, que são canais entre os processos nos quais um processo pode escrever um fluxo de bytes para outro processo ler.

Comentários:

Uma interrupção de software (ou trap) é uma forma de interrupção gerada intencionalmente pelo próprio programa ou pelo sistema operacional para lidar com eventos específicos ou condições de erro. Normalmente são usadas para controlar o fluxo do programa, realizar operações privilegiadas ou lidar com exceções. Ou seja, interrupções de softwares não são formas de comunicação entre processos.

Gabarito: Errado

6. (CEBRASPE/FUB/2018) Julgue o item seguinte, a respeito dos sistemas operacionais Windows e Linux.

O Linux é um sistema operacional em que cada usuário consegue ter apenas um processo ativo por vez, processo esse que é iniciado automaticamente quando o sistema é carregado.

Comentários:

O Linux é um sistema operacional multitarefa e multiusuário. Cada usuário pode executar N processos. A chamada de sistema fork() faz o clone de um processo, sendo que o gerado passa a ser o filho daquele que foi clonado. A partir desse ponto o texto (código) do filho é alterado, junto com os dados.

Gabarito: Errado

7. (CEBRASPE/EBSERH/2018) No sistema operacional Linux, é possível alterar a prioridade de um processo já iniciado com o uso do comando nice.

Comentários:

O nice é utilizado para executar um programa e o renice é utilizado para alterar a prioridade do processo já em execução.



Gabarito: Errado

8. (CEBRASPE/TJ-PA/2020) No Linux, um processo, por si só, não é elegível para receber tempo de CPU. Essa ação depende, basicamente, do seu estado da execução. O processo está administrativamente proibido de executar no estado

- A) pronto.
- B) dormente.
- C) executável.
- D) parado.
- E) zumbi.

Comentários:

Os estados dos processos Linux podem ser:

Estado	Significado
Executável	Pode ser executado (pronto).
Dormente	Aguardando algum recurso (ex.: E/S).
Zumbi	Tentando se destruir.
Parado	Suspenso (sem permissão para ser executado).

Gabarito: D

9. (CEBRASPE/TJ-RJ/2021) O sistema de monitoramento de ativos computacionais alertou quanto à elevada quantidade de processos que estavam rodando em um servidor com o sistema operacional RedHat Enterprise Linux. Ao acessar diretamente pela linha de comando, foram identificados vários processos filhos sem a presença de pais, com consumos pequenos de recursos, tratando-se de processos zumbis.

10. Considerando essa situação hipotética, assinale a opção que apresenta os comandos utilizados no RedHat Enterprise Linux 5, 6 ou 7 para identificar os PIDs dos processos zumbis.

- A) `ps aux | awk {'print $8'} wc -l`
- B) `kill PID zumbi`





C) `ps aux | awk '$8=="Z" { print $0 }'`

D) `abrt-cli info zumbi`

E) `abrt-cli list zumbi`

### Comentários:

A linguagem de programação AWK permite a manipulação de textos a partir de uma sequência de padrões.

Print \$0: imprime a linha corrente.

abrt-cli: Lista, remove, imprime, analisa e reporta problemas. Para a questão, não faz sentido.

`ps aux | awk '$8=="Z" { print $0 }'`: gera o resultado de "ps aux" (tela abaixo) e joga para a AWK, que filtra apenas as linhas em que a coluna 8 seja igual a Z, depois imprime a linha corrente (\$0). Ou seja, para todos os processos no estado (STAT) zumbi, imprime na tela.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.2  0.3 101820 11444 ?        Ss   16:26   0:00 /sbin/init splash
root         2  0.0  0.0     0     0 ?        S    16:26   0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        I<   16:26   0:00 [rcu_gp]
root         4  0.0  0.0     0     0 ?        I<   16:26   0:00 [rcu_par_gp]
root         5  0.0  0.0     0     0 ?        I    16:26   0:00 [kworker/0:0-events]
root         6  0.0  0.0     0     0 ?        I<   16:26   0:00 [kworker/0:0H-kbloc
```

Gabarito: C



## QUESTÕES COMENTADAS – FGV

1. (FGV/TJ-RO/2015) Um desenvolvedor Java construiu um programa que, ao ser executado em sua máquina Linux, entrou em loop infinito. Considerando que há outros processos java rodando nessa máquina, o(s) comando(s) que esse desenvolvedor deveria executar para terminar a execução deste processo é(são):

- A) top e free;
- B) killall;
- C) ps e kill;
- D) ls e kill;
- E) shutdown.

### Comentários:

Para enviar um sinal ao processo para terminar a execução, pode-se utilizar kill ou killall. Kill deve especificar o PID do processo e killall o nome. Como a questão menciona que há outros processos JAVA e quer encerrar somente um, temos que usar o kill. Mas para saber qual o PID do processo, deve-se utilizar o ps (ou top). Então a combinação que atende ao que a questão quer é aplicar o ps para saber o PID dos processos e na sequência o kill para enviar o sinal a tal processo.

Gabarito: C

2. (FGV/IBGE/2017) Em ambiente Linux, o comando ps -axl permite visualizar informações sobre todos os processos existentes no sistema, em especial, os estados dos processos. Se o estado informado é Z, isto indica que o processo:

- A) é recém-criado e está pronto para executar;
- B) está no estado bloqueado, aguardando por um recurso;
- C) está morto, aguardando que o processo pai providencie sua extinção;
- D) está órfão, aguardando reparentização;
- E) sofreu preempção.

### Comentários:

Vamos aos parâmetros do comando ps:

- a = all;
- x = não terminais;
- l = mais campos no resultado.



Vimos na aula que zumbi é o estado quando o processo está tentando se destruir., ou seja, “está morto, aguardando que o processo pai providencie sua extinção”.

Gabarito: C

3. (FGV/IBGE/2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:

- A) create\_process;
- B) new\_process;
- C) fork;
- D) spawn;
- E) duplicate.

Comentários:

Não há chamada de sistema que crie um processo no Linux, o que acontece é que um processo existente realiza seu clone - chamada de sistema fork() - e o processo clone pode trocar o programa por um diferente. O original é o pai e a cópia gerada é o filho.

Gabarito: C

4. (FGV/Prefeitura de Niterói-RJ/2018) Sobre o gerenciamento de processos no sistema operacional Linux, assinale a afirmativa correta.

- A) O comando ps lista os processos em execução no sistema, trazendo informações sobre o quanto de processamento ou de memória cada um deles está consumindo.
- B) O ID do processo não pode ser reutilizado depois que o processo termina.
- C) Por meio do comando renice é possível alterar a prioridade de execução de um processo.
- D) Por padrão, os processos executados por um usuário iniciam com nível de prioridade -20, isto é, a prioridade mais baixa possível.
- E) Processos órfãos não possuem PPID.

Comentários:

(A) O ps pode exibir PID, UID, prioridade, terminal de controle de processos, memória consumida, estado atual. O quanto de processamento está consumindo, não! (B) Ex.: se o processo 4000 é terminado, depois de algum tempo outro processo pode utilizar o PID 4000, sem problemas. (C) Isso aí! O renice altera a prioridade de um processo já em execução, enquanto o nice apenas atribui um valor de prioridade a um programa a ser executado. (D) O padrão é iniciar com prioridade 0 e -20 é a maior prioridade possível! A menor é 19 (isso mesmo, invertido). (E) Possuem PPID de um processo que já não existe mais. Quando isso é constatado o processo init assume a “paternidade” desse processo órfão.



5. (FGV/MPE-AL/2018) Sobre gerenciamento de serviços e processos no sistema operacional Linux, analise as afirmativas a seguir.

I. Processo órfão é todo processo que já finalizou a execução, mas ainda possui uma entrada na tabela de processos.

II. O PID usado por um processo fica indisponível para reutilização enquanto o processo está presente na tabela de processos.

III. Daemons são processos zombies que rodam em background e não possuem um PPID associado.

Está correto o que se afirma em

A) I, apenas.

B) II, apenas.

C) III, apenas.

D) II e III, apenas.

E) I, II e III.

Comentários:

I. ERRADA - Processo órfão é todo processo filho cujo processo pai encerrou antes dele. Quando um processo pai termina antes do processo filho, o processo filho é herdado pelo processo init (processo com PID = 1), tornando-se um processo órfão.

II. CORRETA - O PID deve ser único, por isso, enquanto um PID é usado por um processo, ele fica indisponível para reutilização enquanto o processo está presente na tabela de processos (ou seja, está em uso ainda).

III. ERRADA - Daemons são processos em segundo plano, sem a intervenção direta do usuário. *Daemons* são frequentemente iniciados durante o boot do sistema operacional e continuam a ser executados enquanto o sistema está em funcionamento, realizando diversas tarefas e serviços em *background* (segundo plano).

6. (FGV/MPE-AL/2018) Sobre gerenciamento de processos no sistema operacional Linux, analise as afirmativas a seguir.

I. O PID do processo init é automaticamente atribuído ao PPID de processos que se tornam órfãos.



II. Processo zombie é a nomenclatura atribuída ao processo que possui autonomia própria, comportamento errático e que nunca termina sua execução, consumindo recursos do sistema indistintamente.

III. A prioridade de execução de um processo é indicada por um número que tem variação de -20 a 19 e quanto menor for este número, maior será a prioridade do processo.

Está correto o que se afirma em

- A) I, apenas.
- B) II, apenas.
- C) III, apenas.
- D) I e III, apenas.
- E) I, II e III.

Comentários:

I. CORRETA - Todo processo que se torna órfão, pega o init como processo pai, ou seja, o PID do processo init é atribuído ao PPID de processos órfãos.

II. ERRADA - Zumbi é o estado de processo que está tentando se destruir.

III. CORRETA - Exato! É invertido, mesmo, quando menor melhor a prioridade, ou seja, -20 é a maior prioridade e 19 é a menor. Cuidado com isso!

Gabarito: D

7. (FGV/Banestes/2021) No Linux, a função fork cria um novo processo, referido como filho, pela duplicação do processo que inicia a operação, referido como pai, de tal forma que:

- I. O processo filho herda o PID do processo pai.
- II. Pai e filho rodam (run) em espaços de memória diferentes.
- III. O filho herda todos os sinais de tempo do pai, no estado em que se encontram na execução da função fork.
- IV. Em caso de sucesso, a função fork retorna zero no processo pai.

O número de afirmativas corretas é:

- A) zero;
- B) uma;
- C) duas;
- D) três;



E) quatro.

Comentários:

I. ERRADA - O processo filho recebe um PID novo. Não pode haver dois processos com o mesmo PID!

II. CORRETA - A partir da chamada de sistema fork() há a clonagem e cada um fica em um espaço de memória diferente.

III. ERRADA - A partir da chamada de sistema fork(), cada um fica com seu código e seus dados. Não há nada de filho herdar qualquer coisa do pai.

IV. ERRADA - Em caso de sucesso, a função fork retorna zero no processo FILHO.

Gabarito: B

8. (FGV/TRT16/2022) No sistema operacional Linux, um técnico em informática digita o comando ps aux em uma janela de terminal. O nome da coluna que exibe o status dos processos é

A) STAT.

B) START.

C) STATS.

D) PID.

E) CMD.

Comentários:

Essa questão é boa para quem usa muito o comando ps com os argumentos aux. Sabendo que STATUS é estado e que o Linux gosta de abreviar bastante ou usar siglas, STAT seria uma boa resposta! Vamos ver na figura:

```
onworks@onworks:~$ ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT   START       TIME     COMMAND
root         1   0.5  0.4 167988 13004 ?        Ss     17:37   0:01   /sbin/init sp
root         2   0.0  0.0     0     0 ?        S      17:37   0:00   [kthreadd]
root         3   0.0  0.0     0     0 ?        I<    17:37   0:00   [rcu_gp]
```

Gabarito: A

9. (FGV/TJ-TO/2022) A ferramenta utilitária do sistema operacional GNU/Linux que exibe os processos em execução no sistema que mais utilizam a CPU é:

A) df;

B) grep;



- C) sed;
- D) stat;
- E) top.

**Comentários:**

Os comandos que exibem os processos são: top e ps. O comando jobs mostra os processos que estão parados ou rodando em segundo plano.

A banca ajudou, porque dentre os comandos que exibem processos, somente tem o top como opção. Como o nome sugere, ele mostra no "topo" os que mais utilizam o processador (CPU) e a atualização ocorre, por padrão, a cada dez segundos.

Gabarito: E





## QUESTÕES COMENTADAS – VUNESP

1. (VUNESP/UNESP/2013) Os sistemas operacionais FreeBSD, Linux e Unix compartilham grande parte dos comandos, pois são originários de um mesmo conceito. Dentre esses comandos comuns, está o que permite a apresentação dos estados dos processos correntes, denominado

- A) cd
- B) lpr
- C) ls
- D) ps
- E) proc

### Comentários:

Os comandos para exibir os processos são:

- ps: pode exibir PID, UID, prioridade, terminal de controle de processos, memória consumida, estado atual;
- top: mostra um resumo dos processos ativos e o uso de recursos, atualizando regularmente (processos mais ativos aparecem no topo, padrão de atualização = 10 segundos);
- pstree: exibe uma representação gráfica da hierarquia de processos em execução no sistema. Mostra os processos e seus relacionamentos em forma de árvore, facilitando a visualização da estrutura do sistema;
- jobs: mostra os processos que estão parados ou rodando em segundo plano.

Gabarito: D

2. (VUNESP/Pref. de P. Prudente-SP/2016) O administrador de um servidor com sistema operacional Linux deseja alterar a prioridade de execução de um processo. Para isso, ele deve utilizar o comando:

- A) ps
- B) top
- C) lsof
- D) pstree
- E) renice



### Comentários:

“Niceness” é a prioridade de um processo. O comando renice serve para alterar essa prioridade, que pode ir de -20 a 19, sendo que -20 é a maior prioridade e 19 é a menor. Isso, mesmo, é invertido!

Gabarito: E

3. (VUNESP/TJM-SP/2017) O administrador do computador com sistema operacional Linux executou o comando `kill -9 254`, no prompt de comandos de um terminal shell, para, corretamente,

A) enviar o sinal de interrupção para o processo 254, para avisar que ele será eliminado em 9 segundos.

B) interromper e eliminar imediatamente o processo de número 254.

C) enviar o sinal de interrupção para 9 processos a partir do processo de número 254.

D) sinalizar todos os processos ativos no sistema para interrupção imediata e eliminação em 254 segundos.

E) eliminar todos os processos executados pelo administrador após 9 segundos.

### Comentários:

`kill` envia um sinal a um processo, que pode ser SIGTERM, SIGKILL etc.

O valor de SIGKILL é 9.

O comando `kill` exige que o PID seja informado.

Então,

```
kill -9 254
```

significa que o sinal SIGKILL será enviado ao processo com PID = 254, ou seja, interrompe e elimina imediatamente o processo de número 254.

Gabarito: B

4. (VUNESP/Câm. de Indaiatuba-SP/2018) O administrador de um servidor com sistema operacional Linux executou o seguinte comando no prompt do terminal bash:

```
sudo renice -20 -p 1024
```

O resultado da execução desse comando será

A) a alteração da resolução do ambiente gráfico para 1024 pixels.

B) a apresentação de um novo terminal gráfico de cor verde e 1024 pixels de resolução.



- C) a redução da prioridade de execução do processo 1024.
- D) o aumento da prioridade de execução do processo 1024.
- E) o deslocamento da posição do ambiente gráfico para a esquerda.

#### Comentários:

“Niceness” é a prioridade de um processo. O comando `renice` serve para alterar essa prioridade, que pode ir de -20 a 19, sendo que -20 é a maior prioridade e 19 é a menor. Isso, mesmo, é invertido! Sabendo disso, o comando

```
sudo renice -20 -p 1024
```

faz com que o processo com PID tenha sua prioridade alterada para -20 (a maior prioridade possível). Supondo que a prioridade anterior era algo entre -19 e 19, então a prioridade aumentou. Mas e se antes fosse -20? Aí permaneceria a mesma, e não teria resposta! Vamos relevar esse detalhe e marcar a letra D, mesmo 😊.

Gabarito: D

5. (VUNESP/UNICAMP/2022) No sistema operacional Linux, o comando `top` pode exibir uma coluna cujo título é `TIME+`. No modo não-cumulativo, essa coluna exibe

- A) o tempo transcorrido desde o início da tarefa até o instante corrente.
- B) o tempo total de CPU que a tarefa usou desde seu início.
- C) a porcentagem total de tempo de CPU usada pela tarefa desde a última atualização das informações na tela.
- D) a porcentagem total de tempo de CPU usada pela tarefa desde seu início.
- E) o tempo total de ociosidade da tarefa desde que ela usou a CPU pela última vez.

#### Comentários:

O comando `top` exibe colunas de informações relacionadas aos processos e à utilização de recursos do sistema. Algumas das colunas mais comuns são:

- PID (Process ID): Identificador único atribuído a cada processo em execução;
- USER: Nome do usuário associado ao processo;
- PR (Priority): Prioridade do processo;
- NI (Nice value): Valor "nice" atribuído ao processo (prioridade);
- VIRT (Virtual Memory): Quantidade total de memória virtual utilizada pelo processo;



- RES (Resident Memory): Quantidade de memória física (RAM) usada pelo processo;
- SHR (Shared Memory): Quantidade de memória compartilhada utilizada pelo processo;
- S (%CPU): Percentual de uso do processador (CPU) pelo processo;
- MEM (%MEM): Percentual de uso da memória pelo processo;
- **+TIME**: Tempo total de processador (CPU) usado pelo processo desde o início;
- COMMAND: Nome do comando ou programa associado ao processo.

Gabarito: B



## QUESTÕES COMENTADAS – MULTIBANCAS

1. (CS-UFG/UFG/2019) Com respeito aos processos em execução no sistema Linux, o comando para listar processos em execução, sem se limitar aos processos executados a partir do shell corrente, é:

- A) ps
- B) ps -g
- C) ps -e
- D) ps -y

Comentários:

-e = every (idêntico ao -A = all).

Gabarito: C

2. (COMPERVE/Câm. de Parn.-RN/2019) Um processo é uma instância de um software (programa) em execução utilizando recursos do computador, tais como, processador e memória, para desempenhar as tarefas atribuídas a ele. Os sistemas operacionais oferecem ferramentas para gerenciamento de processos. Uma das formas de realizar esse gerenciamento é através de linha de comando utilizando um prompt de comando.

Sobre o gerenciamento de processos no Linux Ubuntu 18.04, utilizando um prompt de comando, considere as descrições de execução de comandos apresentadas abaixo.

I Lista os processos em ordem decrescente de consumo de recursos.

II Interrompe o processo de forma definitiva. É necessário informar o PID.

III Define a prioridade de um processo. Pode atualizar a prioridade de um processo existente ou criar um novo processo com prioridade definida.

I, II e III referem-se, respectivamente, aos comandos

- A) top, kill e nice.
- B) kill, top e ps.
- C) ps, nice e kill.
- D) top, ps e nice.

Comentários:

I - Mostra os processos que mais utilizam recursos primeiro, e atualiza a cada 10s, por padrão = top;



II - kill -SIGKILL (ou -9) interrompe o processo de forma definitiva (através do PID);

III - O nice define a prioridade de um processo (novo, ou seja, a ser executado). Em relação a atualizar a prioridade de um processo existente (em execução), o comando seria o renice, mas não há essa opção nas alternativas!

Gabarito: A

3. (Avança SP/Câm. de R. Pires-SP/2021) Assinale a alternativa que apresenta um comando do Linux Ubuntu capaz de "matar" um processo travado por meio do PID:

- A) kill.
- B) force.
- C) mix.
- D) death.
- E) exit.

Comentários:

Deram uma forçada nessa, heim? O kill envia um sinal ao processo, mas claro que tem a opção de "matar", com o sinal SIGKILL (ou valor 9).

Gabarito: A

4. (AOCP/BANESE/2022) O sistema operacional é responsável pela definição de quais processos têm prioridade sobre a CPU. No sistema operacional Linux, distribuição Ubuntu, qual comando pode ser utilizado para alterar a prioridade de execução de um programa?

- A) jobs
- B) nice
- C) ps
- D) top
- E) fg

Comentários:

jobs: lista os processos em segundo plano;

**nice**: altera a prioridade de um programa a ser executado (se o processo já estivesse em execução, aí seria o renice);

ps: lista os processos;



top: lista os processos em ordem decrescente de uso de recursos;

fg: coloca um processo que estava em background (segundo plano) para foreground (primeiro plano).

Gabarito: B

5. (CS-UFG/UFG/2023) Dentre os comandos para a manutenção de sistemas Linux, um comando que apresenta os processos em execução, com opções para fornecer o nome de usuário e a hora de início de processo, é o comando

A) which.

B) vim.

C) ping.

D) ps.

Comentários:

Dos comandos apresentados, apenas o ps mostra os processos. Poderia ser o jobs, top, pstree também, cada um com suas características.

Gabarito: D



## QUESTÕES COMENTADAS - CEBRASPE

1. (CEBRASPE/TCE-RN/2015) O Linux tem a capacidade de montar sistemas de arquivos do dispositivo `/dev/sdc3`, que tem o sistema de arquivos do tipo `ext4` e, no ponto de montagem `/teste`, deve ser utilizado o comando `mount -t ext4 /dev/sdc3 /teste`.

Comentários:

A forma básica do comando `mount` possui dois parâmetros:

- Dispositivo contendo o sistema de arquivos a ser montado (ex.: `/dev/sda3`);
- Ponto de montagem (ex.: `/teste`).

Além disso, temos o parâmetro `-t` que serve para definir o tipo de sistema de arquivos.

Gabarito: Certo

2. (CEBRASPE/TCE-PA/2016) A opção `mount -t vfat` permite fazer o becape dentro de uma partição de disco DOS/Windows, caso em que a hierarquia de diretório aparece como parte do sistema de arquivos Linux.

Comentários:

Vimos que `-t` é o parâmetro para especificar o tipo de sistema de arquivos e que `"vfat"` especifica a família FAT, a qual é utilizada em sistemas DOS/Windows.

Gabarito: Certo

3. (CEBRASPE/ABIN/2018) O Linux não impede a alteração do nome do superusuário, nem a criação de contas com UID igual a 0, embora essas ações não sejam recomendadas.

Comentários:

Como vimos na aula, não é recomendado fazer isso, mas é possível alterar o nome do root para qualquer coisa, e, também é possível criar novas contas de superusuário (com UID = 0).

Gabarito: Certo

4. (CEBRASPE/CODEVASF/2021) Com relação à administração do sistema operacional Linux, julgue o item seguinte.

Para atribuir permissões de leitura, escrita e execução ao proprietário do arquivo de nome `documento1.txt`, sem qualquer permissão para grupos ou qualquer outro usuário, é correto o uso do comando `chmod 0177 documento1.txt`.

Comentários:

Vamos deixar o 0 à esquerda de lado e partir para o "UGO":





U = 1 = 001 = --x (proprietário, somente execução);

G = 7 = 111 = rwx (grupo, completo);

O = 7 = 111 = rwx (outros, completo).

O correto seria:

```
chmod 700 documento1.txt
```

Gabarito: Errado

5. (CEBRASPE/TRT8/2022) Um administrador de rede precisa modificar as permissões na lista de controle de acesso, utilizando o Red Hat Enterprise Linux 8, para o arquivo TRT8 de propriedade do usuário root, que, por sua vez, pertence ao grupo root. Nesse caso, apenas Paulo deve ter permissão rw- e os demais usuários devem ter permissão r--.

Com base nessa situação hipotética, assinale a opção que indica o comando correto para executar a configuração referida.

A) # chmod rw- paulo TRT8

B) # getfacl TRT8 -m u:paulo rw

C) # setfacl -m u:paulo:rw- TRT8

D) # setfacl TRT8 -m u:paulo rw

E) # chmod paulo 640 TRT8

Comentários:

setfacl: define listas de controle de acesso (ACLs) de arquivos e diretórios. Na linha de comando, uma sequência de comandos é seguida por uma sequência de arquivos (que por sua vez pode ser seguida por outra sequência de comandos, e assim por diante). As opções -m (-modify) e -x (-remove) esperam uma ACL na linha de comandos. A diretiva "u" define o usuário, "g" define o grupo.

Assim, temos que modificar (-m) para o usuário (u) paulo. Logo, a resposta é:

```
# setfacl -m u:paulo:rw- TRT8
```

Gabarito: C

6. (CEBRASPE/POLITEC-RO/2022) Em Linux, o comando equivalente a chmod a-rxw,u+r,g+rw arquivo para alterar as permissões de um arquivo é

A) chmod 465 arquivo.

B) chmod 456 arquivo.

C) chmod 460 arquivo.

D) chmod 654 arquivo.



E) chmod 046 arquivo.

Comentários:

Vamos analisar por partes:

chmod a-rwx, → primeiro remove todas permissões para todos (a = all);

u+r, → na sequência, adiciona a permissão de leitura (r) para o dono (u);

g+rw → por fim, adiciona as permissões de leitura e escrita (rw) para o grupo do dono.

No fim de tudo isso, temos:

U = r-- = 100 = 4;

G = rw- = 110 = 6;

O = não temos nada = 0.

Resposta: chmod 460 arquivo

Gabarito: C

7. (CEBRASPE/PO-AL/2023) A respeito de sistemas operacionais, julgue o item subsecutivo.

O sistema Linux possibilita atribuir permissões de acesso diferentes em um mesmo diretório para distintas categorias: usuário dono, grupo dono do arquivo e demais usuários.

Comentários:

É o famoso UGO! Com o comando chmod podemos definir as permissões rwx (leitura, escrita, execução) para o usuário dono (U), grupo (G) ou outros (O).

Gabarito: Certo



## QUESTÕES COMENTADAS - FGV

1. (FGV/PGE-RO/2015) Para verificar quais sistemas de arquivos estão correntemente montados em ambiente Linux, o administrador deve utilizar o comando:

- A) fsmount;
- B) mountfs;
- C) mount;
- D) showfs;
- E) fsck.

Comentários:

O comando mount, sem parâmetros, mostra quais sistemas de arquivos estão montados:

```
[root@localhost ~]# mount
root on / type 9p (rw,relatime,dirsync,access=client,trans=virtio)
devtmpfs on /dev type devtmpfs (rw,relatime,size=125948k,nr_inodes=31487,mode=755)
none on /proc type proc (rw,relatime)
none on /sys type sysfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600,ptmxmode=000)
[root@localhost ~]#
```

Gabarito: C

2. (FGV/SEPOG-RO/2017) Considere o comando Linux a seguir aplicado a um arquivo regular chamado teste:

```
chmod 546 teste
```

As novas características do arquivo chamado teste, após a execução do comando, serão

- A) permissão de escrita para todos os usuários e execução somente pelo proprietário e pelos membros do grupo.
- B) permissão de leitura para todos os usuários, execução somente pelo proprietário e escrita pelos outros usuários.
- C) somente o proprietário pode executar, os componentes do grupo podem ler e escrever e os outros usuários somente podem listar seu nome no diretório.
- D) todos os usuários podem executar o arquivo, somente o proprietário pode ler e escrever e os componentes do grupo podem ler.
- E) todos os usuários podem ler e executar mas somente o proprietário pode modificar o arquivo.

Comentários:



Vamos ao famoso UGO para os valores 546:

$$U = 5 = 101 = r-x$$

$$G = 4 = 100 = r--$$

$$O = 6 = 110 = rw-$$

Então, o usuário (dono) do arquivo teste tem permissão para ler e executar, o grupo tem permissão para ler e os outros têm permissão para ler e escrever.

Gabarito: B

3. (FGV/Prefeitura de Niterói-RJ/2018) Pedro é o proprietário do arquivo header.txt em um sistema Linux e gostaria de assegurar que somente ele tivesse permissão de leitura, gravação e execução a este arquivo, enquanto que todos os demais usuários com acesso ao sistema tivessem somente a permissão de leitura.

Assinale a opção que indica o comando que pode ser usado para conseguir esse objetivo.

A) `chmod ug+r header.txt`

B) `chmod 766 header.txt`

C) `chmod 722 header.txt`

D) `chmod +r header.txt`

E) `chmod 744 header.txt`

Comentários:

Vamos fazer na sequência o UGO (usuário, grupo, outros):

$$U = rwx = 111 = 7$$

$$G = r-- = 100 = 4$$

$$O = r-- = 100 = 4$$

Gabarito: E

4. (FGV/MPE-AL/2018) Para definir o script `/usr/local/bin/meuscript` como executável no sistema operacional Linux, devemos usar o comando

A) `chmod og /usr/local/bin/meuscript`

B) `chmod 000 /usr/local/bin/meuscript`

C) `chmod +x /usr/local/bin/meuscript`

D) `chmod ugo-x /usr/local/bin/meuscript`

E) `chmod 666 /usr/local/bin/meuscript`

Comentários:



A única que adiciona a execução "x" é a letra C. Como não especifica para "quem", todo o UGO recebe a permissão para a execução.

Na letra E, se transformarmos 666 em binário, temos: 110 110 110 = rw- rw- rw- (não tem "x" para ninguém!).

Gabarito: C

5. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando que permite estabelecer permissões para arquivos e diretórios.

- A) chmod
- B) chper
- C) dir
- D) permit
- E) useradd

#### Comentários:

Questão bastante manjada. O chmod é utilizado para dar/alterar permissões nos arquivos e diretórios. Um exemplo é:

`chmod 751 /home/evandro/arquivo.txt` vamos analisar o "UGO":

- Usuário = 7 = 111 = completo (*read, write, execute*);
- Grupo = 5 = 101 = leitura e execução (*read, execute*);
- Outros = 1 = 001 = apenas execução (*execute*).

Um outro exemplo:

`chmod +x teste` dá a permissão de execução ao arquivo "teste", mas para quem? U? G? O? Como não está explícito para quem é, todos recebem o "+x"!

Gabarito: A

6. (FGV/Câmara de Taubaté-SP/2022) No ambiente Linux, assinale a função do comando `sudo`.

- A) Buscar na Internet atualizações para o sistema operacional.
- B) Configurar as opções de idiomas e padrões de codificação de caracteres de texto.
- C) Executar comandos com privilégios mais altos, ou como outro usuário.
- D) Gerenciar as contas e permissões de usuários.
- E) Remover arquivos temporários e memória de trabalho, tanto originados pelo uso da Internet como dos aplicativos comuns.



## Comentários:

O comando sudo permite a usuários comuns obter privilégios de outro usuário, em geral o root, para executar tarefas específicas dentro do sistema de maneira segura e controlável pelo administrador. Ao ser executado, há uma consulta ao arquivo /etc/sudoers, que possui os usuários e comandos habilitados em cada host. É solicitada a senha do próprio usuário que executou o sudo e, na sequência, é executado o comando que se encontra após o sudo. É importante salientar que é mantido um log dos comandos executados, hosts, usuários, diretório, data/hora etc. O exemplo mais comum é sem a especificação do usuário:

```
$ sudo apt-get install abc
```

Gabarito: C



## QUESTÕES COMENTADAS - FCC

1. (FCC/TRT4/2015) A possibilidade de compartilhar arquivos entre diferentes sistemas operacionais é fundamental para aumentar a produtividade computacional. A montagem automática de uma partição com sistema de arquivos CIFS, durante o boot do servidor com sistema operacional Linux, deve ser configurada no arquivo

- A) /etc/fstab.
- B) /boot/mount.
- C) /etc/mount.
- D) /boot/inittab.
- E) /etc/initd.

### Comentários:

O arquivo /etc/fstab possui informações sobre os sistemas de arquivos que o sistema pode montar, e o arquivo /etc/mtab possui uma lista dos dispositivos montados (através do comando mount).

Gabarito: A

2. (FCC/TRT23/2016) Uma partição NFS remota deve ser montada em um computador com sistema operacional Linux. Para especificar, no comando mount, que a partição é NFS, deve-se utilizar a opção:

- A) -n.
- B) -f.
- C) -i.
- D) -s.
- E) -t.

### Comentários:

- n: monta s/ escrever em /etc/mtab;
- f: "fakes", útil pra debugar;
- i: não chama o /sbin/mount;
- s: tolera opções "sloopy", ao invés de falhar;
- t: especifica o tipo de sistema de arquivos.

Gabarito: E



3. (FCC/TJ-MA/2019) Um Analista digitou o comando `chmod u=rwx,g=rx,o=r processo` para definir as permissões de acesso ao arquivo processo. O comando equivalente usando a notação octal é:

- A) `chmod 713 processo`
- B) `chmod 777 processo`
- C) `chmod 134 processo`
- D) `chmod 754 processo`
- E) `chmod 671 processo`

Comentários:

$$U = rwx = 111 = 7$$

$$G = r-x = 101 = 5$$

$$O = 1-- = 100 = 4$$

Então, temos "`chmod 754 processo`".

Gabarito: D

4. (FCC/SANASA Campinas/2019) No Linux, para definir somente para o owner do arquivo teste.txt apenas a permissão de read e write, utiliza-se o comando

- A) `chmod 700 teste.txt`
- B) `chmod 724 teste.txt`
- C) `chmod 666 teste.txt`
- D) `chmod 600 teste.txt`
- E) `chmod 777 teste.txt`

Comentários:

Vamos ao UGO:

$$U = \text{apenas "read" e "write"} = rw- = 110 = 6;$$

$$G = \text{nada} = 0;$$

$$O = \text{nada} = 0.$$

Resposta: `chmod 600 teste.txt`

Gabarito: D

5. (FCC/TRF4/2019) No Linux, para atribuir somente permissão de leitura e escrita para o dono do arquivo trf4r.sh e somente leitura para usuários do mesmo grupo, utiliza-se o comando





- A) `chmod 640 trf4r.sh`
- B) `chmod 755 trf4r.sh`
- C) `chmod 777 trf4r.sh`
- D) `chmod 341 trf4r.sh`
- E) `chmod 666 trf4r.sh`

#### Comentários:

Vamos ao UGO:

U = leitura e escrita = `rw-` =  $110 = 6$ ;

G = leitura = `r--` =  $100 = 4$ ;

O = nada = `0`.

Resposta: `chmod 640 trf4r.sh`

Gabarito: A

6. (FCC/TRT5/2022) Um analista, utilizando o usuário padrão definido no Red Hat, listou o conteúdo de um diretório após ter criado os objetos `arquivo.txt` e `diretório`, obtendo a saída abaixo.

```
-rw--w-r-- 1 analista analista 0 Oct 12 18:32 arquivo.txt  
drwx-wxr-x 2 analista analista 4096 Oct 12 18:32 diretório
```

Com base nesta saída, o valor do `umask` foi definido em

- A) 0002.
- B) 0244.
- C) 0202.
- D) 0042.
- E) 0642.

#### Comentários:

A utilização do `umask` com números octais funciona da seguinte forma: pega-se `777` e diminui pela máscara definida, no caso de diretórios, ou `666` diminui pela máscara definida, no caso de arquivos. Por exemplo, se a máscara definida for `"447"`, então teremos diretórios criados com as permissões `330 (-wx-wx---)` e arquivos com permissões `220 (-w--w----`). Obs.: Se ficar negativo, considere `0`, como é o caso de `666 - 447`, ficaria `2 2 -1`, então consideramos `2 2 0`).

Vamos analisar o que foi passado na questão (um arquivo e um diretório):

- `-rw--w-r-- 1 analista analista 0 Oct 12 18:32 arquivo.txt`



Temos as permissões  $rw- -w- r-- = 110\ 010\ 100 = 624$ .

Como é arquivo, eu pego 666 e diminuo o 624 = 042.

- drwx-wxr-x 2 analista analista 4096 Oct 12 18:32 diretório

Temos as permissões  $rx- -wx r-x = 111\ 011\ 101 = 735$

Como é diretório, eu pego 777 e diminuo o 735 = 042.

Então a resposta é: 0042 (o 0 à esquerda são aquelas permissões especiais, mas como é 0 nem nos interessa).

Gabarito: D



## QUESTÕES COMENTADAS - VUNESP

1. (VUNESP/PRODEST-ES/2014) Nos sistemas operacionais Linux, o programa "sudo" pode ser utilizado para que um usuário possa executar comandos com privilégios diferentes dos definidos para seu usuário. No entanto, para o seu correto funcionamento, é fundamental que o usuário em questão

- A) esteja autorizado nas configurações do "sudo".
- B) seja membro do grupo "admin".
- C) seja o único usuário autenticado na máquina.
- D) possua um único terminal aberto.
- E) conheça a senha do superusuário (root).

### Comentários:

Ao contrário do que muitos pensam, o usuário que for executar com privilégios de outro (o root, por exemplo), não precisa saber a senha desse outro usuário, senão seria uma falha grave de segurança! Então, quem for usar o "sudo" deve estar cadastrado em um arquivo de configurações do "sudo", o arquivo "/etc/sudoers".

Gabarito: A

2. (VUNESP/Câm. de Itatiba-SP/2015) Um novo disco rígido foi adquirido, instalado, particionado e formatado em um computador com sistema operacional Linux. Para que a partição do disco rígido seja montada automaticamente quando o sistema operacional é inicializado, o comando de montagem dessa partição deve ser incorporado no arquivo

- A) dconf.
- B) fstab.
- C) init.
- D) mount.
- E) udisk.

### Comentários:

O arquivo /etc/fstab possui informações sobre os sistemas de arquivos que o sistema pode montar, e o arquivo /etc/mstab possui uma lista dos dispositivos montados (através do comando mount).

Gabarito: B

3. (VUNESP/Pref. de Birigui-SP/2019) No ambiente Linux, considere a seguinte permissão:

--WX-WX---



Ela equivale, em notação octal, a:

- A) 0220
- B) 0330
- C) 0440
- D) 0550
- E) 0770

#### Comentários:

Observe que o primeiro são nove espaços, então não tem o primeiro para dizer se é arquivo, diretório etc. Vamos separar em UGO:

$$U = -wx = 011 = 3;$$

$$G = -wx = 011 = 3;$$

$$O = --- = 000 = 0.$$

Resposta: 0330 (Obs.: o zero à esquerda se refere às permissões especiais, mas como é zero, não tem nenhuma!).

Gabarito: B

4. (VUNESP/Câmara de Mauá-SP/2019) Um usuário de um computador com sistema operacional Linux criou um arquivo chamado backup.sh, que contém um script para efetuar uma rotina de backup de arquivos importantes da máquina. Porém, mesmo sendo dono do arquivo backup.sh e tendo permissão de leitura e escrita, o usuário observou que não consegue executar esse script por falta de permissão de execução.

Assinale a alternativa que contém um comando a ser executado pelo mesmo usuário que resolveria esse problema.

- A) `chmod u-wx backup.sh`
- B) `chmod ugo-x backup.sh`
- C) `chmod 744 backup.sh`
- D) `chmod 666 backup.sh`
- E) `chmod 644 backup.sh`

#### Comentários:

O foco está aqui: "o usuário observou que não consegue executar esse script por falta de permissão de execução." Vamos analisar a alternativa que dá a permissão para execução (x).

A) `chmod u-wx backup.sh` → "u-wx" quer dizer que o usuário não terá permissão para escrever (w) nem executar (x).

B) `chmod ugo-x backup.sh` → "ugo-x" quer dizer que ninguém terá permissão para executar (x).



- C) `chmod 744 backup.sh` → "744" quer dizer 111 100 100, ou seja, U = rwx, G = r--, O = r--.  
Neste caso, o usuário dono terá a permissão de execução (x).
- D) `chmod 666 backup.sh` → "666" quer dizer 110 110 110, ou seja, U = rw-, G = rw-, O = rw-.  
Neste caso, ninguém terá a permissão de execução (x).
- E) `chmod 644 backup.sh` → "644" quer dizer 110 100 100, ou seja, U = rw-, G = r--, O = r--.  
Neste caso, ninguém terá a permissão de execução (x).

Gabarito: C

5. (VUNESP/UNICAMP/2022) Por meio da execução do comando `ls -l` em um terminal Linux, observou-se que o arquivo `myfile` apresenta as seguintes permissões:

`-rw-rw-r--`

Após a execução com sucesso do comando `chmod 644 myfile` as permissões de `myfile` passam a ser:

- A) `-r--r--r--`  
B) `-r--rw-rw-`  
C) `-rw-rw-rw-`  
D) `-rw-r--r--`  
E) `-rw-rw-r--`

Comentários:

São dez espaços, então o primeiro só diz que é um arquivo ("-"). Agora vamos ver como fica o UGO com 644:

U = 6 = 110 = "rw-";

G = 4 = 100 = "r--";

O = 4 = 100 = "r--".

Então, temos como resposta: "- rw- r-- r--".

Gabarito: D

6. (VUNESP/Pref. de Sorocaba-SP/2022) Em um computador com o sistema operacional Linux, um subdiretório contém um arquivo chamado `arq.sh`. Nesse subdiretório, o usuário dono do arquivo, que não possui privilégios de root, executou com sucesso o seguinte comando em um terminal:

`chmod ugo-rwx arq.sh`

Em seguida, o comando a seguir foi executado:

`cat arq.sh`



O resultado obtido foi

- A) a exibição do conteúdo de arq.sh.
- B) a exclusão de arq.sh.
- C) uma mensagem de erro informando que não há permissão.
- D) a execução de arq.sh, uma vez que é um arquivo de shell script
- E) a exibição do conteúdo de arq.sh concatenado com ele mesmo.

Comentários:

- `chmod ugo-rwx arq.sh`

Para U, G e O: retirou as permissões r, w e x. Ou seja, retirou tudo para todos!

- `cat arq.sh`

Como não há permissão para leitura (r), deve aparecer mensagem informando isso!

Gabarito: C

7. (VUNESP/Pref. de Pindamonhangaba-SP/2023) Um usuário de sistema operacional Linux utilizou o comando `chmod 321` em um determinado arquivo. Assinale a alternativa que apresenta as permissões do arquivo após o comando.

- A) `r--w---x`
- B) `rw-rw-r--`
- C) `-wx-w---x`
- D) `rw-wx--x`
- E) `rw--w-r--`

Comentários:

Vamos ver o UGO:

U = 3 = 011 = -wx;

G = 2 = 010 = -w-;

O = 1 = 001 = --x.

Então, temos: "-wx-w---x".

Gabarito: C



## QUESTÕES COMENTADAS - CESGRANRIO

1. (CESGRANRIO/B. da Amazônia/2022) O administrador de um sistema operacional Linux mudou o atributo de dono (owner) do arquivo de um programa para o usuário root e o atributo de grupo desse arquivo para root.

O comando a ser utilizado para que esse programa seja executado com permissão de root por qualquer usuário do sistema é o

- A) chmod 0555
- B) chmod 1555
- C) chmod 2555
- D) chmod 3555
- E) chmod 4555

### Comentários:

Podemos ver que envolve o uso dos privilégios especiais (o primeiro valor em octal), coisa rara, mas aqui está um belo exemplo! Valor ver o conceito do setuid abaixo.

setuid: privilégio do dono e não de quem executou, ex.: passwd executado pelo usuário evandro  privilégio de root, mas quem executou foi o usuário evandro!

Na figura abaixo vemos que para o setuid estar habilitado, o primeiro octeto deve estar com o valor 100 = 4. Então a resposta é "chmod 4555".

			U			G			O		
uid	gid	sticky	r	w	x	r	w	x	r	w	x

Gabarito: E



## QUESTÕES COMENTADAS - MULTIBANCAS

1. (FUNCAB/MDA/2014) No sistema operacional Linux, a criação de um grupo permite que um conjunto de usuários diferentes possua acesso a um mesmo arquivo. O comando do Linux que tem como objetivo alterar o grupo de um arquivo/diretório é o

- A) chown.
- B) chgroup.
- C) chset.
- D) chfile.
- E) chgrp.

### Comentários:

Sabemos que o chown permite a alteração de usuário/grupo, mas a questão fala somente em grupo. Tem uma pegadinha na letra B, mas seguindo o padrão de 5 letras (chown), a resposta é "chgrp".

Gabarito: E

2. (UFRRJ/UFRRJ/2015) Qual o comando para agregar uma mídia formatada em NTFS no sistema de arquivos de um sistema operacional UNIX-like?

- A) umount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.
- B) mount /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.
- C) umout /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.
- D) mount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.
- E) umount /dev/diskid/DISK-JP1440HA196TKS -t ntfs /mnt.

### Comentários:

A forma básica do comando mount possui dois parâmetros:

- Dispositivo contendo o sistema de arquivos a ser montado (ex.: /dev/sda3);
- Ponto de montagem (ex.: /teste).

Além disso, temos o parâmetro -t que serve para definir o tipo de sistema de arquivos.

Gabarito: D

3. (FEPESE/Pref. de Lages-SC/2016) Ao tentar recuperar um sistema Linux Ubuntu, o administrador de sistemas constata que a partição /boot foi montada em modo somente leitura.

Assinale a alternativa cujo comando possibilitará a gravação de arquivos nesta partição.





- A) mount -o remount,rw /boot
- B) mount -a remount,rw /boot
- C) mount -a remount,ro /boot
- D) mount -a remount,wr /boot
- E) mount -o remount,raw /boot

#### Comentários:

Comando mount, o parâmetro **-o** possibilita definir opções, separadas por vírgulas, como por exemplo:

- ro: somente leitura;
- **rw: leitura e escrita;**
- exec: permite a execução de binários;
- noexec: não permite a execução de binários;
- remount: tenta remontar um sistema de arquivos já montado.

Gabarito: A

4. (FCM/IF Sudeste-MG/2016) O Linux permite ao superusuário (root) executar qualquer operação válida em qualquer arquivo ou processo, inclusive algumas chamadas de sistema podem ser executadas somente pelo superusuário. A operação que NÃO é exclusiva do superusuário é

- A) configurar interfaces de rede.
- B) configurar o relógio do sistema.
- C) alterar as permissões de um arquivo.
- D) aumentar os limites de uso dos recursos.
- E) definir o nome de host (hostname) do sistema.

#### Comentários:

De todas as alternativas mostradas, sabemos que alterar as permissões de um arquivo qualquer um pode fazer, não é? Claro que qualquer um pode alterar as permissões (comando chmod) do seu arquivo e não de outros (a não ser que tenha permissão para isso).

Gabarito: C

5. (IBFC/EBSERH/2016) O comando sudo do sistema operacional Linux é muito poderoso, permitindo que usuários comuns obtenham privilégios de super usuário. Por questões de segurança, o administrador precisa definir no arquivo \_\_\_\_\_, quais usuários podem executar sudo, em quais computadores podem fazê-lo e quais comandos podem executar através dele. Assinale a alternativa que complete correta e respectivamente a lacuna:

- A) /etc/sudoers
- B) /root/sudouser



- C) /root/sudoers
- D) /usr/sudouser
- E) /etc/sudouser

#### Comentários:

Pense o seguinte: os arquivos de configuração geralmente ficam no diretório "/etc". E para executar o sudo existe um arquivo com as configurações de quem e do que pode ser executado com o sudo, que é o arquivo "sudoers". Em inglês, quem caminha ("walk") é um "walker", então quem faz um "sudo" é um "suoer", assim fica mais fácil lembrar 😊.

Gabarito: A

6. (INAZ do Pará/DPE-PR/2017) Qual a alternativa que corresponde à linha de comando para que usuários comuns possam utilizar o comando administrativo apt-get dist-upgrade?

- A) # apt-get dist-upgrade
- B) \$ sudo apt-get gw dist-upgrade
- C) # sudo apt-get gw up dist-upgrade
- D) \$ sudo apt-get dist-upgrade
- E) # sudo apt-get dist-upgrade gw now

#### Comentários:

Usuário comum tem o prompt "\$", então temos duas alternativas possíveis. Para executar o "apt-get dist-upgrade" com o "sudo" é só colocar o "sudo" na frente.

Gabarito: D

7. (IF-ES/IF-ES/2019) Os administradores de redes precisam entender com clareza sobre o gerenciamento de contas de usuários nos Sistemas Operacionais de Servidores Linux. Em relação ao sistema de contas Linux, analise as afirmativas abaixo:

I – O arquivo /etc/shadow serve para manter senhas criptografadas seguras quanto a acesso não autorizado.

II – Um novo usuário pode ser adicionado utilizando-se ferramentas, como adduser ou useradd.

III – Os usuários dos sistemas Linux recebem um número UID referente a sua identificação, porém o usuário root, por questões de segurança, não dispõe de UID.

É CORRETO o que se afirma em:

- A) I, II e III.
- B) I e II, apenas
- C) I e III, apenas.



- D) II e III, apenas.  
E) Nenhuma das afirmativas está correta.

**Comentários:**

(I) O arquivo /etc/shadow serve para manter o hash das senhas, entre outras informações relacionadas. (II) Um novo usuário pode ser adicionado utilizando-se ferramentas, como adduser ou useradd e automaticamente os dados serão acrescentados nos arquivos /etc/passwd e /etc/shadow. (III) Os usuários dos sistemas Linux recebem um número UID referente a sua identificação. O root recebe o UID 0.

Gabarito: B

8. (CS-UFG/IF Goiano/2019) Na configuração do Sistema Linux, o comando chmod modifica as permissões de arquivos com respeito à execução, leitura e escrita. A expressão chmod 751 arq2 é equivalente a:

- A) chmod u=rx,g=rwx,o=x arq2  
B) chmod u=rx,g=rwx,o=r arq2  
C) chmod u=rwx,g=rx,o=x arq2  
D) chmod u=rwx,g=rx,o=r arq2

**Comentários:**

U = 7 = 111 = rwx

G = 5 = 101 = r-x

O = 1 = 001 = --x

Tirando os "tracinhos": u=rwx,g=rx,o=x

Gabarito: C

9. (CCV-UFC/UFC/2019) Qual dos itens abaixo contém o comando do sistema operacional Linux capaz de mudar o grupo de um arquivo ou diretório do sistema?

- A) df  
B) sed  
C) chown  
D) chmod  
E) passwd

**Comentários:**

Questão interessante porque não mostra o comando "chgrp". Mas sabemos que o chown permite a alteração de usuário/grupo, ou seja, é possível trocar o grupo também!



Gabarito: C

10. (COSEAC/UFF/2019) No Sistema Operacional Linux o diretório local padrão do superusuário é:

- A) /home.
- B) /usr.
- C) /root.
- D) /lib.
- E) /dev.

Comentários:

/home – home dos usuários “comuns”.

/usr – onde a maioria dos programas ficam instalados, executáveis e bibliotecas de todos os principais programas.

/root – é o home do root!

/lib – bibliotecas necessárias para os binários essenciais para as pastas /bin e /sbin.

/dev – links para dispositivos de hardware.

Gabarito: C

11. (CONSULPLAN/MPE-PA/2022) Um administrador de sistema Linux, conectado como superusuário, realizou o seguinte comando: `chmod 751 /home/file.sh`. Sobre tal comando, analise as afirmativas a seguir.

- I. O usuário dono do arquivo terá permissão para ler, escrever e executar o arquivo.
- II. Os usuários do grupo do arquivo terão permissão para ler e executar o arquivo.
- III. Todos os demais usuários (exceto os citados anteriormente) não terão nenhum tipo de permissão sobre o arquivo.

Está correto o que se afirma em

- A) I, II e III.
- B) I e II, apenas.
- C) I e III, apenas.
- D) II e III, apenas.

Comentários:

I. CORRETA -  $U = 7 = 111 = rwx$ .



II. CORRETA - G = 5 = 101 = r-x.

III. ERRADA - O = 1 = 001 = --x.

Gabarito: B

12. (IBADE/RBPREV/2023) Qual a função do comando `chmod go-rx`, utilizado no terminal do sistema operacional Linux?

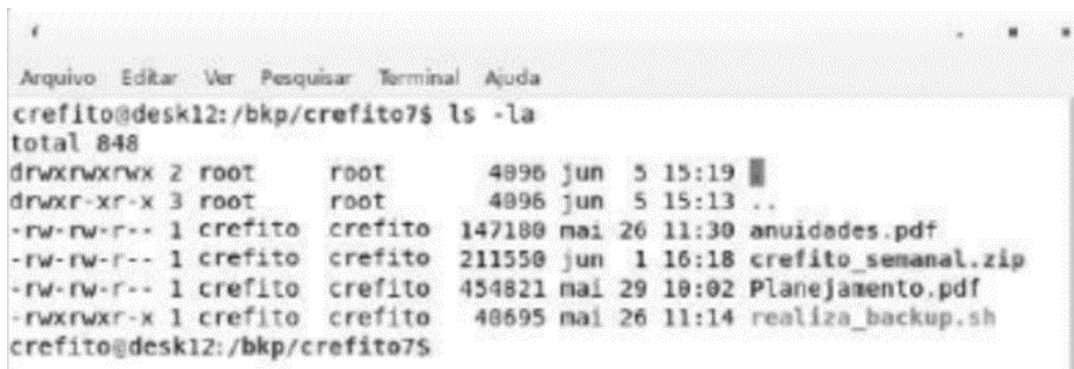
- A) Dar permissão de leitura, execução e escrita para todos os usuários.
- B) Remover a permissão de escrita e execução aos usuários do grupo e outros.
- C) Dar permissão de leitura e execução para todos os usuários.
- D) Remover a permissão de leitura e execução aos usuários do grupo e outros.
- E) Remover a permissão apenas de execução aos usuários do grupo e outros.

Comentários:

"go-rx" significa que o G (grupo) e O (outros) terão as permissões "rx" (leitura e execução) removidas (- que aparece antes de rx).

Gabarito: D

13. (Quadrix/REFITO-7ª Região/2023)



```
Arquivo Editar Ver Pesquisar Terminal Ajuda
crefito@desk12:/bkp/crefito7$ ls -la
total 848
drwxrwxrwx 2 root    root    4896 jun  5 15:19 
drwxr-xr-x 3 root    root    4896 jun  5 15:13 ..
-rw-rw-r-- 1 crefito crefito 147188 mai 26 11:30 anuidades.pdf
-rw-rw-r-- 1 crefito crefito 211550 jun  1 16:18 crefito_semanal.zip
-rw-rw-r-- 1 crefito crefito 454821 mai 29 19:02 Planejamento.pdf
-rwxrwxr-x 1 crefito crefito 48695 mai 26 11:14 realiza_backup.sh
crefito@desk12:/bkp/crefito7$
```

Quanto à imagem acima e aos comandos no Ubuntu a ela relacionados, julgue o item.

Para a remoção de todas as permissões sobre o arquivo `anuidades.pdf`, tornando-o inacessível, o usuário `root` deverá executar o comando `chmod -x anuidades.pdf`.

Comentários:

O arquivo PDF mostrado pertence ao usuário "crefito", grupo "crefito". Se o `root` aplicar o comando "`chmod -x anuidades.pdf`", a permissão de execução (x) de UGO (usuário dono, grupo e outros) será removida. Mas isso não quer dizer que as outras permissões (rw) serão removidas!

Curiosidade: Na realidade, o arquivo nem apresenta a permissão "x", logo, nada será alterado!

Gabarito: Errado



14. (CS-UFG/UFG/2023) Sistemas de arquivos promovem o armazenamento estruturado de arquivos, ao agregar características como: identificador de arquivo, permissões de acesso ao arquivo, partições do arquivo no meio de armazenamento etc. Um comando Linux para atribuir permissões a arquivos é

- A) chmod.
- B) diff.
- C) du.
- D) df.

**Comentários:**

Já vimos bastante sobre o **chmod**, mas vamos ao significado: chmod = change file mode bits;  
diff = verifica as diferenças entre arquivos (linha a linha);  
du = "disk use": verifica o uso do disco;  
df = "disk free": verifica o disco livre, semelhante ao "du".

Gabarito: A

15. (FUNCERN/Câmara de Natal-RN/2023) Um técnico de informática da Prefeitura, por meio do sistema Linux, precisou dar permissão de leitura, edição e execução para determinado arquivo. Para isso, utilizou o comando

- A) chmod 755
- B) chmod 777
- C) chmod 750
- D) chmod 700

**Comentários:**

"precisou dar permissão de leitura, edição e execução para determinado arquivo" - não ficou claro para "quem". Então, pode-se entender que para todos (UGO). Nesse caso, temos:

U = rwx = 111 = 7;  
G = rwx = 111 = 7;  
O = rwx = 111 = 7.

Gabarito: B

16. (CETREDE/Câmara de Ipu-CE/2023) Assinale a alternativa que corresponda ao comando utilizado para mudar a proteção de um arquivo ou diretório.

- A) diff.
- B) file.



- C) grep.
- D) chmod.
- E) quota.

#### Comentários:

Já vimos bastante sobre o **chmod**, mas vamos ao significado: chmod = change file mode bits;  
diff = verifica as diferenças entre arquivos (linha a linha);  
file = verifica e mostra o tipo de arquivo (ex.: PDF);  
grep = procura por trechos de texto (strings) dentro de arquivos ou diretórios;  
quota = mostra o uso de disco por usuário.

Gabarito: D

17. (UFPR/IF-PR/2023) Para o controle de um arquivo no Sistema Linux Ubuntu foi criado um usuário "supervisor" que pertence ao grupo "gerencia". O usuário "supervisor" criou um arquivo "metas.txt" para incluir e atualizar os dados das vendas feitas pelos funcionários. No entanto, o arquivo precisa ter restrição de acesso; dessa forma, o administrador alterou a permissão do arquivo para que o usuário "supervisor" tenha total acesso e o grupo "gerencia" consiga somente ler. Sabendo-se disso, qual a opção correta para a saída do comando "ls -l" após a permissão?

- A) -rwx----- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- B) -rwxr----- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- C) -rx-rw---- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- D) -rwxrwx--- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- E) -rwx---r-- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt

#### Comentários:

Em resumo, a questão quer que o usuário "supervisor" (dono) tenha total acesso e o grupo "gerencia" (grupo que e o dono pertence) consiga somente ler. Então, temos:

U = rwx;

G = r--;

O = ---.

Ou seja: "rwx r-- ---".

Gabarito: B

18. (SELECON/IF-RJ/2023) No sistema operacional LINUX, o comando CHMOD altera a permissão de acesso a um arquivo ou diretório. Um administrador de sistemas LINUX, conectado



como usuário root, executou o comando "chmod 764 teste". As permissões concedidas no arquivo "teste" após a execução do comando acima foram as seguintes:

- A) outros (leitura, gravação); grupo (leitura, gravação e execução); dono (leitura)
- B) outros (leitura, gravação e execução); grupo (leitura e gravação); dono (leitura)
- C) dono (leitura e gravação); grupo (leitura); outros (leitura)
- D) dono (leitura); grupo (leitura, gravação e execução); outros (leitura e gravação)
- E) dono (leitura, gravação e execução); grupo (leitura e gravação); outros (leitura)

#### Comentários:

Vamos focar no valor octal 764:

U = 7 = 111 = rwx → usuário dono (leitura, gravação, execução);

G = 6 = 110 = rw- → grupo (leitura, gravação);

O = 4 = 100 = r-- → outros (leitura).

Gabarito: E





## QUESTÕES COMENTADAS - CEBRASPE

1. (CEBRASPE/FUNPRESP-JUD/2016) Os comandos `find` e `which` podem ser utilizados para a obtenção de informações acerca de programas instalados no Linux; o primeiro localiza o caminho (path) onde o comando está instalado, e o segundo, a versão do programa.

### Comentários:

`find`: procura por arquivos em uma hierarquia de diretórios.

`which` ("qual"): mostra o caminho completo de determinado comando.

O `find` procura qualquer tipo de arquivo/diretório, e o `which` mostra o caminho completo de um comando (não a sua versão).

Gabarito: Errado

2. (CEBRASPE/PG-DF/2021) Quanto à administração de sistemas operacionais, julgue o item seguinte.

No Linux, o comando `sed 's/windows/linux/' maria > pedro` faz a substituição da palavra `windows` pela palavra `linux` contida no arquivo `maria` e o resultado é enviado para o arquivo de nome `pedro`.

### Comentários:

`sed`: ferramenta que ajuda a analisar e transformar textos. Existem diversas opções, mas vamos ver apenas alguns exemplos de uso abaixo.

- `s/regex/substituicao/`: procura expressões REGEX e as substitui:

`$ sed 's/bonito/feio/' entrada.txt > saida.txt` → substitui toda palavra "bonito" para "feio" no arquivo `entrada.txt`, gerando o resultado no arquivo `saida.txt`.

Gabarito: Certo

3. (CEBRASPE/CODEVASF/2021) Com relação à administração do sistema operacional Linux, julgue o item seguinte.

O comando `lastlog` é capaz de mostrar o último login de usuários e, se o login tiver sido feito de um computador remoto, informará também o endereço IP de origem.

### Comentários:

`lastlog`: mostra o login mais recente de um usuário específico ou todos do sistema. Formata e imprime o conteúdo do último arquivo de log `/var/log/lastlog`.

Gabarito: Certo



4. (CEBRASPE/PGE-RJ/2022) Acerca da administração do sistema operacional Linux e da linguagem de script Shell, julgue o item subsequente.

No Linux, o comando `vmstat` apresenta um relatório de consumo de CPU. Nesse relatório, a coluna `id` mostra o número do processo que está consumindo o referido recurso.

Comentários:

`vmstat`: mostra estatísticas sobre processos, memória, swap, blocos de E/S, traps, discos e atividades do processador.



```
onworks@onworks-Standard-PC-i440FX-PIIX-1996: ~  
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ vmstat  
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----  
r b swpd free buff cache si so bi bo in cs us sy id wa st  
3 0 0 910812 69676 982576 0 0 1851 841 331 763 15 7 77 2 0  
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

Gabarito: Errado

5. (CEBRASPE/DPE-DF/2022) Julgue o item subsequente, a respeito da administração de sistemas operacionais Linux, considerando que o sistema esteja logado com o usuário `root`.

Se `022` é o `umask` padrão para todos os usuários, então o comando a seguir modificará o `umask` para `227` somente para o usuário `roberto`.

```
echo 'umask 227' >> /home/roberto/.bashrc
```

Comentários:

Essa questão exige conhecimentos diversos, mas nosso foco será no comando `echo`, que mostra um texto na tela. Ex.:

`$ echo oi` → mostra “oi” na saída padrão, geralmente a tela do computador.

Pode criar um arquivo de texto, se utilizado o redirecionamento de saída “>” ou “>>”, exemplos:

o `echo oi > arquivo.txt` → cria um arquivo com o conteúdo “oi”;

o `echo oi >> arquivo.txt` → adiciona no fim do arquivo.txt o conteúdo “oi” (se não existir, cria o arquivo).

Não foi mostrado nesta aula, mas o que for configurado em `/home/usuario/.bashrc` será aplicado apenas ao usuário “usuario”.

Gabarito: Certo

6. (CEBRASPE/SEPLAN-RR/2023) Julgue o próximo item a respeito dos conceitos de Linux.

Suponha-se o seguinte código bash.



```
tail nome_arquivo.txt -n2 | head -n1
```

A execução do código em tela resultará na apresentação da penúltima linha do arquivo nome\_arquivo.txt.

#### Comentários:

tail: a tradução literal é “cauda” (cauda de um animal, mesmo), então pense no rabo de um cachorro, é o que fica por último, certo? Esse é o propósito do comando, mostrar a última parte de um arquivo. Então, se um arquivo é pequeno, tanto faz usar o “cat” ou o “tail”, mas se for grande, depende o que o usuário quer ver! Obs.: por padrão, o comando tail mostra as 10 últimas linhas do arquivo. A opção “-n NUMERO” define uma quantidade das N últimas linhas a serem mostradas.

head: mostra as primeiras 10 linhas de um arquivo na saída padrão (o oposto do comando tail). A opção -n, ou --lines=NUM mostra as primeiras NUM linhas, ao invés das primeiras 10.

Vamos analisar a linha de comando por partes:

```
tail nome_arquivo.txt -n2 | head -n1
```

“tail nome\_arquivo.txt -n2” mostra as duas últimas linhas de “nome\_arquivo.txt”, a saída é enviada para o segundo comando “head -n1”, que mostra a primeira linha. Então, no fim das contas, é mostrada a penúltima linha do arquivo “nome\_arquivo.txt”.

Gabarito: Certo



## QUESTÕES COMENTADAS - FGV

1. (FGV/Câmara Municipal do Recife-PE/2014) No Linux, o comando

```
cat something | grep 'something'
```

tem o efeito de:

- A) mostrar as linhas do arquivo something que incluem a palavra something;
- B) adicionar a palavra something ao conteúdo do arquivo something;
- C) criar uma cópia do arquivo something denominada grpe;
- D) mover o arquivo something para um novo diretório denominado something;
- E) restringir o acesso ao arquivo something ao usuário something.

**Comentários:**

O `cat NOME_ARQUIVO` mostra o conteúdo desse arquivo como saída, mas como há um pipe (`|`) em seguida, essa saída vira a entrada para o comando seguinte. Então, o `grep` realiza a pesquisa do termo 'something' sobre o conteúdo do arquivo e mostra todas as linhas que possuem tal termo.

Gabarito: A

2. (FGV/TJ-BA/2015) O seguinte comando, invocado em uma shell do Linux:

```
find . -type f
```

escreverá na saída padrão:

- A) o número de arquivos regulares descendentes do diretório corrente;
- B) os caminhos (pathnames) relativos para os arquivos regulares descendentes do diretório corrente;
- C) os caminhos (pathnames) absolutos para os arquivos regulares descendentes do diretório corrente;
- D) os caminhos (pathnames) relativos para os diretórios descendentes do diretório corrente;
- E) os caminhos (pathnames) absolutos para os arquivos regulares e diretórios descendentes do diretório corrente.

**Comentários:**

O ponto (`.`) indica diretório corrente e o `"-type f"` indica apenas os arquivos (files), sem os diretórios:



```
[root@localhost teste]# ls
numeros.txt  oi.txt          opa            tchau.txt
[root@localhost teste]# find . -type f
./oi.txt
./tchau.txt
./numeros.txt
[root@localhost teste]# █
```

Gabarito: B

3. (FGV/Câmara de Aracaju-SE/2021) Observe a linha de comando para um sistema Linux

```
tar xzf myfile.tar.gz
```

A ação realizada pela sua execução é:

- A) criar o arquivo de backup myfile.tar.gz;
- B) descompactar o arquivo myfile.tar.gz no diretório corrente;
- C) habilitar a execução do arquivo myfile.tar.gz;
- D) executar o arquivo myfile.tar.gz com permissão temporária;
- E) apagar o conteúdo do arquivo myfile.tar.gz.

Comentários:

tar: utilizado para compactação de arquivos. Gera um arquivo ".tar". Para maior compactação é utilizado o gzip, gerando um arquivo .tar.gz. Vamos a alguns parâmetros:

- c: cria um novo arquivo .tar;
- v: mostra uma descrição do progresso de compactação;
- f: nome do arquivo;
- z: compressão gzip;
- x: descompacta um arquivo;
- C: realiza a extração para um diretório diferente do atual.

Gabarito: B

4. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando que informa o path absoluto do diretório corrente.

- A) cd
- B) find
- C) mv
- D) mkdir
- E) pwd

Comentários:

- A) cd: (change directory) - muda o diretório.



- B) find: procura arquivo(s).
- C) mv: move arquivos (ou renomeia).
- D) mkdir: (make directory) - cria diretório.
- E) pwd: (print working directory) - mostra o diretório corrente. Ou como aparece no enunciado: mostra o "path absoluto do diretório corrente".

cat: concatena arquivos e mostra na saída padrão. Sintaxe: cat ARQ1 ARQ2. Mas se for utilizado o direcionador "simples" (>), coloca o conteúdo do primeiro arquivo sobre o conteúdo do segundo e não mostra na tela. Se for utilizado o direcionador "duplo" (>>), concatena o primeiro ao segundo e não mostra na tela. Quando usado com apenas um arquivo, mostra seu conteúdo. Abaixo um exemplo com o direcionador duplo e depois somente com um arquivo.

chmod: atribui permissões aos arquivos/diretórios.

finger: pesquisa e exibe informações sobre os usuários do sistema.

fsck: significa "file system check". Usado para verificar e, opcionalmente, reparar um ou mais sistemas de arquivos.

grep: mostra as linhas de um arquivo que "batem" (match) com um determinado padrão.

Gabarito: E

5. (FGV/FUNSAÚDE-CE/2021) No Linux, a saída (output) de um comando pode ser direcionada para um arquivo ou processo. Nesse contexto, assinale o redirecionamento que é determinado pelo símbolo >>.

- A) Adicionar o conteúdo direcionado a um arquivo existente, ou a um novo arquivo se não existir o arquivo referido.
- B) Descartar o conteúdo direcionado; não gravar e não mostrar.
- C) Direcionar o conteúdo para outro comando ou processo.
- D) Gravar um novo arquivo com o conteúdo direcionado; não pode existir eventual arquivo homônimo.
- E) Gravar um novo arquivo com o conteúdo direcionado, sobrescrevendo eventual arquivo homônimo.

#### Comentários:

O símbolo > direciona para um arquivo, criando-o. Se já existir, sobrescreve-o.

O símbolo >> direciona para o fim de um arquivo, ou seja, adiciona ao conteúdo já existente. Se não existir, o arquivo é criado.

Gabarito: A

6. (FGV/TCE-AM/2021) No Linux, o comando usado para verificar e, opcionalmente, reparar um sistema de arquivos é denominado:



- A) cat;
- B) chmod;
- C) finger;
- D) fsck;
- E) grep.

#### Comentários:

cat: concatena arquivos e mostra na saída padrão. Sintaxe: cat ARQ1 ARQ2. Mas se for utilizado o direcionador "simples" (>), coloca o conteúdo do primeiro arquivo sobre o conteúdo do segundo e não mostra na tela. Se for utilizado o direcionador "duplo" (>>), concatena o primeiro ao segundo e não mostra na tela. Quando usado com apenas um arquivo, mostra seu conteúdo. Abaixo um exemplo com o direcionador duplo e depois somente com um arquivo.

chmod: atribui permissões aos arquivos/diretórios.

finger: pesquisa e exibe informações sobre os usuários do sistema.

**fsck**: significa "file system check". Usado para verificar e, opcionalmente, reparar um ou mais sistemas de arquivos.

grep: mostra as linhas de um arquivo que "batem" (match) com um determinado padrão.

Gabarito: D

7. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando que busca linhas num arquivo.

- A) cd
- B) lp
- C) ls
- D) tar
- E) grep

#### Comentários:

cd: (change directory) - muda o diretório corrente.

lp: envia trabalhos de impressão para uma impressora.

ls: lista o conteúdo de um diretório.

tar: compacta/descompacta arquivos.

**grep**: mostra as linhas de um arquivo que "batem" (match) com um determinado padrão.

Gabarito: E



8. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando utilizado para listar o conteúdo de um diretório.

- A) cf
- B) lp
- C) ls
- D) tar
- E) grep

**Comentários:**

cf: não existe.

lp: envia trabalhos de impressão para uma impressora.

**ls:** lista o conteúdo de um diretório.

tar: compacta/descompacta arquivos.

grep: mostra as linhas de um arquivo que "batem" (match) com um determinado padrão.

Gabarito: C

9. (FGV/SEAD-AP/2022) O formato de arquivo chamado Raw, padrão para a saída gerada pelo comando dd do Linux/UNIX, é uma cópia bit a bit dos dados brutos da unidade que está sob investigação, que pode ser usado na criação de uma imagem da unidade completa ou de um único volume (partição).

Assinale a opção que apresenta a principal desvantagem do formato Raw.

- A) Não ser suportado na maioria dos programas de computação forense.
- B) Requerer o mesmo espaço de armazenamento da unidade de origem.
- C) Ignorar erros de leitura menores a partir da unidade de origem.
- D) Realizar uma rápida transferência de dados.
- E) Realizar a compactação dos dados.

**Comentários:**

**dd:** ferramenta poderosa usada para copiar e converter dados. Frequentemente utilizada para operações de cópia de blocos brutos de dados (raw copy), como criar imagens de disco, copiar partições ou zerar dispositivos. A cópia realizada é bit-a-bit, portanto a cópia gerada possuirá o mesmo tamanho da origem.

\$ dd if=/dev/sdb of=/mnt/destino → faz uma cópia bruta (raw) de um dispositivo (mídia) para o arquivo "destino", em /mnt.

Gabarito: B





10. (FGV/SEAD-AP/2022) João está investigando uma possível violação de segurança e precisa saber quais usuários fizeram login no sistema recentemente.

O utilitário de linha de comando do sistema operacional Linux que mostra essas informações é

- A) alias
- B) last
- C) nice
- D) tty
- E) wc

#### Comentários:

alias: cria atalhos ou apelidos para outros comandos. Pode ser útil para simplificar comandos longos, adicionar opções padrão a comandos frequentemente utilizados ou até mesmo criar atalhos mais fáceis de lembrar. Exemplo de um alias simples: `$ alias l='ls -l'`

**last**: exibe informações sobre os últimos logins no sistema. Ele mostra uma lista dos usuários que fizeram login, quando o fizeram e de onde (endereço IP ou terminal).

nice: define a prioridade (niceness) de execução de processos.

tty: imprime o nome do terminal associado ao processo que o está chamando. Ele exibe o caminho do dispositivo do terminal (ex.: `/dev/tty1` ou `pts/0`).

wc: significa "word count", mas além de "contar palavras", conta também as linhas e os caracteres em um arquivo.

Gabarito: B

11. (FGV/TCE-TO/2022) No contexto do Linux, a simbologia de redirecionamento de saída de dados que adiciona dados no final de um arquivo é:

- A) |
- B) ||
- C) >
- D) ->
- E) >>

#### Comentários:

O símbolo `>` direciona para um arquivo, criando-o. Se já existir, sobrescreve-o.

O símbolo `>>` direciona para o fim de um arquivo, ou seja, adiciona ao conteúdo já existente. Se não existir, o arquivo é criado.

Gabarito: E



12. (FGV/MPE-GO/2022) Assinale a função do comando man do Linux.

- A) Acompanhar passo a passo a execução de um comando.
- B) Documentar os comandos do sistema.
- C) Executar rotinas do Kernel.
- D) Executar tarefas de administração de contas/usuários.
- E) Gerenciar os processos ativos no sistema.

Comentários:

man: fornece um manual completo para comandos ou utilitários que podem ser executados no terminal. É só digitar man NOME\_COMANDO que aparece o manual na tela.

Gabarito: B

13. (FGV/DPE-RS/2023) O Ubuntu é um sistema operacional considerado fácil de usar, instalar e configurar, além de possuir uma vasta documentação. A respeito do Ubuntu, é correto afirmar que:

- A) com o ls é possível interagir com o sistema por meio de comandos digitados no teclado;
- B) o comando shell lista os arquivos e diretórios;
- C) o comando pwd fornece a identificação efetiva do usuário e o comando date retorna a data atual;
- D) o comando whoami retorna exatamente o diretório corrente;
- E) o comando cd / serve para ir ao diretório raiz, informando ao shell para navegar até o diretório /.

Comentários:

- A) ERRADA - O correto seria: com o shell é possível interagir com o sistema por meio de comandos digitados no teclado;
- B) ERRADA - O correto seria: o comando ls lista os arquivos e diretórios;
- C) ERRADA - O correto seria: o comando whoami fornece a identificação efetiva do usuário e o comando date retorna a data atual;
- D) ERRADA - O correto seria: o comando pwd retorna exatamente o diretório corrente;
- E) CORRETA - o comando cd serve para mudar o diretório corrente. cd / muda para o diretório raiz ("/").

Gabarito: E



## QUESTÕES COMENTADAS - FCC

1. (FCC/TRF4/2014) Marcos, usuário de um computador com sistema operacional Linux Red Hat listou o conteúdo do seu diretório home e observou a presença do arquivo manual.txt com 31.251 bytes de tamanho, o que representa cerca de 20 páginas de texto se visualizado em um terminal Linux padrão. Para que Marcos possa visualizar diretamente o final do arquivo manual.txt, sem a necessidade de iniciar a visualização a partir do começo do arquivo, ele deve executar o comando:

- A) `cat manual.txt | more`
- B) `more manual.txt`
- C) `list manual.txt | end`
- D) `tail manual.txt`
- E) `cat manual.txt | end`

### Comentários:

tail: a tradução literal é "rabo", então pense no rabo de um cachorro, é o que fica por último...é o propósito desse comando, mostrar a última parte de um arquivo. Então, se um arquivo é pequeno, tanto faz usar o "cat" ou o "tail", mas se for grande, depende o que o usuário quer ver!

Gabarito: D

2. (FCC/Prefeitura de São Luís-MA/2018) Um Auditor deseja configurar e habilitar em um servidor Linux a interface de rede eth0 com o IP 172.16.25.125 e a máscara 255.255.255.224. Para isso, ele terá que utilizar o comando

- A) `ipconfig eth0 172.16.25.125 mask 255.255.255.224 up`
- B) `netconfig eth0 ip 172.16.25.125 netmask 255.255.255.224 on`
- C) `netstat eth0 ip 172.16.25.125 netmask 255.255.255.224 on`
- D) `ifconfig eth0 172.16.25.125 netmask 255.255.255.224 up`
- E) `netstart eth0 172.16.25.125 mask 255.255.255.224`

### Comentários:

ifconfig: configura as interfaces de rede e seus parâmetros. Parâmetros:

- interface: nome da interface de rede. Ex.: eth0 para o primeiro dispositivo de rede ethernet;
- up: habilita a interface de rede para uso;
- down: desabilita a interface de rede;
- address: configura um endereço IP para a interface;
- netmask máscara: configura uma determinada máscara de sub-rede para a interface.



Para configurar o IP, máscara e broadcast de uma placa de rede:

```
# ifconfig eth0 10.0.0.1 netmask 255.255.0.0 broadcast 10.0.0.255
```

Para levantar uma interface de rede:

```
# ifconfig eth0 up
```

Para visualizar uma determinada interface de rede:

```
# ifconfig eth0
```

Gabarito: D

3. (FCC/Câmara Legislativa-DF/2018) Considere, hipoteticamente, que a Câmara Legislativa do DF receberá um funcionário temporário por um período de 3 meses. Para adicionar esse funcionário como usuário com período de tempo determinado no servidor com sistema operacional Linux, o Administrador deve utilizar o comando `useradd` com a opção

- A) -r.
- B) -d.
- C) -o.
- D) -s.
- E) -e.

Comentários:

`useradd`, opções:

- `-e, --expiredate DATA_EXPIRAÇÃO`: define a data de expiração da conta;
- `-d diretorio_home`: fornece o caminho completo do diretório home do usuário;
- `-s shell`: fornece o caminho completo do shell utilizado pela conta (ex.: `/bin/bash`);

Gabarito: E

4. (FCC/TRT2/2018) No shell do Linux, para ativar a interface de rede `eth0` utiliza-se a instrução

- A) `netstart eth0`
- B) `netstart --interface=eth0`
- C) `nslookup --interface=eth0`
- D) `ifconfig eth0 up`
- E) `ipconfig eth0 up`

Comentários:



ifconfig: configura as interfaces de rede e seus parâmetros. Parâmetros:

- interface: nome da interface de rede. Ex.: eth0 para o primeiro dispositivo de rede ethernet;
- up: habilita a interface de rede para uso;
- down: desabilita a interface de rede;
- address: configura um endereço IP para a interface;
- netmask máscara: configura uma determinada máscara de sub-rede para a interface.

Para configurar o IP, máscara e broadcast de uma placa de rede:

```
# ifconfig eth0 10.0.0.1 netmask 255.255.0.0 broadcast 10.0.0.255
```

Para **levantar uma interface** de rede:

```
# ifconfig eth0 up
```

Para visualizar uma determinada interface de rede:

```
# ifconfig eth0
```

Gabarito: D

5. (FCC/METRÔ-SP/2019) Um Analista que usa o Linux Red Hat, em condições ideais, editou um arquivo e colocou os comandos abaixo, para indicar os servidores que atuarão na resolução de nomes para o servidor que está configurando.

```
search localdomain nameserver 8.8.8.8 nameserver 8.8.4.4
```

Para editar o arquivo em questão, utilizou a instrução

- A) vi /etc/dns.conf
- B) nano /conf/dns.conf
- C) vi /etc/resolv.conf
- D) nano /bin/resolv.conf
- E) vi /lib/dnsconf.conf

**Comentários:**

Para responder essa questão é necessário ter conhecimento do arquivo de resolução de nomes (resolv.conf), que fica no diretório /etc. Para editar esse arquivo pode ser utilizado um editor de texto (vi, nano, entre outros).

Gabarito: C

6. (FCC/TRF3/2019) No Linux Red Hat, para mostrar o espaço em disco, incluindo espaço usado e disponível, utiliza-se o comando



- A) df -h
- B) dir- s
- C) tune2fs - d
- D) free -h
- E) top - a

**Comentários:**

df (disk free): além de mostrar quanto tem livre no disco, mostra também o espaço usado. Opção -h ou --human-readable: mostra os tamanhos em potência de 1024, ex.: 1020M.

Gabarito: A

7. (FCC/TJ-MA/2019) Para descompactar, no Linux, um arquivo chamado data.tar.gz, mostrando o nome dos arquivos durante a descompactação, deve-se usar o comando:

- A) tar -xsf data.tar.gz
- B) unzip -xsf data.tar.gz
- C) tar -xjf data.tar.gz
- D) gzip -xmjf data.tar.gz
- E) tar -xvzf data.tar.gz

**Comentários:**

tar: utilizado para compactação de arquivos. Gera um arquivo ".tar". Para maior compactação é utilizado o gzip, gerando um arquivo .tar.gz. Vamos a alguns parâmetros:

- c: cria um novo arquivo .tar;
- v: mostra uma descrição do progresso de compactação;
- f: nome do arquivo;
- z: compressão gzip;
- x: descompacta um arquivo;
- C: realiza a extração para um diretório diferente do atual.

\$ tar -cvzf arquivo.tgz /home/evandro → Compacta usando o gzip, mostrando o processo.

\$ tar -xvf compactado.tar -C /home/evandro/ → descompactação para um diretório diferente do atual, mostrando o processo.

\$ tar -xvf compactado.tar.gz → descompactação, mostrando o processo.

Gabarito: E

8. (FCC/TJ-MA/2019) Uma Técnica Judiciária, ao utilizar um computador com o sistema operacional Linux, identificou um comando que faz uma busca no Linux por termos ligados a



determinado assunto que um usuário deseja pesquisar, e exibe os comandos e utilitários relacionados àquele assunto, com sua descrição. O comando descrito é o:

- A) cal
- B) pwd
- C) apropos
- D) manual
- E) whatisthis

**Comentários:**

apropos: procura por uma determinada expressão nas páginas de documentação.

Gabarito: C

9. (FCC/TRT19/2022) Utilizando o app Terminal do Linux, um Analista conectou-se com sucesso ao servidor Web de sua organização por intermédio do SSH. Para realizar a manutenção de um servidor web, dentre os principais comandos desse protocolo (SSH), o

- A) touch permite a criação de um arquivo com a extensão escolhida.
- B) ls -hal permite a visualização do conteúdo de determinado arquivo.
- C) rm serve para abrir e editar um arquivo dentro do Terminal.
- D) nano mostra o caminho do diretório atual.
- E) cat serve para redirecionar ou apagar um arquivo ou diretório.

**Comentários:**

O comando touch pode ser usado para criar um arquivo em branco. Pode ser com ou sem extensão. Exemplos (um com e outro sem extensão):

```
$ touch script.sh  
$touch teste
```

Gabarito: A

10. (FCC/Copergás-PE/2023) Considere que um usuário de um computador com o sistema operacional Linux funcionando em condições ideais usou, de forma adequada, os comandos df, cat e ls. Esses comandos são utilizados, correta e respectivamente, para

- A) mostrar a quantidade de memória usada – abrir um arquivo – mostrar as últimas 10 linhas de um arquivo texto.
- B) deletar um arquivo – copiar um arquivo – verificar o status da conexão do servidor.
- C) mostrar a quantidade de espaço usada por um arquivo – remover um arquivo – listar todos os arquivos do diretório.



D) mostrar a quantidade de espaço usada no disco rígido – abrir um arquivo – listar todos os arquivos do diretório.

E) deletar um diretório e todos os seus arquivos – copiar um arquivo – mostrar as últimas 10 linhas de um arquivo texto.

#### Comentários:

df (disk free): mostra a quantidade de disco livre e também a quantidade ocupada;

cat: serve para concatenar arquivos e mostrar o resultado, mas se houver apenas um, mostra o conteúdo deste único arquivo;

ls: lista o conteúdo de um diretório.

Gabarito: D

11. (FCC/TRT18/2023) A partir do terminal do RedHat Linux, funcionando em condições ideais, um técnico precisa procurar pela palavra "técnico" no arquivo trt.txt e direcionar a saída do comando para o arquivo trtsaída.txt

A) ls -s técnico trt.txt || trtsaída.txt

B) grep técnico trt.txt > trtsaída.txt

C) search trt.txt (técnico) && trtsaída.txt

D) cat trt.txt técnico >> trtsaída.txt

E) touch técnico trt.txt >> trtsaída.txt

#### Comentários:

O comando grep mostra a linha com o conteúdo que você deseja. Então, grep técnico trt.txt mostraria na tela a linha com o conteúdo. Ex.: "Um técnico de futebol ganha bem." Se for usado > e em seguida o nome de um arquivo, aquilo que seria mostrado na tela será gravado no arquivo. Se o arquivo não existir, ele é criado. Se existir com algum conteúdo, ele será sobrescrito.

Gabarito: B





## QUESTÕES COMENTADAS - VUNESP

1. (VUNESP/DESENVOLVESP/2014) O superusuário do sistema operacional Linux deseja verificar a informação da última linha do arquivo `/var/log/dmesg`. Para efetivar essa ação por meio do uso de apenas um comando, ou seja, sem qualquer outra ação, deve-se executar, na linha de comando:

- A) `$ cat /var/log/dmesg`
- B) `$ cat /var/log/dmesg >`
- C) `$ more /var/log/dmesg >>`
- D) `$ tail /var/log/dmesg`
- E) `$ top /var/log/dmesg`

### Comentários:

`tail`: a tradução literal é "rabo", então pense no rabo de um cachorro, é o que fica por último...é o propósito desse comando, mostrar a última parte de um arquivo. Então, se um arquivo é pequeno, tanto faz usar o "cat" ou o "tail", mas se for grande, depende o que o usuário quer ver!

Como a questão fala em ver a última linha, o `tail` pode ser utilizado, mostrando não só a última, mas as últimas linhas do arquivo.

Gabarito: D

2. (VUNESP/SP-URBANISMO/2014) Um usuário padrão do sistema operacional Linux, "logado" no seu diretório `home`, deseja saber o diretório no qual está instalado o comando `mount`. Para obter tal informação, esse usuário poderá utilizar o seguinte comando:

- A) `cat mount`
- B) `find mount`
- C) `grep mount`
- D) `where mount`
- E) `which mount`

### Comentários:

`which` ("qual"): mostra o caminho completo de determinado comando. Abaixo vemos, por exemplo, o caminho completo dos comandos `mount`, `ls` e `cat`.

Gabarito: E

3. (VUNESP/Câmara de Mauá-SP/2019) O seguinte comando foi executado em um computador com sistema operacional Linux:



ls -l meudir

O resultado obtido foi:

```
-rw-r--r-- g b 18 Nov 29 17:50 meudir
```

De acordo com esse resultado, é correto afirmar que

- A) o grupo dono (owning group) de meudir é g.
- B) o tamanho de meudir é 29 bytes.
- C) meudir é um diretório.
- D) meudir pode ser executado.
- E) meudir pode ser escrito pelo usuário g.

Comentários:

```
-rw-r--r-- g b 18 Nov 29 17:50 meudir
```

- A) ERRADA - o grupo dono (owning group) de meudir é b.
- B) ERRADA - o tamanho de meudir é 18 bytes.
- C) ERRADA - meudir é um arquivo (se fosse diretório teria um "d" no primeiro caractere que aparece).
- D) ERRADA - meudir pode não ser executado (não tem permissão "x" para ninguém!).
- E) CORRETA - meudir pode ser escrito pelo usuário g, pois tem permissão "w" para o usuário.

Gabarito: E

4. (VUNESP/Câm. de M. Alto-SP/2019) Deseja-se atribuir um endereço IPv4 alias para a interface eth0 em um computador com sistema operacional Linux utilizando o comando ifconfig. A correta sintaxe para efetivar essa operação é:

- A) #ifconfig eth0:192.168.0.100
- B) #ifconfig eth0:0 192.168.0.100
- C) #ifconfig alias eth0 192.168.0.100
- D) #ifconfig eth0 alias 192.168.0.100
- E) #ifconfig eth0/alias 192.168.0.100

Comentários:

Para configurar um alias para uma interface de rede (alias numérico):

```
# ifconfig eth0:1 192.168.1.2 netmask 255.255.255.0 up
```



Gabarito: B

5. (VUNESP/Câm. de M. Alto-SP/2019) Durante o uso de um terminal Bash do sistema operacional Linux, o usuário digitou !! (dupla exclamação) seguido de Enter. O resultado dessa ação será

- A) a apresentação do PID do terminal.
- B) a limpeza (clear) do terminal.
- C) a reexecução do último comando.
- D) o bloqueio do terminal.
- E) o fechamento do terminal.

Comentários:

Não tem muito o que falar, né? Dupla exclamação (!! ) repete o último comando.

Gabarito: C

6. (VUNESP/Câm. de M. Alto-SP/2019) Utilizando o comando shutdown, o Administrador de um servidor com sistema operacional Linux deseja avisar os usuários e realizar a reinicialização do servidor em 10 minutos. Para isso, o Administrador deve utilizar o comando shutdown com as opções:

- A) -h 10.
- B) -k 10.
- C) -p 10.
- D) -r 10.
- E) -t 10.

Comentários:

shutdown: desliga ou reinicia o sistema com horários determinados. Parâmetros:

- -r: reinicia (reboot) o sistema depois de terminada a sequência de desligamento;
- -h: paralisa (halt) o sistema depois de terminada a sequência de desligamento. Não reinicia a máquina;
- -k: manda a mensagem de desligamento, mas não inicia a sequência de desligamento;
- -f: faz a carga de sistema rápida sem a checagem de discos;
- -F: força uma checagem dos discos quando for reiniciar o sistema.

Gabarito: D

7. (VUNESP/UFABC/2019) O administrador de um servidor com sistema operacional Linux está testando um programa e deseja armazenar, em um arquivo, as informações da saída de erro



padrão (stderr) do programa. Para fazer esse redirecionamento, ou seja, stderr para um arquivo, o administrador deve utilizar os seguintes caracteres de redirecionamento:

- A) 0>
- B) 1>
- C) 2>
- D) >>
- E) &>

#### Comentários:

"2>" utilizado para redirecionar a saída de erro padrão para um arquivo ou outro local específico.

Sintaxe:

\$ comando\_que\_pode gerar\_erro 2> erro.txt → Se o comando gerar um erro, não aparecerá na tela, será gravado no arquivo erro.txt.

Gabarito: C

8. (VUNESP/Câmara de Tatuí-SP/2019) Considere a execução da sequência de comandos a seguir, em um terminal shell do Linux:

```
# cd /root  
  
# mkdir -p /root/foo/bar  
  
# pwd
```

O resultado impresso na tela após a execução do último comando será:

- A) /root
- B) /root/foo/bar
- C) bar
- D) /root/foo
- E) /foo/bar

#### Comentários:

O comando cd troca de diretório, "indo" para /root.

O comando mkdir cria o diretório /root/foo/bar. O argumento "-p" faz com que, se houver erro no comando, que tente o diretório pai ("parent"). Vamos considerar que não houve erro.

O comando pwd mostra o diretório corrente. Aí tem uma leve pegadinha, pois foi criado o diretório /root/foo/bar, mas o diretório corrente ainda é o /root, pois o comando cd fez o "deslocamento" até esse diretório e não saiu mais! Logo, pwd deve retornar /root.



Gabarito: A

9. (VUNESP/CODEN-SP/2021) No sistema operacional Linux, o comando que cria um arquivo vazio, ou atualiza a data/hora (timestamp) de um arquivo já existente, é:

- A) cat
- B) less
- C) tail
- D) touch
- E) head

Comentários:

touch: cria um arquivo vazio, e também atualiza as informações de data e hora de um arquivo existente, ex.: touch nome\_arquivo.

Gabarito: D

10. (VUNESP/PRUDENCO/2022) Em uma instalação do sistema operacional Linux, executou-se o comando `ls -l` e observou-se a seguinte designação, composta por 10 caracteres, à esquerda de um item listado:

`lrwxrwxrwx`

Com base nessa informação, pode-se afirmar que se trata de um

- A) diretório.
- B) link simbólico.
- C) shell script executável.
- D) arquivo executável correspondente a um comando do Linux.
- E) arquivo pertencente ao usuário root, uma vez que todas as permissões estão habilitadas.

Comentários:

O "l" à esquerda indica ser um link simbólico. Se fosse diretório, seria "d". Se fosse arquivo, seria "-".

Por ser um link com permissão de execução (x) não quer dizer que seja um shell script ou um comando Linux! Também não quer dizer que pertença ao root, por ser tudo "x"!

Gabarito: B

11. (VUNESP/Câm. de Olímpia-SP/2022) Em um computador com o sistema operacional Linux, foi encontrado um arquivo chamado `arq` em um certo diretório. Nesse diretório, executou-se, com sucesso, o seguinte comando:



ln arq arq2

É correto afirmar que arq2 é

- A) um hard link para arq.
- B) um soft link para arq.
- C) uma cópia de arq.
- D) o subdiretório para onde arq foi movido.
- E) o grupo que passou a ser dono de arq.

Comentários:

ln: cria ligações (*links*) entre arquivos. Pode ser:

- -d: cria um **hard link (é o padrão)**. Ligação direta (hard link): define mais de um nome para um arquivo. O arquivo será removido do disco quando o último nome for removido;
- -L ou --logical: cria uma ligação para um link simbólico. Ligação simbólica (dymlink): define um caminho para um arquivo. Pode apontar para arquivos em diferentes sistemas de arquivo. Não necessita apontar para arquivos existentes;
- -s: cria uma ligação simbólica (soft link).

Como o ln foi usado sem parâmetro algum, o padrão é a criação de um hard link.

Gabarito: A

12. (VUNESP/Câm. de Olímpia-SP/2022) No sistema operacional Linux, os comandos que têm a finalidade de

- (i) alterar a senha de um usuário,
- (ii) alterar o usuário dono de um arquivo e
- (iii) copiar arquivos, respectivamente, são:

- A) pwd, chown e cp.
- B) pwd, chown e copy.
- C) pwd, chattr e cp.
- D) passwd, chmod e copy.
- E) passwd, chown e cp.

Comentários:

- (i) alterar a senha de um usuário: **passwd** (abreviação de password);



- (ii) alterar o usuário dono de um arquivo: **chown** (abreviação de change owner);
- (iii) copiar arquivos: **cp** (abreviação de copy).

Gabarito: E

13. (VUNESP/UNICAMP/2022) Em um terminal do sistema operacional Linux, a execução do comando

```
echo "teste" 2> arquivo.txt
```

exibe a seguinte saída no próprio terminal:

- A) Nenhuma (saída vazia).
- B) teste
- C) arquivo.txt
- D) 2> arquivo.txt
- E) "teste" 2> arquivo.txt

Comentários:

Como vimos na aula:

`echo oi 2> erro.txt` □ "2>" utilizado para redirecionar a saída de erro padrão para um arquivo ou outro local específico. Dificilmente um comando `echo` gera erro, mas se gerar, o erro será escrito no arquivo `erro.txt` (exemplo). Neste caso, "oi" será mostrado na tela e o eventual erro será gravado em `erro.txt`.

Gabarito: B

14. (VUNESP/UNICAMP/2022) Algumas operações de recuperação de instalações do sistema operacional Linux podem requerer a utilização do comando `chroot`. Este comando tem por finalidade

- A) executar comandos que requerem privilégio de root, sem alterar o usuário da sessão já aberta.
- B) alterar o usuário de uma sessão já aberta, permitindo executar comandos com os privilégios do outro usuário.
- C) alterar o usuário dono de um arquivo.
- D) alterar o diretório raiz, criando um ambiente modificado no qual os programas não podem acessar arquivos fora desse diretório.
- E) alterar o nome do usuário root no sistema, permitindo que o root de fato utilize outro login.

Comentários:



chroot: utilizado para alterar o diretório raiz do processo corrente e dos seus filhos para o diretório especificado. "chroot" significa "change root" (mudar raiz) e geralmente é usado para criar um ambiente isolado, conhecido como "chroot jail", onde um processo acredita que o diretório especificado é o seu diretório raiz. A sintaxe básica é:

```
$ sudo chroot novo_diretorio comando
```

- novo\_diretorio: O diretório para o qual se deseja alterar a raiz;
- comando: O comando a ser executado no novo ambiente.

Por exemplo, para iniciar um shell dentro de um diretório específico:

```
$ sudo chroot /caminho/para/chroot /bin/bash
```

O comando acima inicia um shell dentro do diretório especificado como o novo diretório raiz.

Gabarito: D

15. (VUNESP/UNICAMP/2022) Após a instalação do sistema operacional Linux em um computador, é desejável verificar as portas TCP e UDP em estado listening por padrão, podendo-se, então, desinstalar ou desativar os aplicativos correspondentes àquelas portas que não se pretende utilizar. Dentre as a seguir, o comando do Linux que permite fazer essa verificação (utilizando os devidos parâmetros) é:

- A) ifconfig
- B) df
- C) ps
- D) nslookup
- E) netstat

Comentários:

netstat: exibe conexões de rede, tabelas de roteamento, interfaces de rede e estatísticas de protocolo de rede.

Gabarito: E

16. (VUNESP/EPC/2023) Considere um arquivo com nome prova.txt que contém o seguinte conteúdo:

```
Boa  
  
prova  
  
!
```

Após um usuário de sistema operacional Linux digitar o comando no terminal





```
grep -v a prova.txt
```

será apresentado

- A) Boa  
prova  
!
- B) Boa  
prova
- C) a  
a
- D) 2
- E) !

#### Comentários:

grep: mostra as linhas de um arquivo que “batem” (*match*) com um determinado padrão. Um exemplo comum é utilizar o cat para “visualizar” o arquivo e a saída dele passar como entrada para o grep, através do pipe (|). Uma opção que pode causar confusão é a “-v”, que **inverte a correspondência**, ou seja, ela seleciona todas as linhas que não contêm o padrão especificado. Cuidado!!!

Então, com o comando

```
grep -v a prova.txt
```

procuramos as linhas que **não tenham a letra “a”**. Só uma linha não tem a letra “a”, a última, que possui apenas um sinal de exclamação.

Gabarito: E

17. (VUNESP/EPC/2023) É possível obter o valor do horário atual em um sistema operacional Linux por meio de um comando escrito no terminal. Esse comando é o

- A) time
- B) date
- C) hour
- D) now
- E) clock.

#### Comentários:

Uma pegadinha interessante. Quem não conhece o comando, poderia ficar tentado a marcar a opção “time”, mas vamos ver a seguir.



date: utilizado para exibir a data e hora atuais do sistema. Além disso, ele também pode ser usado para configurar a data e hora do sistema, mas geralmente isso requer permissões de root.

Gabarito: B



## QUESTÕES COMENTADAS - MULTIBANCAS

1. (IF-RR/IF-RR/2015) O comando find permite buscar arquivos filtrando por nome, tamanho, data, dono, grupo, permissões, tipo e outros. A busca é recursiva, por padrão, mas pode ser restringida a uma determinada profundidade. O comando GNU find faz parte das findutils GNU e está instalado em cada sistema Ubuntu. Um usuário possui uma distribuição Ubuntu 14.04 LTS e precisa encontrar um arquivo de áudio com tamanho superior a 5MB, dentre as opções abaixo, qual alternativa melhor representa o comando para a busca pelo arquivo.

- A) find \$HOME -iname '\*.mp4' -size -5M
- B) find \$HOME -iname '\*.mp4' -size +5M
- C) find \$HOME -iname '\*.jpeg' -size +5M
- D) find \$HOME -iname '\*.ogg' +size -5M
- E) find \$HOME -iname '\*.ogg' -size +5M

### Comentários:

\$HOME indica o caminho do diretório home do usuário, -iname indica o nome com case insensitive. A questão fala em tamanho maior que 5MB, então o parâmetro correto é "-size +5M", e das extensões mostradas, o referente a áudio é o ".ogg". Poderia haver confusão com o ".mp4", mas lembre-se que esse formato é de áudio e vídeo, enquanto o ".ogg" é apenas de áudio!

Gabarito: E

2. (FCM/IF Sudeste-MG/2016) Alguns comandos em Linux podem combinar diferentes parâmetros, como o comando a seguir:

```
$ find /home/usuario/ -name '*.txt' -mtime 2;
```

Esta linha de comando procura

- A) todos os arquivos com extensão ".txt", dentro da pasta "/home/usuario".
- B) e remove todos os arquivos com extensão ".txt" dentro da pasta "/home/usuario".
- C) todos os arquivos com extensão ".txt", modificados a mais de 2 dias, dentro da pasta "/home/usuario" e subpastas.
- D) todos os arquivos modificados a mais de 2 dias, dentro da pasta "/home/usuario", e subpastas, e os move para a pasta "name".
- E) e copia para o diretório corrente todos os arquivos com nome "\*.txt", criados a menos de 2 dias, dentro da pasta "/home/usuario".

### Comentários:



Procura em `"/home/usuario/"` os arquivos com qualquer nome, desde que terminem com `".txt"`, com a data de modificação (mtime) 2 dias (incluindo os "quebrados", ou seja, 2 dias e pouco...).

Gabarito: C

3. (FCM/IFF-RS/2016) Em um ambiente Linux, desejam-se encontrar todos os arquivos com a extensão `".sh"` a partir do diretório corrente. O comando a ser utilizado é o

- A) `find -all *.sh /`
- B) `grep - all *.sh`
- C) `find -user -i .sh`
- D) `ps -name \ *.sh`
- E) `find . -name \*.sh`

#### Comentários:

Para encontrar os caminhos de determinados arquivos, utilizamos o `find`. O ponto `(.)` indica o diretório corrente e o parâmetro `-name` indica o nome dos arquivos que se procura, ou seja, `"*.sh"`.

Gabarito: E

4. (IESES/BAHIAGÁS/2016) O comando `find` está presente em vários sistemas operacionais Unix-like, sendo muito conhecido pela sua versatilidade. Considere o uso do `bash` em uma distribuição de Linux Ubuntu 14.04, na qual o comando `find` esteja disponível. Um usuário gostaria então de listar somente arquivos de extensões `'java'` e `'py'` em seu diretório `"/home/usuario"`. Assinale a alternativa que executa esta tarefa corretamente.

- A) `find /home/usuario -type f -iname '*.java' -iname '*.py'`
- B) `find /home/usuario -type d -a \( -iname '*.java' -o -iname '*.py' \)`
- C) `find /home/usuario -type f -o \( -iname '*.java' -o -iname '*.py' \)`
- D) `find /home/usuario -type f -a \( -iname '*.java' -o -iname '*.py' \)`
- E) `find /home/usuario -type f -a \( -iname '*.java' -a -iname '*.py' \)`

#### Comentários:

O foco são os arquivos, então utiliza-se o `"-type f"`. Os arquivos desejados são os com extensão `.java` e `.py`, então deve-se pesquisar pelos arquivos `.java` OU `.py`, ou seja, utiliza-se o parâmetro `"-o"` para indicar o OU.

Gabarito: D

5. (INSTITUTO AOCP/EBSERH/2016) Um administrador de sistema operacional Linux necessita procurar determinados textos, dada uma palavra, dentro de um arquivo texto. O comando que possibilita executar essa operação é o



- A) find.
- B) more.
- C) whereis.
- D) grep.
- E) pipe.

**Comentários:**

grep: mostra as linhas de um arquivo que “batem” (match) com um determinado padrão. Um exemplo comum é utilizar o cat para “visualizar” o arquivo e a saída dele passar como entrada para o grep, através do pipe (|). Ao lado podemos ver um exemplo em que é aplicado o cat no arquivo numeros.txt e em cima dessa saída (visualização do conteúdo de numeros.txt) é aplicado o grep 3, ou seja, mostra todas as linhas do arquivo que possuem o número 3.

Gabarito: D

6. (IESES/BAHIAGÁS/2016) Qual dos comandos apresentados a seguir pode ser utilizado para juntar o conteúdo de um arquivo texto de log do Linux, denominado de log1, ao conteúdo de um arquivo texto de log, denominado log2?

- A) cat log2 << log1
- B) cat log2 < log1
- C) cat log1 || log2
- D) cat log1 > log2
- E) cat log1 >> log2

**Comentários:**

cat: concatena arquivos e mostra na saída padrão. Sintaxe: cat ARQ1 ARQ2. Mas se for utilizado o direcionador “simples” (>), coloca o conteúdo do primeiro arquivo sobre o conteúdo do segundo e não mostra na tela. Se for utilizado o direcionador “duplo” (>>), concatena o primeiro ao segundo e não mostra na tela. Quando usado com apenas um arquivo, mostra seu conteúdo. Abaixo um exemplo com o direcionador duplo e depois somente com um arquivo.

```
[root@localhost teste]# cat oi.txt >> tchau.txt
[root@localhost teste]# cat tchau.txt
tchau
oi
[root@localhost teste]#
```

Gabarito: E

7. (IESES/BAHIAGÁS/2016) Um administrador de sistema gostaria de acompanhar o arquivo de log ‘/var/log/apache.log’ em tempo real. Este arquivo está em um servidor Linux Ubuntu 14.04. Supondo que o administrador possua acesso de superusuário ao sistema através do bash. Qual dos comandos a seguir, executado uma única vez, poderia auxiliar o administrador a acompanhar este log dinamicamente (sem interferência adicional)?



- A) tail -rt /var/log/apache.log
- B) more -rt /var/log/apache.log
- C) more -f /var/log/apache.log
- D) cat /var/log/apache.log
- E) tail -f /var/log/apache.log

**Comentários:**

Para ver os dados no fim do arquivo o comando é tail. Uma opção para atualizar a exibição de dados que são inseridos no arquivo é a "-f" (follow = seguir).

Gabarito: E

8. (CONSULPLAN/Câmara de Belo Horizonte-MG/2018) No Sistema Operacional GNU/Linux, o comando cat é utilizado para concatenar arquivos. Sua sintaxe básica é: cat [opções] arquivo. Uma das opções utilizadas é -A, que tem como função:

- A) Numerar todas as linhas na saída.
- B) Exibir todos os caracteres especiais.
- C) Numerar apenas as linhas não vazias.
- D) Exibir caracteres não passíveis de impressão (caracteres de controle).

**Comentários:**

Se o usuário quiser ver caracteres "não imprimíveis", pode utilizar alguns parâmetros:

- -v, --show-nonprinting: utiliza "^" e "M-" para mostrar caracteres não imprimíveis, exceto para Line Feed (quebra de linha) e TAB;
- -E, --show-ends: mostra "\$" no fim de cada linha;
- -T, --show-tabs: mostra TAB como "^I"
- -A, --show-all (mostra tudo): equivalente aos três anteriores juntos (-vET).

Gabarito: B

9. (IDECAN/IF-PB/2019) A respeito dos comandos que podem ser utilizados para verificar o uso de recursos em um sistema operacional Linux, assinale a alternativa que indica, respectivamente, o nome do comando responsável por exibir informações referentes ao uso da memória RAM e uso do espaço em disco.

- A) cat e du
- B) locate e df
- C) cat e free
- D) free e df
- E) du e free



### Comentários:

df: mostra a quantidade de espaço em disco disponível no sistema de arquivos contendo cada argumento de nome de arquivo. Se nenhum nome de arquivo é fornecido, o espaço disponível em todos os sistemas de arquivos montados é mostrado.

free: mostra a quantidade de memória utilizada e livre no sistema.

Gabarito: D

10. (COSEAC/UFF/2019) No Linux, o comando para listar o conteúdo de um diretório e o comando que mostra o caminho por inteiro do diretório em que o usuário se encontra são, respectivamente:

- A) find e more.
- B) cat e grep.
- C) lpr e cat.
- D) df e finger.
- E) ls e pwd.

### Comentários:

ls: lista o conteúdo de um diretório. Os parâmetros mais cobrados são:

- -a (all): todos os arquivos, não ignora as entradas que começam com ".";
- -l (long): utiliza um formato de listagem longa;
- -s (size): imprime o tamanho de cada arquivo.

pwd (print working directory): imprime o caminho completo do diretório de trabalho corrente.

Gabarito: E

11. (CS-UFG/IF Goiano/2019) No sistema operacional GNU/Linux, usando a linha de comando, deseja-se executar o seguinte: a) criar um diretório chamado IFG; b) criar um arquivo de texto chamado Concurso.txt; c) apagar o arquivo de texto Concurso.txt; d) apagar o diretório IFG. Qual é a sequência de comandos a ser empregada? Obs.: o sinal "," é um mero separador entre os comandos

- A) mkdir IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- B) mkdir IFG , create Concurso.txt , kill Concurso.txt , killdir IFG
- C) create IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- D) create IFG , create Concurso.txt , kill Concurso.txt , killdir IFG

### Comentários:



Questão bem objetiva. O `mkdir` serve para criar um diretório. Uma forma de criar um arquivo texto é através do comando `touch`. Para remover um arquivo utilizamos o comando `rm`. E para excluir um diretório, temos o comando `rmdir`.

Gabarito: A

12. (FUNDATEC/IF Farroupilha-RS/2019) O comando "`cd`", no sistema operacional Linux, serve para:

- A) Mudar do diretório atual para o especificado.
- B) Criar um arquivo.
- C) Criar um diretório.
- D) Renomear um arquivo.
- E) Renomear um diretório.

Comentários:

`cd` = "changed directory", serve para mudar o diretório atual. Ex.: `cd /home/evandro` muda do diretório atual (digamos que esteja em `/home`) para `cd /home/evandro`.

Gabarito: A

13. (AMAUC/Pref. de Arabutã-SC/2021) Qual atalho usado pelo SO Linux, é global, e repete o último comando?

- A) \$\$
- B) %%
- C) !!
- D) ##
- E) \*\*

Comentários:

Questão recorrente, mas em outras bancas. Não tem o que comentar. É decoreba! Para repetir o último comando utiliza-se dupla exclamação (!!).

Gabarito: C

14. (UFAM/UFAM/2021) No sistema operacional GNU/LINUX, o comando que cria uma partição `ext2` na partição `/dev/hda3` é:

- A) `# mkfs -t ext2 /dev/hda3`
- B) `# mdfs -s ext3 /dev/hda3`
- C) `# mkfp -t ext2 /dev/hda3`





D) # mkfs -s ext3 /dev/hda2

E) # mkfs -t ext2 /dev/hda2

#### Comentários:

mkfs: cria um sistema de arquivos em um dispositivo de armazenamento. A estrutura básica do comando mkfs é:

```
# mkfs -t tipo_do_sistema_de_arquivos dispositivo
```

Ex. com EXT4:

```
# mkfs -t ext4 /dev/sdb1
```

Gabarito: A

15. (FURB/Prefeitura de Gaspar-SC/2022) O Sistema Operacional Linux possui diversos utilitários que são executados através de linha de comando. Associe a segunda coluna de acordo com a primeira relacionando o comando e sua respectiva função:

Primeira coluna: Comando

1.vi

2.chmod

3.pwd

4.ip addr

Segunda coluna: Função

Permite alterar as permissões de um diretório ou arquivo.

Permite configurar e gerenciar endereços de rede associados a uma interface.

Editor de textos ASCII utilizado para criar e editar arquivos.

Mostra o caminho completo do diretório corrente.

Assinale a alternativa que apresenta a correta associação entre as colunas:

A) 2 - 3 - 1 - 4

B) 2 - 4 - 1 - 3

C) 3 - 1 - 2 - 4

D) 3 - 2 - 1 - 4



E) 1 - 2 - 3 - 4

Comentários:

- 1.vi - editor de texto simples (bem simples, mesmo!).
- 2.chmod - (change mode) atribui/altera permissões de arquivos/diretórios.
- 3.pwd - (print working directory) mostra o diretório corrente.
- 4.ip addr - mostra/manipula configurações de roteamento, dispositivos de rede, entre outros.

Gabarito: B

16. (AOCP/BANESE/2022) Em um sistema operacional Unix-like, como o Linux Ubuntu, qual comando lê as cinco primeiras linhas de um arquivo de texto?

- A) tail -f
- B) tail -n 5
- C) tac -n 5
- D) cat -n 5
- E) head -n 5

Comentários:

tail ("cauda") mostra as últimas linhas (padrão = 10).

head ("cabeça" ou "cabeçalho") lê as primeiras (padrão = 10).

A opção -n (tanto para tail como para head) define quantas linhas deseja mostrar, se não quiser o padrão (10).

Gabarito: E

17. (FUNDATEC/Pref. de S. J. A.-RS/2022) No Linux, o comando \_\_ pode ser utilizado para identificar a quantidade de espaço em disco usado e disponível em cada partição do sistema.

Assinale a alternativa que preenche corretamente a lacuna do trecho acima.

- A) df
- B) du
- C) cd
- D) cp
- E) ls

Comentários:

Existe uma sutil diferença entre df e du. Vamos ver abaixo.



df ("disk free") mostra a quantidade de espaço em disco disponível no sistema de arquivos contendo cada argumento de nome de arquivo. Se nenhum nome de arquivo for fornecido, o espaço disponível em todos os sistemas de arquivos montados é mostrado.

du ("disk usage"): mostra o espaço ocupado por diretórios e arquivos no sistema de arquivos. Por exemplo, para verificar o espaço ocupado por um diretório:

```
du -h /caminho_do_diretorio
```

O argumento -h torna a saída mais legível para humanos, mostrando os tamanhos em formato legível, como KB, MB ou GB.

Gabarito: A

18. (FUNDATEC/IF-RS/2022) No sistema operacional Linux, o comando pwd:

- A) Lista arquivos de um diretório.
- B) Altera a senha do usuário corrente.
- C) Cria um diretório.
- D) Exibe o nome do diretório corrente.
- E) Procura arquivos por conteúdo.

Comentários:

pwd = *print working directory* - mostra o diretório corrente.

Gabarito: D

19. (FUNDATEC/Pref. de Viamão-RS/2022) O comando abaixo foi executado em um sistema operacional Linux:

```
echo -n "prefeitura" | wc -c
```

Qual é o seu retorno?

- A) Prefeitura
- B) PREFEITURA.
- C) arutieferp
- D) 10
- E) 11

Comentários:

Vamos ver um exemplo do comando echo:



\$ echo oi → mostra "oi" na saída padrão, geralmente a tela do computador. O parâmetro -n é usado para imprimir uma linha de texto sem adicionar uma nova linha ao final, ou seja, é para suprimir a quebra de linha automática que o comando echo geralmente adiciona.

Em relação ao wc, esse comando conta a quantidade de caracteres, linhas ou palavras. Com o parâmetro -c conta a quantidade de caracteres.

Então, a resposta é 10 (quantidade de caracteres da palavra prefeitura).

Gabarito: D

20. (FUNDATEC/Pref. de Esteio-RS/2022) Observe a Figura 2 abaixo, obtida da tela do terminal de comandos de um sistema operacional, após a execução de uma ferramenta de administração:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sdb	251G	22G	218G	9%	/
tmpfs	13G	0	13G	0%	/mnt/xxx
tools	466G	261G	205G	56%	/init
none	13G	0	13G	0%	/dev
none	13G	4.0K	13G	1%	/run
none	13G	0	13G	0%	/run/lock
none	13G	0	13G	0%	/run/shm
none	13G	0	13G	0%	/run/user
tmpfs	13G	0	13G	0%	/sys/fs/cgroup

Assinale a alternativa que indica a ferramenta que gerou a saída da figura, assim como o sistema operacional (S.O.) onde ela foi utilizada.

- A) Ferramenta mount e S.O. Linux.
- B) Ferramenta mount e S.O. Windows.
- C) Ferramenta diskpart e S.O. Windows.
- D) Ferramenta df e S.O. Windows.
- E) Ferramenta df e S.O. Linux.

Comentários:

df ("disk free") mostra a quantidade de espaço em disco disponível no sistema de arquivos contendo cada argumento de nome de arquivo. Se nenhum nome de arquivo for fornecido, o espaço disponível em todos os sistemas de arquivos montados é mostrado.

Não existe essa ferramenta no Windows!

Gabarito: E

21. (FUNDATEC/IF Farroupilha-RS/2023) No sistema operacional Linux, o comando \_\_\_\_\_ é utilizado para trocar o proprietário de um arquivo.



Assinale a alternativa que preenche corretamente a lacuna do trecho acima.

- A) chown
- B) man
- C) apropos
- D) who
- E) finger

#### Comentários:

chown (change owner): troca o proprietário de um arquivo.

man: mostra o "manual" de um comando ou ferramenta.

apropos: procura por uma determinada expressão nas páginas de documentação.

who: exibe informações sobre os usuários que estão atualmente conectados ao sistema. Ele mostra detalhes como nome de usuário, terminal de onde estão logados, endereço IP, hora de login e outros detalhes relacionados à sessão de login.

finger: exibe informações sobre usuários do sistema. Normalmente é utilizado para fornecer detalhes sobre um usuário específico, como nome, diretório home, informações sobre a última vez que o usuário fez login etc. O formato básico do comando finger é:

```
# finger nome_do_usuario
```

Gabarito: A

22. (FUNDATEC/GHC-RS/2023) Qual comando do GNU Linux pode ser usado para criar um usuário?

- A) su
- B) sudo
- C) newuser
- D) useradd
- E) passwd

#### Comentários:

Poderia surgir uma dúvida entre newuser e useradd, certo? Bom, newuser não existe e o useradd cria contas de usuário no sistema. Único parâmetro obrigatório é login do usuário. Alguns outros são:

- -c "nome do usuário": grava no arquivo passwd o nome do proprietário da conta ou qualquer outra observação e comentário importantes;
- -d diretório\_home: fornece o caminho completo do diretório home do usuário;
- -m: cria o diretório home fornecido na opção "-d";
- -g número\_do\_grupo: fornece o grupo padrão da conta do usuário;



- -s shell: fornece o caminho completo do shell utilizado pela conta (ex.: /bin/bash);
- -e, --expiredate DATA\_EXPIRAÇÃO: define a data de expiração da conta.

Gabarito: D

23. (AOCP/IF-MA/2023) Com a utilização dos serviços de terminal do Linux, qual comando pode ser utilizado para contar linhas, palavras ou bytes de um arquivo específico ou do que for escrito no terminal?

- A) sudo
- B) wc
- C) chmod
- D) clear
- E) echo

Comentários:

wc: significa "word count", mas além de "contar palavras", conta também as linhas e os caracteres em um arquivo. As opções mais frequentes são:

- -c: conta o número de caracteres de um ou mais arquivos;
- -l: conta o número de linhas de um ou mais arquivos;
- -L: conta o número de caracteres da maior linha do arquivo;
- -w: conta as palavras de um ou mais arquivos.

Gabarito: B

24. (AOCP/IF-MA/2023) O sistema operacional Linux possui uma vasta lista de comandos. Porém, existem alguns que são essenciais para um usuário desse sistema operacional. Esses comandos são indicados para aqueles que não conseguem suprir suas necessidades por meio da interface gráfica ou que estejam usando o sistema em modo texto.

Manoel, técnico de informática, está trabalhando em modo texto no sistema operacional Linux e precisa verificar qual é o espaço livre no disco. Qual comando ele deve digitar?

- A) cd
- B) cp
- C) mv
- D) rm
- E) df

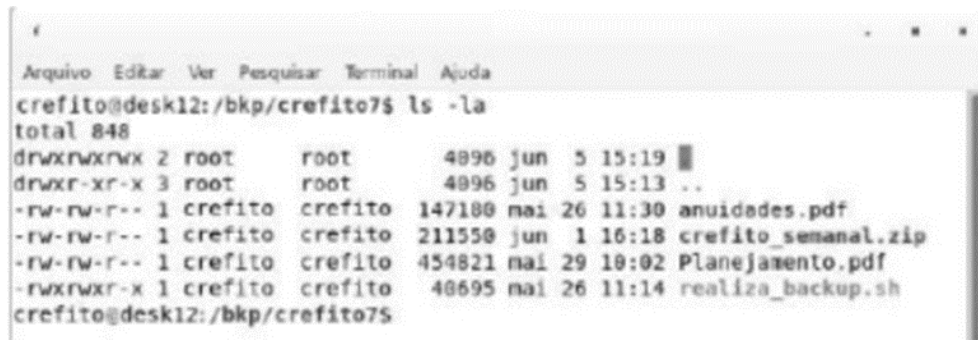
Comentários:



df ("disk free") mostra a quantidade de espaço em disco disponível no sistema de arquivos contendo cada argumento de nome de arquivo. Se nenhum nome de arquivo for fornecido, o espaço disponível em todos os sistemas de arquivos montados é mostrado.

Gabarito: E

25. (Quadrix/CREFITO-7ª Região/2023)



```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
crefito@desk12:/bkp/crefito7$ ls -la
total 848
drwxrwxrwx 2 root    root    4096 jun  5 15:19
drwxr-xr-x 3 root    root    4096 jun  5 15:13 ..
-rw-rw-r-- 1 crefito crefito 147180 mai 26 11:30 anuidades.pdf
-rw-rw-r-- 1 crefito crefito 211550 jun  1 16:18 crefito_semanal.zip
-rw-rw-r-- 1 crefito crefito 454821 mai 29 10:02 Planejamento.pdf
-rwxrwxr-x 1 crefito crefito 40695 mai 26 11:14 realiza_backup.sh
crefito@desk12:/bkp/crefito7$
```

Quanto à imagem acima e aos comandos no Ubuntu a ela relacionados, julgue o item.

Caso o comando `rm -f anuidades.pdf crefito_semanal.zip` seja executado, tanto o arquivo `anuidades.pdf` quanto o arquivo `crefito_semanal.zip` serão removidos.

Comentários:

`rm`: remove um arquivo, ex.: `rm nome_arquivo`. O parâmetro `-f` é frequentemente usado para forçar a remoção sem solicitar confirmação, ignorando possíveis mensagens de aviso.

Mais de um arquivo pode ser removido com a aplicação do `rm` uma única vez, bastando colocar os nomes deles separados por espaço.

Gabarito: Certo

26. (Quadrix/CREFITO-7ª Região/2023) Quanto à imagem acima e aos comandos no Ubuntu a ela relacionados, julgue o item.

Com o comando `vi realiza_backup.sh`, o usuário terá acesso às funcionalidades de edição no arquivo `realiza_backup.sh`.

Comentários:

O `vi` é um editor de texto e o arquivo `realiza_backup.sh` possui permissão de leitura (`r`) para todos (usuário dono, grupo e outros), então a questão está correta.

Gabarito: Certo

27. (FUNCERN/Câmara de Natal-RN/2023) Um técnico em informática recém-contratado necessitou consultar a distribuição e versão do Linux que está sendo executada em uma máquina. Felizmente, existem várias maneiras de descobrir essas informações usando a linha de comando. Uma delas é



- A) `cat /etc/os-release`
- B) `sb_release -x`
- C) `cat /etc/ -release`
- D) `lsb_release -os`

#### Comentários:

Para responder esta questão é necessário ter o conhecimento prévio que a distribuição e versão do Linux ficam no arquivo `/etc/os-release`. A partir disso é só usar o comando `cat` para mostrar o conteúdo de tal arquivo.

Gabarito: A

28. (FUNCERN/Câmara de Natal-RN/2023) Existe um comando que é utilizado no terminal do Linux para criar um arquivo vazio ou atualizar o carimbo de data/hora de um arquivo existente. Esse comando recebe como argumento o nome do arquivo que deseja criar/atualizar e denomina-se

- A) `find`.
- B) `mkdir`.
- C) `touch`.
- D) `cat`.

#### Comentários:

`find`: procura por arquivos;

`mkdir`: cria um diretório;

`touch`: cria um arquivo vazio, e também atualiza as informações de data e hora de um arquivo existente, ex.: `touch nome_arquivo`;

`cat`: mostra o conteúdo de um arquivo (se for apenas um) ou concatena o conteúdo de arquivos, mostrando o resultado da concatenação.

Gabarito: C

29. (FUNDATEC/Pref. de Maçambará-RS/2023) Em um sistema operacional Linux, um usuário deseja criar um novo diretório pelo prompt de comandos. Qual comando ele deve utilizar?

- A) `file`
- B) `mkdir`
- C) `dirname`
- D) `rmdir`
- E) `cd`





## Comentários:

file: mostra o tipo de arquivo (baseado na assinatura, os famosos "magic numbers");

**mkdir**: significa "make directory" (cria um diretório), ex.: mkdir /home/evandro/teste cria o diretório "teste" dentro de /home/evandro (se não houver esse caminho, ocorre um erro);

dirname: extrai o nome do diretório de um caminho de arquivo especificado ;

rmdir: remove um diretório vazio, ex.: rmdir /root/teste;

cd: significa "change directory" (muda o diretório corrente).

Gabarito: B

30. (FUNDATEC/Pref. de Maçambara-RS/2023) Em relação ao prompt de comandos do sistema operacional Linux, assinale a alternativa que corresponde a uma das funções do comando cat.

- A) Mostrar as diferenças entre dois arquivos de texto.
- B) Exibir o conteúdo de um ou mais arquivos.
- C) Alterar o diretório de trabalho corrente.
- D) Modificar as permissões de um ou mais arquivos.
- E) Mostrar o nome do diretório de trabalho corrente.

## Comentários:

cat: concatena arquivos e mostra na saída padrão. Sintaxe: cat ARQ1 ARQ2. Mas se for utilizado o direcionador "simples" (>), coloca o conteúdo do primeiro arquivo sobre o conteúdo do segundo e não mostra na tela. Se for utilizado o direcionador "duplo" (>>), concatena o primeiro ao segundo e não mostra na tela (se não existir o segundo, este é criado). Quando usado com apenas um arquivo, mostra seu conteúdo. Abaixo um exemplo com o direcionador duplo e depois somente com um arquivo (importante saber que o arquivo "oi.txt" tem o conteúdo "oi" e que "tchau.txt" tem o conteúdo "tchau").

```
[root@localhost teste]# cat oi.txt >> tchau.txt
[root@localhost teste]# cat tchau.txt
tchau
oi
[root@localhost teste]#
```

Gabarito: B

31. (FURB/Pref. de Jaraguá do Sul-SC/2023) Associe a segunda coluna de acordo com a primeira que relaciona utilitários de linha de comando no Linux às suas funções e características:

Primeira coluna: Utilitário de linha de comando

1.nano



2.cat

3.systemctl

4.chmod

Segunda coluna: Funções e características

Comando permite gerenciar serviços do Linux.

Comando permite unir, criar e exibir arquivos

É um editor de textos para sistemas Linux.

É utilizado para lidar com permissões de arquivos do sistema Linux.

Assinale a alternativa que apresenta a correta associação entre as colunas:

A) 3 - 1 - 4 - 2.

B) 3 - 2 - 1 - 4.

C) 1 - 4 - 2 - 3.

D) 4 - 2 - 3 - 1.

E) 1 - 3 - 4 - 2.

**Comentários:**

nano: editor de texto (mais elaborado que o vi);

cat: mostra o conteúdo de um arquivo, ou concatena e mostra o conteúdo concatenado;

systemctl: ferramenta de controle para o Systemd, um sistema de inicialização e gerenciamento de serviços amplamente utilizado em sistemas operacionais Linux modernos;

chmod: atribui permissões a arquivos.

Gabarito: B

32. (FUNDATEC/GHC-RS/2023) Um técnico estava utilizando um sistema operacional GNU Linux e executou os seguintes comandos:

1. cat arquivo.txt

2. wc --words arquivo.txt

O primeiro comando apresentou a saída do quadro abaixo:

Grupo Hospitalar Conceição



Qual a saída correta para o segundo comando?

- A) GrupoHospitalarConceição
- B) Grupo,Hospitalar,Conceição
- C) 26 arquivo.txt
- D) 1 arquivo.txt
- E) 3 arquivo.txt

Comentários:

A partir da saída do primeiro comando: "Grupo Hospitalar Conceição", ao aplicar o comando:

```
wc --words arquivo.txt
```

é mostrada a quantidade de palavras, seguido do nome do arquivo, ou seja:

```
3 arquivo.txt
```

Gabarito: E

33. (FUNDATEC/PROCERGS/2023) Suponha que você está em um sistema Linux e precisa criar o diretório "fotos". Considere que você já está no local raiz para o novo diretório. Qual alternativa apresenta o comando a ser executado para criar o diretório "fotos"?

- A) cd fotos
- B) cp fotos
- C) file fotos
- D) mkdir fotos
- E) rmdir fotos

Comentários:

Se o usuário já estiver na raiz, é só criar o diretório "fotos", com o comando mkdir (make directory):

```
mkdir fotos
```

Gabarito: D

34. (FUNDATEC/PROCERGS/2023) O sistema operacional Linux possui diversos comandos. Um desses comandos é o cat. Considerando a execução do comando abaixo, qual o resultado esperado?

```
cat arquivo1 arquivo2
```

- A) O arquivo1 recebe o conteúdo do arquivo2
- B) Exibe conteúdo de arquivo1 e arquivo2



- C) Envia para a impressora o arquivo1 e o arquivo2
- D) Faz uma cópia de arquivo1 com nome arquivo2
- E) Renomeia arquivo1 para arquivo2

**Comentários:**

cat aplicado a dois arquivos, primeiro concatena os dois, depois mostra o conteúdo (já concatenado). Ex.: se o arquivo1 tiver o conteúdo "a" e o arquivo2 tiver "b", ao aplicar

```
cat arquivo1 arquivo2
```

o que deve aparecer na tela é (a concatenação):

a  
b

Gabarito: B

35. (FUNDEPES/Pref. de M. Deodoro-AL/2023) O sistema operacional é o software utilizado para intermediar a operacionalização do computador, por parte de seus usuários. O sistema operacional Linux é um software de código aberto que permite a operacionalização tanto em modo gráfico, quanto em modo textual. O terminal do Linux é a forma mais utilizada para executar ações em modo textual, via linha de comando. Tal característica favorece o desenvolvimento de scripts para automatizar tarefas. Dado o exemplo de uma linha de comando que pode ser executada em um terminal Linux, `cat documento.txt | grep "forma"` assinale a alternativa que descreve a ação realizada pela execução dessa linha de comando.

- A) Atribuir à categoria do arquivo "documento.txt" o valor "forma".
- B) Adicione a palavra "forma" na última linha do arquivo "documento.txt".
- C) Listar as categorias do arquivo "documento.txt" que iniciam com a palavra "forma".
- D) Ler o conteúdo do arquivo "documento.txt" e apresente na tela as linhas que contêm o texto "forma".
- E) Realizar a contagem do número de vezes que a palavra "forma" aparece no arquivo "documento.txt".

**Comentários:**

`cat documento.txt` → mostra o conteúdo do arquivo documento.txt.

O pipe (|) faz com que a saída do primeiro comando seja a entrada para o segundo comando (`grep "forma"`).

O segundo comando vai mostrar as linhas que possuem o conteúdo "forma".

Gabarito: D



## QUESTÕES COMENTADAS - CEBRASPE

1. (CEBRASPE/TRT8/2013) Para alterar o arquivo de configuração do serviço de DHCP no Linux, deve-se acessar o diretório

- A) /dev.
- B) /mnt.
- C) /etc.
- D) /usr/bin.
- E) /boot.

### Comentários:

Por padrão, os arquivos de configuração ficam no diretório "/etc", como é o caso do "dhcpd.conf".

Gabarito: C

2. (CEBRASPE/FUB/2015) O protocolo DNS (domain name service), localizado no nível de aplicação da camada de transporte do TCP, é responsável pelo mapeamento de nomes e de endereços.

### Comentários:

O DNS está localizado na camada de aplicação. A questão misturou "nível de aplicação da camada de transporte do TCP".

Gabarito: Errado

3. (CEBRASPE/TRE-MT/2015) Acerca do servidor DNS/BIND (Domain Name System/Berkley Internet Domain), cuja funcionalidade é resolver nomes da rede, assinale a opção correta.

- A) Cada domínio tem seus registros de recursos e o registro de domínio denominado NS (name server), o qual é utilizado para definir propriedades básicas do domínio e sua zona.
- B) Um servidor DNS utiliza LDAP para fazer armazenamento das zonas de domínio para uma rápida resolução de um nome.
- C) O BIND, que utiliza a porta 53, é um programa de código aberto utilizado pela maior parte dos servidores DNS.
- D) Os domínios de um servidor DNS são organizados na Internet sobre uma estrutura de dados do tipo lista encadeada, sendo o primeiro elemento da lista um ponto.
- E) O protocolo HTTP implementa, por padrão, um servidor de resolução de nomes amplamente utilizado na Internet conhecido como DNS.

### Comentários:



Para quem não está acostumado com essa parte de servidores, memorize o seguinte: servidor DNS é quase sinônimo de BIND! Se você ainda não o conhece e não leu sobre ele, recomendo uma passada rápida na URL <https://www.isc.org/downloads/bind/>.

Gabarito: C

4. (CEBRASPE/TCE-PR/2016) Uma consulta DNS inicial típica, originada de uma máquina de usuário e encaminhada ao servidor de nomes local para um nome de domínio externo não armazenado em cache DNS, será do tipo

- A) raiz.
- B) domínio de alto nível.
- C) iterativa.
- D) recursiva.
- E) direta.

Comentários:

O foco da questão é a consulta realizada de um equipamento ao servidor DNS local, supondo que a informação não esteja na memória cache desse servidor local. Nesse caso, o servidor DNS local irá requisitar ao servidor DNS TLD, receberá como resposta o próximo servidor DNS a ser consultado e repetirá o processo até ter toda a informação desejada. Armazenará em sua cache e responderá ao cliente (equipamento que fez a solicitação inicial). Note que o cliente fez uma requisição apenas e obteve uma resposta, mas o servidor DNS local realizou consultas recursivas.

Gabarito: D

5. (CEBRASPE/TCE-SC/2016) Após o servidor local SMTP aceitar uma mensagem para subsequente envio, é necessário determinar o endereço do servidor de email do destinatário. Essa etapa é realizada mediante consulta DNS a um servidor de nomes capaz de prover a informação, no qual serão verificados os registros especiais MX (mail exchange).

Comentários:

Vamos relembrar a tabela, focando na linha sobre "MX":

Tipo	Significado	Valor
SOA	Início de autoridade ( <i>Start of Authority</i> ).	Parâmetros para essa zona.
A	Endereço IPv4.	Inteiro de 32 bits.
AAAA	Endereço IPv6.	Inteiro de 128 bits.
MX	Troca de mensagens de e-mail.	Prioridade, domínio disposto a aceitar e-mails.



NS	Servidor de nomes.	Nome de um servidor para este domínio.
CNAME	Nome canônico ( <i>alias</i> = apelido).	Nome de domínio.
PTR	Ponteiro (usado para o DNS reverso <sup>1</sup> )	Nome alternativo de um end. IP.
SPF	Estrutura de política do transmissor.	Codificação de texto da política de envio de mensagens de e-mail.
SRV	Identifica computadores que hospedam serviços específicos.	Host que o oferece.
TXT	Informações sobre um servidor, rede, <i>datacenter</i> etc.	Texto ASCII com descrições.

Gabarito: Certo

6. (CEBRASPE/Polícia Científica-PE/2016) Na operação de um serviço DNS, se uma consulta a um servidor de alto nível retornar uma resposta positiva para o servidor DNS local, o mapeamento nome/endereço será

- A) armazenado definitivamente no banco de dados do servidor DNS local.
- B) replicado para todos os servidores DNS alcançáveis para que estejam disponíveis a outras consultas.
- C) repassado pelo módulo tradutor a um servidor raiz para confirmar a resposta recebida.
- D) armazenado em cache local e pode ser mantido por um tempo definido por configuração.
- E) repassado ao módulo tradutor do provedor de serviços para realizar a entrega à aplicação.

Comentários:

Quando o servidor DNS local busca uma informação nova, ele guarda na cache, por tempo determinado. Esse tempo varia de segundos/minutos a um dia, por exemplo. Depende do tempo definido no campo TTL.

Gabarito: D

7. (CEBRASPE/Polícia Federal/2018) As atualizações entre servidores DNS utilizam o UDP, enquanto as consultas feitas a servidores DNS utilizam o TCP (ou, opcionalmente, o SCTP).

Comentários:

<sup>1</sup> Envia um endereço IP como consulta e recebe o nome como resposta.



Vamos tentar utilizar a lógica: o UDP é bem mais leve, então deve ser utilizado em consultas DNS, pois tais consultas ocorrem em grande número na Internet e se alguma mensagem for perdida não há grandes problemas, bastando realizar uma nova consulta. As atualizações entre servidores DNS são bem menos frequentes que as consultas e é importante que mensagens não sejam perdidas pela rede, logo, é interessante que se utilize TCP, correto? Bom, essa lógica é a utilizada no DNS! E, como podemos ver, a questão inverteu o TCP com o UDP.

Gabarito: Errado

8. (CEBRASPE/EMAP/2018) Em um servidor DNS que esteja utilizando Bind, o arquivo `/etc/named.conf` mantém as configurações de funcionamento do serviço DNS. Nesse servidor, é possível configurar a porta que deve escutar o serviço; assim, para que o serviço escute a porta 53, deve-se configurar a opção seguinte.

```
Listening 53 { any;};
```

Comentários:

No arquivo `named.conf` são configuradas informações de zonas. O próprio nome do arquivo já sugere isso (`named` = nomes de domínios). Não há a configuração da porta em que o serviço deve "escutar" nesse arquivo!

Gabarito: Errado

9. (CEBRASPE/TRT-8ª Região/2022) Caso se queira bloquear, no Linux, o IP 10.1.1.3 por meio do iptables, o comando a ser utilizado é o

- A) `iptables -a INPUT -i 10.1.1.3 -r BLOCK`.
- B) `iptables -A INPUT -s 10.1.1.3 -j DROP`.
- C) `iptables -I INPUT -s 10.1.1.3 -r BLOCK`.
- D) `iptables -I INPUT -s 10.1.1.3 -r DENY`.
- E) `iptables -i INPUT -d 10.1.1.3 -b REFUSED`.

Comentários:

Vamos por partes:

- Para adicionar (add) uma regra deve ser utilizado `-A`;
- As chains são `INPUT`, `OUTPUT`, `FORWARD`, entre outras. Só a letra B tem uma chain existente (`INPUT`, que indica que o pacote está chegando);
- A ação a ser efetuada é indicada por `"-j"`. "Bloquear" foi forçado, por parte da banca. Pode-se descartar (`DROP`), rejeitar (`REJECT`), aceitar (`ACCEPT`), entre outras. `DROP` ainda seria a melhor opção.

A questão não está bem elaborada, mas dá para descartar as alternativas absurdas e a letra B é a melhor opção.

Gabarito: B







## QUESTÕES COMENTADAS - FGV

1. (FGV/MEC/2009) No UNIX, o DNS é implementado por meio do software Berkeley Internet Name Domain (BIND), que segue a filosofia cliente/servidor.

O cliente do BIND é denominado "resolver", responsável por gerar as "queries" enquanto que o servidor do BIND é chamado por "named", responsável por responder às "queries".

Configurações BIND são descritas pelo tipo de serviço a ser executado pelo software, existindo quatro níveis de serviço que podem ser definidos nessas configurações: "resolver-only systems", "caching-only servers", "primary servers" e "secondary servers".

Os parâmetros que definem a configuração do "resolver-only systems" são inseridas no seguinte arquivo:

- A) /etc/resolv.sys
- B) /etc/resolv.cfg
- C) /etc/resolv.bind
- D) /etc/resolv.conf
- E) /etc/resolv.parm

### Comentários:

O arquivo resolv.conf também serve para especificar o tipo (nível) de serviço a ser executado pelo software, a saber:

- resolver-only system: não requer que o sistema local tenha um servidor DNS em execução, requer apenas o resolver;
- caching-only server: servidor que não possui uma cópia da tabela de zonas, ou seja é não-autoritativo, possuindo uma grande quantidade de registros em cache;
- primary server (master server): onde todas atualizações manuais devem ser realizadas, é autoritativo;
- secondary server: busca informações de zona em um master server, é autoritativo.

Gabarito: D

2. (FGV/TJ-GO/2014) A empresa Y passou a adotar uma política de uso de software livre e resolveu mudar o seu correio eletrônico, que era baseado em Microsoft Exchange. O novo ambiente deveria rodar em ambiente Linux. Uma das possíveis escolhas do novo software de correio eletrônico recai no programa:

- A) nginx;
- B) puppet;
- C) apache;



- D) postfix;
- E) bind.

**Comentários:**

Alguns exemplos de servidores de e-mail no Linux: qmail, postfix e sendmail.

Gabarito: D

3. (FGV/TCE-SE/2015) Um programa precisa simular o comportamento de um cliente DNS. Para funcionar adequadamente, o programa precisa enviar as consultas para um servidor DNS, especificamente para a sua porta:

- A) udp/23
- B) icmp/34
- C) tcp/22
- D) ip/50
- E) udp/53

**Comentários:**

Como já vimos, o protocolo de transporte utilizado para consultas DNS é o UDP e a porta é a 53. Essas informações devem estar enraizadas em seu cérebro! Para a transferência de zonas de DNS, é utilizado o TCP (na mesma porta: 53) como protocolo de transporte.

Gabarito: E

4. (FGV/TCE-SE/2015) Em um equipamento rodando Linux, um administrador deseja alterar o endereço default do servidor DNS usado para consultas. Para isso, ele precisa alterar o arquivo:

- A) /etc/named.conf
- B) /etc/bind.conf
- C) /etc/network.conf
- D) /etc/dns.conf
- E) /etc/resolv.conf

**Comentários:**

Um exemplo de arquivo resolv.conf:

```
domain ti.teste.com.br
search ti.teste.com.br teste.com.br
nameserver 192.168.1.1
nameserver 192.168.2.1
```



Gabarito: E

5. (FGV/COMPESA/2016) O sistema de Nomes de Domínio - DNS permite transformar nomes digitados em um navegador WEB em um endereço de rede. O nome do host e o nível do domínio para o domínio "system.master.com" são, respectivamente,

- A) system e segundo nível.
- B) system e terceiro nível.
- C) master e terceiro nível.
- D) master e segundo nível.
- E) .com e terceiro nível.

#### Comentários:

Um servidor DNS raiz "aponta" para servidores de primeiro nível (TLD). No domínio mostrado, o TLD é o ".com", que aponta para um servidor DNS responsável por "master" (segundo nível), que por sua vez, aponta para um terceiro nível, o host "system", no caso da questão.

Gabarito: B

6. (FGV/AL-RO/2018) Wallace é administrador de um servidor com sistema operacional CentOS e deseja compartilhar um diretório com os integrantes da empresa por meio do protocolo NFS.

Para que a configuração desse compartilhamento possa ser efetuada, Wallace deve editar o arquivo /etc/

- A) services
- B) modules
- C) hosts
- D) export
- E) fstab

#### Comentários:

O NFS tem a ideia de "exportar" arquivos, ou seja, deixar disponíveis para outros usuários. O nome do arquivo de configuração é "exports" e fica dentro do diretório /etc. A resposta correta não tem nas alternativas, faltou um "s", mas tudo bem.

Gabarito: D

7. (FGV/TRT-MA/2022) O comando do sistema operacional Linux utilizado por administradores de redes para configurar ou examinar informações sobre tabelas de roteamento e interfaces de redes ou criar túneis em um servidor se denomina

- A) traceroute.



- B) ifshow.
- C) netstat.
- D) ping.
- E) ip.

#### Comentários:

ip: ferramenta para configurar e exibir informações de rede relacionadas a interfaces, endereços IP, tabelas de roteamento, túneis, QoS (Qualidade de Serviço) etc. É a opção mais moderna em relação aos comandos ifconfig e route.

Para exibir informações sobre todas as interfaces de rede:

```
$ ip a
```

ou

```
$ ip addr show
```

Para exibir informações detalhadas sobre determinada interface:

```
$ ip a show eth0
```

Para ativar uma interface de rede:

```
$ sudo ip link set dev eth0 up
```

Para desativar uma interface de rede:

```
$ sudo ip link set dev eth0 down
```

Para configurar um endereço IP em uma interface:

```
$ sudo ip addr add 192.168.1.150/255.255.255.0 dev eth0
```

Para exibir informações sobre a tabela de roteamento:

```
$ ip route
```

Gabarito: E

8. (FGV/MPE-SC/2022) Um analista obteve acesso a um terminal Linux para realizar o teste de um sistema, mas não tem certeza se o computador tem acesso à Internet, o que será necessário para o teste.

A forma mais simples de o analista verificar a conectividade do computador à Internet é:

- A) nc -vzu 127.0.0.1 80;
- B) mtr 192.168.0.1;
- C) ping 1.1.1.1;
- D) nmap -A 127.0.0.1;



E) telnet 10.0.0.255 22.

Comentários:

ping: ferramenta usada para testar a conectividade de rede e verificar a disponibilidade de um host ou dispositivo. São enviados pacotes ICMP (Internet Control Message Protocol) para o host de destino e as respostas são aguardadas. Pode-se "pingar" através do nome (o DNS é utilizado para a tradução) ou o endereço IP diretamente:

```
$ ping www.estrategiaconcursos.com.br
```

```
$ ping 192.168.1.150
```

Gabarito: C



## QUESTÕES COMENTADAS - FCC

1. (FCC/TCE-SP/2010) Em Linux, o `/etc/resolv.conf` é um arquivo texto simples, com um parâmetro por linha e especificações de endereços de servidores DNS. Nesse arquivo existem três palavras chaves normalmente usadas, que são

- A) domain, search e resolv.
- B) search, resolv e order.
- C) domain, search e nameserver.
- D) search, nameserver e order.
- E) search nameserver e resolv.

### Comentários:

Mais uma vez vamos ver um exemplo do `resolv.conf`, para memorizar 😊

```
domain ti.teste.com.br
search ti.teste.com.br teste.com.br
nameserver 192.168.1.1
nameserver 192.168.2.1
```

Gabarito: C

2. (FCC/TCE-SP/2010) No Linux, ele é muito útil para máquinas que são acessadas frequentemente, pois a inclusão de um computador neste arquivo dispensa a consulta a um servidor DNS para obter um endereço IP, sendo muito útil para máquinas que são acessadas frequentemente. Trata-se de:

- A) `/etc/nameserver`
- B) `/etc/localhost`
- C) `/etc/ipsec`
- D) `/etc/hosts`
- E) `/etc/dnshost`

### Comentários:

Um exemplo do arquivo `hosts` (que também existe no Windows!):

```
# Aqui são colocados comentários
# End. IP      Nome do host      Alias (opcional)
127.0.0.1     localhost         localmente
192.168.0.1   www.teste.com.br teste
```

Gabarito: D



3. (FCC/CNMP/2015) O serviço de nome de domínios (DNS) possui uma arquitetura do tipo cliente/servidor na qual a base de dados é distribuída por toda internet. Nessa arquitetura, o acesso ao servidor DNS para buscar o relacionamento IP/Domínio é feito pelo cliente que é o

- A) Browser.
- B) DNS Cache.
- C) DNS Resolver.
- D) DNS Searcher.
- E) Gateway.

#### Comentários:

Conforme vimos, o browser (navegador) não resolve o nome! Supondo que uma URL seja solicitada através de um browser, o "resolvedor DNS" terá o papel de resolver o DNS (o próprio nome já diz isso). Pode ser através do arquivo hosts (máquina local) ou através de solicitação ao servidor DNS "mais próximo" (mais comum - geralmente o servidor DNS utilizado pelo provedor de acesso à Internet). Se o servidor DNS tiver a informação em cache, ele responde, senão faz uma busca nos servidores superiores na hierarquia DNS, atualiza sua cache e responde ao "resolvedor".

Gabarito: C

4. (FCC/PRODATER/2016) No Serviço de Nomes de Domínio – DNS existem diferentes tipos de servidores distribuídos hierarquicamente que armazenam informações também de forma hierárquica. Considerando o nome: www.empresa.com, o domínio .com é gerenciado pelo servidor

- A) Global.
- B) PDR.
- C) Authoritative.
- D) TLD.
- E) Root.

#### Comentários:

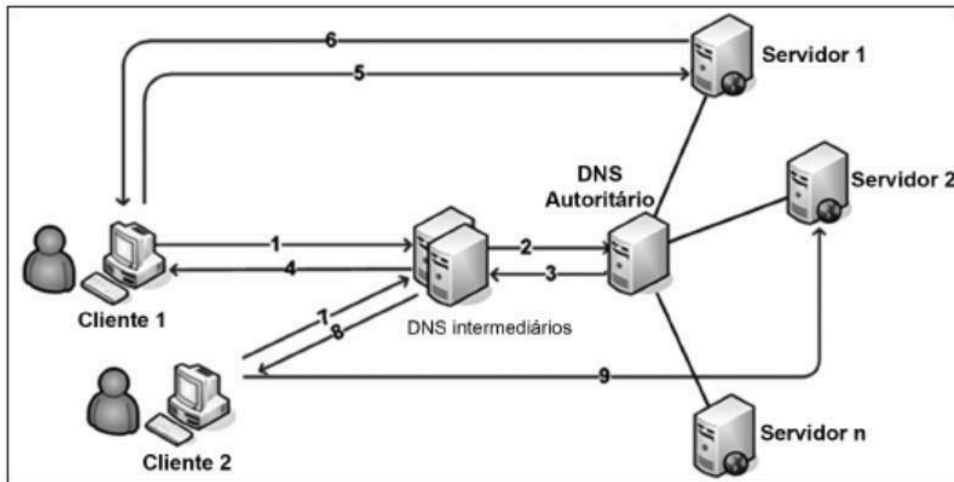
Os domínios de mais alto nível são classificados em genéricos (".com", ".edu", ".bradesco", etc.) ou códigos de países (".br", ".mx", ".uy", ".jp", etc.). Em inglês é conhecido como TLD (Top Level Domain) e os servidores DNS TLD gerenciam tais domínios. Abaixo podemos ver mais exemplos de TLDs genéricos e dois exemplos de códigos de países.

Gabarito: D

5. (FCC/ELETROSUL/2016) O balanceamento de carga por DNS é uma abordagem popular e simples para as solicitações de balanceamento de servidores. Considere a figura abaixo.







De acordo com a figura, geralmente são estes os passos que ocorrem assim que é feita uma consulta DNS:

1. Quando um cliente tenta acessar o site, é realizada uma pesquisa no DNS local para determinar qual é o endereço I..... correspondente;
2. O pedido de endereço chega ao servidor de DNS II..... do domínio;
3. A primeira vez que esta consulta é feita, o servidor DNS remoto pode retornar todos os registros de endereços que ele tem para o site;
4. O servidor DNS III....., em seguida, determina o endereço de registro para retornar ao cliente;
5. Se todos os registros são retornados, o cliente utilizará o primeiro que lhe é atribuído;
6. O servidor responde ao cliente e atende ao pedido;
7. A cada pedido, o algoritmo Round Robin roda os endereços e retorna pela ordem em que eles estão;
8. Cada consulta DNS irá resultar em um cliente usando um endereço IV.....;
9. Esta rotação de endereços irá distribuir pedidos para os servidores.

As lacunas de I a IV são, correta e respectivamente, preenchidas com:

- A) TCP – autoritário – local – TCP igual ao obtido em 1
- B) IP – autoritário – local – IP diferente
- C) do servidor – local – autoritário – de um servidor
- D) do servidor – autoritário – local – do servidor igual ao obtido em 1
- E) IP – local – autoritário – IP igual ao obtido em 1

Comentários:



Quando um cliente tenta acessar o site, é realizada uma pesquisa no DNS local para determinar qual é o endereço IP (DNS traduz nome em endereço IP) correspondente; O pedido de endereço chega ao servidor de DNS autoritário (após o servidor DNS local, chega ao servidor que possui os registros atualizados = autoritário) do domínio; O servidor DNS local (depois de retornar do servidor autoritário, o servidor local atualiza a cache e envia ao cliente), em seguida, determina o endereço de registro para retornar ao cliente; Cada consulta DNS irá resultar em um cliente usando um endereço IP diferente (essa é a ideia do algoritmo Round Robin, o balanceamento de carga).

Gabarito: B

6. (FCC/TRT-SP/2018) No shell do Linux, para ativar a interface de rede eth0 utiliza-se a instrução

- A) netstart eth0
- B) netstart --interface=eth0
- C) nslookup --interface=eth0
- D) ifconfig eth0 up
- E) ipconfig eth0 up

#### Comentários:

ifconfig: usado para exibir e configurar informações de interfaces de rede. Gradualmente está sendo substituído pelo comando ip, que oferece mais recursos.

Para exibir informações sobre todas as interfaces de rede ativas:

```
$ ifconfig
```

Para exibir informações detalhadas sobre determinada interface:

```
$ ifconfig eth0
```

Para ativar uma interface de rede:

```
$ sudo ifconfig eth0 up
```

Para desativar uma interface de rede:

```
$ sudo ifconfig eth0 down
```

Para configurar um endereço IP em uma interface:

```
$ sudo ifconfig eth0 192.168.1.150 netmask 255.255.255.0
```

Gabarito: D

7. (FCC/Prefeitura de Manaus-AM/2019) Um Assistente de TI foi incumbido de configurar o protocolo para ser utilizado na troca de mensagens entre dois servidores de e-mail. A escolha correta do protocolo para essa finalidade é:



- A) IMAP.
- B) POP3.
- C) SNMP.
- D) SMTP.
- E) POP4.

**Comentários:**

SMTP (Simple Mail Transfer Protocol): utilizado quando o e-mail é entregue a partir de um cliente de e-mail (Outlook, Thunderbird, entre outros) a um servidor de e-mail ou quando o e-mail é entregue a partir de um servidor de e-mail para outro servidor. Ou seja, se você enviar um e-mail a partir de um software cliente, ele utilizará o SMTP até o seu servidor e depois o seu servidor utilizará também o SMTP para entregar ao servidor destino, quando este ficará na caixa de e-mails do usuário destinatário. A porta padrão de um servidor SMTP é a 25.

Gabarito: D

8. (FCC/METRÔ-SP/2019) Um Analista que usa o Linux Red Hat, em condições ideais, editou um arquivo e colocou os comandos abaixo, para indicar os servidores que atuarão na resolução de nomes para o servidor que está configurando.

```
search localdomain nameserver 8.8.8.8 nameserver 8.8.4.4
```

Para editar o arquivo em questão, utilizou a instrução

- A) vi /etc/dns.conf
- B) nano /conf/dns.conf
- C) vi /etc/resolv.conf
- D) nano /bin/resolv.conf
- E) vi /lib/dnsconf.conf

**Comentários:**

O arquivo /etc/resolv.conf identifica os locais dos computadores de servidor DNS. Nesse arquivo deve ser indicado o nome de domínio DNS da rede, e qual o servidor DNS irá resolver as consultas de nomes. Um exemplo é mostrado a seguir.

```
domain ti.teste.com.br
search ti.teste.com.br teste.com.br
nameserver 192.168.1.1
nameserver 192.168.2.1
```

Tanto vi como nano são editores de texto, então tanto faz usar um ou o outro para editar o arquivo /etc/resolv.conf.



Gabarito: C

9. (FCC/TJ-MA/2019) O firewall do Linux Iptables é usado para monitorar entradas e saídas de tráfego de dados para o servidor. Em um computador com uma distribuição Linux em condições ideais, para aceitar pacotes do endereço 192.168.1.5, utiliza-se o comando:

- A) iptables -B INPUT -p 192.168.1.5 -j ALLOW
- B) iptables -A INPUT -s 192.168.1.5 -j ACCEPT
- C) iptables -C INPUT -d 192.168.1.5 -p ACCEPT
- D) iptables -A INPUT -d 192.168.1.5 -j ALLOW
- E) iptables -X FORWARD -i 192.168.1.5 -p ACCEPT

#### Comentários:

Para aceitar pacotes (vindos) do endereço 192.168.1.5, temos que adicionar (add = A) uma regra que tenha como origem (source = s) o endereço 192.168.1.5. E como ação (-j), o pacote deve ser aceito (ACCEPT).

Gabarito: B



## QUESTÕES COMENTADAS - VUNESP

1. (VUNESP/SAAE de Barretos-SP/2018) Os serviços de acesso a dados da internet são disponibilizados em uma arquitetura Cliente/Servidor. Por exemplo, considerando um servidor com sistema operacional Linux, o serviço padrão que responde às requisições ftp é denominado

- A) ftpd.
- B) ftps.
- C) ftpsrv.
- D) samba.
- E) smb.

### Comentários:

Seguindo a lógica adotada para servidores no Linux, o servidor FTP é executado através de um daemon, o ftpd (geralmente o daemon tem o nome do serviço, acrescido de um "d" no fim).

Gabarito: A

2. (VUNESP/Câm. de Itaq.-SP/2018) No sistema operacional Linux, o arquivo /etc/hosts é utilizado para

- A) listar os IPs com acesso permitido ao sistema.
- B) listar os servidores DNS externos em ordem de prioridade.
- C) mapear um nome (domínio) para um endereço IP.
- D) relacionar os IPs dos computadores da mesma rede local.
- E) relacionar os IPs dos roteadores vinculados.

### Comentários:

Um arquivo útil para máquinas que são acessadas frequentemente: o /etc/hosts. A inclusão de um computador nesse arquivo dispensa a consulta a um servidor DNS para obter um endereço IP. Ou seja, a tradução ocorre automaticamente ao ler tal arquivo, sem ter que consultar o serviço de DNS. Um exemplo é mostrado abaixo.

# Aqui são colocados comentários		
# End. IP	Nome do host	Alias
(opcional)		
127.0.0.1	localhost	localmente
192.168.0.1	www.teste.com.br	teste

Gabarito: C



3. (VUNESP/Câm. de M. Alto-SP/2019) Em um computador com sistema operacional Linux e iptables instalado, a apresentação de todas as regras atuais que estão configuradas pode ser realizada utilizando o comando iptables com a opção:

- A) -A
- B) -C
- C) -F
- D) -L
- E) -P

#### Comentários:

Vamos ver alguns exemplos de iptables:

\$ iptables -A INPUT -i eth0 -j ACCEPT → Adiciona a regra para aceitar pacotes que chegam pela interface eth0.

\$ iptables -A INPUT -p tcp --dport 80 -j DROP → Adiciona a regra que descarta pacotes TCP que chegam, tendo como porta destino a 80.

\$ iptables -D OUTPUT -p udp --dport 53 -j REJECT → Exclui a regra que rejeita pacotes UDP (com aviso à origem) originados no dispositivo, com porta destino 53.

\$ iptables -L → Lista todas as regras atuais configuradas.

\$ iptables -F → Limpa todas as regras atuais configuradas.

Gabarito: D

4. (VUNESP/UNICAMP/2022) Após a instalação do sistema operacional Linux em um computador, é desejável verificar as portas TCP e UDP em estado listening por padrão, podendo-se, então, desinstalar ou desativar os aplicativos correspondentes àquelas portas que não se pretende utilizar. Dentre as alternativas a seguir, o comando do Linux que permite fazer essa verificação (utilizando os devidos parâmetros) é:

- A) ifconfig
- B) df
- C) ps
- D) nslookup
- E) netstat

#### Comentários:

netstat: usado para exibir informações sobre as conexões de rede, tabelas de roteamento, estatísticas de interface de rede etc. Útil para diagnosticar problemas de rede, monitorar o tráfego e obter informações sobre as portas e protocolos em uso.

Para exibir informações detalhadas sobre todas as conexões e sockets:



```
$ netstat -a
```

Para listar todas as conexões de rede ativas (endereços IP e portas):

```
$ netstat -tuln
```

- -t: mostra as conexões TCP;
- -u: mostra as conexões UDP;
- -l: mostra as portas em escuta (listen);
- -n: exibe endereços IP e números de porta em formato numérico (evita a resolução de nomes).

Para exibir estatísticas de interface de rede:

```
$ netstat -i
```

Para exibir tabelas de roteamento IP:

```
$ netstat -r
```

Gabarito: E



## QUESTÕES COMENTADAS - MULTIBANCAS

1. (FUNIVERSA/PC-DF/2012) Em um servidor de e-mail com sistema operacional Linux, especialmente quando se utiliza o serviço Postfix, pode-se auditar o tráfego de entrada e saída de mensagens de correio eletrônico por meio dos registros conhecidos por logs. Em que arquivo desse sistema operacional encontram-se os logs das atividades de envio e recebimento de mensagens de correio eletrônico (e-mail)?

- A) No diretório "/etc/postfix".
- B) No arquivo "/var/log/maillog".
- C) Na pasta "/var/mail/<nome\_do\_usuario>".
- D) No arquivo "/usr/local/bin/postfix".
- E) No arquivo de registro "/dev/tty".

### Comentários:

Uma característica importante no Linux são os logs, que, na sua maioria, são armazenados em /var/log. Esse é o caso também dos logs relacionados aos servidores de e-mail, útil quando são necessárias informações sobre postfix, smtpd ou qualquer serviço relacionado ao e-mail que esteja sendo executado no servidor. Por padrão, esses logs ficam em /var/log/maillog ou /var/log/mail.log.

Gabarito: B

2. (FUNCAB/MDA/2014) São exemplos de servidores de correio eletrônico do Linux:

- A) postfix e sendmail.
- B) fetchmail e sendmail.
- C) pine e postfixe.
- D) mail e pine.
- E) samba e postfixe.

### Comentários:

Alguns exemplos de servidores de e-mail no Linux: qmail, postfix e sendmail.

Gabarito: A

3. (AOC/UFMG/2014) Em algumas distribuições Linux encontramos um arquivo que identifica os locais dos computadores de servidor de nome DNS, computadores estes que são usados pelo TCP/IP para traduzir os nomes de Internet. Assinale a alternativa que apresenta o nome deste arquivo.

- A) DNS.conf.





- B) resolver.conf.
- C) resolv.conf.
- D) translate.conf.
- E) change.conf.

**Comentários:**

O arquivo resolv.conf, que fica dentro do /etc, é o arquivo com as configurações para resolver nomes. Note que o nome é sugestivo: resolv.conf.

Gabarito: C

4. (IADES/TRE-PA/2014) Um servidor Linux pode hospedar o serviço de resolução de nomes de uma rede de computadores. Conhecido por DNS, esse serviço é indispensável em uma rede que possua conexão com a internet. O nome de um pacote que implementa o DNS, muito utilizado em sistemas operacionais Linux, é

- A) Firefox.
- B) Apache.
- C) Squid.
- D) Postfix.
- E) BIND.

**Comentários:**

O pacote é conhecido como BIND (versão 8, ou bind9 - versão 9).

Gabarito: E

5. (INSTITUTO AOCP/UFPB/2014) Um estagiário executou o comando FTP no servidor Linux. Esse comando possibilita

- A) permitir a abertura de portas para um acesso remoto a um servidor da rede.
- B) realizar a manutenção de disco e de memória de um computador remotamente.
- C) trocar informações de desempenho de redes utilizando um canal de comunicação protegido e exclusivo.
- D) realizar uma verificação de permissões de acesso na rede e no computador que o usuário está utilizando.
- E) trocar arquivos entre computadores interligados, independente do sistema operacional e da rede utilizada.

**Comentários:**

O comando ftp, tanto no Linux como no Windows, chama o cliente FTP, que permite trocar arquivos entre computadores interligados, independente do sistema operacional e da rede



utilizada. Por exemplo, um cliente FTP Linux pode se conectar a um servidor FTP Windows e tudo irá funcionar tranquilamente.

Gabarito: E

6. (IADES/CRC-MG/2015) Quanto ao programa que é normalmente utilizado para testar se um servidor DNS (Domain Name Server) está funcionando corretamente, ou seja, resolvendo nomes para os endereços IP, assinale a alternativa correta.

- A) ping
- B) tracert
- C) nslookup
- D) ipconfig
- E) net host

Comentários:

(A) Serve para ver se "está vivo"; (B) Serve para traçar uma rota (roteadores intermediários); (C) Exatamente! ns = name server, lookup = "dar uma olhada", se você nunca usou, experimente agora no prompt: nslookup estrategiaconcursos.com.br; (D) verifica configurações das interfaces de rede; (E) o comando net não possui o parâmetro host!

Gabarito: C

7. (IBFC/EBSERH-HUAP/2016) O Domain Name System (DNS) é um sistema de gerenciamento de nomes hierárquicos e distribuídos. A funcionalidade do DNS Reverso vem a ser:

- A) reverter a funcionalidade básica de um DNS padrão obtendo o endereço MAC.
- B) resolver um endereço IP, buscando o nome de domínio associado ao host.
- C) resolver o nome do domínio de um host qualquer para seu endereço IP correspondente.
- D) com base na região geográfica, obter automaticamente um endereço IP local.
- E) resolver o problema da reversão do IPv6 para o IPv4 de uma forma rápida e automática.

Comentários:

A questão começa com a definição de DNS, segundo o Tanenbaum. O DNS tem a função de traduzir um nome em um endereço, então o DNS reverso faz o contrário: traduz um endereço em um nome. É definido através do registro PTR:

PTR    Ponteiro (usado para o DNS reverso)    Nome alternativo de um end. IP.

Gabarito: B

8. (FUNRIO/CM Nova Iguaçu/2016) Um administrador de rede está configurando o DNS de um servidor e vai trabalhar com o registro que define as características da zona a ser configurada, tais como, o nome da zona e o nome do servidor, que é a autoridade para a



referida zona, ou seja, o servidor DNS no qual está a zona que foi criada originalmente. Esse registro é conhecido pela sigla

- A) HINFO.
- B) MX.
- C) SOA.
- D) CNAME.

#### Comentários:

Mais uma vez aquela tabela, agora copiei apenas a linha sobre o "SOA":

SOA Início de autoridade (Start of Authority). Parâmetros para essa zona.

Ou seja, se falar sobre características/parâmetros de uma zona, trata-se do registro Start of Authority (SOA).

Gabarito: C

9. (IBFC/Polícia Científica-PR/2017) Servidores de DNS (Domain Name System) têm como função converter endereços IP em seu respectivo nome e vice-versa. Para sua configuração, são utilizados arquivos denominados mapas de domínio (zone). Esses arquivos são compostos por entradas chamadas RR (Resource Record). O tipo básico de RR que estabelece a correspondência entre um nome canônico e um endereço IP é indicado por:

- A) PTR
- B) MX
- C) NS
- D) A
- E) SOA

#### Comentários:

Quando é realizada uma consulta pelo nome, esperando como resposta um endereço IP, os RRs possíveis são:

A	Endereço IPv4.	Inteiro de 32 bits.
AAAA	Endereço IPv6.	Inteiro de 128 bits.

Como a questão não mostra "AAAA" como alternativa, sobrou apenas o "A", que retorna um endereço IPv4.

Gabarito: D

10. (IBFC/Polícia Científica-PR/2017) Servidores DNS (Domain Name Server) são responsáveis pela conversão do nome dos diversos servidores espalhados pela Internet para seu número IP e vice-versa. Servidores de DNS trabalham de forma colaborativa e hierárquica. Assinale a



alternativa a que apresenta o nome dado aos servidores que se encontram no topo da hierarquia de DNS:

- A) Root Name Servers
- B) Main servers
- C) International Name Servers
- D) Controllers Servers
- E) Master Servers

#### Comentários:

A hierarquia DNS é uma árvore invertida, ou seja, os servidores raiz (Root Name Servers) ficam no topo, logo abaixo ficam os TLDs (Top Level Domains), depois os de segundo nível e assim por diante.

Gabarito: A

11. (IF-RS/IF-RS/2018) Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?

- A) dns 8.8.8.8
- B) echo 8.8.8.8 > /etc/hosts.conf
- C) echo nameserver 8.8.8.8 > /etc/resolv.conf
- D) nslookup 8.8.8.8
- E) nameserver 8.8.8.8

#### Comentários:

nameserver: Endereço IP do servidor de nomes que o resolvedor deve pesquisar, utilizados na ordem em que estão listados. Tal diretiva deve estar no arquivo /etc/resolv.conf, então com o comando echo é possível escrever "nameserver 8.8.8.8" em "/etc/resolv.conf" sem ter que abrir o arquivo.

Gabarito: C

12. (FAURGS/BANRISUL/2018) Um cliente DNS, ao fazer a requisição DNS para resolver o nome www.banrisul.com.br, recebe como resposta uma mensagem do tipo non authoritative. Esse tipo de resposta é obtido por meio de um registro DNS (resource record - RR) armazenado

- A) em um cache de DNS.
- B) em um servidor interativo sem cache DNS.
- C) em um servidor recursivo sem cache DNS.
- D) no arquivo de zona do DNS primário.



E) no arquivo de zona do DNS secundário.

#### Comentários:

Um servidor DNS autoritativo possui autoridade sobre um nome de domínio. O DNS autoritativo dita qual será o apontamento da tabela de DNS do seu sítio. Uma resposta autoritativa de um servidor é a garantia de estar atualizada, enquanto uma resposta não-autoritativa pode estar desatualizada (cache com informações antigas, por exemplo). Existe um percentual elevado de respostas não autoritativas que estão perfeitamente corretas, casos em que mudanças de endereçamento são raros.

Servidores primários e secundários são autoritativos para os seus domínios, porém não o são sobre informações a respeito de outros domínios mantidas em cache. Servidores caching-only nunca são autoritativos, mas possuem a vantagem de reduzir a quantidade de tráfego DNS na rede.

Gabarito: A

13. (FAPEC/UFMS/2018) Considere as afirmações a seguir sobre a instalação e configuração de um servidor de DNS (Bind9) no sistema operacional Debian 9.

I - Quando o servidor local de DNS não consegue traduzir o nome, ele deve solicitar a um servidor público (forwarders). O Google Public DNS é um exemplo de forwarders e deve ser configurado no arquivo `named.conf.options`.

II - A instalação do servidor pode ser realizada pelo usuário root com o comando: `apt install bind9 bind9-doc dnsutils`

III - A lista de controle de acessos ou ACL é definida no arquivo `named.conf.options`.

Está(ão) correta(s):

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas II e III.
- E) I, II e III.

#### Comentários:

Todas as afirmativas foram abordadas no decorrer da aula.

Gabarito: E

14. (SUGEP-UFRPE/UFRPE/2018) Para acesso à Internet, servidores Windows e Linux utilizam um serviço para atribuição dinâmica de IPs aos microcomputadores conectados à rede, em que é marcada a opção associada a "o IP será atribuído automaticamente pelo servidor", na configuração.



Esse serviço é conhecido pela sigla:

- A) DNS
- B) HTTP
- C) DHCP
- D) TCP
- E) IP

Comentários:

O DHCP (Dynamic Host Configuration Protocol) é um protocolo que tem a função de configurar os endereços IP dos computadores de uma rede de forma dinâmica. Ou seja, deve haver pelo menos um servidor DHCP pré-configurado para receber solicitações de computadores que não possuem endereço IP (clientes), o servidor verifica qual endereço IP disponível e envia ao solicitante.

Gabarito: C

15. (IBADE/IPM-JP/2018) Arquivo de configuração do sistema operacional Linux que contém uma relação entre endereços IP e nomes, que pode ser utilizado para acessar computadores por um nome sem utilizar um servidor de nomes:

- A) /etc/names
- B) /etc/ips
- C) /etc/mapping
- D) /etc/hosts
- E) /etc/list

Comentários:

Um arquivo útil para máquinas que são acessadas frequentemente: o /etc/hosts. A inclusão de um computador nesse arquivo dispensa a consulta a um servidor DNS para obter um endereço IP. Ou seja, a tradução ocorre automaticamente ao ler tal arquivo, sem ter que consultar o serviço de DNS. Um exemplo é mostrado abaixo.

# Aqui são colocados comentários		
# End. IP	Nome do host	Alias (opcional)
127.0.0.1	localhost	localmente
192.168.0.1	www.teste.com.br	teste

Gabarito: D

16. (AOCP/ITEP-RN/2018) Na falta de um serviço de nomes em uma rede local, qual dos arquivos a seguir é utilizado para fazer a tradução de nomes de computadores para endereços IP em um computador Linux?



- A) /etc/hosts.allow
- B) /etc/hosts
- C) /etc/hostname
- D) /etc/host.conf
- E) /usr/local/hostname

#### Comentários:

Um arquivo útil para máquinas que são acessadas frequentemente: o /etc/hosts. A inclusão de um computador nesse arquivo dispensa a consulta a um servidor DNS para obter um endereço IP. Ou seja, a tradução ocorre automaticamente ao ler tal arquivo, sem ter que consultar o serviço de DNS. Um exemplo é mostrado abaixo.

# Aqui são colocados comentários		
# End. IP	Nome do host	Alias (opcional)
127.0.0.1	localhost	localmente
192.168.0.1	www.teste.com.br	teste

Gabarito: B

17. (INAZ do Pará/CREFITO-MA/2018) As configurações de aplicativos e do próprio sistema operacional Linux podem ser realizadas por meio da manipulação dos arquivos de configuração através de um editor de texto simples. Caso o usuário deseje ingressar em um grupo doméstico ou domínio, qual dos arquivos abaixo ele deve editar para efetuar essa operação?

- A) Dhcp.conf
- B) Samba.conf
- C) Squid.conf
- D) Lilo.conf
- E) Access.conf

#### Comentários:

O SAMBA é um software de código aberto que implementa o protocolo SMB/CIFS (Server Message Block/Common Internet File System). O SMB/CIFS é usado para compartilhar arquivos e recursos em redes locais, permitindo a interoperabilidade entre sistemas Windows, Linux/Unix e outros. Também é realizado o compartilhamento de impressão.

smbd: um dos principais processos do Samba.

etc/samba/**smb.conf**: arquivo de configuração principal.

Podemos ver que a banca se equivocou no nome, mas colocou parecido (Samba.conf).

Gabarito: B



18. (CONSULPLAN/Câm. de B.H.-MG/2018) O programa iptables fornece uma interface para que o usuário possa manipular o filtro de pacotes do kernel, ou seja, permite a administração de tabelas no firewall do kernel Linux. A sintaxe do iptables é a seguinte: iptables [ -t tabela ] <comando>[opção<parâmetro>]<destino>. Os comandos permitem que tarefas sejam executadas com o iptables. Um desses comandos é o [-D]. Assinale a alternativa que apresenta a descrição correta desse comando.

- A) Apaga uma regra.
- B) Apaga todas as regras.
- C) Estabelece a observação de pacotes.
- D) Adiciona uma regra (chave + opção + destino).

#### Comentários:

O iptables é usado para configurar regras de firewall e controlar o tráfego de rede. Ele permite a definição de políticas de segurança, regras de encaminhamento de pacotes, o mapeamento de portas (NAT) etc.

Cadeias (chains):

- INPUT: pacotes destinados ao próprio sistema (chegando);
- OUTPUT: pacotes gerados pelo próprio sistema (saindo);
- FORWARD: pacotes que são roteados pelo sistema (não destinados nem gerados pelo sistema).

Regras (executa o primeiro "match", ou seja, se deu "match", não executa o que vem depois):

- -A: adiciona uma regra;
- -D: "deleta" (apaga) uma regra;
- -j: ação a ser tomada (ACCEPT, DROP, REJECT etc.);
- -s: endereço IP origem (source);
- -d: endereço IP destino (destination);
- -i: interface de entrada (input);
- -o: interface de saída (output);
- -p: protocolo;
- --dport: porta destino;
- --sport: porta origem;
- etc.

Gabarito: A

19. (IBADE/IPM-JP/2018) O Sistema Operacional Linux oferece um conjunto de comandos que podem ser utilizados para realizar a administração de redes. O comando que mostra o caminho percorrido por um pacote até chegar a seu destino é:

- A) traceroute.
- B) netpoints.
- C) traceway.





- D) routeway.
- E) identification.

#### Comentários:

traceroute: usado para rastrear a rota que os pacotes seguem para alcançar um destino. Mostra uma lista de saltos (hops) que os pacotes fazem, além dos tempos de resposta de cada salto. É útil para diagnosticar problemas de roteamento e identificar os nós intermediários (roteadores) pelos quais os pacotes de dados passam ao viajar da origem para o destino. A sintaxe é idêntica ao ping:

```
$ traceroute www.estrategiaconcursos.com.br
```

```
$ traceroute 192.168.1.150
```

Importante lembrar que no Windows a ferramenta que tem o mesmo propósito é o tracert. Cuidado com as pegadinhas!!! Nomes semelhantes, mas são diferentes!

Gabarito: A

20. (CCV-UFC/UFC/2019) Uma das formas de controlar o acesso de rede em um dispositivo utilizando o sistema operacional Linux é através do iptables. Qual dos itens abaixo contém a configuração para não permitir que o comando ping, executado em um dispositivo com o sistema operacional Linux, consiga enviar requisições para outros dispositivos?

- A) iptables -A INPUT -p icmp -j DROP
- B) iptables -A OUTPUT -p icmp --icmp-type 0 -j DROP
- C) iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
- D) iptables -D INPUT -p icmp --icmp-type echo-request -j DROP
- E) iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP

#### Comentários:

Esta questão exige um pouco de conhecimento de redes de computadores. O comando ping utiliza mensagens do tipo "echo-request" do protocolo ICMP (Internet Control Message Protocol). Sabendo disso, vamos por partes:

- "não permitir" - deve-se rejeitar o pacote, ou seja: -j DROP (ou REJECT, se quisesse um retorno, mas nenhuma alternativa tem o REJECT);
- "comando ping" - utiliza mensagens "echo-request" do protocolo ICMP, ou seja: -p icmp --icmp-type echo-request;
- "consiga enviar requisições" - adiciona a regra à chain OUTPUT, ou seja: -A OUTPUT.

Mesmo que você não soubesse o detalhe do ICMP, tipo echo-request, já poderia eliminar as três alternativas com a chain INPUT, o que facilitaria muito a decisão final.

Gabarito: E



21. (COVEST-COPSET/UFPE/2019) No Windows e no Linux, qual utilitário de linha de comando é usado para exibir o caminho que um pacote faz até o seu destino?

- A) Netpath, no Windows e no Linux.
- B) Traceroute, no Windows e no Linux
- C) Tracert, no Windows e no Linux.
- D) Netpath, no Windows; e Tracert, no Linux.
- E) Tracert, no Windows; e Traceroute, no Linux.

#### Comentários:

traceroute: usado para rastrear a rota que os pacotes seguem para alcançar um destino. Mostra uma lista de saltos (hops) que os pacotes fazem, além dos tempos de resposta de cada salto. É útil para diagnosticar problemas de roteamento e identificar os nós intermediários (roteadores) pelos quais os pacotes de dados passam ao viajar da origem para o destino. A sintaxe é idêntica ao ping:

```
$ traceroute www.estrategiaconcursos.com.br
```

```
$ traceroute 192.168.1.150
```

Importante lembrar que no Windows a ferramenta que tem o mesmo propósito é o tracert. Cuidado com as pegadinhas!!! Nomes semelhantes, mas são diferentes!

Gabarito: E

22. (IBFC/MGS/2019) Um recurso técnico que permite, ao ser instalado no Linux, simular um servidor Windows para permitir o gerenciamento e compartilhamento de arquivos e de impressoras em uma rede Microsoft é o programa chamado:

- A) NetBIOS
- B) SAMBA
- C) OPERA
- D) NetBEUI

#### Comentários:

O SAMBA é um software de código aberto que implementa o protocolo SMB/CIFS (Server Message Block/Common Internet File System). O SMB/CIFS é usado para compartilhar arquivos e recursos em redes locais, permitindo a interoperabilidade entre sistemas Windows, Linux/Unix e outros. Também é realizado o compartilhamento de impressão.

smbd: um dos principais processos do Samba.

etc/samba/smb.conf: arquivo de configuração principal.

Gabarito: B



23. (Quadrix/CRO-GO/2019) Julgue o item, relativo à administração de sistemas Linux.

Os comandos `ifup` e `ifdown` são destinados, respectivamente, a ativar e a desativar uma interface de rede especificada.

#### Comentários:

`ifup`: usado para ativar (subir) uma interface de rede de acordo com o que estiver definido nos arquivos de configuração. Por exemplo, se houver uma interface chamada `eth0` e as configurações estiverem definidas em `/etc/network/interfaces`, pode-se usar:

```
$ sudo ifup eth0
```

`ifdown`: usado para desativar (baixar) uma interface de rede

```
$ sudo ifdown eth0
```

Gabarito: Certo

24. (AOCP/IBGE/2019) Lucas está administrando remotamente um servidor GNU/Linux através de Shell Seguro (SSH). Ele inseriu duas regras no IPTABLES na ordem apresentada a seguir. Supondo que Lucas tenha plena certeza do efeito dessas regras após a inclusão delas nessa ordem, qual das seguintes alternativas explica corretamente o que ele espera que ocorra com as conexões desse servidor?

```
Regra 1: iptables -A INPUT -p tcp -s 10.0.0.0/8 --dport 22 -j ACCEPT
Regra 2: iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 22 -j DROP
```

A) Que, após a inclusão da Regra 2, independentemente da rede de origem da sua conexão via SSH, a conexão seja interrompida imediatamente.

B) Que, após a inclusão da Regra 2, independentemente da origem, qualquer tentativa de conexão com esse servidor através da porta 22 seja descartada.

C) Que, após a inclusão da Regra 2, o servidor só aceite conexões na porta 22 a partir de hosts da rede 10.0.0.0 com máscara 255.0.0.0.

D) Que, após a inclusão da Regra 1, qualquer host que não tenha IP 10.x.x.x tenha sua conexão recusada na porta 22.

E) Que, após a inclusão da Regra 2, o servidor continue permitindo conexões na porta 22 com origem em qualquer rede, uma vez que as regras não foram aplicadas com o comando `'iptables -commit'`.

#### Comentários:

A ordem das regras é importante, então primeiro é realizado o "teste" para saber se a 1ª regra é satisfeita, caso contrário verifica se a 2ª é satisfeita e por aí vai. Quando uma regra é satisfeita, a verificação das regras seguintes não é realizada.

Sabendo disso, vamos analisar a primeira: foi adicionada (-A) uma regra para a chain INPUT (pacotes chegando), protocolo TCP (-p tcp), rede origem (-s) 10.0.0.0/8, porta destino (--dport) 22 (serviço SSH), ação (-j) ACCEPT (aceita, "deixa passar").



Se não se enquadrar na regra 1, então vai para a 2: foi adicionada (-A) uma regra para a chain INPUT (pacotes chegando), protocolo TCP (-p tcp), qualquer rede origem (-s) 0.0.0.0/0, porta destino (--dport) 22 (serviço SSH), ação (-j) DROP (descarta).

Ou seja, se não se encaixar na regra 1, então descarta o pacote!

Gabarito: C

25. (IF-PE/IF-PE/2019) Observe, a seguir, os itens referentes ao servidor de DNS Berkeley Internet Name Domain (BIND) do Linux Ubuntu 18.04.

I. Digitando o comando `# apt-get install bind` no terminal, a instalação do bind é iniciada.

II. O arquivo de configuração do "bind", no Linux, é o `named.conf`, que se encontra na pasta `/etc`.

III. Para que o servidor externo receba solicitações não atendidas pelo DNS, devemos alterar o arquivo `named.conf.options` que se encontra na pasta `/etc/bind`.

IV. Para a verificação do funcionamento do servidor DNS previamente configurado, basta abrir o terminal do DNS server e dar um "ping" para algum site, como: `# ping www.ifpe.edu.br`.

São verdadeiras, apenas, as proposições

A) I e II.

B) II, III.

C) III e IV.

D) I e IV.

E) II, III e IV.

Comentários:

I. ERRADA - O correto seria `bind9`.

II. ERRADA - O arquivo correto seria o `resolv.conf`.

III. CORRETA - Servidores DNS públicos (forwarders), como por exemplo o Google Public DNS, podem ser configurados no arquivo `named.conf.options`. Nesse arquivo também pode ser definida a lista de controle de acessos (ACL).

IV. CORRETA - Se o DNS estiver funcionando, vai resolver o nome "www.ifpe.edu.br" para o endereço IP correspondente.

Gabarito: C

26. (UFU-MG/UFU-MG/2020) Considerando-se o sistema operacional Linux Ubuntu e os diversos serviços que podem ser instalados nesse sistema, assinale a alternativa INCORRETA.



- A) O serviço Samba é uma implementação do protocolo SMB/CIFS e provê suporte para compartilhamento de arquivos e de impressoras com o sistema operacional Microsoft Windows.
- B) A ferramenta de configuração de firewall “ufw” permite filtrar pacotes de rede, liberando ou bloqueando serviços específicos de modo a proteger o computador de acessos indesejados, por exemplo.
- C) NFS (Network File System) permite compartilhar uma pasta de um computador com acesso à rede com os demais computadores na mesma rede com a restrição de que tanto o servidor quanto os clientes devem usar um sistema operacional Linux.
- D) LDAP é um protocolo de rede para acessar informação armazenada em um serviço de diretório com aplicações como single sign-on, autenticação em aplicações web e de usuários locais.

#### Comentários:

O NFS é mais comumente associado a sistemas operacionais baseados em Unix-like. No entanto, para integração com sistemas Windows, existem implementações do NFS para Windows.

Gabarito: C

27. (CEFET-MG/CEFET-MG/2021) Um arquivo de configuração de rede no Debian GNU/Linux 10 é:

- A) /etc/network.conf
- B) /etc/interfaces.conf
- C) /etc/interfaces/network
- D) /etc/network/interfaces
- E) /etc/network-interfaces

#### Comentários:

Vimos na explicação do comando ifup, que é usado para ativar (subir) uma interface de rede de acordo com o que estiver definido nos arquivos de configuração. Por exemplo, se houver uma interface chamada eth0 e as configurações estiverem definidas em **/etc/network/interfaces**, pode-se usar:

```
$ sudo ifup eth0
```

As demais alternativas não existem.

Gabarito: D

28. (Quadrix/CRC-AP/2021) Em sistemas operacionais Linux derivados da edição Debian, o comando utilizado para fazer uma varredura ou busca por vulnerabilidades em um computador da mesma rede é o comando

- A) vuln.
- B) netstat.



- C) nmap.
- D) arp.
- E) netcraft.

**Comentários:**

nmap: ferramenta usada para descobrir e mapear redes, além de verificar a segurança de sistemas e serviços de rede. Amplamente utilizada para a varredura de portas, detecção de sistemas operacionais e análise de serviços em uma rede.

Para escanear um host específico para descobrir as portas abertas:

```
$ nmap 192.168.1.150
```

Para escanear um intervalo de endereços IP:

```
$ nmap 192.168.1.100-150
```

Para escanear todas as portas em um host (1 a 65535):

```
$ nmap -p1-65535 192.168.1.150
```

Gabarito: C

29. (UFRJ/UFRJ/2021) Um dos modos de se especificar um endereço IP estático para uma interface de rede em um ambiente Linux, como no caso do Debian Linux, é alterando o arquivo:

- A) /etc/network/interfaces
- B) /etc/interfaces/links
- C) /etc/link/interfaces
- D) /etc/interfaces/ipconf
- E) /etc/conf/interfaces

**Comentários:**

Vimos na explicação do comando ifup, que é usado para ativar (subir) uma interface de rede de acordo com o que estiver definido nos arquivos de configuração. Por exemplo, se houver uma interface chamada eth0 e as configurações estiverem definidas em **/etc/network/interfaces**, pode-se usar:

```
$ sudo ifup eth0
```

No arquivo pode-se configurar endereço IP estático ou dinâmico (cliente DHCP).

Gabarito: A

30. (Consulplan/SEFAZ-PI/2021) Utilizando o Sistema Operacional Linux é possível configurar um servidor de rede; e, dessa forma, configurar um servidor DHCP (Dynamic Host Configuration Protocol / Protocolo de Configuração de Host Dinâmico) de modo que aos computadores da rede sejam atribuídos endereços IP (Internet Protocol / Protocolo de Internet) automaticamente.



Um dos arquivos de configuração DHCP no Linux armazena, automaticamente, as informações de aluguel DHCP atribuídas a todo endereço IP concedido a cada cliente. Assinale-o.

- A) dhcpd.conf
- B) dhcpd.leases
- C) dhcpd.option
- D) dhcpd.server

**Comentários:**

O arquivo **dhcpd.leases** é utilizado pelo servidor DHCP para armazenar informações sobre os endereços IP atribuídos a clientes DHCP, o tempo de concessão, entre outras informações. Este arquivo geralmente é mantido pelo servidor DHCP para rastrear os detalhes das concessões de endereços IP.

Gabarito: B

31. (Quadrix/PRODAM-AM/2022) No sistema operacional Linux, para configurar o servidor dhcpd, o administrador deverá editar o arquivo

- A) access.conf.
- B) limits.conf.
- C) host.conf.
- D) inetd.conf.
- E) dhcpd.conf.

**Comentários:**

O arquivo dhcpd.conf contém as configurações. Esse arquivo pode se parecer com:

```
option domain-name-servers 8.8.8.8, 8.8.4.4;  
option subnet-mask 255.255.255.0;  
option routers 139.96.30.100;  
subnet 139.96.30.0 netmask 255.255.255.0 {  
    range 139.96.30.150 139.96.30.250;  
}
```

Gabarito: E

32. (CEFET-MG/CEFET-MG/2022) O caminho e o arquivo para configuração de IP fixo no Linux é

- A) /etc/init.d/network
- B) /etc/network/interfaces





- C) /etc/networking/interfaces
- D) /etc/network/interfaces.conf
- E) /etc/network/interface/interfaces.conf

**Comentários:**

Vimos na explicação do comando ifup, que é usado para ativar (subir) uma interface de rede de acordo com o que estiver definido nos arquivos de configuração. Por exemplo, se houver uma interface chamada eth0 e as configurações estiverem definidas em /etc/network/interfaces, pode-se usar:

```
$ sudo ifup eth0
```

No arquivo pode-se configurar endereço IP estático ou dinâmico (cliente DHCP).

Gabarito: B

33. (Consulplan/IF-PA/2023) O arquivo de configuração de rede no Linux é utilizado pelos programas ifup e ifdown para configurar as interfaces de rede do sistema. No Ubuntu, uma das distribuições Linux, este arquivo é:

- A) /etc/hosts
- B) /etc/resolv.conf
- C) /etc/networks.conf
- D) /etc/network/interfaces

**Comentários:**

Vimos na explicação do comando ifup, que é usado para ativar (subir) uma interface de rede de acordo com o que estiver definido nos arquivos de configuração. Por exemplo, se houver uma interface chamada eth0 e as configurações estiverem definidas em /etc/network/interfaces, pode-se usar:

```
$ sudo ifup eth0
```

O mesmo serve para o comando ifdown.

Gabarito: D

34. (CS-UFG/UFG/2023) O Sistema Operacional Linux possui diversos arquivos relacionados à configuração de redes de computadores, dos quais, podemos citar:

- A) /etc/network/dns.conf e /etc/hosts/network.
- B) /etc/hosts e /etc/interface/host.
- C) /etc/network/interfaces e /etc/resolv.conf.
- D) /etc/sysconfig/interfaces e /etc/interfaces/network.

**Comentários:**





/etc/network/interfaces: configurações das interfaces de rede.

/etc/resolv.conf: identifica os locais dos computadores de servidor DNS. Nesse arquivo deve ser indicado o nome de domínio DNS da rede, e qual o servidor DNS irá resolver as consultas de nomes. Um exemplo é mostrado a seguir.

```
domain ti.teste.com.br
search ti.teste.com.br teste.com.br
nameserver 192.168.1.1
nameserver 192.168.2.1
```

Gabarito: C

35. (Quadrix/CRM-MG/2023) Assinale a alternativa que apresenta o comando do sistema operacional Red Hat Enterprise Linux que suspende o compartilhamento de arquivos NFS, enquanto mantém todos os serviços NFS ativos.

- A) exportfs -r
- B) ls -la
- C) chmod 750
- D) exportfs -ua
- E) grep | more

#### Comentários:

exportfs: ferramenta usada para gerenciar e listar os sistemas de arquivos compartilhados para acesso via NFS. Os parâmetros mais conhecidos são:

- -r: usado para reexportar (reexportar recursivamente) todos os compartilhamentos definidos no arquivo /etc/exports;
- -ua: usado para desativar (unexport) todos (all) os compartilhamentos NFS exportados pelo servidor. Isso quer dizer que ele revoga temporariamente todas as configurações de exportação definidas no arquivo /etc/exports, tornando os compartilhamentos indisponíveis até que sejam reexportados.

Gabarito: D

36. (FURB/Pref. de J. do Sul-SC/2023) No Linux, o IPTABLES pode ser configurado para bloquear o acesso a sites indesejados na Internet. Um técnico em informática pode, então, definir políticas de restrição de acesso e criar regras de acesso à Internet da Empresa. A respeito da sintaxe do comando, assinale a alternativa correta que configura um bloqueio de um endereço IP específico na Internet:

- A) # iptables -A OUTPUT -p tcp -d 157.240.226.174 -j DROP
- B) # iptables -A INPUT -p tcp -d 157.240.226.2174 -j DROP
- C) # iptables -A OUTPUT -p tcp -d 157.240.226.174 -j BLOCK



- D) # iptables -A INPUT -p tcp -d 157.240.226.174 -j BLOCK  
E) # iptables -D OUTPUT -p tcp -d 157.240.226.174 -j BLOCK

#### Comentários:

Pegando essas partes do enunciado: "acesso à Internet da Empresa", "bloqueio de um endereço IP específico na Internet", podemos deduzir que o bloqueio deve ser de saída, ou seja, de dentro da empresa para a Internet. Então, podemos concluir que:

- o foco é na chain OUTPUT;
- queremos adicionar uma regra (-A);
- o endereço destino (-d) específico não sabemos qual é, mas descartamos a letra B, pois 157.240.226.2174 não é um endereço válido;
- o protocolo (-p) deve ser TCP, pois é um site (HTTP);
- não existe ação (-j) BLOCK. Para descartar um pacote é utilizado -j DROP.

Juntando tudo isso, temos: # iptables -A OUTPUT -p tcp -d 157.240.226.174 -j DROP.

Gabarito: A

37. (FUNDATEC/PROCERGS/2023) Assinale a alternativa que apresenta o comando nativo do Linux utilizado em todas as versões para visualizar o caminho de um pacote de rede em tempo real.

- A) traceroute  
B) ping,  
C) netstat  
D) ping  
E) ifconfig

#### Comentários:

traceroute: usado para rastrear a rota que os pacotes seguem para alcançar um destino. Mostra uma lista de saltos (hops) que os pacotes fazem, além dos tempos de resposta de cada salto. É útil para diagnosticar problemas de roteamento e identificar os nós intermediários (roteadores) pelos quais os pacotes de dados passam ao viajar da origem para o destino. A sintaxe é idêntica ao ping:

```
$ traceroute www.estrategiaconcursos.com.br
```

```
$ traceroute 192.168.1.150
```

Gabarito: A

38. (IDECAN/SEFAZ-RR/2023) Você recebe a demanda de verificar todos os saltos feitos, na camada de rede, desde o IP da sua máquina (com sistema operacional Linux) até o servidor Web da sua empresa. Selecione o comando que você utilizará na sua máquina para executar essa demanda.



- A) traceroute
- B) ipconfig
- C) netstat
- D) tracert
- E) ping

#### Comentários:

traceroute: usado para rastrear a rota que os pacotes seguem para alcançar um destino. Mostra uma lista de saltos (hops) que os pacotes fazem, além dos tempos de resposta de cada salto. É útil para diagnosticar problemas de roteamento e identificar os nós intermediários (roteadores) pelos quais os pacotes de dados passam ao viajar da origem para o destino. A sintaxe é idêntica ao ping:

```
$ traceroute www.estrategiaconcursos.com.br
```

```
$ traceroute 192.168.1.150
```

Cuidado!!!! O comando tracert é usado no Windows!!!! Não caia nessa pegadinha!

Gabarito: A

39. (FUNDATEC/PROCERGS/2023) Assinale a alternativa que apresenta o comando do Linux utilizado para visualizar as conexões de rede estabelecidas, portas utilizadas pela rede e suas estatísticas.

- A) ifconfig
- B) netstat
- C) route
- D) arp
- E) ip

#### Comentários:

netstat: usado para exibir informações sobre as conexões de rede, tabelas de roteamento, estatísticas de interface de rede etc. Útil para diagnosticar problemas de rede, monitorar o tráfego e obter informações sobre as portas e protocolos em uso.

Para exibir informações detalhadas sobre todas as conexões e sockets:

```
$ netstat -a
```

Para listar todas as conexões de rede ativas (endereços IP e portas):

```
$ netstat -tuln
```

- -t: mostra as conexões TCP;
- -u: mostra as conexões UDP;
- -l: mostra as portas em escuta (listen);



- -n: exibe endereços IP e números de porta em formato numérico (evita a resolução de nomes).

Para exibir estatísticas de interface de rede:

```
$ netstat -i
```

Para exibir tabelas de roteamento IP:

```
$ netstat -r
```

Gabarito: B



## QUESTÕES COMENTADAS - FGV

1. (FGV/AL-PR/2024) Um desenvolvedor de software precisa utilizar um sistema operacional FreeBSD para melhor instalar, atualizar e remover pacotes por meio da linha de comando.

Nesse caso, ele vai utilizar o sistema de gerenciamento de pacotes conhecido como

- A) apt.
- B) dnf.
- C) gnu.
- D) pkg.
- E) pacman.

### Comentários:

Após a instalação, é desejável que se atualize o sistema para garantir que as últimas correções de segurança estejam instaladas. Para a instalação e alterações de aplicativos, **Pkg é o gerenciador de pacotes no FreeBSD**. Os seguintes comandos podem ser executados:

```
# freebsd-update fetch  
  
# freebsd-update install  
  
# pkg update  
  
# pkg upgrade
```

Para a instalação de pacotes adicionais (com o gerenciador de pacotes **pkg**), é utilizado o seguinte comando (ex.: para instalar o vim):

```
# pkg install vim
```

Gabarito: D

2. (FGV/AL-PR/2024) Ao instalar um sistema operacional FreeBSD em um computador x86, o administrador da máquina configurou o sistema de arquivos para um tipo específico do FreeBSD, que opera em 64 bits, com implementação de snapshots e a possibilidade de fazer a verificação de integridade do sistema de arquivos em segundo plano. Esse sistema de arquivos é o

- A) FAT.
- B) APFS.
- C) HPFS.
- D) UFS2.



E) UJNFS.

#### Comentários:

O UFS2 é uma versão aprimorada do sistema de arquivos UFS (Unix File System), amplamente utilizado no FreeBSD e outros sistemas operacionais derivados do BSD. UFS2 foi introduzido no FreeBSD 5.0 como uma evolução do UFS1, adicionando várias melhorias e capacidades para atender às necessidades de sistemas modernos. Vamos às características:

- Suporte a volumes maiores: aumenta o tamanho máximo de arquivos e sistemas de arquivos, permitindo volumes de até 16 exabytes e arquivos de até 2 exabytes, em comparação aos limites muito menores do UFS1;
- Timestamps melhorados: suporta timestamps de 64 bits, oferecendo maior precisão e eliminando o problema do "Ano 2038" para timestamps de arquivos, que afetava sistemas de arquivos mais antigos;
- Atributos estendidos: inclui suporte a atributos estendidos, permitindo armazenar metadados adicionais para arquivos, como informações de segurança e permissões mais granulares;
- *Snapshots*: suporta *snapshots*, uma funcionalidade que permite tirar "fotografias" do sistema de arquivos em um determinado momento, extremamente útil para backups e recuperação de sistemas;
- Compatibilidade retroativa: mantém compatibilidade com UFS1, permitindo que sistemas existentes sejam migrados para o novo formato sem a necessidade de reformatação completa;
- Desempenho: oferece melhorias de desempenho em relação ao UFS1, especialmente em sistemas de arquivos maiores ou com grandes quantidades de arquivos.

Gabarito: D



## QUESTÕES COMENTADAS - MULTIBANCAS

1. (FEPESE/UEDESC/2022) São todos elementos que caracterizam uma jail no FreeBSD:

1. Um conjunto de arquivos
2. Uma subárvore de diretórios
3. Um nome de host
4. Um endereço IP
5. Um comando

Assinale a alternativa que indica todas as afirmativas corretas.

- A) São corretas apenas as afirmativas 1, 2, 3 e 4.
- B) São corretas apenas as afirmativas 1, 2, 3 e 5.
- C) São corretas apenas as afirmativas 1, 2, 4 e 5.
- D) São corretas apenas as afirmativas 1, 3, 4 e 5.
- E) São corretas apenas as afirmativas 2, 3, 4 e 5.

### Comentários:

*Jails*: São uma forma de virtualização leve que permite a criação de ambientes isolados dentro de um único sistema operacional.

Os principais elementos que caracterizam uma *jail* são:

- Sistema de arquivos Isolado: cada *jail* tem seu próprio sistema de arquivos, que é separado do sistema host. O diretório raiz (/) da *jail* é montado em um diretório específico do sistema host (ex.: /usr/jails/nome\_da\_jail);
- Isolamento de processos: processos dentro de uma *jail* são isolados e não podem ver ou interagir diretamente com processos fora dela ou em outras jails;
- Configuração de rede isolada: uma *jail* pode ter seu próprio endereço IP, que é configurado separadamente do sistema host;
- Isolamento de usuários e permissões: *jails* possuem seu próprio conjunto de usuários e grupos, que são independentes dos usuários e grupos no sistema host;
- Configuração e limitações de recursos: é possível configurar limites de recursos para *jails*, como limites de uso de processador, memória, e entrada/saída de disco, para controlar o



impacto de cada jail no sistema host;

- Segurança e controle de acesso: são os principais motivos para usar *jails*, pois elas oferecem um ambiente isolado onde vulnerabilidades de segurança são confinadas à jail, sem afetar o sistema host ou outras *jails*;
- Facilidade de gerenciamento: *jails* são relativamente fáceis de criar, configurar e gerenciar usando ferramentas nativas do FreeBSD (ex.: ezjail, iocage, ou comandos, como jail e jexec);
- Isolamento de sistema: cada *jail* pode executar uma versão específica do FreeBSD, independentemente da versão do sistema host, desde que seja compatível com a ABI (*Application Binary Interface*);
- Integração com o sistema host: mesmo sendo isoladas, as *jails* ainda compartilham o kernel do sistema host. Ou seja, não há overhead significativo como ocorre em máquinas virtuais, resultando em uma virtualização muito leve;
- Flexibilidade de configuração: o administrador pode configurar uma *jail* para ser extremamente restritiva, onde apenas um conjunto muito limitado de comandos e acessos são permitidos, ou mais permissiva, com acesso a mais recursos do sistema host.

Gabarito: E

2. (FEPESE/UDESC/2022) No contexto do sistema operacional FreeBSD, a coleção de arquivos projetados para automatizar o processo de compilar uma aplicação a partir do código fonte é denominada:

- A) build
- B) make
- C) package
- D) yarn
- E) port

Comentários:

*Ports Collection* (ou simplesmente, *ports*): Uma grande coleção de aplicativos que podem ser instalados a partir do código-fonte de maneira simplificada. Também existem pacotes pré-compilados disponíveis para uma instalação mais rápida.

Gabarito: E





## LISTA DE QUESTÕES - CEBRASPE

1. (CEBRASPE/EMAP/2018) O kernel do sistema operacional Linux tem a função de interpretar os comandos executados em um terminal.

2. (CEBRASPE/FUB/2018) O Linux é um sistema operacional em que cada usuário consegue ter apenas um processo ativo por vez, processo esse que é iniciado automaticamente quando o sistema é carregado.

3. (CEBRASPE/SLU-DF/2019) Windows e Linux são classificados como sistemas operacionais de tempo real crítico, porque fornecem garantias absolutas de que todas as suas ações ocorrerão dentro de intervalos de tempo determinados.

4. (CEBRASPE/PG-DF/2021) Acerca da administração de sistemas Windows e Unix/Linux, julgue o seguinte item.

No sistema Linux Ubuntu, o comando `sudo apt-get upgrade` apresenta as atualizações de sistema disponíveis com opção para seleção e execução individualizada de cada atualização.

5. (CEBRASPE/TCE-RJ/2021) Acerca dos sistemas operacionais Windows e Linux e da interoperabilidade entre esses sistemas, julgue o item que se segue.

No processo de inicialização de um sistema Linux, a função de carregar e descompactar a imagem do kernel é responsabilidade do carregador de boot de primeiro estágio.

6. (CEBRASPE/TRT8/2022) Assinale a opção que apresenta o arquivo utilizado para inicializar um ambiente de computador em um sistema operacional Linux.

- a) /etc/group
- b) /etc/passwd
- c) /etc/syslog.conf
- d) /etc/startup
- e) /etc/profile



## GABARITO

01	02	03	04	05	06
Errado	Errado	Errado	Errado	Errado	E



## LISTA DE QUESTÕES – FGV

1. (FGV/IBGE/2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:

- A) create\_process;
- B) new\_process;
- C) fork;
- D) spawn;
- E) duplicate.

2. (FGV/MPE-AL/2018) Instalar, atualizar e remover pacotes no sistema operacional CentOS 7 é uma tarefa frequente para desenvolvedores de sistemas. Por isso, eventualmente podem ocorrer dúvidas sobre se determinado pacote está instalado ou qual é a versão que está sendo utilizada. Para dirimir essas dúvidas, sobre o pacote httpd devemos utilizar o comando

- A) rpm -q httpd
- B) yum check httpd
- C) apt-get find httpd
- D) find -iname httpd
- E) crontab -l httpd

3. (FGV/COMPESA/2018) Com relação às características e tarefas de administração do sistema operacional Linux, analise as afirmativas a seguir.

I. O esquema de partição GPT (Guid Partition Table) oferece a possibilidade de até 128 partições primárias, com tamanhos maiores do que 2TB cada.

II. Ext4, XFS e exFAT são exemplos de sistemas de arquivo com jornal, utilizados pelo Linux.

III. A área de swap no Linux pode ser uma partição dedicada de swap, um arquivo de swap ou uma combinação de arquivos e partições de swap.

Está correto o que se afirma em

- A) I, somente.
- B) II, somente.
- C) III, somente.
- D) I e III, somente.



E) I, II e III.

4. (FGV/AL-RO/2018) Wallace é administrador de um servidor com sistema operacional CentOS e deseja compartilhar um diretório com os integrantes da empresa por meio do protocolo NFS.

Para que a configuração desse compartilhamento possa ser efetuada, Wallace deve editar o arquivo /etc/

- A) services
- B) modules
- C) hosts
- D) export
- E) fstab

5. (FGV/AL-RO/2018) Roberto trabalha como administrador de redes em uma empresa de cosméticos e sua principal função é a configuração de estações de trabalho dos seus colaboradores. Rotineiramente, Roberto instala e remove pacotes nas estações de trabalho às quais ele dá suporte.

Assinale a opção que indica o comando utilizado por Roberto para desinstalar pacotes RPM nas estações de trabalho com sistema operacional Linux CentOS.

- A) rpm -Uvh pacote
- B) rpm -e pacote
- C) rpm -ivh pacote
- D) rpm -qi pacote
- E) rpm -qa | grep pacote

6. (FGV/TRT-MA/2022) O RPM Package Manager é um sistema executado em várias distribuições Linux e UNIX. O conjunto completo de modos básicos de operação do RPM é

- A) consulta, atualização e instalação de pacotes.
- B) instalação, atualização, criação e depuração de pacotes.
- C) valoração, reinstalação, consulta e desinstalação de pacotes.
- D) instalação, desinstalação, valoração, consulta, monitoramento e depuração de pacotes.
- E) criação, instalação, desinstalação, atualização, consulta e verificação de pacotes.



## GABARITO

01	02	03	04	05	06
C	A	D	D	B	E



## LISTA DE QUESTÕES – FCC

1. (FCC/TRF5/2017) Considere a figura abaixo que mostra a arquitetura do sistema operacional Linux



A caixa

A) I representa a camada responsável pela interface entre o hardware e as aplicações. Dentre suas funções encontram-se gerenciamento de I/O, manutenção do sistema de arquivos, gerenciamento de memória e swapping, controle da fila de processos etc.

B) II representa a camada que permite o acesso a recursos através da execução de chamadas feitas por processos. Tais chamadas são geradas por funções padrão suportadas pelo kernel. Dentre suas funções estão habilitar funções padrão como open, read, write e close e manter a comunicação entre as aplicações e o kernel.

C) I é um processo que executa funções de leitura de comandos de entrada de um terminal, interpreta-os e gera novos processos, sempre que requisitados. É conhecido também como interpretador de comandos.

D) II é um processo que realiza modificações no shell, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o shell, o usuário torna o Linux enxuto e adaptável.

E) I é um processo que realiza modificações no kernel, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o kernel, o usuário torna o Linux enxuto e adaptável.

2. (FCC/SEFAZ-BA/2019) Um Auditor Fiscal da área de Tecnologia da Informação deseja desinstalar um pacote chamado java-1.6.0-openjdk.x86\_64 em linha de comando, como usuário root, no Red Hat Enterprise Linux 6. Para isso, terá que utilizar o comando

A) apt-get uninstall java-1.6.0-openjdk.x86\_64

B) apt-get remove -rf java-1.6.0-openjdk.x86\_64

C) rm -rf java-1.6.0-openjdk.x86\_64



D) yum remove java-1.6.0-openjdk.x86\_64

E) apt-get -rf java-1.6.0-openjdk.x86\_64

3. (FCC/Prefeitura de Manaus-AM/2019) Em computador com sistema operacional Linux e o OpenLDAP instalado, o programador editou o arquivo ldap.conf para configurar

A) a autenticação do servidor.

B) a execução do serviço do LDAP.

C) o banco de dados a ser utilizado.

D) a geração de índices do banco de dados.

E) o acesso dos clientes ao diretório.

## GABARITO

01	02	03
C	D	E



## LISTA DE QUESTÕES – VUNESP

1. (VUNESP/Câmara de Sertãozinho-SP/2019) Considere que o usuário de um computador com sistema operacional Linux deseja executar um comando em segundo plano (background) em um terminal Shell. Para efetivar essa ação, o usuário deve executar o comando

- A) e teclar a combinação Ctrl+c
- B) e teclar a combinação Ctrl+x
- C) seguido de bg
- D) seguido de &
- E) seguido de !

## GABARITO

01

D





## LISTA DE QUESTÕES – MULTIBANCAS

1. (CESGRANRIO/Petrobras/2011) Em um sistema Unix, um arquivo de script chamado teste.sh foi copiado para o diretório /tmp. No shell do sistema, o usuário submeteu dois comandos: cd /tmp e teste.sh. Após a execução do segundo comando, o shell informou uma mensagem de erro, indicando comando não encontrado. O que deve ser feito para corrigir o problema que gerou essa mensagem?

- A) Certificar que o usuário não entrou com letras maiúsculas.
- B) Certificar que o arquivo tem permissão para ser executado.
- C) Omitir a extensão .sh ao entrar com o nome do script.
- D) Incluir ./ antes do nome do script.
- E) Mover o arquivo para o diretório home e executá-lo.

2. (CESGRANRIO/Petrobras/2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado.

Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.
- E) na linha de comando, após o nome do arquivo que contém o script.

3. (CEPERJ/Rio previdência/2014) Shell script é uma linguagem de script para Linux, nada mais do que comandos do próprio Linux que são executados em uma determinada sequência para uma determinada finalidade. Nesse contexto, duas situações são listadas a seguir.

I- No terminal ou modo gráfico, deseja-se criar um arquivo que possa ser editado para que se torne o primeiro shell script a ser criado, sendo necessário utilizar um comando CMD1.

II- Para que seja possível executar o shell script criado, é preciso atribuir a este o direito de execução; para isso é necessário usar um comando CMD2.

Exemplos de CMD1 e de CMD2 são, respectivamente:



- A) touch shell1.sh e exec +x shell1.sh
- B) touch shell1.sh e chmod +x shell1.sh
- C) create shell1.sh e chmod +x shell1.sh
- D) new shell1.sh e chmod +x shell1.sh
- E) new shell1.sh e exec +x shell1.sh

4. (FUMARC/AL-MG/2014) No Linux, considere a existência de um arquivo "arquivo.txt" no diretório corrente, com o seguinte conteúdo:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Deseja-se produzir, a partir do arquivo, a seguinte saída:

1 - 10 2 - 20 3 - 30

O script shell usado com comandos AWK que produz a saída desejada é:

- A) awk arquivo.txt '{ print \$2" - "\$4 }'
- B) cat arquivo.txt | awk '{ print \$2" - "\$4 }'
- C) cat arquivo.txt | awk '{ print \$1" - "\$3"\n" }'
- D) cat arquivo.txt | awk '{ printf(\$2" - "\$4) }'

5. (INSTITUTO AOCP/EBSERH/2016) Em sistemas operacionais Linux é possível criar scripts para automatizar tarefas rotineiras. A extensão de arquivo utilizada para Shell Script é

- A) .bin
- B) .sl
- C) .bash
- D) .sh
- E) .gz

6. (AOCP/EBSERH/2016) O NFS (Network File System) permite compartilhar sistemas de arquivos entre computadores conectados em rede e pode ser parte fundamental da infraestrutura da tecnologia da informação. Sobre NFS, analise as assertivas a seguir e assinale a alternativa que aponta a(s) correta(s).



I. O NFS é considerado sem estado (stateless) e, portanto, quando um servidor NFS volta a funcionar, o estado anterior é restaurado e a informação não é perdida.

II. O NFS pode ser implementado do lado servidor e do lado cliente.

III. O NFS roda sobre o protocolo RPC (Remote Procedure Call), que define um modo independente do sistema para processos se comunicarem através de uma rede de computadores.

IV. O NFS suporta apenas UDP como protocolo de transporte, pois ele apresenta desempenho significativamente melhor que o TCP em redes locais.

A) Apenas I e II.

B) Apenas II e IV.

C) Apenas III.

D) Apenas III e I.

E) Apenas I, II e III.

7. (CS-UFG/DEMAE/2017) Na maior parte das distribuições Linux, a memória virtual é uma partição denominada

A) EXT

B) SWAP

C) SETUP

D) BIOS

8. (Quadrix/COFECI/2017) Não é possível realizar uma instalação automática, por meio de uma rede, no sistema operacional Linux.

9. (Colégio Pedro II/Colégio Pedro II/2017) Sistema operacional é um conjunto de programas básicos e utilitários que fazem seu computador funcionar. O Debian é um sistema operacional, em cujo núcleo está o Kernel, o programa mais fundamental no computador e que faz todas as operações mais básicas, permitindo que você execute os outros programas. O Debian atualmente usa o Kernel Linux. Mas um sistema operacional não funciona somente com o Kernel, são necessários utilitários e aplicativos. O Debian utiliza estas ferramentas do projeto GNU. Por esse motivo, muitos utilizadores defendem que devemos chamar o sistema de "Debian GNU/Linux". Reconheça (1) o gerenciador de pacotes (programas) usado no Debian e em distribuições derivadas do Debian (Ubuntu, Knoppix, Big Linux etc.) que utiliza uma lista de dependências para instalar tudo o mais automaticamente possível, e (2) o arquivo de configuração utilizado por este gerenciador de pacote.



- A) Gerenciador (1) - apt-get  
Arquivo (2) - source.list
- B) Gerenciador (1) - apt-get  
Arquivo (2) - package.cfg
- C) Gerenciador (1) - make install  
Arquivo (2) - package.cfg
- D) Gerenciador (1) - dpkg  
Arquivo (2) - source.list

10. (CS-UFG/SANEAGO-GO/2018) Um utilitário GNU, presente em diversos sistemas operacionais linux, que lista as partições dos discos rígidos, é:

- A) fdisk.
- B) mcdisk.
- C) parted.
- D) gedit.

11. (IF-RS/IF-RS/2018) São tipos de sistemas de arquivos utilizados no Linux:

- A) FAT 16, FAT 32, NTFS
- B) FAT 32, HFS, ExFAT
- C) Ext2, Ext3, Btrfs
- D) Ext2, Ext3, HFS
- E) Ext2, Ext3, NFS, SMB

12. (COPESE-UFT/UFT/2018) Embora seja possível realizar boot de um sistema Linux a partir de um pendrive, a maioria das instalações do Linux o realiza a partir do disco rígido do computador. Esse processo consiste em duas fases básicas:

1. Executar o carregador de boot a partir do dispositivo de boot;
2. Iniciar o kernel do Linux e iniciar os processos.



Assinale a alternativa que contém um gerenciador de boot para sistemas Linux:

- A) CISC (Complex Instruction Set Computer).
- B) GRUB (Grand Unified Bootloader).
- C) ORM (Object Relational Mapping).
- D) CMS (Content Management System).

13. (COPESE-UFT/UFT/2018) Sistemas operacionais Linux permitem logins simultâneos de diferentes usuários. Assinale a alternativa que contém o comando a ser utilizado para visualizar os usuários logados em um determinado instante.

- A) listusers
- B) Idconfig
- C) who
- D) uname

14. (IF-RS/IF-RS/2018) Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?

- A) dns 8.8.8.8
- B) echo 8.8.8.8 > /etc/hosts.conf
- C) echo nameserver 8.8.8.8 > /etc/resolv.conf
- D) nslookup 8.8.8.8
- E) nameserver 8.8.8.8

15. (COPEVE-UFAL/UFAL/2018) O assistente de tecnologia da informação precisa atualizar o sistema operacional GNU/Linux do servidor WEB de uma empresa. Qual comando, usando o modo root, deve ser utilizado?

- A) upgrade distro-all.
- B) upgrade apt-distro.
- C) apt-get dist-upgrade.



- D) apt-get purge "nome da distro".
- E) apt-cache showpkg "nome da distro".

16. (COPEVE-UFAL/UFAL/2018) Dadas as afirmativas quanto aos conceitos de Kernel no sistema operacional GNU/Linux,

- I. Compilar o Kernel permite ao usuário remover drivers inúteis diminuindo o tempo de arranque do sistema operacional.
- II. Os módulos do Kernel do sistema estão armazenados no diretório /boot/.
- III. O comando uname -a exibe as informações do kernel do sistema.
- IV. Módulos são as partes do kernel que são carregadas somente quando são requisitadas por um aplicativo ou dispositivo.

Verifica-se que está(ão) correta(s)

- A) II, apenas.
- B) I e II, apenas.
- C) III e IV, apenas.
- D) I, III e IV, apenas.
- E) I, II, III e IV.

17. (COPEVE-UFAL/UFAL/2018) Analise a configuração de rede no GNU/Linux.

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
allow-hotplug eth0
```

```
iface eth0 inet static address 192.168.100.1 netmask 255.255.255.0 network 192.168.100.0  
broadcast 192.168.100.255
```

Dadas as afirmativas quanto à configuração de rede,

- I. O "iface eth0 inet static" informa ao sistema que a placa de rede terá o endereço IP estático.
- II. O "iface lo" informa ao sistema que a placa de rede receberá um endereço IP via servidor DHCP.



III. Uma segunda placa de rede com fio instalada nesse computador irá receber a denominação eth1.

IV. O "allow-hotplug" permite conectar dispositivos hotplug no computador.

Verifica-se que está(ão) correta(s)

- A) II, apenas.
- B) I e III, apenas.
- C) II e IV, apenas.
- D) I, III e IV, apenas.
- E) I, II, III e IV.

18. (Quadrix/CFBio/2018) Carregamento e inicialização do kernel e execução dos scripts de inicialização do sistema são algumas das etapas do processo de inicialização de um Linux típico.

19. (Quadrix/CRM-PR/2018) Um computador pessoal com sistema operacional Linux, versão desktop, pode ter no máximo cinco usuários simultâneos.

20. (FADESP/IF-PA/2018) Sobre os pacotes do Sistema Operacional (SO) Debian 9.5, é correto afirmar que

- A) o apt é um programa que pode substituir o dpkg para o gerenciamento de pacotes do SO.
- B) os pacotes devem ser distribuídos no formato .tar.gz.
- C) o dpkg-source compila e executa código fonte de um pacote.
- D) um pacote marcado como Essencial não pode ser desinstalado do Sistema Operacional.
- E) cada pacote deve especificar as dependências sobre outros pacotes para seu funcionamento.

21. (CONSULPAM/Câmara de Juiz de Fora-MG/2018) Em relação ao Sistema Operacional Linux, marque o item INCORRETO:

- A) Sua arquitetura é composta por um núcleo monolítico cujas funções são: gerenciar a memória, operar as entradas e saídas e o acesso aos arquivos.
- B) Outra característica do Linux é com relação aos drivers de dispositivos e suporte a rede, os quais podem ser compactadas e utilizadas como se fossem módulos ou bibliotecas (LKM em



inglês Loadable Kernel Modules), separados pela parte principal, cujo carregamento pode ser ativado após a execução do núcleo.

C) No quesito portabilidade, o Linux funciona com eficiência em plataformas como x64 da Intel (EM64T e AMD64) PowerPC, Alpha, SPARC, porém é de difícil instalação nos sistemas embarcados como PVR, celulares, Tv's e Handhelds.

D) A partir da década de 90, ao passo que a distribuição do Linux se popularizou, foi também limitada, pois se torna uma alternativa no uso de software livre, contra os sistemas operacionais da Apple (Mac OS) e Microsoft (Windows).

22. (COPESE/UFT/2018) O sistema operacional Linux é reconhecido por permitir diversos níveis de personalização, inclusive de suportar o uso de vários ambientes gráficos. Assinale a alternativa que NÃO constituiu uma interface gráfica usada no Linux

A) Pantheon.

B) XFCM.

C) MATE Desktop.

D) Cinnamon Desktop.

23. (CCV-UFC/UFC/2018) De acordo com a linha do arquivo '/ etc / passwd' exibida abaixo, qual das seguintes afirmações é verdadeira?

```
jferreira:x:502:1000:Joao Ferreira:/home/jferreira:/bin/bash
```

A) O usuário Joao Ferreira possui a senha 'x'.

B) Shadow passwords (senhas ocultas) são utilizados no sistema.

C) O usuário Joao Ferreira pertence ao grupo com groupID 502.

D) Membros do groupID 502 podem ler o diretório /home/jferreira.

E) O nome de usuário (username) 'jferreira' pertence ao grupo 'jferreira'.

24. (Quadrix/CRM-PR/2018) É possível instalar, em um mesmo computador desktop, os sistemas operacionais Windows 7 e Linux, de forma independente.

25. (Quadrix/CRM-PR/2018) O usuário root do sistema Linux tem autonomia para acessar arquivos de outros usuários, com exceção dos usuários administradores.





26. (IDECAN/CRF-SP/2018) "O arquivo \_\_\_\_\_ só é legível pelo superusuário e serve para manter senhas criptografadas protegidas contra o acesso não autorizado. Ele também fornece informações sobre contas que não são disponíveis em \_\_\_\_\_." Assinale a alternativa que completa correta e sequencialmente a afirmativa sobre o Sistema Operacional Linux.

- A) /etc/passwd; /etc/group
- B) /etc/group; /etc/passwd
- C) /etc/passwd; etc/shadow
- D) /etc/shadow; /etc/passwd

27. (UERR/IPERON/2018) Um administrador de uma rede baseada em Linux deseja guardar no servidor os scripts especiais para iniciar ou interromper módulos e programas diversos. Para isso, ele precisa guardar essas informações dentro do diretório:

- A) /srv
- B) /opl
- C) /etc
- D) /lib
- E) home

28. (UERR/IPERON/2018) Em um computador com Linux, deseja-se instalar o sistema de arquivos ext3, mas com o nível de Journaling no qual se grava mudanças em arquivos de metadados, que força que a escrita do conteúdo dos arquivos seja feita após a marcação de seus metadados. Esse nível de Journaling é denominado:

- A) ordered
- B) cshell.
- C) iso9660
- D) journal.
- E) fsreiser.



29. (CS-UFG/IF Goiano/2019) Os sistemas Linux buscam padronizar diretórios para a localização de arquivos. Comandos essenciais do sistema operacional, tais como cat, tar, su, rm e pwd, em geral, estão localizados nos seguintes diretórios:

- A) /lib e /var
- B) /usr/lib e /lib
- C) /bin e /usr/bin
- D) /usr/bin e /var

30. (IF-MS/IF-MS/2019) O sistema operacional Linux possui várias partições (áreas) em sua estrutura, cada uma com uma função definida. Assinale a alternativa que apresenta a partição que abriga a pasta raiz do sistema que contém arquivos essenciais ao seu pleno funcionamento:

- A) /boot
- B) /bin
- C) /
- D) /etc
- E) /root

31. (COSEAC/UFF/2019) No Sistema Operacional Linux o diretório local padrão do superusuário é:

- A) /home.
- B) /usr.
- C) /root.
- D) /lib.
- E) /dev.

32. (INAZ do Pará/CORE-SP/2019) "Todos os arquivos e diretórios do sistema Linux instalado no computador partem de uma única origem: o diretório raiz. Mesmo que estejam armazenados em outros dispositivos físicos." - Disponível em: <https://canaltech.com.br/linux/entendendo-a-estrutura-de-diretorios-dolinux/>. Acesso em: 13.12.2018.



Qual pasta no sistema operacional LINUX tem a função de armazenar arquivos de dispositivos do sistema?

- A) /dev.
- B) /etc.
- C) /lib.
- D) /var.
- E) /proc.

33. (INAZ do Pará/CORE-SP/2019) "O agendamento de tarefas é um recurso muito interessante para a administração de sistemas operacionais. É possível programar a execução de scripts de manutenção do sistema, disparar envio de newsletters, gerar relatórios de análises de logs, entre outros" - Disponível em: [https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-\(cron-e-at\)](https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-(cron-e-at)). Acesso em: 13.12.2018.

Qual programa do sistema operacional LINUX executa tarefas de modo automático, em horários pré-determinados, e executa tarefas que não foram executadas, enquanto o sistema esteve desligado?

- A) Fcron.
- B) Cron.
- C) Anacron.
- D) Schedule.
- E) Crontab.

34. (IDECAN/IF-PB/2019) O Debian, uma das distribuições do sistema operacional Linux, possui um conjunto de pastas com nome pré-estabelecidos. Essas pastas possuem significado específico para o sistema operacional, visando atender necessidades do mesmo. Assinale a alternativa que indica corretamente o nome da pasta que armazena arquivos referentes às instalações de programas não oficiais da distribuição do sistema operacional.

- A) /bin
- B) /lib
- C) /opt
- D) /var



E) /proc

35. (UFRR/UFRR/2019) A respeito de diretórios no sistema operacional Linux, a estrutura é conhecida como árvore invertida, isto é, a raiz da árvore de diretórios é o topo. Qual é a representação do diretório raiz, e nome do usuário que possui privilégio para escrever neste diretório.

A) (raiz) e usuário root

B) \\ (raiz) e usuário administrador

C) / (raiz) e usuário root

D) [ (raiz) e usuário administrador

E) ]/ (raiz) e usuário root

36. (IF-ES/IF-ES/2019) O Network File System (NFS) – Sistemas de Arquivos em Rede – tem como um dos principais propósitos dar suporte a um sistema heterogêneo, no qual clientes e servidores estejam possivelmente executando sistemas operacionais e hardwares diferentes. Sobre o NFS, é CORRETO afirmar:

A) O NFS utiliza dois protocolos cliente-servidor, em que o primeiro é responsável pela montagem e o segundo é para acesso de diretório e arquivos.

B) O servidor tem total gerência sobre o ponto de montagem nos clientes.

C) Os serviços NFS são implementados apenas nos servidores Linux.

D) Como critério de segurança, os clientes não podem ter acesso aos atributos dos arquivos.

E) O NFS faz uso de máquinas distintas para servidores e clientes, impossibilitando que a mesma máquina seja tanto cliente quanto servidor.

37. (FUNDATEC/Prefeitura de Maçambará-RS/2023) No Linux, a camada conhecida por ser o núcleo do sistema operacional, administrando o hardware e atuando, principalmente, como uma interface entre esse e qualquer programa em execução, é o:

A) kernel

B) system-manager

C) shell

D) bus

E) socket-level



38. (CS-UFG/UFG/2023) O Sistema Operacional Linux possui diversos arquivos relacionados à configuração de redes de computadores, dos quais, podemos citar:

- A) /etc/network/dns.conf e /etc/hosts/network.
- B) /etc/hosts e /etc/interface/host.
- C) /etc/network/interfaces e /etc/resolv.conf.
- D) /etc/sysconfig/interfaces e /etc/interfaces/network.

39. (FUNCERN/Câmara de Natal-RN/2023) Levando-se em consideração distribuições de sistemas operacionais Linux, o conceito FHS (File Hierarchy Standard) é um padrão para que softwares e usuários possam prever a localização de arquivos e diretórios instalados. Sendo assim, o diretório sugerido para conter arquivos de configuração do sistema, os quais são usados para controlar a operação de um programa, é o

- A) /etc
- B) /boot
- C) /tmp
- D) /root

40. (Quadrix/CRM-MG/2023) Assinale a alternativa que apresenta o sistema de inicialização (init system), do sistema operacional Linux, composto de um conjunto de programas executado em segundo plano.

- A) Init
- B) Bash
- C) Cron
- D) Systemd
- E) Syslogd



## GABARITO

01	02	03	04	05	06
D	A	B	B	D	E
07	08	09	10	11	12
B	Errado	A	C	C	B
13	14	15	16	17	18
C	C	C	D	D	Certo
19	20	21	22	23	24
Errado	E	C	B	B	Certo
25	26	27	28	29	30
Errado	D	C	A	C	C
31	32	33	34	35	36
C	A	A	C	C	A
37	38	39	40		
A	C	D			



## LISTA DE QUESTÕES – CEBRASPE

1. (CEBRASPE/TJ-SE/2014) No que se refere à lógica e à programação, julgue os itens que se seguem.

Em um comando Shell Script do Linux, é possível combinar diversos comandos em sequência utilizando-se apenas o comando +.

2. (CEBRASPE/ANATEL/2014) Julgue os próximos itens, acerca do ambiente Linux.

Em um script shell, se o retorno do último comando obtido pela variável interrogação (?) for igual a 1, então o comando teve sucesso na sua execução.

3. (CEBRASPE/TCE-PA/2016) Acerca dos sistemas operacionais Linux e Windows Server, julgue o item que se segue.

No Linux, o tipo de permissão r possibilita ao usuário executar o comando ls e listar o conteúdo de um diretório, ao passo que o tipo w admite a gravação e exclusão de um diretório, cabendo ao tipo x facultar ao usuário a execução de um arquivo script.

4. (CEBRASPE/PGE-RJ/2022) Acerca da administração do sistema operacional Linux e da linguagem de script Shell, julgue o item subsequente.

Em ambiente Linux, ao se executar um arquivo que contenha a instrução a seguir.

```
echo PGE-RJ celebra o fim das obras de restauro \ do Convento do Carmo
```

Será obtido o resultado seguinte.

```
PGE-RJ celebra o fim das obras de restauro do Convento do Carmo
```



## GABARITO

01	02	03	04
Errado	Errado	Certo	Certo





## LISTA DE QUESTÕES – FGV

1. (FGV/TRT-MA/2022) Analise o script abaixo.

Assinale a opção que indica quantas linhas serão impressas na tela de um terminal Linux e quais os valores de  $\${NR}$  e de count ao término da execução do script.

- a) 7, 273 e 6.
- b) 14, 312 e 7.
- c) 14, 263 e 7.
- d) 21, 361 e 7.
- e) 21, 312 e 6.

2. (FGV/Senado Federal/2022) Assinale a opção que indica o resultado da execução do script Shell a seguir em um prompt de comando do Linux.

- a) AREPO TENET
- b) OPERA SATOR ARE
- c) ENET OP AREP
- d) SATOR OPERA
- e) R AREP ROTAS

3. (FGV/Prefeitura de Manaus-AM/2022) Dado o programa bash abaixo,

```
#!/bin/bash
```

```
a=12
```

```
b=a+15
```

```
echo  $\${b+20}$ 
```

assinale a opção que mostra a saída correta do programa após sua execução.

- A) 0.



- B) 15.
- C) 20.
- D) 32.
- E) 47.

4. (FGV/TRT13/2022) Analise o código a seguir escrito em shell script:

Ao executar esse programa em um terminal será escrito na saída padrão:

- A) www.website\_trt.com.br
- B) www.website\_trt.cxm.br
- C) Website
- D) http://www.website\_trt
- E) http://www.website\_trt.com.br

## GABARITO

01	02	03	04
E	C	C	B



## LISTA DE QUESTÕES – FCC

1. (FCC/TRE-SP/2012) No sistema Linux, para se executar um arquivo texto contendo comandos de interpretador como um script é necessário que o arquivo

- A) seja compilado.
- B) possua permissão de execução.
- C) esteja no diretório /usr/bin
- D) tenha a extensão .exe
- E) tenha a extensão .bat

2. (FCC/Prefeitura de Teresina-PI/2016) O Administrador de um servidor com sistema operacional Red Hat Linux deseja escrever um script para automatizar o processo de backup do servidor. A primeira linha do arquivo de script deve ser:

- A) #/sbin/bash
- B) %#/sbin/bash
- C) #!/bin/bash
- D) \*#/bin/bash
- E) \$/sbin/bash

3. (FCC/TRT23/2016) Bento, administrador de um servidor com sistema operacional Linux, escreveu um shell script para automatizar o processo de backup do sistema. Para que apenas Bento possa executar o shell script criado, as permissões devem ser alteradas utilizando o comando chmod com o parâmetro

- A) 707.
- B) 660.
- C) 700.
- D) 006.
- E) 077.

4. (FCC/ELETRONBRAS-ELETROSUL/2016) Um profissional de TI está usando um computador com sistema operacional Linux que utiliza no shell o interpretador de comandos bash. Ele está logado como usuário teste e criou o seguinte arquivo shell script:



```
1 #!/bin/bash
2 echo 'Eletrosul- Centrais Elétricas S.A.'
3 $ variavel= 'Eu estou logado como usuário $user'
4 $ echo $variavel
```

Considerando que 1, 2, 3 e 4 indicam as linhas do arquivo e que este tenha sido salvo com o nome exemplo, é correto afirmar:

- A) Para o arquivo ser executável, é necessário acionar o comando `$ chmod +x exemplo`. Depois disto o arquivo poderá ser executado com `./exemplo`.
- B) A linha 1 indica que todas as outras linhas abaixo deverão ser executadas pelo compilador sh, que se localiza em `/bin/bash`.
- C) Após ser executado, o arquivo imprimirá na tela apenas frase "Eletrosul – Centrais Elétricas S. A. " utilizando o comando echo.
- D) Ao acionar o comando file arquivo é possível ver que a definição dele é Bourne-Again Shell Script, que se refere ao bash script.
- E) As linhas 3 e 4 farão com que seja impresso na tela Eu estou logado como usuário \$teste.

5. (FCC/SANASA/2019) Considerando o uso do Linux Ubuntu, é possível criar um shell script com objetivo de encadear comandos para realizar tarefas. Considere o trecho de shell script abaixo, criado em um diretório que contém arquivos.

Para mostrar o número de arquivos encontrados após a compilação e execução a lacuna I deve ser preenchida por

- A) `c=c+1`
- B) `c=$((c+1))`
- C) `c=#[c+1]`
- D) `c=Math(soma(c+1))`
- E) `c++`

6. (FCC/SANASA/2019) Em linha de comando, no Linux, para verificar os problemas e possíveis causas de erros em um script chamado sanasa.sh, que já possui permissão de execução, utiliza-se o comando

- A) `sh -x sanasa.sh`
- B) `kdsh -l sanasa.sh`



- C) `bashrun -y sanasa.sh`
- D) `chmod -x sanasa.sh`
- E) `ssh +x sanasa.sh`

7. (FCC/TJ-SC/2021) Uma Analista de Sistemas criou o script bash, para ser executado em ambiente Linux em condições ideais, cujo código é apresentado abaixo.

```
#!/bin/bash
```

```
DATA=`date +%y%m%d-%H%M`
```

```
cp $1 $DATA-$2
```

Considerando que o código não apresenta erros,

- A) a Analista deve atribuir permissão de execução ao script criado com o comando `chmod a+x copiar.ba`; considerando que o nome do arquivo de script seja `copiar.ba`.
- B) o script copia um arquivo e, durante a operação, acrescenta a data e a hora local ao nome do novo arquivo.
- C) o script copia um arquivo e, durante a operação, acrescenta a data e a hora local para o novo arquivo, após seu último registro.
- D) o script usa os objetos `$1` e `-$2`. `$1` representa o primeiro parâmetro passado na linha de comando usada para executar o script e o `-$2` representa uma cópia do primeiro parâmetro com um `-` no começo do nome.
- E) o bash não faz diferença entre aspas simples, aspas duplas e crases, por isso o comando da 2ª linha do código poderia ser escrito como `DATA="date +%y%m%d-%H%M"` sem alterar a lógica do script.



## GABARITO

01	02	03	04	05	06
B	C	C	A	B	A
07					
B					



## LISTA DE QUESTÕES – VUNESP

1. (VUNESP/CRO-SP/2015) No sistema operacional UNIX, uma aplicação pode ser desenvolvida por meio de um script que é um conjunto de comandos que podem ser executados pelo interpretador de comandos (shell). Considerando a existência do interpretador /bin/sh, a primeira linha de um arquivo de script deve conter:

- A) \*/bin/sh
- B) \$/bin/sh
- C) &/bin/sh
- D) #!/bin/sh
- E) #\*/bin/sh

2. (VUNESP/Câmara de Sertãozinho-SP/2019) Considere que o usuário de um computador com sistema operacional Linux deseja executar um comando em segundo plano (background) em um terminal Shell. Para efetivar essa ação, o usuário deve executar o comando

- A) e teclar a combinação Ctrl+c
- B) e teclar a combinação Ctrl+x
- C) seguido de bg
- D) seguido de &
- E) seguido de !

## GABARITO

01	02
D	D



## LISTA DE QUESTÕES – MULTIBANCAS

1. (CESGRANRIO/Petrobras/2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado.

Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.
- E) na linha de comando, após o nome do arquivo que contém o script.

2. (CEPERJ/Rio previdência/2014) Shell script é uma linguagem de script para Linux, nada mais do que comandos do próprio Linux que são executados em uma determinada sequência para uma determinada finalidade. Nesse contexto, duas situações são listadas a seguir.

I- No terminal ou modo gráfico, deseja-se criar um arquivo que possa ser editado para que se torne o primeiro shell script a ser criado, sendo necessário utilizar um comando CMD1.

II- Para que seja possível executar o shell script criado, é preciso atribuir a este o direito de execução; para isso é necessário usar um comando CMD2.

Exemplos de CMD1 e de CMD2 são, respectivamente:

- A) touch shell1.sh e exec +x shell1.sh
- B) touch shell1.sh e chmod +x shell1.sh
- C) create shell1.sh e chmod +x shell1.sh
- D) new shell1.sh e chmod +x shell1.sh
- E) new shell1.sh e exec +x shell1.sh

3. (FUMARC/AL-MG/2014) No Linux, considere a existência de um arquivo "arquivo.txt" no diretório corrente, com o seguinte conteúdo:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Deseja-se produzir, a partir do arquivo, a seguinte saída:





1 - 10 2 - 20 3 - 30

O script shell usado com comandos AWK que produz a saída desejada é:

- A) `awk arquivo.txt '{ print $2" - "$4 }'`
- B) `cat arquivo.txt | awk '{ print $2" - "$4 }'`
- C) `cat arquivo.txt | awk '{ print $1" - "$3"\n" }'`
- D) `cat arquivo.txt | awk '{ printf($2" - "$4) }'`

4. (INSTITUTO AOCP/EBSERH/2016) Em sistemas operacionais Linux é possível criar scripts para automatizar tarefas rotineiras. A extensão de arquivo utilizada para Shell Script é

- A) .bin
- B) .sl
- C) .bash
- D) .sh
- E) .gz

5. (FUNECE/UECE/2017) Considere o seguinte script Bash prog.sh em um sistema operacional Linux. A numeração das linhas à esquerda é meramente ilustrativa.

```
1.  #!/bin/bash
2.  x=-9
3.  echo -n "( "
4.  exit x
5.  echo -n x
6.  echo -n " )"
```

Assinale a opção que indica corretamente o que acontece após a execução do comando abaixo.  
`./prog.sh && echo " fim"`

- A) Será exibido `"(-9) fim"`.
- B) Serão mostrados `"( "` e uma mensagem de erro na linha 4, afirmando que x não é um argumento numérico do comando exit.
- C) Será exibido `"( "` seguido do final com sucesso da execução do script.
- D) Serão mostrados `"( "` e uma mensagem de erro na linha 4, afirmando que a variável x não pode ser negativa.



6. (AOCP/UFOB/2018) Shellscrip é uma linguagem de scrip utilizada em diversos sistemas operacionais. Sobre Shellscrip, incluindo Scrip Bash, Groovy e Powershell, julgue o item a seguir.

Para que um arquivo .sh seja executado em um sistema Linux, este deve ter permissão de escrita para o usuário atual.

7. (SUGEP-UFRPE/UFRPE/2018) Um scrip executável no interpretador de comandos Bash do GNU/Linux deve começar com os seguintes caracteres na primeira linha:

- A) { /bin/bash }
- B) #/bin/bash
- C) #!bin/bash
- D) <scrip language="bash">
- E) #!/bin/bash

8. (CONSESP/Câm. de Castelo-ES/2018) Sob a utilização de uma distribuição Linux, leia as afirmações que se seguem.

I. Um scrip de shell é um grupo de comandos, funções, variáveis, ou qualquer outra coisa que pode ser usado em um shell.

II. Estes itens (grupo de comandos, funções, variáveis, ou qualquer outra coisa que pode ser usado em um shell) são digitados em um arquivo de texto simples que pode ser executado como um comando.

III. A maioria dos sistemas Linux utiliza scrips de shell de inicialização do sistema para executar comandos necessários para fazer os serviços funcionarem, durante a inicialização do sistema.

IV. Scrips de shell estão presentes na criação do kernel e não podem ser criados pelos usuários.

Assinale a alternativa correspondente.

- A) Todas as afirmações estão corretas.
- B) Somente três das afirmações estão corretas.
- C) Somente duas das afirmações estão corretas.
- D) Nenhuma das afirmações está correta.

9. (FURB/Pref. de Massaranduba-SC/2020) Em ambiente Linux, observa-se:

- 1. #!/bin/bash



2. `i=3`
3. `k=2`
4. `let j=k**i`
5. `echo $j`
6. `exit 0`

Isso posto, ao executar o script bash acima, obter-se-á a seguinte saída:

- A) 6.
- B) NaN.
- C) 8.
- D) 0.
- E) 2.

10. (FUNDATEC/PGE-RS/2021) Um usuário criou um arquivo chamado "script.sh" no seu "home directory" de uma instalação padrão do Linux Ubuntu 18 Desktop. Esse arquivo contém alguns comandos que ele costuma executar com frequência, mas, sempre que tenta executar o script, ele obtém uma mensagem de erro. Ele pediu para alguém ajudá-lo e enviou uma cópia dos comandos que executou e seus retornos, como segue abaixo:

```
$ cd
$ ls -l script.sh
-rwxr-xr-x 1 user user 11 Oct 03 18:31 script.sh
$ script.sh
script.sh: command not found
```

Com base nos comandos apresentados pelo usuário, qual seria uma forma de executar o script.sh sem erros?

- A) `chmod a+x script.sh; script.sh`
- B) `./script.sh`
- C) `.script.sh`
- D) `./script.sh`
- E) `../script.sh`



11. (FUNDATEC/PROCERGS/2023) Analise as assertivas abaixo sobre shell scripts no sistema operacional Linux:

I. Uma maneira de criar um literal em um shell script e fazer com que o shell deixe a string intocada é colocar a string toda entre aspas simples.

II. Um shell script deve sempre iniciar pela sequência de caracteres "barra" e "asterisco".

III. Um caractere "cifrão" no início de uma linha indica que a linha é um comentário, ou seja, o shell script irá ignorar tudo o que estiver na linha após o caractere.

Quais estão corretas?

- A) Apenas I.
- B) Apenas I e II.
- C) Apenas I e III.
- D) Apenas II e III.
- E) I, II e III.

12. (FEPESE/Pref. de Balneário Camboriú-SC/2023) Qual o nome do comando ou argumento do loop while em programação shell script que possibilita o parse, ou o correto processamento e obtenção dos valores passados como argumentos na forma de opções, ou flags; isto é, na forma de letras precedidas por hífen (por ex. sh ./meuscript.sh -a abc -b def), ao invocar, via linha de comando, um shell script bash no Linux?

- A) arglist
- B) getopt
- C) parseopts
- D) parseargs
- E) optsparse

13. (UNIFAL-MG/UNIFAL-MG/2023) Considere a seguinte tela de terminal em um servidor Linux Debian:

```
usuario@servidor:~/prova$ ls -l
total 172
-rw-r--r-- 1 usuario usuario 28672 ago  9 2022 Bibliografia.doc
-rw-r--r-- 1 usuario usuario 34463 ago  9 2022 Bibliografia.odt
-rw-r--r-- 1 usuario usuario  2402 jul 23 2022 bibliografia.txt
-rw-r--r-- 1 usuario usuario 49664 ago  9 2022 prova.doc
-rw-r--r-- 1 usuario usuario 32748 ago  7 2022 prova.odt
-rw-r--r-- 1 usuario usuario  8122 ago  6 2022 prova.txt
```



Considere ainda, o seguinte script bash criado pelo usuário (os números apenas identificam as linhas e não fazem parte do script):

```
1 #!/bin/bash
2 cd ~
3 *** -czf /tmp/prova.tgz prova/*.doc prova/*.odt prova/*.txt
4 *** go-rwx /tmp/prova.tgz
5 *** -C /tmp/prova.tgz usuario@123.123.123.123:~/backup/prova.tgz
6 *** /tmp/prova.tgz
7 exit
```

São, respectivamente, comandos corretos para substituir os "\*\*\*\*" nas linhas 3, 4, 5 e 6 do script, de acordo com a descrição dos comandos abaixo:

- I. comando que funciona como um "arquivador" (ele junta vários arquivos em um só) mas pode ser usado em conjunto com um compactador (como o gzip ou zip);
- II. comando utilizado para mudar a permissão de acesso a um arquivo ou diretório;
- III. comando que permite a cópia de arquivos entre o cliente/servidor ssh;
- IV. comando utilizado para apagar arquivos e também pode ser usado para apagar diretórios e subdiretórios vazios ou que contenham arquivos.

- A) tac, pwd, ftp, grep
- B) cmp, tail, rsh, clear
- C) tar, chmod, scp, rm
- D) cat, chattr, sync, cut

## GABARITO

01	02	03	04	05	06
A	B	B	D	B	Errado
07	08	09	10	11	12
E	B	C	D	A	B
13					
C					



## LISTA DE QUESTÕES – CEBRASPE

1. (CEBRASPE/ANTT/2013) No ambiente Linux, um usuário comum pode terminar seu próprio processo por meio do comando kill, ação que não se restringe ao superusuário.

2. (CEBRASPE/TJ-SE/2014) Alguns programas podem apresentar problemas que resultem no travamento do sistema operacional, o que pode ser resolvido, no Linux, por meio do comando Kill, que finaliza o processo, funcionalidade que pode ser acessada por meio de outro terminal.

3. (CEBRASPE/MEC/2015) Julgue o item a seguir, relativo ao sistema operacional.

No Linux, se o processo pai alterar suas variáveis após a criação do processo filho, o filho detecta automaticamente a mudança, pois eles compartilham a mesma imagem da memória privada.

4. (CEBRASPE/STJ/2015) Julgue o próximo item, que se refere a sistemas operacionais.

No Linux, o comando `renice -10 -u jdev -p 650` pode ser usado para reiniciar, no sistema, todos os processos do usuário `jdev` e o processo PID 650.

5. (CEBRASPE/MPE-PI/2018) Julgue o item a seguir, acerca de sistemas operacionais.

Em sistemas operacionais Linux, os processos podem comunicar-se com outros processos por interrupções de software, que são canais entre os processos nos quais um processo pode escrever um fluxo de bytes para outro processo ler.

6. (CEBRASPE/FUB/2018) Julgue o item seguinte, a respeito dos sistemas operacionais Windows e Linux.

O Linux é um sistema operacional em que cada usuário consegue ter apenas um processo ativo por vez, processo esse que é iniciado automaticamente quando o sistema é carregado.

7. (CEBRASPE/EBSERH/2018) No sistema operacional Linux, é possível alterar a prioridade de um processo já iniciado com o uso do comando `nice`.



8. (CEBRASPE/TJ-PA/2020) No Linux, um processo, por si só, não é elegível para receber tempo de CPU. Essa ação depende, basicamente, do seu estado da execução. O processo está administrativamente proibido de executar no estado

- A) pronto.
- B) dormente.
- C) executável.
- D) parado.
- E) zumbi.

9. (CEBRASPE/TJ-RJ/2021) O sistema de monitoramento de ativos computacionais alertou quanto à elevada quantidade de processos que estavam rodando em um servidor com o sistema operacional RedHat Enterprise Linux. Ao acessar diretamente pela linha de comando, foram identificados vários processos filhos sem a presença de pais, com consumos pequenos de recursos, tratando-se de processos zumbis.

Considerando essa situação hipotética, assinale a opção que apresenta os comandos utilizados no RedHat Enterprise Linux 5, 6 ou 7 para identificar os PIDs dos processos zumbis.

- A) `ps aux | awk {'print $8'} wc -l`
- B) `kill PID zumbi`
- C) `ps aux | awk '$8=="Z" { print $0 }'`
- D) `abrt-cli info zumbi`
- E) `abrt-cli list zumbi`

## GABARITO

01	02	03	04	05	06
Certo	Certo	Errado	Errado	Errado	Errado
07	08	09			
Errado	D	C			



## LISTA DE QUESTÕES – FGV

1. (FGV/TJ-RO/2015) Um desenvolvedor Java construiu um programa que, ao ser executado em sua máquina Linux, entrou em loop infinito. Considerando que há outros processos java rodando nessa máquina, o(s) comando(s) que esse desenvolvedor deveria executar para terminar a execução deste processo é(são):

- A) top e free;
- B) killall;
- C) ps e kill;
- D) ls e kill;
- E) shutdown.

2. (FGV/IBGE/2017) Em ambiente Linux, o comando ps -axl permite visualizar informações sobre todos os processos existentes no sistema, em especial, os estados dos processos. Se o estado informado é Z, isto indica que o processo:

- A) é recém-criado e está pronto para executar;
- B) está no estado bloqueado, aguardando por um recurso;
- C) está morto, aguardando que o processo pai providencie sua extinção;
- D) está órfão, aguardando reparentização;
- E) sofreu preempção.

3. (FGV/IBGE/2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:

- A) create\_process;
- B) new\_process;
- C) fork;
- D) spawn;
- E) duplicate.





4. (FGV/Prefeitura de Niterói-RJ/2018) Sobre o gerenciamento de processos no sistema operacional Linux, assinale a afirmativa correta.

- A) O comando ps lista os processos em execução no sistema, trazendo informações sobre o quanto de processamento ou de memória cada um deles está consumindo.
- B) O ID do processo não pode ser reutilizado depois que o processo termina.
- C) Por meio do comando renice é possível alterar a prioridade de execução de um processo.
- D) Por padrão, os processos executados por um usuário iniciam com nível de prioridade -20, isto é, a prioridade mais baixa possível.
- E) Processos órfãos não possuem PPID.

5. (FGV/MPE-AL/2018) Sobre gerenciamento de serviços e processos no sistema operacional Linux, analise as afirmativas a seguir.

- I. Processo órfão é todo processo que já finalizou a execução, mas ainda possui uma entrada na tabela de processos.
- II. O PID usado por um processo fica indisponível para reutilização enquanto o processo está presente na tabela de processos.
- III. Daemons são processos zombies que rodam em background e não possuem um PPID associado.

Está correto o que se afirma em

- A) I, apenas.
- B) II, apenas.
- C) III, apenas.
- D) II e III, apenas.
- E) I, II e III.

6. (FGV/MPE-AL/2018) Sobre gerenciamento de processos no sistema operacional Linux, analise as afirmativas a seguir.

- I. O PID do processo init é automaticamente atribuído ao PPID de processos que se tornam órfãos.



II. Processo zombie é a nomenclatura atribuída ao processo que possui autonomia própria, comportamento errático e que nunca termina sua execução, consumindo recursos do sistema indistintamente.

III. A prioridade de execução de um processo é indicada por um número que tem variação de -20 a 19 e quanto menor for este número, maior será a prioridade do processo.

Está correto o que se afirma em

- A) I, apenas.
- B) II, apenas.
- C) III, apenas.
- D) I e III, apenas.
- E) I, II e III.

7. (FGV/Banestes/2021) No Linux, a função fork cria um novo processo, referido como filho, pela duplicação do processo que inicia a operação, referido como pai, de tal forma que:

- I. O processo filho herda o PID do processo pai.
- II. Pai e filho rodam (run) em espaços de memória diferentes.
- III. O filho herda todos os sinais de tempo do pai, no estado em que se encontram na execução da função fork.
- IV. Em caso de sucesso, a função fork retorna zero no processo pai.

O número de afirmativas corretas é:

- A) zero;
- B) uma;
- C) duas;
- D) três;
- E) quatro.

8. (FGV/TRT16/2022) No sistema operacional Linux, um técnico em informática digita o comando ps aux em uma janela de terminal. O nome da coluna que exibe o status dos processos é

- A) STAT.



- B) START.
- C) STATS.
- D) PID.
- E) CMD.

9. (FGV/TJ-TO/2022) A ferramenta utilitária do sistema operacional GNU/Linux que exibe os processos em execução no sistema que mais utilizam a CPU é:

- A) df;
- B) grep;
- C) sed;
- D) stat;
- E) top.

## GABARITO

01	02	03	04	05	06
C	C	C	C	B	D
07	08	09			
B	A	E			



## LISTA DE QUESTÕES – VUNESP

1. (VUNESP/UNESP/2013) Os sistemas operacionais FreeBSD, Linux e Unix compartilham grande parte dos comandos, pois são originários de um mesmo conceito. Dentre esses comandos comuns, está o que permite a apresentação dos estados dos processos correntes, denominado

- A) cd
- B) lpr
- C) ls
- D) ps
- E) proc

2. (VUNESP/Pref. de P. Prudente-SP/2016) O administrador de um servidor com sistema operacional Linux deseja alterar a prioridade de execução de um processo. Para isso, ele deve utilizar o comando:

- A) ps
- B) top
- C) lsof
- D) pstree
- E) renice

3. (VUNESP/TJM-SP/2017) O administrador do computador com sistema operacional Linux executou o comando `kill -9 254`, no prompt de comandos de um terminal shell, para, corretamente,

- A) enviar o sinal de interrupção para o processo 254, para avisar que ele será eliminado em 9 segundos.
- B) interromper e eliminar imediatamente o processo de número 254.
- C) enviar o sinal de interrupção para 9 processos a partir do processo de número 254.
- D) sinalizar todos os processos ativos no sistema para interrupção imediata e eliminação em 254 segundos.
- E) eliminar todos os processos executados pelo administrador após 9 segundos.



4. (VUNESP/Câm. de Indaiatuba-SP/2018) O administrador de um servidor com sistema operacional Linux executou o seguinte comando no prompt do terminal bash:

```
sudo renice -20 -p 1024
```

O resultado da execução desse comando será

- A) a alteração da resolução do ambiente gráfico para 1024 pixels.
- B) a apresentação de um novo terminal gráfico de cor verde e 1024 pixels de resolução.
- C) a redução da prioridade de execução do processo 1024.
- D) o aumento da prioridade de execução do processo 1024.
- E) o deslocamento da posição do ambiente gráfico para a esquerda.

5. (VUNESP/UNICAMP/2022) No sistema operacional Linux, o comando top pode exibir uma coluna cujo título é TIME+. No modo não-cumulativo, essa coluna exibe

- A) o tempo transcorrido desde o início da tarefa até o instante corrente.
- B) o tempo total de CPU que a tarefa usou desde seu início.
- C) a porcentagem total de tempo de CPU usada pela tarefa desde a última atualização das informações na tela.
- D) a porcentagem total de tempo de CPU usada pela tarefa desde seu início.
- E) o tempo total de ociosidade da tarefa desde que ela usou a CPU pela última vez.

## GABARITO

01	02	03	04	05
D	E	B	D	B



## LISTA DE QUESTÕES – MULTIBANCAS

1. (CS-UFG/UFG/2019) Com respeito aos processos em execução no sistema Linux, o comando para listar processos em execução, sem se limitar aos processos executados a partir do shell corrente, é:

- A) ps
- B) ps -g
- C) ps -e
- D) ps -y

2. (COMPERVE/Câm. de Parn.-RN/2019) Um processo é uma instância de um software (programa) em execução utilizando recursos do computador, tais como, processador e memória, para desempenhar as tarefas atribuídas a ele. Os sistemas operacionais oferecem ferramentas para gerenciamento de processos. Uma das formas de realizar esse gerenciamento é através de linha de comando utilizando um prompt de comando.

Sobre o gerenciamento de processos no Linux Ubuntu 18.04, utilizando um prompt de comando, considere as descrições de execução de comandos apresentadas abaixo.

I Lista os processos em ordem decrescente de consumo de recursos.

II Interrompe o processo de forma definitiva. É necessário informar o PID.

III Define a prioridade de um processo. Pode atualizar a prioridade de um processo existente ou criar um novo processo com prioridade definida.

I, II e III referem-se, respectivamente, aos comandos

- A) top, kill e nice.
- B) kill, top e ps.
- C) ps, nice e kill.
- D) top, ps e nice.

3. (Avança SP/Câm. de R. Pires-SP/2021) Assinale a alternativa que apresenta um comando do Linux Ubuntu capaz de “matar” um processo travado por meio do PID:

- A) kill.
- B) force.



- C) mix.
- D) death.
- E) exit.

4. (AOCP/BANESE/2022) O sistema operacional é responsável pela definição de quais processos têm prioridade sobre a CPU. No sistema operacional Linux, distribuição Ubuntu, qual comando pode ser utilizado para alterar a prioridade de execução de um programa?

- A) jobs
- B) nice
- C) ps
- D) top
- E) fg

5. (CS-UFG/UFG/2023) Dentre os comandos para a manutenção de sistemas Linux, um comando que apresenta os processos em execução, com opções para fornecer o nome de usuário e a hora de início de processo, é o comando

- A) which.
- B) vim.
- C) ping.
- D) ps.

## GABARITO

01	02	03	04	05
C	A	A	B	D



## LISTA DE QUESTÕES - CEBRASPE

1. (CEBRASPE/TCE-RN/2015) O Linux tem a capacidade de montar sistemas de arquivos do dispositivo `/dev/sdc3`, que tem o sistema de arquivos do tipo `ext4` e, no ponto de montagem `/teste`, deve ser utilizado o comando `mount -t ext4 /dev/sdc3 /teste`.

2. (CEBRASPE/TCE-PA/2016) A opção `mount -t vfat` permite fazer o backup dentro de uma partição de disco DOS/Windows, caso em que a hierarquia de diretório aparece como parte do sistema de arquivos Linux.

3. (CEBRASPE/ABIN/2018) O Linux não impede a alteração do nome do superusuário, nem a criação de contas com UID igual a 0, embora essas ações não sejam recomendadas.

4. (CEBRASPE/CODEVASF/2021) Com relação à administração do sistema operacional Linux, julgue o item seguinte.

Para atribuir permissões de leitura, escrita e execução ao proprietário do arquivo de nome `documento1.txt`, sem qualquer permissão para grupos ou qualquer outro usuário, é correto o uso do comando `chmod 0177 documento1.txt`.

5. (CEBRASPE/TRT8/2022) Um administrador de rede precisa modificar as permissões na lista de controle de acesso, utilizando o Red Hat Enterprise Linux 8, para o arquivo TRT8 de propriedade do usuário `root`, que, por sua vez, pertence ao grupo `root`. Nesse caso, apenas Paulo deve ter permissão `rw-` e os demais usuários devem ter permissão `r--`.

Com base nessa situação hipotética, assinale a opção que indica o comando correto para executar a configuração referida.

- A) `# chmod rw- paulo TRT8`
- B) `# getfacl TRT8 -m u:paulo rw`
- C) `# setfacl -m u:paulo:rw- TRT8`
- D) `# setfacl TRT8 -m u:paulo rw`
- E) `# chmod paulo 640 TRT8`





6. (CEBRASPE/POLITEC-RO/2022) Em Linux, o comando equivalente a `chmod a-rwx,u+r,g+rw arquivo` para alterar as permissões de um arquivo é

- A) `chmod 465 arquivo`.
- B) `chmod 456 arquivo`.
- C) `chmod 460 arquivo`.
- D) `chmod 654 arquivo`.
- E) `chmod 046 arquivo`.

7. (CEBRASPE/PO-AL/2023) A respeito de sistemas operacionais, julgue o item subsecutivo.

O sistema Linux possibilita atribuir permissões de acesso diferentes em um mesmo diretório para distintas categorias: usuário dono, grupo dono do arquivo e demais usuários.

## GABARITO

01	02	03	04	05	06
Certo	Certo	Certo	Errado	C	C
07					
Certo					



## LISTA DE QUESTÕES - FGV

1. (FGV/PGE-RO/2015) Para verificar quais sistemas de arquivos estão correntemente montados em ambiente Linux, o administrador deve utilizar o comando:

- A) fsmount;
- B) mountfs;
- C) mount;
- D) showfs;
- E) fsck.

2. (FGV/SEPOG-RO/2017) Considere o comando Linux a seguir aplicado a um arquivo regular chamado teste:

```
chmod 546 teste
```

As novas características do arquivo chamado teste, após a execução do comando, serão

- A) permissão de escrita para todos os usuários e execução somente pelo proprietário e pelos membros do grupo.
- B) permissão de leitura para todos os usuários, execução somente pelo proprietário e escrita pelos outros usuários.
- C) somente o proprietário pode executar, os componentes do grupo podem ler e escrever e os outros usuários somente podem listar seu nome no diretório.
- D) todos os usuários podem executar o arquivo, somente o proprietário pode ler e escrever e os componentes do grupo podem ler.
- E) todos os usuários podem ler e executar mas somente o proprietário pode modificar o arquivo.

3. (FGV/Prefeitura de Niterói-RJ/2018) Pedro é o proprietário do arquivo header.txt em um sistema Linux e gostaria de assegurar que somente ele tivesse permissão de leitura, gravação e execução a este arquivo, enquanto que todos os demais usuários com acesso ao sistema tivessem somente a permissão de leitura.

Assinale a opção que indica o comando que pode ser usado para conseguir esse objetivo.

- A) chmod ug+r header.txt
- B) chmod 766 header.txt
- C) chmod 722 header.txt



- D) `chmod +r header.txt`
- E) `chmod 744 header.txt`

4. (FGV/MPE-AL/2018) Para definir o script `/usr/local/bin/meuscript` como executável no sistema operacional Linux, devemos usar o comando

- A) `chmod og /usr/local/bin/meuscript`
- B) `chmod 000 /usr/local/bin/meuscript`
- C) `chmod +x /usr/local/bin/meuscript`
- D) `chmod ugo-x /usr/local/bin/meuscript`
- E) `chmod 666 /usr/local/bin/meuscript`

5. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando que permite estabelecer permissões para arquivos e diretórios.

- A) `chmod`
- B) `chper`
- C) `dir`
- D) `permit`
- E) `useradd`

6. (FGV/Câmara de Taubaté-SP/2022) No ambiente Linux, assinale a função do comando `sudo`.

- A) Buscar na Internet atualizações para o sistema operacional.
- B) Configurar as opções de idiomas e padrões de codificação de caracteres de texto.
- C) Executar comandos com privilégios mais altos, ou como outro usuário.
- D) Gerenciar as contas e permissões de usuários.
- E) Remover arquivos temporários e memória de trabalho, tanto originados pelo uso da Internet como dos aplicativos comuns.



## GABARITO

01	02	03	04	05	06
C	B	E	C	A	C



## LISTA DE QUESTÕES - FCC

1. (FCC/TRT4/2015) A possibilidade de compartilhar arquivos entre diferentes sistemas operacionais é fundamental para aumentar a produtividade computacional. A montagem automática de uma partição com sistema de arquivos CIFS, durante o boot do servidor com sistema operacional Linux, deve ser configurada no arquivo

- A) /etc/fstab.
- B) /boot/mount.
- C) /etc/mount.
- D) /boot/inittab.
- E) /etc/initd.

2. (FCC/TRT23/2016) Uma partição NFS remota deve ser montada em um computador com sistema operacional Linux. Para especificar, no comando mount, que a partição é NFS, deve-se utilizar a opção:

- A) -n.
- B) -f.
- C) -i.
- D) -s.
- E) -t.

3. (FCC/TJ-MA/2019) Um Analista digitou o comando `chmod u=rwx,g=rx,o=r processo` para definir as permissões de acesso ao arquivo processo. O comando equivalente usando a notação octal é:

- A) `chmod 713 processo`
- B) `chmod 777 processo`
- C) `chmod 134 processo`
- D) `chmod 754 processo`
- E) `chmod 671 processo`

4. (FCC/SANASA Campinas/2019) No Linux, para definir somente para o owner do arquivo teste.txt apenas a permissão de read e write, utiliza-se o comando



- A) chmod 700 teste.txt
- B) chmod 724 teste.txt
- C) chmod 666 teste.txt
- D) chmod 600 teste.txt
- E) chmod 777 teste.txt

5. (FCC/TRF4/2019) No Linux, para atribuir somente permissão de leitura e escrita para o dono do arquivo trf4r.sh e somente leitura para usuários do mesmo grupo, utiliza-se o comando

- A) chmod 640 trf4r.sh
- B) chmod 755 trf4r.sh
- C) chmod 777 trf4r.sh
- D) chmod 341 trf4r.sh
- E) chmod 666 trf4r.sh

6. (FCC/TRT5/2022) Um analista, utilizando o usuário padrão definido no Red Hat, listou o conteúdo de um diretório após ter criado os objetos arquivo.txt e diretório, obtendo a saída abaixo.

```
-rw--w-r-- 1 analista analista 0 Oct 12 18:32 arquivo.txt  
drwx-wxr-x 2 analista analista 4096 Oct 12 18:32 diretório
```

Com base nesta saída, o valor do umask foi definido em

- A) 0002.
- B) 0244.
- C) 0202.
- D) 0042.
- E) 0642.



## GABARITO

01	02	03	04	05	06
A	E	D	D	A	D



## LISTA DE QUESTÕES - VUNESP

1. (VUNESP/PRODEST-ES/2014) Nos sistemas operacionais Linux, o programa "sudo" pode ser utilizado para que um usuário possa executar comandos com privilégios diferentes dos definidos para seu usuário. No entanto, para o seu correto funcionamento, é fundamental que o usuário em questão

- A) esteja autorizado nas configurações do "sudo".
- B) seja membro do grupo "admin".
- C) seja o único usuário autenticado na máquina.
- D) possua um único terminal aberto.
- E) conheça a senha do superusuário (root).

2. (VUNESP/Câm. de Itatiba-SP/2015) Um novo disco rígido foi adquirido, instalado, particionado e formatado em um computador com sistema operacional Linux. Para que a partição do disco rígido seja montada automaticamente quando o sistema operacional é inicializado, o comando de montagem dessa partição deve ser incorporado no arquivo

- A) dconf.
- B) fstab.
- C) init.
- D) mount.
- E) udisk.

3. (VUNESP/Pref. de Birigui-SP/2019) No ambiente Linux, considere a seguinte permissão:

--WX-WX---

Ela equivale, em notação octal, a:

- A) 0220
- B) 0330
- C) 0440
- D) 0550
- E) 0770





4. (VUNESP/Câmara de Mauá-SP/2019) Um usuário de um computador com sistema operacional Linux criou um arquivo chamado backup.sh, que contém um script para efetuar uma rotina de backup de arquivos importantes da máquina. Porém, mesmo sendo dono do arquivo backup.sh e tendo permissão de leitura e escrita, o usuário observou que não consegue executar esse script por falta de permissão de execução.

Assinale a alternativa que contém um comando a ser executado pelo mesmo usuário que resolveria esse problema.

- A) `chmod u-wx backup.sh`
- B) `chmod ugo-x backup.sh`
- C) `chmod 744 backup.sh`
- D) `chmod 666 backup.sh`
- E) `chmod 644 backup.sh`

5. (VUNESP/UNICAMP/2022) Por meio da execução do comando `ls -l` em um terminal Linux, observou-se que o arquivo myfile apresenta as seguintes permissões:

```
-rw-rw-r--
```

Após a execução com sucesso do comando `chmod 644 myfile` as permissões de myfile passam a ser:

- A) `-r--r--r--`
- B) `-r--rw-rw-`
- C) `-rw-rw-rw-`
- D) `-rw-r--r--`
- E) `-rw-rw-r-`

6. (VUNESP/Pref. de Sorocaba-SP/2022) Em um computador com o sistema operacional Linux, um subdiretório contém um arquivo chamado arq.sh. Nesse subdiretório, o usuário dono do arquivo, que não possui privilégios de root, executou com sucesso o seguinte comando em um terminal:

```
chmod ugo-rwx arq.sh
```

Em seguida, o comando a seguir foi executado:

```
cat arq.sh
```

O resultado obtido foi



- A) a exibição do conteúdo de arq.sh.
- B) a exclusão de arq.sh.
- C) uma mensagem de erro informando que não há permissão.
- D) a execução de arq.sh, uma vez que é um arquivo de shell script
- E) a exibição do conteúdo de arq.sh concatenado com ele mesmo.

7. (VUNESP/Pref. de Pindamonhangaba-SP/2023) Um usuário de sistema operacional Linux utilizou o comando `chmod 321` em um determinado arquivo. Assinale a alternativa que apresenta as permissões do arquivo após o comando.

- A) `r--w---x`
- B) `rwxrw-r--`
- C) `-wx-w---x`
- D) `rwx-wx--x`
- E) `rw--w-r-`

## GABARITO

01	02	03	04	05	06
A	B	B	C	D	C
07					
C					



## LISTA DE QUESTÕES - CESGRANRIO

1. (CESGRANRIO/B. da Amazônia/2022) O administrador de um sistema operacional Linux mudou o atributo de dono (owner) do arquivo de um programa para o usuário root e o atributo de grupo desse arquivo para root.

O comando a ser utilizado para que esse programa seja executado com permissão de root por qualquer usuário do sistema é o

- A) chmod 0555
- B) chmod 1555
- C) chmod 2555
- D) chmod 3555
- E) chmod 4555

## GABARITO

01

E



## LISTA DE QUESTÕES - MULTIBANCAS

1. (FUNCAB/MDA/2014) No sistema operacional Linux, a criação de um grupo permite que um conjunto de usuários diferentes possua acesso a um mesmo arquivo. O comando do Linux que tem como objetivo alterar o grupo de um arquivo/diretório é o

- A) chown.
- B) chgroup.
- C) chset.
- D) chfile.
- E) chgrp.

2. (UFRRJ/UFRRJ/2015) Qual o comando para agregar uma mídia formatada em NTFS no sistema de arquivos de um sistema operacional UNIX-like?

- A) `umount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.`
- B) `mount /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.`
- C) `umout /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.`
- D) `mount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.`
- E) `umount /dev/diskid/DISK-JP1440HA196TKS -t ntfs /mnt.`

3. (FEPESE/Pref. de Lages-SC/2016) Ao tentar recuperar um sistema Linux Ubuntu, o administrador de sistemas constata que a partição /boot foi montada em modo somente leitura.

Assinale a alternativa cujo comando possibilitará a gravação de arquivos nesta partição.

- A) `mount -o remount,rw /boot`
- B) `mount -a remount,rw /boot`
- C) `mount -a remount,ro /boot`
- D) `mount -a remount,wr /boot`
- E) `mount -o remount,raw /boot`

4. (FCM/IF Sudeste-MG/2016) O Linux permite ao superusuário (root) executar qualquer operação válida em qualquer arquivo ou processo, inclusive algumas chamadas de sistema



podem ser executadas somente pelo superusuário. A operação que NÃO é exclusiva do superusuário é

- A) configurar interfaces de rede.
- B) configurar o relógio do sistema.
- C) alterar as permissões de um arquivo.
- D) aumentar os limites de uso dos recursos.
- E) definir o nome de host (hostname) do sistema.

5. (IBFC/EBSERH/2016) O comando sudo do sistema operacional Linux é muito poderoso, permitindo que usuários comuns obtenham privilégios de super usuário. Por questões de segurança, o administrador precisa definir no arquivo \_\_\_\_\_, quais usuários podem executar sudo, em quais computadores podem fazê-lo e quais comandos podem executar através dele. Assinale a alternativa que complete correta e respectivamente a lacuna:

- A) /etc/sudoers
- B) /root/sudouser
- C) /root/sudoers
- D) /usr/sudouser
- E) /etc/sudouser

6. (INAZ do Pará/DPE-PR/2017) Qual a alternativa que corresponde à linha de comando para que usuários comuns possam utilizar o comando administrativo apt-get dist-upgrade?

- A) # apt-get dist-upgrade
- B) \$ sudo apt-get gw dist-upgrade
- C) # sudo apt-get gw up dist-upgrade
- D) \$ sudo apt-get dist-upgrade
- E) # sudo apt-get dist-upgrade gw now

7. (IF-ES/IF-ES/2019) Os administradores de redes precisam entender com clareza sobre o gerenciamento de contas de usuários nos Sistemas Operacionais de Servidores Linux. Em relação ao sistema de contas Linux, analise as afirmativas abaixo:

I – O arquivo /etc/shadow serve para manter senhas criptografadas seguras quanto a acesso não autorizado.



II – Um novo usuário pode ser adicionado utilizando-se ferramentas, como adduser ou useradd.

III – Os usuários dos sistemas Linux recebem um número UID referente a sua identificação, porém o usuário root, por questões de segurança, não dispõe de UID.

É CORRETO o que se afirma em:

- A) I, II e III.
- B) I e II, apenas
- C) I e III, apenas.
- D) II e III, apenas.
- E) Nenhuma das afirmativas está correta.

8. (CS-UFG/IF Goiano/2019) Na configuração do Sistema Linux, o comando `chmod` modifica as permissões de arquivos com respeito à execução, leitura e escrita. A expressão `chmod 751 arq2` é equivalente a:

- A) `chmod u=rx,g=rwx,o=x arq2`
- B) `chmod u=rx,g=rwx,o=r arq2`
- C) `chmod u=rwx,g=rx,o=x arq2`
- D) `chmod u=rwx,g=rx,o=r arq2`

9. (CCV-UFC/UFC/2019) Qual dos itens abaixo contém o comando do sistema operacional Linux capaz de mudar o grupo de um arquivo ou diretório do sistema?

- A) `df`
- B) `sed`
- C) `chown`
- D) `chmod`
- E) `passwd`

10. (COSEAC/UFF/2019) No Sistema Operacional Linux o diretório local padrão do superusuário é:

- A) `/home`.
- B) `/usr`.



- C) /root.
- D) /lib.
- E) /dev.

11. (CONSULPLAN/MPE-PA/2022) Um administrador de sistema Linux, conectado como superusuário, realizou o seguinte comando: `chmod 751 /home/file.sh`. Sobre tal comando, analise as afirmativas a seguir.

- I. O usuário dono do arquivo terá permissão para ler, escrever e executar o arquivo.
- II. Os usuários do grupo do arquivo terão permissão para ler e executar o arquivo.
- III. Todos os demais usuários (exceto os citados anteriormente) não terão nenhum tipo de permissão sobre o arquivo.

Está correto o que se afirma em

- A) I, II e III.
- B) I e II, apenas.
- C) I e III, apenas.
- D) II e III, apenas.

12. (IBADE/RBPREV/2023) Qual a função do comando `chmod go-rx`, utilizado no terminal do sistema operacional Linux?

- A) Dar permissão de leitura, execução e escrita para todos os usuários.
- B) Remover a permissão de escrita e execução aos usuários do grupo e outros.
- C) Dar permissão de leitura e execução para todos os usuários.
- D) Remover a permissão de leitura e execução aos usuários do grupo e outros.
- E) Remover a permissão apenas de execução aos usuários do grupo e outros.

13. (Quadrix/CREFITO-7ª Região/2023)



```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
crefito@desk12:/bkp/crefito7$ ls -la
total 848
drwxrwxrwx 2 root    root    4896 jun  5 15:19 .
drwxr-xr-x 3 root    root    4896 jun  5 15:13 ..
-rw-rw-r-- 1 crefito crefito 147180 mai 26 11:30 anuidades.pdf
-rw-rw-r-- 1 crefito crefito 211550 jun  1 16:18 crefito_semanal.zip
-rw-rw-r-- 1 crefito crefito 454821 mai 29 18:02 Planejamento.pdf
-rwxrwxr-x 1 crefito crefito 48695  mai 26 11:14 realiza_backup.sh
crefito@desk12:/bkp/crefito7$
```

Quanto à imagem acima e aos comandos no Ubuntu a ela relacionados, julgue o item.

Para a remoção de todas as permissões sobre o arquivo anuidades.pdf, tornando-o inacessível, o usuário root deverá executar o comando `chmod -x anuidades.pdf`.

14. (CS-UFG/UFG/2023) Sistemas de arquivos promovem o armazenamento estruturado de arquivos, ao agregar características como: identificador de arquivo, permissões de acesso ao arquivo, partições do arquivo no meio de armazenamento etc. Um comando Linux para atribuir permissões a arquivos é

- A) `chmod`.
- B) `diff`.
- C) `du`.
- D) `df`.

15. (FUNCERN/Câmara de Natal-RN/2023) Um técnico de informática da Prefeitura, por meio do sistema Linux, precisou dar permissão de leitura, edição e execução para determinado arquivo. Para isso, utilizou o comando

- A) `chmod 755`
- B) `chmod 777`
- C) `chmod 750`
- D) `chmod 700`

16. (CETREDE/Câmara de Ipu-CE/2023) Assinale a alternativa que corresponda ao comando utilizado para mudar a proteção de um arquivo ou diretório.

- A) `diff`.





- B) file.
- C) grep.
- D) chmod.
- E) quota.

17. (UFPR/IF-PR/2023) Para o controle de um arquivo no Sistema Linux Ubuntu foi criado um usuário "supervisor" que pertence ao grupo "gerencia". O usuário "supervisor" criou um arquivo "metas.txt" para incluir e atualizar os dados das vendas feitas pelos funcionários. No entanto, o arquivo precisa ter restrição de acesso; dessa forma, o administrador alterou a permissão do arquivo para que o usuário "supervisor" tenha total acesso e o grupo "gerencia" consiga somente ler. Sabendo-se disso, qual a opção correta para a saída do comando "ls -l" após a permissão?

- A) -rwx----- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- B) -rwxr----- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- C) -rx-rw---- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- D) -rwxrwx--- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt
- E) -rwx---r-- 1 supervisor gerencia 50552 Jan 20 15:00 metas.txt

18. (SELECON/IF-RJ/2023) No sistema operacional LINUX, o comando CHMOD altera a permissão de acesso a um arquivo ou diretório. Um administrador de sistemas LINUX, conectado como usuário root, executou o comando "chmod 764 teste". As permissões concedidas no arquivo "teste" após a execução do comando acima foram as seguintes:

- A) outros (leitura, gravação); grupo (leitura, gravação e execução); dono (leitura)
- B) outros (leitura, gravação e execução); grupo (leitura e gravação); dono (leitura)
- C) dono (leitura e gravação); grupo (leitura); outros (leitura)
- D) dono (leitura); grupo (leitura, gravação e execução); outros (leitura e gravação)
- E) dono (leitura, gravação e execução); grupo (leitura e gravação); outros (leitura)



## GABARITO

01	02	03	04	05	06
E	D	A	C	A	D
07	08	09	10	11	12
B	C	C	C	B	D
13	14	15	16	17	18
E	A	B	D	B	E



## LISTA DE QUESTÕES - CEBRASPE

1. (CEBRASPE/FUNPRESP-JUD/2016) Os comandos `find` e `which` podem ser utilizados para a obtenção de informações acerca de programas instalados no Linux; o primeiro localiza o caminho (path) onde o comando está instalado, e o segundo, a versão do programa.

2. (CEBRASPE/PG-DF/2021) Quanto à administração de sistemas operacionais, julgue o item seguinte.

No Linux, o comando `sed 's/windows/linux/' maria > pedro` faz a substituição da palavra `windows` pela palavra `linux` contida no arquivo `maria` e o resultado é enviado para o arquivo de nome `pedro`.

3. (CEBRASPE/CODEVASF/2021) Com relação à administração do sistema operacional Linux, julgue o item seguinte.

O comando `lastlog` é capaz de mostrar o último login de usuários e, se o login tiver sido feito de um computador remoto, informará também o endereço IP de origem.

4. (CEBRASPE/PGE-RJ/2022) Acerca da administração do sistema operacional Linux e da linguagem de script Shell, julgue o item subsequente.

No Linux, o comando `vmstat` apresenta um relatório de consumo de CPU. Nesse relatório, a coluna `id` mostra o número do processo que está consumindo o referido recurso.

5. (CEBRASPE/DPE-DF/2022) Julgue o item subsequente, a respeito da administração de sistemas operacionais Linux, considerando que o sistema esteja logado com o usuário `root`.

Se `022` é o `umask` padrão para todos os usuários, então o comando a seguir modificará o `umask` para `227` somente para o usuário `roberto`.

```
echo 'umask 227' >> /home/roberto/.bashrc
```

6. (CEBRASPE/SEPLAN-RR/2023) Julgue o próximo item a respeito dos conceitos de Linux.

Suponha-se o seguinte código `bash`.



```
tail nome_arquivo.txt -n2 | head -n1
```

A execução do código em tela resultará na apresentação da penúltima linha do arquivo nome\_arquivo.txt.

## GABARITO

01	02	03	04	05	06
Errado	Certo	Certo	Errado	Certo	Certo



## LISTA DE QUESTÕES - FGV

1. (FGV/Câmara Municipal do Recife-PE/2014) No Linux, o comando

```
cat something | grep 'something'
```

tem o efeito de:

- A) mostrar as linhas do arquivo something que incluem a palavra something;
- B) adicionar a palavra something ao conteúdo do arquivo something;
- C) criar uma cópia do arquivo something denominada grpe;
- D) mover o arquivo something para um novo diretório denominado something;
- E) restringir o acesso ao arquivo something ao usuário something.

2. (FGV/TJ-BA/2015) O seguinte comando, invocado em uma shell do Linux:

```
find . -type f
```

escreverá na saída padrão:

- A) o número de arquivos regulares descendentes do diretório corrente;
- B) os caminhos (pathnames) relativos para os arquivos regulares descendentes do diretório corrente;
- C) os caminhos (pathnames) absolutos para os arquivos regulares descendentes do diretório corrente;
- D) os caminhos (pathnames) relativos para os diretórios descendentes do diretório corrente;
- E) os caminhos (pathnames) absolutos para os arquivos regulares e diretórios descendentes do diretório corrente.

3. (FGV/Câmara de Aracaju-SE/2021) Observe a linha de comando para um sistema Linux

```
tar xzf myfile.tar.gz
```

A ação realizada pela sua execução é:

- A) criar o arquivo de backup myfile.tar.gz;
- B) descompactar o arquivo myfile.tar.gz no diretório corrente;
- C) habilitar a execução do arquivo myfile.tar.gz;



- D) executar o arquivo myfile.tar.gz com permissão temporária;
- E) apagar o conteúdo do arquivo myfile.tar.gz.

4. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando que informa o path absoluto do diretório corrente.

- A) cd
- B) find
- C) mv
- D) mkdir
- E) pwd

5. (FGV/FUNSAÚDE-CE/2021) No Linux, a saída (output) de um comando pode ser direcionada para um arquivo ou processo. Nesse contexto, assinale o redirecionamento que é determinado pelo símbolo >>.

- A) Adicionar o conteúdo direcionado a um arquivo existente, ou a um novo arquivo se não existir o arquivo referido.
- B) Descartar o conteúdo direcionado; não gravar e não mostrar.
- C) Direcionar o conteúdo para outro comando ou processo.
- D) Gravar um novo arquivo com o conteúdo direcionado; não pode existir eventual arquivo homônimo.
- E) Gravar um novo arquivo com o conteúdo direcionado, sobrescrevendo eventual arquivo homônimo.

6. (FGV/TCE-AM/2021) No Linux, o comando usado para verificar e, opcionalmente, reparar um sistema de arquivos é denominado:

- A) cat;
- B) chmod;
- C) finger;
- D) fsck;
- E) grep.



7. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando que busca linhas num arquivo.

- A) cd
- B) lp
- C) ls
- D) tar
- E) grep

8. (FGV/IMBEL/2021) No contexto do Linux, assinale o comando utilizado para listar o conteúdo de um diretório.

- A) cf
- B) lp
- C) ls
- D) tar
- E) grep

9. (FGV/SEAD-AP/2022) O formato de arquivo chamado Raw, padrão para a saída gerada pelo comando dd do Linux/UNIX, é uma cópia bit a bit dos dados brutos da unidade que está sob investigação, que pode ser usado na criação de uma imagem da unidade completa ou de um único volume (partição).

Assinale a opção que apresenta a principal desvantagem do formato Raw.

- A) Não ser suportado na maioria dos programas de computação forense.
- B) Requerer o mesmo espaço de armazenamento da unidade de origem.
- C) Ignorar erros de leitura menores a partir da unidade de origem.
- D) Realizar uma rápida transferência de dados.
- E) Realizar a compactação dos dados.

10. (FGV/SEAD-AP/2022) João está investigando uma possível violação de segurança e precisa saber quais usuários fizeram login no sistema recentemente.

O utilitário de linha de comando do sistema operacional Linux que mostra essas informações é

- A) alias



- B) last
- C) nice
- D) tty
- E) wc

11. (FGV/TCE-TO/2022) No contexto do Linux, a simbologia de redirecionamento de saída de dados que adiciona dados no final de um arquivo é:

- A) |
- B) ||
- C) >
- D) ->
- E) >>

12. (FGV/MPE-GO/2022) Assinale a função do comando man do Linux.

- A) Acompanhar passo a passo a execução de um comando.
- B) Documentar os comandos do sistema.
- C) Executar rotinas do Kernel.
- D) Executar tarefas de administração de contas/usuários.
- E) Gerenciar os processos ativos no sistema.

13. (FGV/DPE-RS/2023) O Ubuntu é um sistema operacional considerado fácil de usar, instalar e configurar, além de possuir uma vasta documentação. A respeito do Ubuntu, é correto afirmar que:

- A) com o ls é possível interagir com o sistema por meio de comandos digitados no teclado;
- B) o comando shell lista os arquivos e diretórios;
- C) o comando pwd fornece a identificação efetiva do usuário e o comando date retorna a data atual;
- D) o comando whoami retorna exatamente o diretório corrente;
- E) o comando cd / serve para ir ao diretório raiz, informando ao shell para navegar até o diretório /.





## GABARITO

01	02	03	04	05	06
A	B	B	E	A	D
07	08	09	10	11	12
E	C	B	B	E	B
13					
E					



## LISTA DE QUESTÕES - FCC

1. (FCC/TRF4/2014) Marcos, usuário de um computador com sistema operacional Linux Red Hat listou o conteúdo do seu diretório home e observou a presença do arquivo manual.txt com 31.251 bytes de tamanho, o que representa cerca de 20 páginas de texto se visualizado em um terminal Linux padrão. Para que Marcos possa visualizar diretamente o final do arquivo manual.txt, sem a necessidade de iniciar a visualização a partir do começo do arquivo, ele deve executar o comando:

- A) `cat manual.txt | more`
- B) `more manual.txt`
- C) `list manual.txt | end`
- D) `tail manual.txt`
- E) `cat manual.txt | end`

2. (FCC/Prefeitura de São Luís-MA/2018) Um Auditor deseja configurar e habilitar em um servidor Linux a interface de rede eth0 com o IP 172.16.25.125 e a máscara 255.255.255.224. Para isso, ele terá que utilizar o comando

- A) `ipconfig eth0 172.16.25.125 mask 255.255.255.224 up`
- B) `netconfig eth0 ip 172.16.25.125 netmask 255.255.255.224 on`
- C) `netstat eth0 ip 172.16.25.125 netmask 255.255.255.224 on`
- D) `ifconfig eth0 172.16.25.125 netmask 255.255.255.224 up`
- E) `netstart eth0 172.16.25.125 mask 255.255.255.224`

3. (FCC/Câmara Legislativa-DF/2018) Considere, hipoteticamente, que a Câmara Legislativa do DF receberá um funcionário temporário por um período de 3 meses. Para adicionar esse funcionário como usuário com período de tempo determinado no servidor com sistema operacional Linux, o Administrador deve utilizar o comando `useradd` com a opção

- A) `-r`.
- B) `-d`.
- C) `-o`.
- D) `-s`.
- E) `-e`.



4. (FCC/TRT2/2018) No shell do Linux, para ativar a interface de rede eth0 utiliza-se a instrução

- A) netstart eth0
- B) netstart --interface=eth0
- C) nslookup --interface=eth0
- D) ifconfig eth0 up
- E) ipconfig eth0 up

5. (FCC/METRÔ-SP/2019) Um Analista que usa o Linux Red Hat, em condições ideais, editou um arquivo e colocou os comandos abaixo, para indicar os servidores que atuarão na resolução de nomes para o servidor que está configurando.

```
search localdomain nameserver 8.8.8.8 nameserver 8.8.4.4
```

Para editar o arquivo em questão, utilizou a instrução

- A) vi /etc/dns.conf
- B) nano /conf/dns.conf
- C) vi /etc/resolv.conf
- D) nano /bin/resolv.conf
- E) vi /lib/dnsconf.conf

6. (FCC/TRF3/2019) No Linux Red Hat, para mostrar o espaço em disco, incluindo espaço usado e disponível, utiliza-se o comando

- A) df -h
- B) dir- s
- C) tune2fs - d
- D) free -h
- E) top - a

7. (FCC/TJ-MA/2019) Para descompactar, no Linux, um arquivo chamado data.tar.gz, mostrando o nome dos arquivos durante a descompactação, deve-se usar o comando:

- A) tar -xsf data.tar.gz



- B) `unzip -xsf data.tar.gz`
- C) `tar -xjf data.tar.gz`
- D) `gzip -xmjf data.tar.gz`
- E) `tar -xvzf data.tar.gz`

8. (FCC/TJ-MA/2019) Uma Técnica Judiciária, ao utilizar um computador com o sistema operacional Linux, identificou um comando que faz uma busca no Linux por termos ligados a determinado assunto que um usuário deseja pesquisar, e exibe os comandos e utilitários relacionados àquele assunto, com sua descrição. O comando descrito é o:

- A) `cal`
- B) `pwd`
- C) `apropos`
- D) `manual`
- E) `whatisthis`

9. (FCC/TRT19/2022) Utilizando o app Terminal do Linux, um Analista conectou-se com sucesso ao servidor Web de sua organização por intermédio do SSH. Para realizar a manutenção de um servidor web, dentre os principais comandos desse protocolo (SSH), o

- A) `touch` permite a criação de um arquivo com a extensão escolhida.
- B) `ls -hal` permite a visualização do conteúdo de determinado arquivo.
- C) `rm` serve para abrir e editar um arquivo dentro do Terminal.
- D) `nano` mostra o caminho do diretório atual.
- E) `cat` serve para redirecionar ou apagar um arquivo ou diretório.

10. (FCC/Copergás-PE/2023) Considere que um usuário de um computador com o sistema operacional Linux funcionando em condições ideais usou, de forma adequada, os comandos `df`, `cat` e `ls`. Esses comandos são utilizados, correta e respectivamente, para

- A) mostrar a quantidade de memória usada – abrir um arquivo – mostrar as últimas 10 linhas de um arquivo texto.
- B) deletar um arquivo – copiar um arquivo – verificar o status da conexão do servidor.
- C) mostrar a quantidade de espaço usada por um arquivo – remover um arquivo – listar todos os arquivos do diretório.



- D) mostrar a quantidade de espaço usada no disco rígido – abrir um arquivo – listar todos os arquivos do diretório.
- E) deletar um diretório e todos os seus arquivos – copiar um arquivo – mostrar as últimas 10 linhas de um arquivo texto.

11. (FCC/TRT18/2023) A partir do terminal do RedHat Linux, funcionando em condições ideais, um técnico precisa procurar pela palavra “técnico” no arquivo trt.txt e direcionar a saída do comando para o arquivo trtsaída.txt

- A) `ls -s técnico trt.txt || trtsaída.txt`
- B) `grep técnico trt.txt > trtsaída.txt`
- C) `search trt.txt (técnico) && trtsaída.txt`
- D) `cat trt.txt técnico >> trtsaída.txt`
- E) `touch técnico trt.txt >> trtsaída.txt`

## GABARITO

01	02	03	04	05	06
D	D	E	D	C	A
07	08	09	10	11	
E	C	A	D	B	



## LISTA DE QUESTÕES - VUNESP

1. (VUNESP/DESENVOLVESP/2014) O superusuário do sistema operacional Linux deseja verificar a informação da última linha do arquivo `/var/log/dmesg`. Para efetivar essa ação por meio do uso de apenas um comando, ou seja, sem qualquer outra ação, deve-se executar, na linha de comando:

- A) `$ cat /var/log/dmesg`
- B) `$ cat /var/log/dmesg >`
- C) `$ more /var/log/dmesg >>`
- D) `$ tail /var/log/dmesg`
- E) `$ top /var/log/dmesg`

2. (VUNESP/SP-URBANISMO/2014) Um usuário padrão do sistema operacional Linux, "logado" no seu diretório `home`, deseja saber o diretório no qual está instalado o comando `mount`. Para obter tal informação, esse usuário poderá utilizar o seguinte comando:

- A) `cat mount`
- B) `find mount`
- C) `grep mount`
- D) `where mount`
- E) `which mount`

3. (VUNESP/Câmara de Mauá-SP/2019) O seguinte comando foi executado em um computador com sistema operacional Linux:

```
ls -l meudir
```

O resultado obtido foi:

```
-rw-r--r-- g b 18 Nov 29 17:50 meudir
```

De acordo com esse resultado, é correto afirmar que

- A) o grupo dono (owning group) de `meudir` é `g`.
- B) o tamanho de `meudir` é 29 bytes.
- C) `meudir` é um diretório.



- D) meudir pode ser executado.
- E) meudir pode ser escrito pelo usuário g.

4. (VUNESP/Câm. de M. Alto-SP/2019) Deseja-se atribuir um endereço IPv4 alias para a interface eth0 em um computador com sistema operacional Linux utilizando o comando ifconfig. A correta sintaxe para efetivar essa operação é:

- A) #ifconfig eth0:192.168.0.100
- B) #ifconfig eth0:0 192.168.0.100
- C) #ifconfig alias eth0 192.168.0.100
- D) #ifconfig eth0 alias 192.168.0.100
- E) #ifconfig eth0/alias 192.168.0.100

5. (VUNESP/Câm. de M. Alto-SP/2019) Durante o uso de um terminal Bash do sistema operacional Linux, o usuário digitou !! (dupla exclamação) seguido de Enter. O resultado dessa ação será

- A) a apresentação do PID do terminal.
- B) a limpeza (clear) do terminal.
- C) a reexecução do último comando.
- D) o bloqueio do terminal.
- E) o fechamento do terminal.

6. (VUNESP/Câm. de M. Alto-SP/2019) Utilizando o comando shutdown, o Administrador de um servidor com sistema operacional Linux deseja avisar os usuários e realizar a reinicialização do servidor em 10 minutos. Para isso, o Administrador deve utilizar o comando shutdown com as opções:

- A) -h 10.
- B) -k 10.
- C) -p 10.
- D) -r 10.
- E) -t 10.



7. (VUNESP/UFABC/2019) O administrador de um servidor com sistema operacional Linux está testando um programa e deseja armazenar, em um arquivo, as informações da saída de erro padrão (stderr) do programa. Para fazer esse redirecionamento, ou seja, stderr para um arquivo, o administrador deve utilizar os seguintes caracteres de redirecionamento:

- A) 0>
- B) 1>
- C) 2>
- D) >>
- E) &>

8. (VUNESP/Câmara de Tatuí-SP/2019) Considere a execução da sequência de comandos a seguir, em um terminal shell do Linux:

```
# cd /root
```

```
# mkdir -p /root/foo/bar
```

```
# pwd
```

O resultado impresso na tela após a execução do último comando será:

- A) /root
- B) /root/foo/bar
- C) bar
- D) /root/foo
- E) /foo/bar

9. (VUNESP/CODEN-SP/2021) No sistema operacional Linux, o comando que cria um arquivo vazio, ou atualiza a data/hora (timestamp) de um arquivo já existente, é:

- A) cat
- B) less
- C) tail
- D) touch
- E) head





10. (VUNESP/PRUDENCO/2022) Em uma instalação do sistema operacional Linux, executou-se o comando `ls -l` e observou-se a seguinte designação, composta por 10 caracteres, à esquerda de um item listado:

`lrwxrwxrwx`

Com base nessa informação, pode-se afirmar que se trata de um

- A) diretório.
- B) link simbólico.
- C) shell script executável.
- D) arquivo executável correspondente a um comando do Linux.
- E) arquivo pertencente ao usuário root, uma vez que todas as permissões estão habilitadas.

11. (VUNESP/Câm. de Olímpia-SP/2022) Em um computador com o sistema operacional Linux, foi encontrado um arquivo chamado `arq` em um certo diretório. Nesse diretório, executou-se, com sucesso, o seguinte comando:

`ln arq arq2`

É correto afirmar que `arq2` é

- A) um hard link para `arq`.
- B) um soft link para `arq`.
- C) uma cópia de `arq`.
- D) o subdiretório para onde `arq` foi movido.
- E) o grupo que passou a ser dono de `arq`.

12. (VUNESP/Câm. de Olímpia-SP/2022) No sistema operacional Linux, os comandos que têm a finalidade de

- (i) alterar a senha de um usuário,
- (ii) alterar o usuário dono de um arquivo e
- (iii) copiar arquivos, respectivamente, são:

- A) `pwd`, `chown` e `cp`.
- B) `pwd`, `chown` e `copy`.
- C) `pwd`, `chattr` e `cp`.



- D) passwd, chmod e copy.
- E) passwd, chown e cp.

13. (VUNESP/UNICAMP/2022) Em um terminal do sistema operacional Linux, a execução do comando

```
echo "teste" 2> arquivo.txt
```

exibe a seguinte saída no próprio terminal:

- A) Nenhuma (saída vazia).
- B) teste
- C) arquivo.txt
- D) 2> arquivo.txt
- E) "teste" 2> arquivo.txt

14. (VUNESP/UNICAMP/2022) Algumas operações de recuperação de instalações do sistema operacional Linux podem requerer a utilização do comando chroot. Este comando tem por finalidade

- A) executar comandos que requerem privilégio de root, sem alterar o usuário da sessão já aberta.
- B) alterar o usuário de uma sessão já aberta, permitindo executar comandos com os privilégios do outro usuário.
- C) alterar o usuário dono de um arquivo.
- D) alterar o diretório raiz, criando um ambiente modificado no qual os programas não podem acessar arquivos fora desse diretório.
- E) alterar o nome do usuário root no sistema, permitindo que o root de fato utilize outro login.

15. (VUNESP/UNICAMP/2022) Após a instalação do sistema operacional Linux em um computador, é desejável verificar as portas TCP e UDP em estado listening por padrão, podendo-se, então, desinstalar ou desativar os aplicativos correspondentes àquelas portas que não se pretende utilizar. Dentre as a seguir, o comando do Linux que permite fazer essa verificação (utilizando os devidos parâmetros) é:

- A) ifconfig
- B) df
- C) ps



- D) nslookup
- E) netstat

16. (VUNESP/EPC/2023) Considere um arquivo com nome prova.txt que contém o seguinte conteúdo:

```
Boa  
  
prova  
  
!
```

Após um usuário de sistema operacional Linux digitar o comando no terminal

```
grep -v a prova.txt
```

será apresentado

- A) Boa  
prova  
!
- B) Boa  
prova
- C) a  
a
- D) 2
- E) !

17. (VUNESP/EPC/2023) É possível obter o valor do horário atual em um sistema operacional Linux por meio de um comando escrito no terminal. Esse comando é o

- A) time
- B) date
- C) hour
- D) now
- E) clock.



## GABARITO

01	02	03	04	05	06
D	E	E	B	C	D
07	08	09	10	11	12
C	A	D	B	A	E
13	14	15	16	17	
B	D	E	E	B	



## LISTA DE QUESTÕES - MULTIBANCAS

1. (IF-RR/IF-RR/2015) O comando find permite buscar arquivos filtrando por nome, tamanho, data, dono, grupo, permissões, tipo e outros. A busca é recursiva, por padrão, mas pode ser restringida a uma determinada profundidade. O comando GNU find faz parte das findutils GNU e está instalado em cada sistema Ubuntu. Um usuário possui uma distribuição Ubuntu14.04 LTS e precisa encontrar um arquivo de áudio com tamanho superior a 5MB, dentre as opções abaixo, qual alternativa melhor representa o comando para a busca pelo arquivo.

- A) find \$HOME -iname '\*.mp4' -size -5M
- B) find \$HOME -iname '\*.mp4' -size +5M
- C) find \$HOME -iname '\*.jpeg' -size +5M
- D) find \$HOME -iname '\*.ogg' +size -5M
- E) find \$HOME -iname '\*.ogg' -size +5M

2. (FCM/IF Sudeste-MG/2016) Alguns comandos em Linux podem combinar diferentes parâmetros, como o comando a seguir:

```
$ find /home/usuario/ -name '*.txt' -mtime 2;
```

Esta linha de comando procura

- A) todos os arquivos com extensão ".txt", dentro da pasta "/home/usuario".
- B) e remove todos os arquivos com extensão ".txt" dentro da pasta "/home/usuario".
- C) todos os arquivos com extensão ".txt", modificados a mais de 2 dias, dentro da pasta "/home/usuario" e subpastas.
- D) todos os arquivos modificados a mais de 2 dias, dentro da pasta "/home/usuario", e subpastas, e os move para a pasta "name".
- E) e copia para o diretório corrente todos os arquivos com nome "\*.txt", criados a menos de 2 dias, dentro da pasta "/home/usuario".

3. (FCM/IFF-RS/2016) Em um ambiente Linux, desejam-se encontrar todos os arquivos com a extensão ".sh" a partir do diretório corrente. O comando a ser utilizado é o

- A) find -all \*.sh /
- B) grep - all \*.sh
- C) find -user -i .sh
- D) ps -name \ \*.sh



E) `find . -name \*.sh`

4. (IESES/BAHIAGÁS/2016) O comando `find` está presente em vários sistemas operacionais Unix-like, sendo muito conhecido pela sua versatilidade. Considere o uso do `bash` em uma distribuição de Linux Ubuntu 14.04, na qual o comando `find` esteja disponível. Um usuário gostaria então de listar somente arquivos de extensões `'.java'` e `'.py'` em seu diretório `'/home/usuario'`. Assinale a alternativa que executa esta tarefa corretamente.

A) `find /home/usuario -type f -iname '*.java' -iname '*.py'`

B) `find /home/usuario -type d -a \( -iname '*.java' -o -iname '*.py' \)`

C) `find /home/usuario -type f -o \( -iname '*.java' -o -iname '*.py' \)`

D) `find /home/usuario -type f -a \( -iname '*.java' -o -iname '*.py' \)`

E) `find /home/usuario -type f -a \( -iname '*.java' -a -iname '*.py' \)`

5. (INSTITUTO AOCP/EBSERH/2016) Um administrador de sistema operacional Linux necessita procurar determinados textos, dada uma palavra, dentro de um arquivo texto. O comando que possibilita executar essa operação é o

A) `find`.

B) `more`.

C) `whereis`.

D) `grep`.

E) `pipe`.

6. (IESES/BAHIAGÁS/2016) Qual dos comandos apresentados a seguir pode ser utilizado para juntar o conteúdo de um arquivo texto de log do Linux, denominado de `log1`, ao conteúdo de um arquivo texto de log, denominado `log2`?

A) `cat log2 << log1`

B) `cat log2 < log1`

C) `cat log1 || log2`

D) `cat log1 > log2`

E) `cat log1 >> log2`



7. (IESES/BAHIAGÁS/2016) Um administrador de sistema gostaria de acompanhar o arquivo de log `'/var/log/apache.log'` em tempo real. Este arquivo está em um servidor Linux Ubuntu 14.04. Supondo que o administrador possua acesso de superusuário ao sistema através do bash. Qual dos comandos a seguir, executado uma única vez, poderia auxiliar o administrador a acompanhar este log dinamicamente (sem interferência adicional)?

- A) `tail -rt /var/log/apache.log`
- B) `more -rt /var/log/apache.log`
- C) `more -f /var/log/apache.log`
- D) `cat /var/log/apache.log`
- E) `tail -f /var/log/apache.log`

8. (CONSULPLAN/Câmara de Belo Horizonte-MG/2018) No Sistema Operacional GNU/Linux, o comando `cat` é utilizado para concatenar arquivos. Sua sintaxe básica é: `cat [opções] arquivo`. Uma das opções utilizadas é `-A`, que tem como função:

- A) Numerar todas as linhas na saída.
- B) Exibir todos os caracteres especiais.
- C) Numerar apenas as linhas não vazias.
- D) Exibir caracteres não passíveis de impressão (caracteres de controle).

9. (IDECAN/IF-PB/2019) A respeito dos comandos que podem ser utilizados para verificar o uso de recursos em um sistema operacional Linux, assinale a alternativa que indica, respectivamente, o nome do comando responsável por exibir informações referentes ao uso da memória RAM e uso do espaço em disco.

- A) `cat` e `du`
- B) `locate` e `df`
- C) `cat` e `free`
- D) `free` e `df`
- E) `du` e `free`

10. (COSEAC/UFF/2019) No Linux, o comando para listar o conteúdo de um diretório e o comando que mostra o caminho por inteiro do diretório em que o usuário se encontra são, respectivamente:

- A) `find` e `more`.



- B) cat e grep.
- C) lpr e cat.
- D) df e finger.
- E) ls e pwd.

11. (CS-UFG/IF Goiano/2019) No sistema operacional GNU/Linux, usando a linha de comando, deseja-se executar o seguinte: a) criar um diretório chamado IFG; b) criar um arquivo de texto chamado Concurso.txt; c) apagar o arquivo de texto Concurso.txt; d) apagar o diretório IFG. Qual é a sequência de comandos a ser empregada? Obs.: o sinal “,” é um mero separador entre os comandos

- A) mkdir IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- B) mkdir IFG , create Concurso.txt , kill Concurso.txt , killdir IFG
- C) create IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- D) create IFG , create Concurso.txt , kill Concurso.txt , killdir IFG

12. (FUNDATEC/IF Farroupilha-RS/2019) O comando “cd”, no sistema operacional Linux, serve para:

- A) Mudar do diretório atual para o especificado.
- B) Criar um arquivo.
- C) Criar um diretório.
- D) Renomear um arquivo.
- E) Renomear um diretório.

13. (AMAUC/Pref. de Arabutã-SC/2021) Qual atalho usado pelo SO Linux, é global, e repete o último comando?

- A) \$\$
- B) %%
- C) !!
- D) ##
- E) \*\*





14. (UFAM/UFAM/2021) No sistema operacional GNU/LINUX, o comando que cria uma partição ext2 na partição /dev/hda3 é:

- A) # mkfs -t ext2 /dev/hda3
- B) # mdfs -s ext3 /dev/hda3
- C) # mkfp -t ext2 /dev/hda3
- D) # mkfp -s ext3 /dev/hda2
- E) # mkfp -t ext2 /dev/hda2

15. (FURB/Prefeitura de Gaspar-SC/2022) O Sistema Operacional Linux possui diversos utilitários que são executados através de linha de comando. Associe a segunda coluna de acordo com a primeira relacionando o comando e sua respectiva função:

Primeira coluna: Comando

- 1.vi
- 2.chmod
- 3.pwd
- 4.ip addr

Segunda coluna: Função

- Permite alterar as permissões de um diretório ou arquivo.
- Permite configurar e gerenciar endereços de rede associados a uma interface.
- Editor de textos ASCII utilizado para criar e editar arquivos.
- Mostra o caminho completo do diretório corrente.

Assinale a alternativa que apresenta a correta associação entre as colunas:

- A) 2 - 3 - 1 - 4
- B) 2 - 4 - 1 - 3
- C) 3 - 1 - 2 - 4
- D) 3 - 2 - 1 - 4
- E) 1 - 2 - 3 - 4



16. (AOCP/BANESE/2022) Em um sistema operacional Unix-like, como o Linux Ubuntu, qual comando lê as cinco primeiras linhas de um arquivo de texto?

- A) tail -f
- B) tail -n 5
- C) tac -n 5
- D) cat -n 5
- E) head -n 5

17. (FUNDATEC/Pref. de S. J. A.-RS/2022) No Linux, o comando \_\_ pode ser utilizado para identificar a quantidade de espaço em disco usado e disponível em cada partição do sistema.

Assinale a alternativa que preenche corretamente a lacuna do trecho acima.

- A) df
- B) du
- C) cd
- D) cp
- E) ls

18. (FUNDATEC/IF-RS/2022) No sistema operacional Linux, o comando pwd:

- A) Lista arquivos de um diretório.
- B) Altera a senha do usuário corrente.
- C) Cria um diretório.
- D) Exibe o nome do diretório corrente.
- E) Procura arquivos por conteúdo.

19. (FUNDATEC/Pref. de Viamão-RS/2022) O comando abaixo foi executado em um sistema operacional Linux:

```
echo -n "prefeitura" | wc -c
```

Qual é o seu retorno?

- A) Prefeitura
- B) PREFEITURA.



- C) arutieferp
- D) 10
- E) 11

20. (FUNDATEC/Pref. de Esteio-RS/2022) Observe a Figura 2 abaixo, obtida da tela do terminal de comandos de um sistema operacional, após a execução de uma ferramenta de administração:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sdb	251G	22G	218G	9%	/
tmpfs	13G	0	13G	0%	/mnt/xxx
tools	466G	261G	205G	56%	/init
none	13G	0	13G	0%	/dev
none	13G	4.0K	13G	1%	/run
none	13G	0	13G	0%	/run/lock
none	13G	0	13G	0%	/run/shm
none	13G	0	13G	0%	/run/user
tmpfs	13G	0	13G	0%	/sys/fs/cgroup

Assinale a alternativa que indica a ferramenta que gerou a saída da figura, assim como o sistema operacional (S.O.) onde ela foi utilizada.

- A) Ferramenta mount e S.O. Linux.
- B) Ferramenta mount e S.O. Windows.
- C) Ferramenta diskpart e S.O. Windows.
- D) Ferramenta df e S.O. Windows.
- E) Ferramenta df e S.O. Linux.

21. (FUNDATEC/IF Farroupilha-RS/2023) No sistema operacional Linux, o comando \_\_\_\_\_ é utilizado para trocar o proprietário de um arquivo.

Assinale a alternativa que preenche corretamente a lacuna do trecho acima.

- A) chown
- B) man
- C) apropos
- D) who
- E) finger



22. (FUNDATEC/GHC-RS/2023) Qual comando do GNU Linux pode ser usado para criar um usuário?

- A) su
- B) sudo
- C) newuser
- D) useradd
- E) passwd

23. (AOCP/IF-MA/2023) Com a utilização dos serviços de terminal do Linux, qual comando pode ser utilizado para contar linhas, palavras ou bytes de um arquivo específico ou do que for escrito no terminal?

- A) sudo
- B) wc
- C) chmod
- D) clear
- E) echo

24. (AOCP/IF-MA/2023) O sistema operacional Linux possui uma vasta lista de comandos. Porém, existem alguns que são essenciais para um usuário desse sistema operacional. Esses comandos são indicados para aqueles que não conseguem suprir suas necessidades por meio da interface gráfica ou que estejam usando o sistema em modo texto.

Manoel, técnico de informática, está trabalhando em modo texto no sistema operacional Linux e precisa verificar qual é o espaço livre no disco. Qual comando ele deve digitar?

- A) cd
- B) cp
- C) mv
- D) rm
- E) df

25. (Quadrix/CREFITO-7ª Região/2023)



```
Arquivo Editar Ver Pesquisar Terminal Ajuda
crefито@desk12:/bkp/crefито7$ ls -la
total 848
drwxrwxrwx 2 root  root  4896 jun  5 15:19
drwxr-xr-x 3 root  root  4896 jun  5 15:13 ..
-rw-rw-r-- 1 crefито  crefито 147180 mai 26 11:30 anuidades.pdf
-rw-rw-r-- 1 crefито  crefито 211550 jun  1 16:18 crefито_semanal.zip
-rw-rw-r-- 1 crefито  crefито 454821 mai 29 10:02 Planejamento.pdf
-rwxrwxr-x 1 crefито  crefито  40695 mai 26 11:14 realiza_backup.sh
crefито@desk12:/bkp/crefито7$
```

Quanto à imagem acima e aos comandos no Ubuntu a ela relacionados, julgue o item.

Caso o comando `rm -f anuidades.pdf crefито_semanal.zip` seja executado, tanto o arquivo `anuidades.pdf` quanto o arquivo `crefито_semanal.zip` serão removidos.

26. (Quadrix/CREFITO-7ª Região/2023) Quanto à imagem acima e aos comandos no Ubuntu a ela relacionados, julgue o item.

Com o comando `vi realiza_backup.sh`, o usuário terá acesso às funcionalidades de edição no arquivo `realiza_backup.sh`.

27. (FUNCERN/Câmara de Natal-RN/2023) Um técnico em informática recém-contratado necessitou consultar a distribuição e versão do Linux que está sendo executada em uma máquina. Felizmente, existem várias maneiras de descobrir essas informações usando a linha de comando. Uma delas é

- A) `cat /etc/os-release`
- B) `sb_release -x`
- C) `cat /etc/ -release`
- D) `lsb_release -os`

28. (FUNCERN/Câmara de Natal-RN/2023) Existe um comando que é utilizado no terminal do Linux para criar um arquivo vazio ou atualizar o carimbo de data/hora de um arquivo existente. Esse comando recebe como argumento o nome do arquivo que deseja criar/atualizar e denomina-se

- A) `find`.
- B) `mkdir`.
- C) `touch`.



D) cat.

29. (FUNDATEC/Pref. de Maçambará-RS/2023) Em um sistema operacional Linux, um usuário deseja criar um novo diretório pelo prompt de comandos. Qual comando ele deve utilizar?

A) file

B) mkdir

C) dirname

D) rmdir

E) cd

30. (FUNDATEC/Pref. de Maçambará-RS/2023) Em relação ao prompt de comandos do sistema operacional Linux, assinale a alternativa que corresponde a uma das funções do comando cat.

A) Mostrar as diferenças entre dois arquivos de texto.

B) Exibir o conteúdo de um ou mais arquivos.

C) Alterar o diretório de trabalho corrente.

D) Modificar as permissões de um ou mais arquivos.

E) Mostrar o nome do diretório de trabalho corrente.

31. (FURB/Pref. de Jaraguá do Sul-SC/2023) Associe a segunda coluna de acordo com a primeira que relaciona utilitários de linha de comando no Linux às suas funções e características:

Primeira coluna: Utilitário de linha de comando

1.nano

2.cat

3.systemctl

4.chmod

Segunda coluna: Funções e características

( ) Comando permite gerenciar serviços do Linux.

( ) Comando permite unir, criar e exibir arquivos



( ) É um editor de textos para sistemas Linux.

( ) É utilizado para lidar com permissões de arquivos do sistema Linux.

Assinale a alternativa que apresenta a correta associação entre as colunas:

A) 3 - 1 - 4 - 2.

B) 3 - 2 - 1 - 4.

C) 1 - 4 - 2 - 3.

D) 4 - 2 - 3 - 1.

E) 1 - 3 - 4 - 2.

32. (FUNDATEC/GHC-RS/2023) Um técnico estava utilizando um sistema operacional GNU Linux e executou os seguintes comandos:

1. `cat arquivo.txt`

2. `wc --words arquivo.txt`

O primeiro comando apresentou a saída do quadro abaixo:

Grupo Hospitalar Conceição

Qual a saída correta para o segundo comando?

A) GrupoHospitalarConceição

B) Grupo,Hospitalar,Conceição

C) 26 arquivo.txt

D) 1 arquivo.txt

E) 3 arquivo.txt

33. (FUNDATEC/PROCERGS/2023) Suponha que você está em um sistema Linux e precisa criar o diretório "fotos". Considere que você já está no local raiz para o novo diretório. Qual alternativa apresenta o comando a ser executado para criar o diretório "fotos"?

A) `cd fotos`

B) `cp fotos`

C) `file fotos`

D) `mkdir fotos`



E) rmdir fotos

34. (FUNDATEC/PROCERGS/2023) O sistema operacional Linux possui diversos comandos. Um desses comandos é o cat. Considerando a execução do comando abaixo, qual o resultado esperado?

```
cat arquivo1 arquivo2
```

- A) O arquivo1 recebe o conteúdo do arquivo2
- B) Exibe conteúdo de arquivo1 e arquivo2
- C) Envia para a impressora o arquivo1 e o arquivo2
- D) Faz uma cópia de arquivo1 com nome arquivo2
- E) Renomeia arquivo1 para arquivo2

35. (FUNDEPES/Pref. de M. Deodoro-AL/2023) O sistema operacional é o software utilizado para intermediar a operacionalização do computador, por parte de seus usuários. O sistema operacional Linux é um software de código aberto que permite a operacionalização tanto em modo gráfico, quanto em modo textual. O terminal do Linux é a forma mais utilizada para executar ações em modo textual, via linha de comando. Tal característica favorece o desenvolvimento de scripts para automatizar tarefas. Dado o exemplo de uma linha de comando que pode ser executada em um terminal Linux, `cat documento.txt | grep "forma"` assinale a alternativa que descreve a ação realizada pela execução dessa linha de comando.

- A) Atribuir à categoria do arquivo "documento.txt" o valor "forma".
- B) Adicione a palavra "forma" na última linha do arquivo "documento.txt".
- C) Listar as categorias do arquivo "documento.txt" que iniciam com a palavra "forma".
- D) Ler o conteúdo do arquivo "documento.txt" e apresente na tela as linhas que contêm o texto "forma".
- E) Realizar a contagem do número de vezes que a palavra "forma" aparece no arquivo "documento.txt".





## GABARITO

01	02	03	04	05	06
E	C	E	D	D	E
07	08	09	10	11	12
E	B	D	E	A	A
13	14	15	16	17	18
A	B	E	E	A	D
19	20	21	22	23	24
D	E	A	D	B	E
25	26	27	28	29	30
Certo	Certo	A	C	B	B
31	32	33	34	35	
B	E	D	B	D	



## LISTA DE QUESTÕES - CEBRASPE

1. (CEBRASPE/TRT8/2013) Para alterar o arquivo de configuração do serviço de DHCP no Linux, deve-se acessar o diretório

- A) /dev.
- B) /mnt.
- C) /etc.
- D) /usr/bin.
- E) /boot.

2. (CEBRASPE/FUB/2015) O protocolo DNS (domain name service), localizado no nível de aplicação da camada de transporte do TCP, é responsável pelo mapeamento de nomes e de endereços.

3. (CEBRASPE/TRE-MT/2015) Acerca do servidor DNS/BIND (Domain Name System/Berkley Internet Domain), cuja funcionalidade é resolver nomes da rede, assinale a opção correta.

- A) Cada domínio tem seus registros de recursos e o registro de domínio denominado NS (name server), o qual é utilizado para definir propriedades básicas do domínio e sua zona.
- B) Um servidor DNS utiliza LDAP para fazer armazenamento das zonas de domínio para uma rápida resolução de um nome.
- C) O BIND, que utiliza a porta 53, é um programa de código aberto utilizado pela maior parte dos servidores DNS.
- D) Os domínios de um servidor DNS são organizados na Internet sobre uma estrutura de dados do tipo lista encadeada, sendo o primeiro elemento da lista um ponto.
- E) O protocolo HTTP implementa, por padrão, um servidor de resolução de nomes amplamente utilizado na Internet conhecido como DNS.

4. (CEBRASPE/TCE-PR/2016) Uma consulta DNS inicial típica, originada de uma máquina de usuário e encaminhada ao servidor de nomes local para um nome de domínio externo não armazenado em cache DNS, será do tipo

- A) raiz.
- B) domínio de alto nível.
- C) iterativa.



- D) recursiva.
- E) direta.

5. (CEBRASPE/TCE-SC/2016) Após o servidor local SMTP aceitar uma mensagem para subsequente envio, é necessário determinar o endereço do servidor de email do destinatário. Essa etapa é realizada mediante consulta DNS a um servidor de nomes capaz de prover a informação, no qual serão verificados os registros especiais MX (mail exchange).

6. (CEBRASPE/Polícia Científica-PE/2016) Na operação de um serviço DNS, se uma consulta a um servidor de alto nível retornar uma resposta positiva para o servidor DNS local, o mapeamento nome/endereço será

- A) armazenado definitivamente no banco de dados do servidor DNS local.
- B) replicado para todos os servidores DNS alcançáveis para que estejam disponíveis a outras consultas.
- C) repassado pelo módulo tradutor a um servidor raiz para confirmar a resposta recebida.
- D) armazenado em cache local e pode ser mantido por um tempo definido por configuração.
- E) repassado ao módulo tradutor do provedor de serviços para realizar a entrega à aplicação.

7. (CEBRASPE/Polícia Federal/2018) As atualizações entre servidores DNS utilizam o UDP, enquanto as consultas feitas a servidores DNS utilizam o TCP (ou, opcionalmente, o SCTP).

8. (CEBRASPE/EMAP/2018) Em um servidor DNS que esteja utilizando Bind, o arquivo /etc/named.conf mantém as configurações de funcionamento do serviço DNS. Nesse servidor, é possível configurar a porta que deve escutar o serviço; assim, para que o serviço escute a porta 53, deve-se configurar a opção seguinte.

```
Listening 53 { any;};
```

9. (CEBRASPE/TRT-8ª Região/2022) Caso se queira bloquear, no Linux, o IP 10.1.1.3 por meio do iptables, o comando a ser utilizado é o

- A) iptables -a INPUT -i 10.1.1.3 -r BLOCK.
- B) iptables -A INPUT -s 10.1.1.3 -j DROP.
- C) iptables -I INPUT -s 10.1.1.3 -r BLOCK.



- D) iptables -I INSERT -s 10.1.1.3 -r DENY.
- E) iptables -i PUT -d 10.1.1.3 -b REFUSED.

## GABARITO

01	02	03	04	05	06
C	Errado	C	D	Certo	D
07	08	09			
Errado	Errado	B			



## LISTA DE QUESTÕES - FGV

1. (FGV/MEC/2009) No UNIX, o DNS é implementado por meio do software Berkeley Internet Name Domain (BIND), que segue a filosofia cliente/servidor.

O cliente do BIND é denominado "resolver", responsável por gerar as "queries" enquanto que o servidor do BIND é chamado por "named", responsável por responder às "queries".

Configurações BIND são descritas pelo tipo de serviço a ser executado pelo software, existindo quatro níveis de serviço que podem ser definidos nessas configurações: "resolver-only systems", "caching-only servers", "primary servers" e "secondary servers".

Os parâmetros que definem a configuração do "resolver-only systems" são inseridas no seguinte arquivo:

- A) /etc/resolv.sys
- B) /etc/resolv.cfg
- C) /etc/resolv.bind
- D) /etc/resolv.conf
- E) /etc/resolv.parm

2. (FGV/TJ-GO/2014) A empresa Y passou a adotar uma política de uso de software livre e resolveu mudar o seu correio eletrônico, que era baseado em Microsoft Exchange. O novo ambiente deveria rodar em ambiente Linux. Uma das possíveis escolhas do novo software de correio eletrônico recai no programa:

- A) nginx;
- B) puppet;
- C) apache;
- D) postfix;
- E) bind.

3. (FGV/TCE-SE/2015) Um programa precisa simular o comportamento de um cliente DNS. Para funcionar adequadamente, o programa precisa enviar as consultas para um servidor DNS, especificamente para a sua porta:

- A) udp/23
- B) icmp/34



- C) tcp/22
- D) ip/50
- E) udp/53

4. (FGV/TCE-SE/2015) Em um equipamento rodando Linux, um administrador deseja alterar o endereço default do servidor DNS usado para consultas. Para isso, ele precisa alterar o arquivo:

- A) /etc/named.conf
- B) /etc/bind.conf
- C) /etc/network.conf
- D) /etc/dns.conf
- E) /etc/resolv.conf

5. (FGV/COMPESA/2016) O sistema de Nomes de Domínio - DNS permite transformar nomes digitados em um navegador WEB em um endereço de rede. O nome do host e o nível do domínio para o domínio "system.master.com" são, respectivamente,

- A) system e segundo nível.
- B) system e terceiro nível.
- C) master e terceiro nível.
- D) master e segundo nível.
- E) .com e terceiro nível.

6. (FGV/AL-RO/2018) Wallace é administrador de um servidor com sistema operacional CentOS e deseja compartilhar um diretório com os integrantes da empresa por meio do protocolo NFS.

Para que a configuração desse compartilhamento possa ser efetuada, Wallace deve editar o arquivo /etc/

- A) services
- B) modules
- C) hosts
- D) export
- E) fstab



7. (FGV/TRT-MA/2022) O comando do sistema operacional Linux utilizado por administradores de redes para configurar ou examinar informações sobre tabelas de roteamento e interfaces de redes ou criar túneis em um servidor se denomina

- A) traceroute.
- B) ifshow.
- C) netstat.
- D) ping.
- E) ip.

8. (FGV/MPE-SC/2022) Um analista obteve acesso a um terminal Linux para realizar o teste de um sistema, mas não tem certeza se o computador tem acesso à Internet, o que será necessário para o teste.

A forma mais simples de o analista verificar a conectividade do computador à Internet é:

- A) nc -vzu 127.0.0.1 80;
- B) mtr 192.168.0.1;
- C) ping 1.1.1.1;
- D) nmap -A 127.0.0.1;
- E) telnet 10.0.0.255 22.

## GABARITO

01	02	03	04	05	06
D	D	E	E	B	D
07	08				
E	C				



## LISTA DE QUESTÕES - FCC

1. (FCC/TCE-SP/2010) Em Linux, o `/etc/resolv.conf` é um arquivo texto simples, com um parâmetro por linha e especificações de endereços de servidores DNS. Nesse arquivo existem três palavras chaves normalmente usadas, que são

- A) domain, search e resolv.
- B) search, resolv e order.
- C) domain, search e nameserver.
- D) search, nameserver e order.
- E) search nameserver e resolv.

2. (FCC/TCE-SP/2010) No Linux, ele é muito útil para máquinas que são acessadas frequentemente, pois a inclusão de um computador neste arquivo dispensa a consulta a um servidor DNS para obter um endereço IP, sendo muito útil para máquinas que são acessadas frequentemente. Trata-se de:

- A) `/etc/nameserver`
- B) `/etc/localhost`
- C) `/etc/ipsec`
- D) `/etc/hosts`
- E) `/etc/dnshost`

3. (FCC/CNMP/2015) O serviço de nome de domínios (DNS) possui uma arquitetura do tipo cliente/servidor na qual a base de dados é distribuída por toda internet. Nessa arquitetura, o acesso ao servidor DNS para buscar o relacionamento IP/Domínio é feito pelo cliente que é o

- A) Browser.
- B) DNS Cache.
- C) DNS Resolver.
- D) DNS Searcher.
- E) Gateway.

4. (FCC/PRODATER/2016) No Serviço de Nomes de Domínio – DNS existem diferentes tipos de servidores distribuídos hierarquicamente que armazenam informações também de forma hierárquica. Considerando o nome: `www.empresa.com`, o domínio `.com` é gerenciado pelo servidor

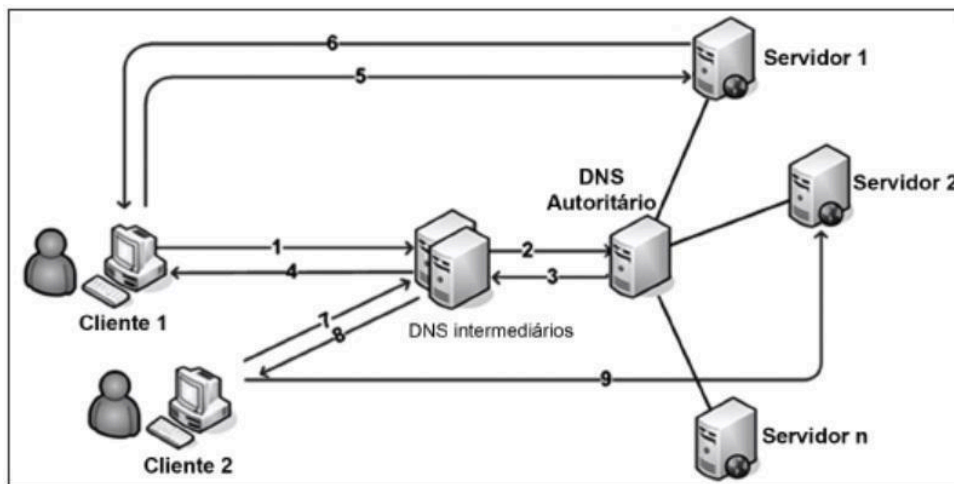
- A) Global.
- B) PDR.





- C) Authoritative.
- D) TLD.
- E) Root.

5. (FCC/ELETROSUL/2016) O balanceamento de carga por DNS é uma abordagem popular e simples para as solicitações de balanceamento de servidores. Considere a figura abaixo.



De acordo com a figura, geralmente são estes os passos que ocorrem assim que é feita uma consulta DNS:

1. Quando um cliente tenta acessar o site, é realizada uma pesquisa no DNS local para determinar qual é o endereço I..... correspondente;
2. O pedido de endereço chega ao servidor de DNS II..... do domínio;
3. A primeira vez que esta consulta é feita, o servidor DNS remoto pode retornar todos os registros de endereços que ele tem para o site;
4. O servidor DNS III....., em seguida, determina o endereço de registro para retornar ao cliente;
5. Se todos os registros são retornados, o cliente utilizará o primeiro que lhe é atribuído;
6. O servidor responde ao cliente e atende ao pedido;
7. A cada pedido, o algoritmo Round Robin roda os endereços e retorna pela ordem em que eles estão;
8. Cada consulta DNS irá resultar em um cliente usando um endereço IV.....;
9. Esta rotação de endereços irá distribuir pedidos para os servidores.

As lacunas de I a IV são, correta e respectivamente, preenchidas com:

- A) TCP – autoritário – local – TCP igual ao obtido em 1
- B) IP – autoritário – local – IP diferente
- C) do servidor – local – autoritário – de um servidor
- D) do servidor – autoritário – local – do servidor igual ao obtido em 1
- E) IP – local – autoritário – IP igual ao obtido em 1

6. (FCC/TRT-SP/2018) No shell do Linux, para ativar a interface de rede eth0 utiliza-se a instrução

- A) netstart eth0
- B) netstart --interface=eth0
- C) nslookup --interface=eth0
- D) ifconfig eth0 up
- E) ipconfig eth0 up

7. (FCC/Prefeitura de Manaus-AM/2019) Um Assistente de TI foi incumbido de configurar o protocolo para ser utilizado na troca de mensagens entre dois servidores de e-mail. A escolha correta do protocolo para essa finalidade é:

- A) IMAP.
- B) POP3.
- C) SNMP.
- D) SMTP.
- E) POP4.

8. (FCC/METRÔ-SP/2019) Um Analista que usa o Linux Red Hat, em condições ideais, editou um arquivo e colocou os comandos abaixo, para indicar os servidores que atuarão na resolução de nomes para o servidor que está configurando.

```
search localdomain nameserver 8.8.8.8 nameserver 8.8.4.4
```

Para editar o arquivo em questão, utilizou a instrução

- A) vi /etc/dns.conf
- B) nano /conf/dns.conf
- C) vi /etc/resolv.conf
- D) nano /bin/resolv.conf



E) vi /lib/dnsconf.conf

9. (FCC/TJ-MA/2019) O firewall do Linux Iptables é usado para monitorar entradas e saídas de tráfego de dados para o servidor. Em um computador com uma distribuição Linux em condições ideais, para aceitar pacotes do endereço 192.168.1.5, utiliza-se o comando:

- A) iptables -B INPUT -p 192.168.1.5 -j ALLOW
- B) iptables -A INPUT -s 192.168.1.5 -j ACCEPT
- C) iptables -C INPUT -d 192.168.1.5 -p ACCEPT
- D) iptables -A INPUT -d 192.168.1.5 -j ALLOW
- E) iptables -X FORWARD -i 192.168.1.5 -p ACCEPT

## GABARITO

01	02	03	04	05	06
C	D	C	D	B	D
07	08	09			
D	C	B			



## LISTA DE QUESTÕES - VUNESP

1. (VUNESP/SAAE de Barretos-SP/2018) Os serviços de acesso a dados da internet são disponibilizados em uma arquitetura Cliente/Servidor. Por exemplo, considerando um servidor com sistema operacional Linux, o serviço padrão que responde às requisições ftp é denominado

- A) ftpd.
- B) ftps.
- C) ftpsrv.
- D) samba.
- E) smb.

2. (VUNESP/Câm. de Itaq.-SP/2018) No sistema operacional Linux, o arquivo /etc/hosts é utilizado para

- A) listar os IPs com acesso permitido ao sistema.
- B) listar os servidores DNS externos em ordem de prioridade.
- C) mapear um nome (domínio) para um endereço IP.
- D) relacionar os IPs dos computadores da mesma rede local.
- E) relacionar os IPs dos roteadores vinculados.

3. (VUNESP/Câm. de M. Alto-SP/2019) Em um computador com sistema operacional Linux e iptables instalado, a apresentação de todas as regras atuais que estão configuradas pode ser realizada utilizando o comando iptables com a opção:

- A) -A
- B) -C
- C) -F
- D) -L
- E) -P

4. (VUNESP/UNICAMP/2022) Após a instalação do sistema operacional Linux em um computador, é desejável verificar as portas TCP e UDP em estado listening por padrão, podendo-se, então, desinstalar ou desativar os aplicativos correspondentes àquelas portas que



não se pretende utilizar. Dentre as alternativas a seguir, o comando do Linux que permite fazer essa verificação (utilizando os devidos parâmetros) é:

- A) ifconfig
- B) df
- C) ps
- D) nslookup
- E) netstat

## GABARITO

01	02	03	04
A	C	D	E



## LISTA DE QUESTÕES - MULTIBANCAS

1. (FUNIVERSA/PC-DF/2012) Em um servidor de e-mail com sistema operacional Linux, especialmente quando se utiliza o serviço Postfix, pode-se auditar o tráfego de entrada e saída de mensagens de correio eletrônico por meio dos registros conhecidos por logs. Em que arquivo desse sistema operacional encontram-se os logs das atividades de envio e recebimento de mensagens de correio eletrônico (e-mail)?

- A) No diretório `"/etc/postfix"`.
- B) No arquivo `"/var/log/maillog"`.
- C) Na pasta `"/var/mail/<nome_do_usuario>"`.
- D) No arquivo `"/usr/local/bin/postfix"`.
- E) No arquivo de registro `"/dev/tty"`.

2. (FUNCAB/MDA/2014) São exemplos de servidores de correio eletrônico do Linux:

- A) postfix e sendmail.
- B) fetchmail e sendmail.
- C) pine e postfixe.
- D) mail e pine.
- E) samba e postfixe.

3. (AOCP/UFGD/2014) Em algumas distribuições Linux encontramos um arquivo que identifica os locais dos computadores de servidor de nome DNS, computadores estes que são usados pelo TCP/IP para traduzir os nomes de Internet. Assinale a alternativa que apresenta o nome deste arquivo.

- A) DNS.conf.
- B) resolver.conf.
- C) resolv.conf.
- D) translate.conf.
- E) change.conf.

4. (IADES/TRE-PA/2014) Um servidor Linux pode hospedar o serviço de resolução de nomes de uma rede de computadores. Conhecido por DNS, esse serviço é indispensável em uma rede



que possua conexão com a internet. O nome de um pacote que implementa o DNS, muito utilizado em sistemas operacionais Linux, é

- A) Firefox.
- B) Apache.
- C) Squid.
- D) Postfix.
- E) BIND.

5. (INSTITUTO AOCP/UFPB/2014) Um estagiário executou o comando FTP no servidor Linux. Esse comando possibilita

- A) permitir a abertura de portas para um acesso remoto a um servidor da rede.
- B) realizar a manutenção de disco e de memória de um computador remotamente.
- C) trocar informações de desempenho de redes utilizando um canal de comunicação protegido e exclusivo.
- D) realizar uma verificação de permissões de acesso na rede e no computador que o usuário está utilizando.
- E) trocar arquivos entre computadores interligados, independente do sistema operacional e da rede utilizada.

6. (IADES/CRC-MG/2015) Quanto ao programa que é normalmente utilizado para testar se um servidor DNS (Domain Name Server) está funcionando corretamente, ou seja, resolvendo nomes para os endereços IP, assinale a alternativa correta.

- A) ping
- B) tracert
- C) nslookup
- D) ipconfig
- E) net host

7. (IBFC/EBSERH-HUAP/2016) O Domain Name System (DNS) é um sistema de gerenciamento de nomes hierárquicos e distribuídos. A funcionalidade do DNS Reverso vem a ser:

- A) reverter a funcionalidade básica de um DNS padrão obtendo o endereço MAC.



- B) resolver um endereço IP, buscando o nome de domínio associado ao host.
- C) resolver o nome do domínio de um host qualquer para seu endereço IP correspondente.
- D) com base na região geográfica, obter automaticamente um endereço IP local.
- E) resolver o problema da reversão do IPv6 para o IPv4 de uma forma rápida e automática.

8. (FUNRIO/CM Nova Iguaçu/2016) Um administrador de rede está configurando o DNS de um servidor e vai trabalhar com o registro que define as características da zona a ser configurada, tais como, o nome da zona e o nome do servidor, que é a autoridade para a referida zona, ou seja, o servidor DNS no qual está a zona que foi criada originalmente. Esse registro é conhecido pela sigla

- A) HINFO.
- B) MX.
- C) SOA.
- D) CNAME.

9. (IBFC/Polícia Científica-PR/2017) Servidores de DNS (Domain Name System) têm como função converter endereços IP em seu respectivo nome e vice-versa. Para sua configuração, são utilizados arquivos denominados mapas de domínio (zone). Esses arquivos são compostos por entradas chamadas RR (Resource Record). O tipo básico de RR que estabelece a correspondência entre um nome canônico e um endereço IP é indicado por:

- A) PTR
- B) MX
- C) NS
- D) A
- E) SOA

10. (IBFC/Polícia Científica-PR/2017) Servidores DNS (Domain Name Server) são responsáveis pela conversão do nome dos diversos servidores espalhados pela Internet para seu número IP e vice-versa. Servidores de DNS trabalham de forma colaborativa e hierárquica. Assinale a alternativa a que apresenta o nome dado aos servidores que se encontram no topo da hierarquia de DNS:

- A) Root Name Servers
- B) Main servers
- C) International Name Servers





- D) Controllers Servers
- E) Master Servers

11. (IF-RS/IF-RS/2018) Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?

- A) dns 8.8.8.8
- B) echo 8.8.8.8 > /etc/hosts.conf
- C) echo nameserver 8.8.8.8 > /etc/resolv.conf
- D) nslookup 8.8.8.8
- E) nameserver 8.8.8.8

12. (FAURGS/BANRISUL/2018) Um cliente DNS, ao fazer a requisição DNS para resolver o nome `www.banrisul.com.br`, recebe como resposta uma mensagem do tipo `non authoritative`. Esse tipo de resposta é obtido por meio de um registro DNS (resource record - RR) armazenado

- A) em um cache de DNS.
- B) em um servidor interativo sem cache DNS.
- C) em um servidor recursivo sem cache DNS.
- D) no arquivo de zona do DNS primário.
- E) no arquivo de zona do DNS secundário.

13. (FAPEC/UFMS/2018) Considere as afirmações a seguir sobre a instalação e configuração de um servidor de DNS (Bind9) no sistema operacional Debian 9.

I - Quando o servidor local de DNS não consegue traduzir o nome, ele deve solicitar a um servidor público (forwarders). O Google Public DNS é um exemplo de forwarders e deve ser configurado no arquivo `named.conf.options`.

II - A instalação do servidor pode ser realizada pelo usuário `root` com o comando: `apt install bind9 bind9-doc dnsutils`

III - A lista de controle de acessos ou ACL é definida no arquivo `named.conf.options`.

Está(ão) correta(s):

- A) Apenas I.



- B) Apenas II.
- C) Apenas III.
- D) Apenas II e III.
- E) I, II e III.

14. (SUGEP-UFRPE/UFRPE/2018) Para acesso à Internet, servidores Windows e Linux utilizam um serviço para atribuição dinâmica de IPs aos microcomputadores conectados à rede, em que é marcada a opção associada a "o IP será atribuído automaticamente pelo servidor", na configuração.

Esse serviço é conhecido pela sigla:

- A) DNS
- B) HTTP
- C) DHCP
- D) TCP
- E) IP

15. (IBADE/IPM-JP/2018) Arquivo de configuração do sistema operacional Linux que contém uma relação entre endereços IP e nomes, que pode ser utilizado para acessar computadores por um nome sem utilizar um servidor de nomes:

- A) /etc/names
- B) /etc/ips
- C) /etc/mapping
- D) /etc/hosts
- E) /etc/list

16. (AOCP/ITEP-RN/2018) Na falta de um serviço de nomes em uma rede local, qual dos arquivos a seguir é utilizado para fazer a tradução de nomes de computadores para endereços IP em um computador Linux?

- A) /etc/hosts.allow
- B) /etc/hosts
- C) /etc/hostname
- D) /etc/host.conf



E) /usr/local/hostname

17. (INAZ do Pará/CREFITO-MA/2018) As configurações de aplicativos e do próprio sistema operacional Linux podem ser realizadas por meio da manipulação dos arquivos de configuração através de um editor de texto simples. Caso o usuário deseje ingressar em um grupo doméstico ou domínio, qual dos arquivos abaixo ele deve editar para efetuar essa operação?

- A) Dhcp.conf
- B) Samba.conf
- C) Squid.conf
- D) Lilo.conf
- E) Access.conf

18. (CONSULPLAN/Câm. de B.H.-MG/2018) O programa Iptables fornece uma interface para que o usuário possa manipular o filtro de pacotes do kernel, ou seja, permite a administração de tabelas no firewall do kernel Linux. A sintaxe do iptables é a seguinte: iptables [ -t tabela ] <comando>[opção<parâmetro>]<destino>. Os comandos permitem que tarefas sejam executadas com o iptables. Um desses comandos é o [-D]. Assinale a alternativa que apresenta a descrição correta desse comando.

- A) Apaga uma regra.
- B) Apaga todas as regras.
- C) Estabelece a observação de pacotes.
- D) Adiciona uma regra (chave + opção + destino).

19. (IBADE/IPM-JP/2018) O Sistema Operacional Linux oferece um conjunto de comandos que podem ser utilizados para realizar a administração de redes. O comando que mostra o caminho percorrido por um pacote até chegar a seu destino é:

- A) traceroute.
- B) netpoints.
- C) traceway.
- D) routeway.
- E) identification.



20. (CCV-UFC/UFC/2019) Uma das formas de controlar o acesso de rede em um dispositivo utilizando o sistema operacional Linux é através do iptables. Qual dos itens abaixo contém a configuração para não permitir que o comando ping, executado em um dispositivo com o sistema operacional Linux, consiga enviar requisições para outros dispositivos?

- A) iptables -A INPUT -p icmp -j DROP
- B) iptables -A OUTPUT -p icmp --icmp-type 0 -j DROP
- C) iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
- D) iptables -D INPUT -p icmp --icmp-type echo-request -j DROP
- E) iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP

21. (COVEST-COPSET/UFPE/2019) No Windows e no Linux, qual utilitário de linha de comando é usado para exibir o caminho que um pacote faz até o seu destino?

- A) Netpath, no Windows e no Linux.
- B) Traceroute, no Windows e no Linux
- C) Tracert, no Windows e no Linux.
- D) Netpath, no Windows; e Tracert, no Linux.
- E) Tracert, no Windows; e Traceroute, no Linux.

22. (IBFC/MGS/2019) Um recurso técnico que permite, ao ser instalado no Linux, simular um servidor Windows para permitir o gerenciamento e compartilhamento de arquivos e de impressoras em uma rede Microsoft é o programa chamado:

- A) NetBIOS
- B) SAMBA
- C) OPERA
- D) NetBEUI

23. (Quadrix/CRO-GO/2019) Julgue o item, relativo à administração de sistemas Linux.

Os comandos ifup e ifdown são destinados, respectivamente, a ativar e a desativar uma interface de rede especificada.

24. (AOCP/IBGE/2019) Lucas está administrando remotamente um servidor GNU/Linux através de Shell Seguro (SSH). Ele inseriu duas regras no IPTABLES na ordem apresentada a seguir. Supondo que Lucas tenha plena certeza do efeito dessas regras após a inclusão delas



nessa ordem, qual das seguintes alternativas explica corretamente o que ele espera que ocorra com as conexões desse servidor?

```
Regra 1: iptables -A INPUT -p tcp -s 10.0.0.0/8 --dport 22 -j ACCEPT  
Regra 2: iptables -A INPUT -p tcp -s 0.0.0.0/0 --dport 22 -j DROP
```

- A) Que, após a inclusão da Regra 2, independentemente da rede de origem da sua conexão via SSH, a conexão seja interrompida imediatamente.
- B) Que, após a inclusão da Regra 2, independentemente da origem, qualquer tentativa de conexão com esse servidor através da porta 22 seja descartada.
- C) Que, após a inclusão da Regra 2, o servidor só aceite conexões na porta 22 a partir de hosts da rede 10.0.0.0 com máscara 255.0.0.0.
- D) Que, após a inclusão da Regra 1, qualquer host que não tenha IP 10.x.x.x tenha sua conexão recusada na porta 22.
- E) Que, após a inclusão da Regra 2, o servidor continue permitindo conexões na porta 22 com origem em qualquer rede, uma vez que as regras não foram aplicadas com o comando 'iptables -commit'.

25. (IF-PE/IF-PE/2019) Observe, a seguir, os itens referentes ao servidor de DNS Berkeley Internet Name Domain (BIND) do Linux Ubuntu 18.04.

- I. Digitando o comando # apt-get install bind no terminal, a instalação do bind é iniciada.
- II. O arquivo de configuração do "bind", no Linux, é o named.conf, que se encontra na pasta /etc.
- III. Para que o servidor externo receba solicitações não atendidas pelo DNS, devemos alterar o arquivo named.conf.options que se encontra na pasta /etc/bind.
- IV. Para a verificação do funcionamento do servidor DNS previamente configurado, basta abrir o terminal do DNS server e dar um "ping" para algum site, como: # ping www.ifpe.edu.br.

São verdadeiras, apenas, as proposições

- A) I e II.
- B) II, III.
- C) III e IV.
- D) I e IV.
- E) II, III e IV.



26. (UFU-MG/UFU-MG/2020) Considerando-se o sistema operacional Linux Ubuntu e os diversos serviços que podem ser instalados nesse sistema, assinale a alternativa INCORRETA.

A) O serviço Samba é uma implementação do protocolo SMB/CIFS e provê suporte para compartilhamento de arquivos e de impressoras com o sistema operacional Microsoft Windows.

B) A ferramenta de configuração de firewall "ufw" permite filtrar pacotes de rede, liberando ou bloqueando serviços específicos de modo a proteger o computador de acessos indesejados, por exemplo.

C) NFS (Network File System) permite compartilhar uma pasta de um computador com acesso à rede com os demais computadores na mesma rede com a restrição de que tanto o servidor quanto os clientes devem usar um sistema operacional Linux.

D) LDAP é um protocolo de rede para acessar informação armazenada em um serviço de diretório com aplicações como single sign-on, autenticação em aplicações web e de usuários locais.

27. (CEFET-MG/CEFET-MG/2021) Um arquivo de configuração de rede no Debian GNU/Linux 10 é:

A) /etc/network.conf

B) /etc/interfaces.conf

C) /etc/interfaces/network

D) /etc/network/interfaces

E) /etc/network-interfaces

28. (Quadrix/CRC-AP/2021) Em sistemas operacionais Linux derivados da edição Debian, o comando utilizado para fazer uma varredura ou busca por vulnerabilidades em um computador da mesma rede é o comando

A) vuln.

B) netstat.

C) nmap.

D) arp.

E) netcraft.

29. (UFRJ/UFRJ/2021) Um dos modos de se especificar um endereço IP estático para uma interface de rede em um ambiente Linux, como no caso do Debian Linux, é alterando o arquivo:

A) /etc/network/interfaces



- B) /etc/interfaces/links
- C) /etc/link/interfaces
- D) /etc/interfaces/ipconf
- E) /etc/conf/interfaces

30. (Consulplan/SEFAZ-PI/2021) Utilizando o Sistema Operacional Linux é possível configurar um servidor de rede; e, dessa forma, configurar um servidor DHCP (Dynamic Host Configuration Protocol / Protocolo de Configuração de Host Dinâmico) de modo que aos computadores da rede sejam atribuídos endereços IP (Internet Protocol / Protocolo de Internet) automaticamente. Um dos arquivos de configuração DHCP no Linux armazena, automaticamente, as informações de aluguel DHCP atribuídas a todo endereço IP concedido a cada cliente. Assinale-o.

- A) dhcpd.conf
- B) dhcpd.leases
- C) dhcpd.option
- D) dhcpd.server

31. (Quadrix/PRODAM-AM/2022) No sistema operacional Linux, para configurar o servidor dhcpd, o administrador deverá editar o arquivo

- A) access.conf.
- B) limits.conf.
- C) host.conf.
- D) inetd.conf.
- E) dhcpd.conf.

32. (CEFET-MG/CEFET-MG/2022) O caminho e o arquivo para configuração de IP fixo no Linux é

- A) /etc/init.d/network
- B) /etc/network/interfaces
- C) /etc/networking/interfaces
- D) /etc/network/interfaces.conf
- E) /etc/network/interface/interfaces.conf



33. (Consulplan/IF-PA/2023) O arquivo de configuração de rede no Linux é utilizado pelos programas ifup e ifdown para configurar as interfaces de rede do sistema. No Ubuntu, uma das distribuições Linux, este arquivo é:

- A) /etc/hosts
- B) /etc/resolv.conf
- C) /etc/networks.conf
- D) /etc/network/interfaces

34. (CS-UFG/UFG/2023) O Sistema Operacional Linux possui diversos arquivos relacionados à configuração de redes de computadores, dos quais, podemos citar:

- A) /etc/network/dns.conf e /etc/hosts/network.
- B) /etc/hosts e /etc/interface/host.
- C) /etc/network/interfaces e /etc/resolv.conf.
- D) /etc/sysconfig/interfaces e /etc/interfaces/network.

35. (Quadrix/CRM-MG/2023) Assinale a alternativa que apresenta o comando do sistema operacional Red Hat Enterprise Linux que suspende o compartilhamento de arquivos NFS, enquanto mantém todos os serviços NFS ativos.

- A) exportfs -r
- B) ls -la
- C) chmod 750
- D) exportfs -ua
- E) grep | more

36. (FURB/Pref. de J. do Sul-SC/2023) No Linux, o IPTABLES pode ser configurado para bloquear o acesso a sites indesejados na Internet. Um técnico em informática pode, então, definir políticas de restrição de acesso e criar regras de acesso à Internet da Empresa. A respeito da sintaxe do comando, assinale a alternativa correta que configura um bloqueio de um endereço IP específico na Internet:

- A) # iptables -A OUTPUT -p tcp -d 157.240.226.174 -j DROP
- B) # iptables -A INPUT -p tcp -d 157.240.226.2174 -j DROP
- C) # iptables -A OUTPUT -p tcp -d 157.240.226.174 -j BLOCK





- D) # iptables -A INPUT -p tcp -d 157.240.226.174 -j BLOCK
- E) # iptables -D OUTPUT -p tcp -d 157.240.226.174 -j BLOCK

37. (FUNDATEC/PROCERGS/2023) Assinale a alternativa que apresenta o comando nativo do Linux utilizado em todas as versões para visualizar o caminho de um pacote de rede em tempo real.

- A) traceroute
- B) ping,
- C) netstat
- D) ping
- E) ifconfig

38. (IDECAN/SEFAZ-RR/2023) Você recebe a demanda de verificar todos os saltos feitos, na camada de rede, desde o IP da sua máquina (com sistema operacional Linux) até o servidor Web da sua empresa. Selecione o comando que você utilizará na sua máquina para executar essa demanda.

- A) traceroute
- B) ipconfig
- C) netstat
- D) tracert
- E) ping

39. (FUNDATEC/PROCERGS/2023) Assinale a alternativa que apresenta o comando do Linux utilizado para visualizar as conexões de rede estabelecidas, portas utilizadas pela rede e suas estatísticas.

- A) ifconfig
- B) netstat
- C) route
- D) arp
- E) ip



## GABARITO

01	02	03	04	05	06
B	A	C	E	E	C
07	08	09	10	11	12
B	C	D	A	C	A
13	14	15	16	17	18
E	C	D	B	B	A
19	20	21	22	23	24
E	E	E	B	Certo	C
25	26	27	28	29	30
C	C	D	C	A	B
31	32	33	34	35	36
E	B	D	C	D	A
37	38	39			
A	A	B			



## LISTA DE QUESTÕES - FGV

1. (FGV/AL-PR/2024) Um desenvolvedor de software precisa utilizar um sistema operacional FreeBSD para melhor instalar, atualizar e remover pacotes por meio da linha de comando.

Nesse caso, ele vai utilizar o sistema de gerenciamento de pacotes conhecido como

- A) apt.
- B) dnf.
- C) gnu.
- D) pkg.
- E) pacman.

2. (FGV/AL-PR/2024) Ao instalar um sistema operacional FreeBSD em um computador x86, o administrador da máquina configurou o sistema de arquivos para um tipo específico do FreeBSD, que opera em 64 bits, com implementação de snapshots e a possibilidade de fazer a verificação de integridade do sistema de arquivos em segundo plano. Esse sistema de arquivos é o

- A) FAT.
- B) APFS.
- C) HPFS.
- D) UFS2.
- E) UJNFS.



## GABARITO

01	02	03	04	05	06
D	D				
07	08				



## LISTA DE QUESTÕES - MULTIBANCAS

1. (FEPESE/UDESC/2022) São todos elementos que caracterizam uma jail no FreeBSD:

1. Um conjunto de arquivos
2. Uma subárvore de diretórios
3. Um nome de host
4. Um endereço IP
5. Um comando

Assinale a alternativa que indica todas as afirmativas corretas.

- A) São corretas apenas as afirmativas 1, 2, 3 e 4.
- B) São corretas apenas as afirmativas 1, 2, 3 e 5.
- C) São corretas apenas as afirmativas 1, 2, 4 e 5.
- D) São corretas apenas as afirmativas 1, 3, 4 e 5.
- E) São corretas apenas as afirmativas 2, 3, 4 e 5.

2. (FEPESE/UDESC/2022) No contexto do sistema operacional FreeBSD, a coleção de arquivos projetados para automatizar o processo de compilar uma aplicação a partir do código fonte é denominada:

- A) build
- B) make
- C) package
- D) yarn
- E) port



## GABARITO

01	02	03	04	05	06
E	E				
07	08				



# ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



**1** Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



**2** Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



**3** Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



**4** Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



**5** Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



**6** Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



**7** Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



**8** O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.