Estratégia Concursos

Aula 00

TCE-SC (Auditor Fiscal - Ciências da Computação) Arquitetura de Computadores

Autor:
Equipe Informática e TI, Evandro Dalla Vecchia Pereira

06 de Julho de 2024

Índice

Índice

Sistemas Operacionais - Conceitos Básicos

Quando falamos em sistema operacional logo pensamos em Windows, Linux, Android etc. Esses são apenas alguns exemplos dos existentes na atualidade, mas o que de fato é um sistema operacional (S.O.)?

Podemos dizer que basicamente um S.O. possui duas funções:

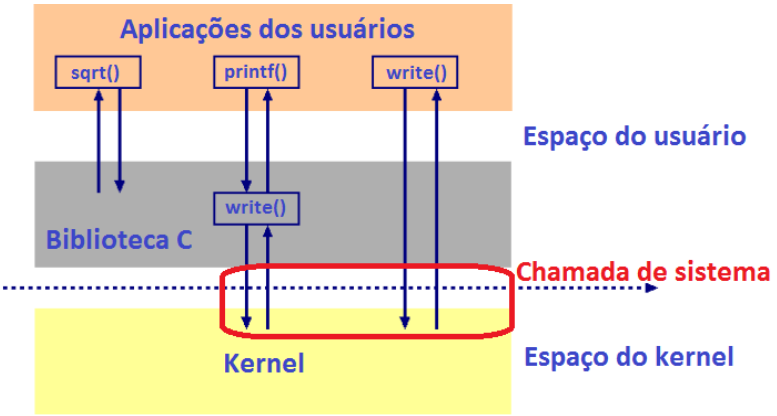
- Apresentar ao usuário uma máquina estendida ou máquina virtual, afinal de contas "alguém" tem que "conversar" com o hardware;

- Gerenciar um sistema complexo: processadores, memórias, discos, dispositivos de entrada e saída (E/S), arquivos etc.

De uma forma mais ampla, algumas funções do S.O. são:

- Permitir aos programas o armazenamento e a obtenção de informações;
- Controlar o fluxo de dados entre os componentes do computador;
- Responder a erros e a pedidos do usuário;
- Impor o escalonamento entre programas que solicitam recursos (memória, disco, entre outros);
- Etc.

Ok, se o S.O. faz o "meio de campo" entre o hardware e os programas do usuário, como um programador faria um acesso a um disco, por exemplo, para ler ou escrever em um arquivo? Para isso existem as chamadas de sistema (*system calls*) que são "instruções estendidas", abstraindo do programador os detalhes de "baixo nível" e garantindo que o programador não faça alguma "bobagem". Vejamos um exemplo para a programação em C:



Conforme vemos na figura acima, existe um espaço do usuário e um espaço do kernel. Mas o que é kernel? Trata-se do núcleo do sistema operacional, "quem" tem um controle total de tudo relacionado ao sistema. O *kernel* é um dos primeiros programas a ser carregado durante a inicialização e assim que começa a ser executado inicia um processo de detecção de todo o hardware necessário para que ocorra um bom funcionamento do computador.

Uma simples alteração da versão do *kernel* pode ser o suficiente para resolver problemas de hardware e de compatibilidade no computador. Além disso, o *kernel* opera solicitações de entrada/saída de software, gerenciamento de memória, aparelhos periféricos, entre outros.

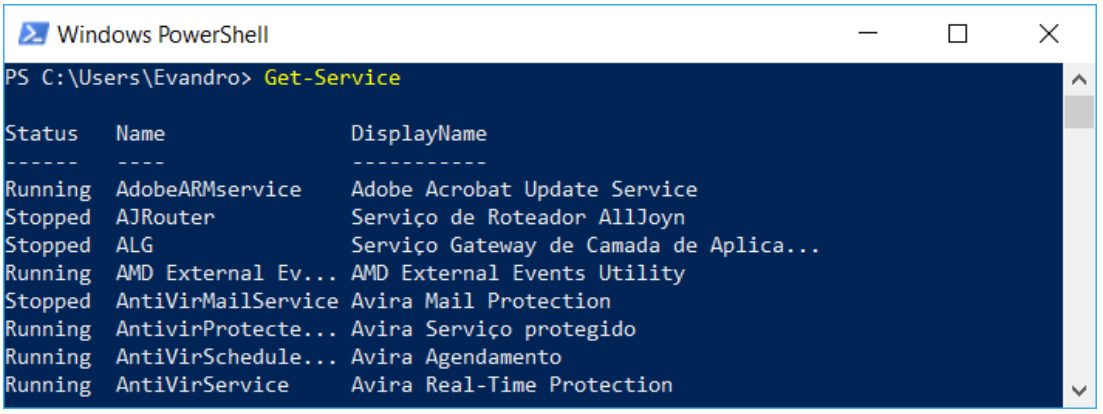
Sabendo disso tudo, vamos voltar à figura. Mesmo para quem não programa em C, os comandos mostrados são intuitivos: sqrt() – *square root* (raiz quadrada) – não precisa realizar uma chamada de sistema, pois recebe um valor e retorna sua raiz quadrada. Mas para escrever em um arquivo – comando write() – é necessário acessar alguma mídia (HD, SSD, pen drive, entre outros) e, para isso, é necessário que o S.O. entre em modo *kernel*. Baixo veremos as definições dos modos kernel e usuário.

Modo kernel: neste modo uma aplicação pode executar instruções não privilegiadas e privilegiadas, ou seja, instruções que oferecem risco ao sistema, ex.: instruções que acessam dados no disco.

Modo usuário: neste modo uma aplicação só pode executar instruções não privilegiadas (instruções que não oferecem riscos ao sistema).

Já falamos do núcleo, agora vamos para a "beirada"...a interface do usuário com o sistema operacional. Na atualidade é muito comum a utilização de GUI (*Graphical User Interface*), ou seja, o usuário apenas clica em janelas, ícones, entre outros elementos para interagir com o S.O. Mas ainda existe o shell, um processo que lê o teclado e espera por comandos, interpreta-os e passa seus parâmetros ao S.O. Por isso também é conhecido como interpretador de comandos.

Antigamente os sistemas operacionais tinham como interface única o shell, mas com o tempo a interface gráfica dominou o mercado (obs.: ainda existem sistemas operacionais em que predomina o shell). Abaixo um exemplo de shell, o PowerShell (Windows):



De uma forma bem específica, encontramos na literatura especializada quatro tipos de gerenciamento realizados por um S.O.:

- Gerenciamento de processos (unidade básica de trabalho do sistema operacional), o que inclui a sua criação, sua exclusão e o fornecimento de mecanismos para a sua comunicação e sincronização;

- Gerenciamento de memória, controlando que partes da memória estão sendo usadas e por quais processos. Além disso, é responsável pela alocação e liberação dinâmica de seu

espaço;

- Gerenciamento de dispositivos de entrada/saída (E/S) ligados ao computador, o que inclui o envio de sinais que informam as ações que o usuário espera que o dispositivo realize, o tratamento das interrupções e erros gerados pelos dispositivos, entre outros;

- Gerenciamento de armazenamento, que inclui o fornecimento do sistema de arquivos para a representação de arquivos e diretórios e o gerenciamento do espaço em dispositivos de armazenamento de dados (HD, SSD, pen drive, entre outros).

Tipos de Kernel e outras classificações

Em relação à arquitetura do *kernel*, o sistema operacional pode ser classificado como monolítico, *microkernel* ou híbrido, conforme veremos a seguir.

Monolítico: os controladores de dispositivos e as extensões de núcleo são executadas no espaço de núcleo, com acesso completo ao hardware. Como todos os módulos são executados em um mesmo espaço de endereçamento, se houver ocorrência de erro em um desses espaços, todo o sistema pode ser afetado. Há um único arquivo objeto, sendo que toda rotina fica visível às demais. Há uma chamada de núcleo (chamada de supervisor) para trocar o modo usuário/núcleo. Alguns exemplos: Linux, BSD e MS-DOS.

Microkernel (micronúcleo): conforme o nome já indica, é um núcleo de tamanho bastante reduzido e, por esse motivo, ele executa o mínimo de processos possível no espaço do *Kernel*. Alguns desses processos são executados no espaço do usuário. Com um micronúcleo, se ocorrer um erro, basta reiniciar o serviço que apresentou o problema. Com isso, evita-se que todo o sistema seja derrubado (como ocorre com o *Kernel* monolítico). Alguns exemplos: AIX, Minix e GNU Hurd.

Híbrido: funciona como um meio-termo, se comparado a sistemas monolíticos e de micronúcleos. O híbrido combina a estabilidade e a segurança do *microkernel* com o desempenho do monolítico. O *kernel* híbrido é semelhante a um micronúcleo, mas tem um código ("não essencial") no espaço do núcleo para que as operações executadas sejam mais rápidas. Alguns exemplos: AmigaOS, Android, Macintosh e Windows.

Sistemas exonúcleos: fornecem um clone do computador real para cada usuário, mas com um subconjunto dos recursos. Por exemplo: uma VM recebe os blocos do disco 0 a 2047 e outra do 2048 a 4095. Na camada inferior existe um programa chamado exonúcleo (*exokernel*).

A ideia é permitir que o desenvolvedor tome todas as decisões relativas ao rendimento do hardware. Os exonúcleos são extremamente pequenos, já que sua função se limita à proteção e à multiplexação dos recursos. Os desenvolvimentos de núcleos clássicos (monolítico ou micronúcleo) abstraem o hardware, deixando esses detalhes "de baixo nível" aos controladores do dispositivo. Nos sistemas clássicos, usando a memória física, ninguém poderá afirmar qual é sua real localização, por exemplo.

A finalidade de um exonúcleo é permitir a uma aplicação que solicite uma região específica da memória, que sejam assegurados os recursos solicitados e que fiquem disponíveis ao programa. Pelo fato do exonúcleo proporcionar uma interface de baixo nível ao hardware, carecendo de todas as funções de alto nível dos outros sistemas operacionais, ele é complementado por uma biblioteca de sistema operacional. Esta biblioteca se comunica com o exonúcleo subjacente e

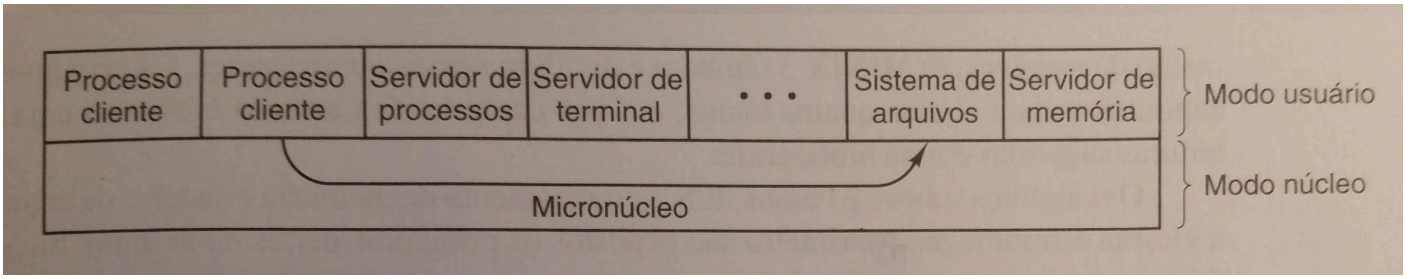
facilita aos programadores de aplicativos com funções que são comuns em outros sistemas operacionais.

Sistemas em camadas: como o nome sugere, é construído sobre uma hierarquia de camadas. O primeiro sistema desenvolvido dessa maneira foi o sistema criado no *Technische Hogeschool Eindhoven* (THE), na Holanda. Tratava-se de um sistema de lote simples para um computador holandês (o Electrologica X8). O S.O. possuía seis camadas:

Camada	Função
5	Operador
4	Programas de usuário
3	Gerenciamento de E/S
2	Comunicação operador-processo
1	Gerenciamento de memória e tambor
0	Alocação do processador e multiprogramação

Máquinas virtuais: Idênticas ao hardware verdadeiro, podendo executar qualquer sistema operacional. É um assunto denso que merece uma aula específica.

Modelo cliente-servidor: possui um núcleo mínimo (*microkernel*), sendo que a maior parte das funções do S.O. ficam em processos de usuário. O cliente obtém o serviço através de mensagens para os processos servidores:



Fonte: Tanenbaum; Woodhull.

Sistemas Monotarefa x Multitarefa

Um processo é uma instância de um programa em execução. Por exemplo, um navegador Web pode estar aberto em cinco janelas diferentes, cada uma com uma página HTML. Cada janela é um processo diferente, pois é uma instância diferente do mesmo programa.

Um sistema multitarefa possui a capacidade de executar vários processos simultaneamente. O sistema operacional divide o tempo do processador (CPU) entre os processos para fornecer a ilusão de execução simultânea. Importante ressaltar que essa ilusão de execução simultânea só ocorre se o sistema for preemptivo (multitarefa preemptiva), ou seja, se tiver a capacidade de interromper a execução de uma tarefa em andamento para que outra tarefa possa fazer uso do

processador. Dessa forma o escalonamento de processos pode ser realizado, através de algum algoritmo específico.

Os sistemas monotarefa não permitem a execução de mais de um processo ao mesmo tempo, portanto não é necessário compartilhar o uso do processador.

Em relação aos processos, eles podem ser executados em primeiro plano (*foreground*) ou em segundo plano (*background*). Um processo em primeiro plano é aquele que está atualmente sendo usado e está visível para o usuário. O usuário pode interagir diretamente com o aplicativo (ex.: navegador Web). Um processo em segundo plano é aquele que está em execução, mas não é visível ou diretamente acessível ao usuário. Ele continua a funcionar, mas suas operações podem ocorrer sem interação direta do usuário (ex.: servidor Web).

Um *daemon* ("serviço") é um tipo especial de processo executado em segundo plano, geralmente sem interação direta com o usuário. O termo *daemon* tem origem do sistema operacional Unix e representa processos que são iniciados durante o boot do sistema e continuam a ser executados enquanto o sistema está ativo. Alguns exemplos são o *daemon* do sistema de impressão (*cupsd*) e o *daemon* de agendamento de tarefas (*cron*).

Conceitos Diversos

Nesta seção vamos ver alguns conceitos diversos cobrados em provas de concurso.

Journaling: técnica usada em sistemas de arquivos (gerenciamento de armazenamento) para melhorar a integridade e recuperação de dados em caso de falhas ou interrupções inesperadas. É comum em sistemas de arquivos utilizados por sistemas operacionais modernos. A ideia principal é manter um "jornal" (ou log) que registra as operações que serão realizadas antes de serem efetivamente aplicadas no sistema de arquivos. Isso ajuda a garantir a consistência dos dados em caso de falhas (falta de energia, panes do sistema etc.).

Spool de impressão: SPOOL é uma abreviação de *Simultaneous Peripheral Operations On-Line* (Operações Periféricas Simultâneas On-line) e geralmente está associado a impressões. O termo "spool de impressão" refere-se a um sistema que permite que vários trabalhos de impressão sejam enviados para uma fila e processados em ordem. A ideia do "spool de impressão" é melhorar a eficiência do processo de impressão, fazendo com que quando um trabalho de impressão é enviado para a fila, ele fique armazenado temporariamente em um local (conhecido como "spool"). Enquanto um trabalho estiver no spool, ele pode ser cancelado, pausado, pode ser dado maior prioridade, entre outras atividades, dependendo do sistema operacional.

Serviços prestados pelo sistema operacional: além dos gerenciamento fundamentais que o sistema operacional realiza (processos, memória, E/S e armazenamento), alguns outros também são realizados, tais como:

- manutenção da data/hora, permitindo inclusive a alteração (se houver permissão para isso);
- lista dos usuários que estão usando o computador;
- spool de impressão, permitindo a alteração (se houver permissão para isso);
- serviços de acessibilidade;
- sistema de segurança relacionado ao acesso de arquivos/diretórios: permissões de acesso a usuários autorizados;
- controle de acesso através de usuários e senhas;
- etc.

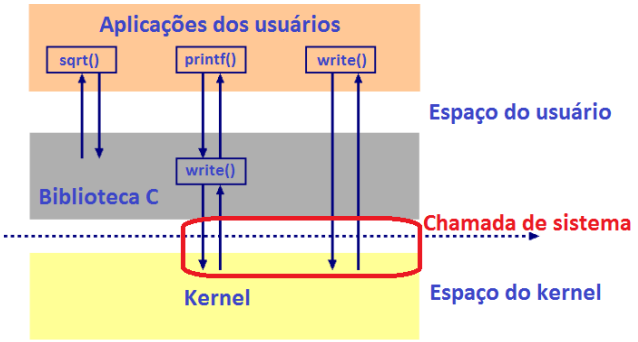
QUESTÕES COMENTADAS - CEBRASPE

1. (CEBRASPE/SEDF/2017) Acerca dos princípios de sistemas operacionais, julgue o item a seguir.

   A interface entre o sistema operacional e os programas de usuário é definida por um conjunto de instruções estendidas denominadas chamadas de sistema.

Comentários:

As chamadas de sistema (*system calls*) são "instruções estendidas", abstraindo do programador os detalhes de "baixo nível" e garantindo que o programador não faça alguma "bobagem". Vejamos um exemplo para a programação em C:



Gabarito: Certo

2. (CEBRASPE/ABIN/2018) Julgue o item a seguir, relativo a sistemas operacionais.

   O início da execução de um programa provocada pelo usuário leva à criação de processos deamons.

Comentários:

Um *daemon* ("serviço") é um tipo especial de processo executado em segundo plano, geralmente sem interação direta com o usuário. O termo *daemon* tem origem do sistema operacional Unix e representa processos que são iniciados durante o boot do sistema e continuam a ser executados enquanto o sistema está ativo. Alguns exemplos são o *daemon* do sistema de impressão (*cupsd*) e o *daemon* de agendamento de tarefas (*cron*).

Gabarito: Errado

3. (CEBRASPE/Min. da Economia/2020) Julgue o próximo item, relativos a sistemas operacionais.

   O sistema operacional atua como alocador e gerenciador dos recursos que um sistema computacional possui, tais como tempo de CPU, espaço de memória e dispositivos de entrada/saída.

Comentários:

De uma forma bem específica, encontramos na literatura especializada quatro tipos de gerenciamento realizados por um S.O.:

- Gerenciamento de processos (unidade básica de trabalho do sistema operacional), o que inclui a sua criação, sua exclusão e o fornecimento de mecanismos para a sua comunicação e sincronização;
- Gerenciamento de memória, controlando que partes da memória estão sendo usadas e por quais processos. Além disso, é responsável pela alocação e liberação dinâmica de seu espaço;

- Gerenciamento de dispositivos de entrada/saída (E/S) ligados ao computador, o que inclui o envio de sinais que informam as ações que o usuário espera que o dispositivo realize, o tratamento das interrupções e erros gerados pelos dispositivos, entre outros;
- Gerenciamento de armazenamento, que inclui o fornecimento do sistema de arquivos para a representação de arquivos e diretórios e o gerenciamento do espaço em dispositivos de armazenamento de dados (HD, SSD, pen drive, entre outros).

Gabarito: Certo

QUESTÕES COMENTADAS - FGV

1. (FGV/TJ-TO/2022) O journaling é um recurso suportado por alguns sistemas de arquivos e sua função é:

A) replicar os volumes de disco lógico em discos rígidos físicos separados para garantir disponibilidade contínua;

B) segmentar os dados e distribuí-los por diferentes dispositivos de armazenamento físico;

C) registrar metadados acerca das operações feitas nas estruturas de dados e diretórios do sistema de arquivo;

D) fazer cache da deduplicação de dados para reduzir as necessidades de armazenamento;

E) garantir que dados excluídos definitivamente do disco rígido sejam irrecuperáveis.

Comentários:

Journaling: técnica usada em sistemas de arquivos (gerenciamento de armazenamento) para melhorar a integridade e recuperação de dados em caso de falhas ou interrupções inesperadas. É comum em sistemas de arquivos utilizados por sistemas operacionais modernos. A ideia principal é manter um "jornal" (ou log) que registra as operações que serão realizadas antes de serem efetivamente aplicadas no sistema de arquivos. Isso ajuda a garantir a consistência dos dados em caso de falhas (falta de energia, panes do sistema etc.).

Gabarito: C

Questões Comentadas - FCC

1. (FCC/DPE-SP/2010) NÃO é uma função do sistema operacional:

A) Permitir aos programas armazenar e obter informações.

B) Controlar o fluxo de dados entre os componentes do computador.

C) Responder a erros e a pedidos do usuário.

D) Impor escalonamento entre programas que solicitam recursos.

E) Gerenciar apenas a base de dados.

Comentários:

De uma forma mais ampla, algumas funções do S.O. são:

- Permitir aos programas o armazenamento e a obtenção de informações;
- Controlar o fluxo de dados entre os componentes do computador;
- Responder a erros e a pedidos do usuário.
- Impor o escalonamento entre programas que solicitam recursos (memória, disco, entre outros).

A alternativa E está bem longe de ser uma função do S.O., ainda mais que expressa "APENAS" e ainda uma "BASE DE DADOS" genérica.

Gabarito: E

2. (FCC/TRT16/2014) Um Sistema Operacional (SO) realiza o gerenciamento

..I.. , que inclui o fornecimento do sistema de arquivos para a representação de arquivos e diretórios e o gerenciamento do espaço em dispositivos com grande capacidade de armazenamento de dados.

..II.. , que são a unidade básica de trabalho do SO. Isso inclui a sua criação, sua exclusão e o fornecimento de mecanismos para a sua comunicação e sincronização.

..III.. , controlando que partes estão sendo usadas e por quem. Além disso, é responsável pela alocação e liberação dinâmica de seu espaço.

As lacunas I, II e III são, correta e respectivamente, preenchidas por:

A) de armazenamento - de processos - de memória

B) em memória secundária - de serviços - em memória principal

C) de arquivos - de barramentos - de discos

D) de discos - de threads - de cache

E) de I/O - de tempos de CPU - de RAM

Comentários:

De uma forma bem específica, encontramos na literatura especializada quatro tipos de gerenciamento realizados por um S.O. (que serão abordados em tópicos específicos nesta aula):

- Gerência de processos (unidade básica de trabalho do sistema operacional), o que inclui a sua criação, sua exclusão e o fornecimento de mecanismos para a sua comunicação e sincronização;
- Gerência de memória, controlando que partes estão sendo usadas e por quem. Além disso, é responsável pela alocação e liberação dinâmica de seu espaço;
- Gerência de dispositivos de entrada/saída (E/S) ligados ao computador, o que inclui o envio de sinais que informam as ações que o usuário espera que o dispositivo realize, o tratamento das interrupções e erros gerados pelos dispositivos, entre outros;
- Gerência de armazenamento, que inclui o fornecimento do sistema de arquivos para a representação de arquivos e diretórios e o gerenciamento do espaço em dispositivos de armazenamento de dados (HD, SSD, pen drive, entre outros).

Gabarito: A

3. (FCC/TRF3/2016) Um Técnico Judiciário de TI do TRF3, ao estudar os princípios dos sistemas operacionais, teve sua atenção voltada ao processo que perfaz a interface do usuário com o sistema operacional. Observou que este processo lê o teclado a espera de comandos, interpreta-os e passa seus parâmetros ao sistema operacional. Entendeu, com isto, que serviços como login/logout, manipulação de arquivos e execução de programas são, portanto, solicitados por meio do interpretador de comandos ou

A) Kernel.

B) System Calls.

C) Shell.

D) Cache.

E) Host.

Comentários:

Na atualidade é muito comum a utilização de GUI (Graphical User Interface), ou seja, o usuário apenas clica em janelas, ícones, entre outros elementos para interagir com S.O. Mas ainda existe o shell, um processo que lê o teclado e espera por comandos, interpreta-os e passa seus parâmetros ao S.O. Por isso também é conhecido como interpretador de comandos.

Gabarito: C

QUESTÕES COMENTADAS - VUNESP

1. (VUNESP/PC-BA/2018) As versões modernas do sistema operacional Windows, como a versão 10, aparentam ao seu usuário que várias tarefas são executadas ao mesmo tempo. Essa característica é conhecida como

A) monotarefa preemptiva.

B) monotarefa sem preempção.

C) multitarefa preemptiva.

D) multitarefa sem preempção.

E) time-sharing sem preempção.

Comentários:

Um sistema multitarefa possui a capacidade de executar vários processos simultaneamente. O sistema operacional divide o tempo do processador (CPU) entre os processos para fornecer a ilusão de execução simultânea. Importante ressaltar que essa ilusão de execução simultânea só ocorre se o sistema for preemptivo (multitarefa preemptiva), ou seja, se tiver a capacidade de interromper a execução de uma tarefa em andamento para que outra tarefa possa fazer uso do processador. Dessa forma o escalonamento de processos pode ser realizado, através de algum algoritmo específico.

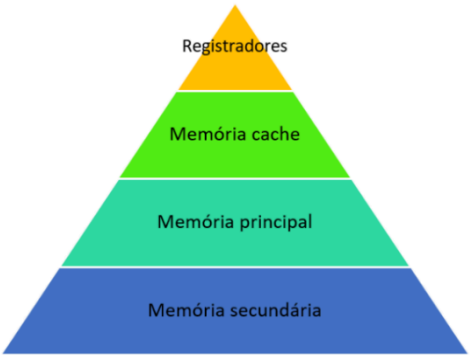
Gabarito: C

QUESTÕES COMENTADAS - MULTIBANCAS

1. (Quadrix/COFECI/2017) O gerenciador de memória é a parte do sistema operacional que gerencia, parcialmente, a hierarquia de memórias.

Comentários:

Vamos ver uma figura simples sobre a hierarquia de memória:



O gerenciador de memória faz o gerenciamento parcial porque não gerencia a hierarquia completa, como por exemplo a memória secundária. Para gravar/ler de um HD é necessário que sinais sejam enviados (gerência de E/S) e que o sistema de arquivos entre em ação, para definir onde está um arquivo (quais blocos do HD), o tamanho do bloco etc.

Gabarito: Certo

2. (UPENET-IAUPE/UPE/2017) O software responsável pelo gerenciamento dos recursos do hardware para o usuário, a fim de que os softwares aplicativos não tenham que interagir diretamente com os dispositivos periféricos, é definido como

A) compilador.

B) driver.

C) sistema operacional.

D) drive.

E) controlador.

Comentários:

Seria muito complicado exigir que todo programador tivesse conhecimento do hardware a ser utilizado. Também seria uma programação muito mais complexa e demorada! Podemos dizer que basicamente um S.O. possui duas funções:

- Apresentar ao usuário uma máquina estendida ou máquina virtual, afinal de contas "alguém" tem que "conversar" com o hardware;
- Gerenciar um sistema complexo: processadores, memórias, discos, dispositivos de E/S, arquivos etc.

Gabarito: C

3. (IESES/IGP-SC/2017) Considere as afirmativas abaixo referentes as funções que são de responsabilidade de um Sistema Operacional Moderno:

I. Controlar os dispositivos de entrada/saída.

II. Efetuar o gerenciamento de programas em execução.

III. Oferecer mecanismos de proteção aos recursos básicos do computador.

Estão corretas as afirmativas:

A) I e III

B) II e III

C) I, II e III

D) I e II

Comentários:

(I) Faz parte da gerência de E/S; (II) Gerência de processos; (III) Como a questão fala em sistema operacional moderno, o oferecimento de mecanismos de proteção poderia ser considerado certo. Aí depende de qual a fonte consultada para elaborar a questão. De qualquer forma a questão foi anulada (eu marcaria a alternativa C), com a justificativa de que esse assunto não estava no edital.

Gabarito: Anulada

4. (AOCP/PRODEB/2018) A difusão dos primeiros computadores pessoais (PCs), feita pela Apple e IBM, aconteceu em 1981, época em que houve um salto muito grande em termos de tecnologia e utilidade dessas máquinas, caracterizadas, já na década de 1990, pela versatilidade extraordinária de transformar o processamento e o armazenamento de dados centralizados em um sistema compartilhado e interativo de computadores em rede. A partir de então, esse processo, veio só a se desenvolver e se disseminar em praticamente todas as áreas de atuação. Desde aquela época até hoje, um dos pontos principais da computação são os Sistemas Operacionais (SO). Assinale a alternativa correta sobre o que são os SO.

A) Um programa ou conjunto de programas cuja função é gerenciar os recursos do sistema, fornecendo uma interface entre o computador e o usuário.

B) São simples interfaces para a comunicação entre o hardware e o usuário.

C) São dispositivos conectados à placa mãe que têm como função gerenciar os recursos de hardware.

D) Sistemas que têm como única e exclusiva função realizar a ligação e a troca de dados entre computadores pessoais.

E) Sistemas para controle e gerenciamento de recursos, que existem exclusivamente em computadores pessoais.

Comentários:

Podemos dizer que basicamente um S.O. possui duas funções:

- Apresentar ao usuário uma máquina estendida ou máquina virtual, afinal de contas "alguém" tem que "conversar" com o hardware;

- Gerenciar um sistema complexo: processadores, memórias, discos, dispositivos de entrada e saída (E/S), arquivos etc.

De uma forma mais ampla, algumas funções do S.O. são:

- Permitir aos programas o armazenamento e a obtenção de informações;
- Controlar o fluxo de dados entre os componentes do computador;
- Responder a erros e a pedidos do usuário;
- Impor o escalonamento entre programas que solicitam recursos (memória, disco, entre outros);
- Etc.

Gabarito: A

5. (Quadrix/CRQ 4ª Região-SP/2018) Quanto a sistemas operacionais, julgue o item.

   Em ambiente multiprogramação, é necessário que exista uma proteção, por exemplo, contra o acesso de dispositivos de E/S ou a alocação de memória por mais de um programa ao mesmo tempo. Assim, o sistema operacional e os programas de usuários operam em modo privilegiado, o que garante que os conflitos no uso dos recursos não ocorram.

Comentários:

Os programas de usuário não operam em modo privilegiado, senão não haveria segurança nenhuma! Quando é necessário "baixar o nível" para ter acesso a funções no modo privilegiado, são realizadas chamadas de sistema (ex.: acessar um disco rígido ou outra mídia de armazenamento).

Gabarito: Errado

6. (Quadrix/CRQ 4ª Região-SP/2018) Quanto a sistemas operacionais, julgue o item.

   No emprego da técnica denominada de multitarefa, mesmo que o sistema computacional possua somente um processador, tem-se a ilusão de que vários programas estão sendo executados simultaneamente.

Comentários:

Um sistema multitarefa possui a capacidade de executar vários processos simultaneamente. O sistema operacional divide o tempo do processador (CPU) entre os processos para fornecer a

ilusão de execução simultânea. Importante ressaltar que essa ilusão de execução simultânea só ocorre se o sistema for preemptivo (multitarefa preemptiva), ou seja, se tiver a capacidade de interromper a execução de uma tarefa em andamento para que outra tarefa possa fazer uso do processador. Dessa forma o escalonamento de processos pode ser realizado, através de algum algoritmo específico.

Gabarito: Certo

7. (COSEAC/UFF - 2019) Os sistemas operacionais normalmente possuem uma casca, que é a parte visível com a qual o usuário entra em contato, e outra parte interna. Essas duas partes são conhecidas, respectivamente, por:

A) API e shell.

B) GUI e cluster.

C) shell e kernel.

D) kernel e CPU.

E) buffers e spooling.

Comentários:

"Casca" poderíamos interpretar como quem faz a interface de quem está "fora" com o núcleo e isso é papel do shell (ou alguma interface gráfica, claro). A parte interna podemos interpretar como o núcleo, ou seja, o kernel.

Gabarito: C

8. (AOCP/UFFS/2019) A interface entre o sistema operacional e os programas de usuários é definida por um conjunto de instruções estendidas disponibilizadas pelo sistema operacional. Essas instruções são denominadas

A) processos.

B) chamadas de sistema.

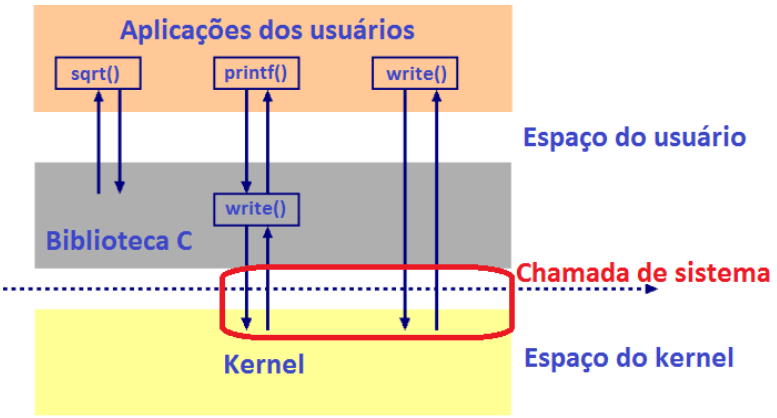
C) prompt de comando.

D) shell.

E) vetores de interrupção.

Comentários:

As chamadas de sistema (*system calls*) são "instruções estendidas", abstraindo do programador os detalhes de "baixo nível" e garantindo que o programador não faça alguma "bobagem". Vejamos um exemplo para a programação em C:



Gabarito: B

9. (IBADE/Pref. de Itapemirim-ES/2019) Existe uma função em diversos sistemas operacionais que consiste em armazenar em disco os arquivos de impressão produzidos pelos aplicativos e promover sua impressão de forma sequencial, permitindo ao usuário intervir na ordem da fila, cancelar arquivos, suspender a impressão, etc. Chama-se essa função de:

A) Buffer.

B) Spooling.

C) Print File.

D) Defrag.

E) Retain.

Comentários:

SPOOL é uma abreviação de *Simultaneous Peripheral Operations On-Line* (Operações Periféricas Simultâneas On-line) e geralmente está associado a impressões. O termo "spool de impressão" refere-se a um sistema que permite que vários trabalhos de impressão sejam enviados para uma fila e processados em ordem. A ideia do "spool de impressão" é melhorar a eficiência do processo de impressão, fazendo com que quando um trabalho de impressão é enviado para a fila, ele fique armazenado temporariamente em um local (conhecido como "spool"). Enquanto um trabalho estiver no spool, ele pode ser cancelado, pausado, pode ser dado maior prioridade, entre outras atividades, dependendo do sistema operacional.

Gabarito: B

10. (UPENET-IAUPE/Prefeitura de Carnaíba-PE/2019) Uma das características para o uso de diferentes programas, sendo executados ao mesmo tempo em um computador, denomina-se

A) Interpretativa.

B) Colegiada.

C) Multitarefa.

D) Conectado.

E) Diretório.

Comentários:

Um sistema multitarefa possui a capacidade de executar vários processos simultaneamente. O sistema operacional divide o tempo do processador (CPU) entre os processos para fornecer a ilusão de execução simultânea. Importante ressaltar que essa ilusão de execução simultânea só ocorre se o sistema for preemptivo (multitarefa preemptiva), ou seja, se tiver a capacidade de interromper a execução de uma tarefa em andamento para que outra tarefa possa fazer uso do processador. Dessa forma o escalonamento de processos pode ser realizado, através de algum algoritmo específico.

Gabarito: C

11. (UFMT/UFT/2019) Os sistemas operacionais constituem interfaces de abstração do hardware e permitem aos usuários de diferentes níveis de capacitação utilizarem adequadamente o computador. Independentemente da interface de usuários, os sistemas operacionais são baseados em princípios comuns que permitem a interoperabilidade. Sobre o assunto, assinale a afirmativa INCORRETA.

A) A manipulação de objetos como arquivos é feita pelas system calls, tornando transparente aos usuários as complexas operações.

B) Os programas dos usuários se comunicam com o sistema operacional por meio de system calls, que são procedimentos que devem ser escritos pelo usuário.

C) Um processo é basicamente um programa em execução que usa os recursos de hardware e do sistema operacional, como pilha, ponteiros, registradores e outros.

D) Os sistemas operacionais implementam os conceitos de processos, arquivos, chamadas de sistemas e interface de usuários (comandos ou gráficas).

Comentários:

A) CORRETA - Para tudo que envolver atividades de mais "baixo nível", como por exemplo, o acesso a mídias de armazenamento, são utilizadas as chamadas de sistema (*system calls*).

B) INCORRETA - *System calls* não são escritas pelo usuário. O programador apenas faz a chamada de sistema (*system call*). Um exemplo clássico é para ler um arquivo.

C) CORRETA - Um processo é uma instância de um programa em execução.

D) CORRETA - Os sistemas operacionais implementam os conceitos de processos (instâncias de programas em execução), arquivos (armazenados em alguma mídia), chamadas de sistemas (para ter acesso ao modo kernel) e interface de usuários (comandos via shell ou gráficas).

Gabarito: B

12. (Consulplan/Pref. de Formiga-MG/2020) Um Sistema Operacional pode ser definido como um programa que gerencia o computador, de modo que trabalhe de maneira correta, gerando um ambiente de comunicação, que faz a interação entre o usuário e a máquina. Dois subsistemas do Sistema Operacional desempenham essas funções; assinale-os.

A) BIOS e Setup.

B) Shell e Kernel.

C) Processador e Memória.

D) Registradores e Memória Cache.

Comentários:

Shell: lê, interpreta e executa comandos. Faz o meio de campo entre o usuário e o kernel.

Kernel: núcleo do sistema operacional.

Gabarito: B

13. (Quadrix/CREFITO-MG/2021) A manutenção de informações a respeito de seu estado (data atual, hora atual, lista dos usuários que estão usando o computador, entre outras) pode ser considerada como um serviço que é oferecido pelo sistema operacional.

Comentários:

Serviços prestados pelo sistema operacional: além dos gerenciamento fundamentais que o sistema operacional realiza (processos, memória, E/S e armazenamento), alguns outros também são realizados, tais como:

- manutenção da data/hora, permitindo inclusive a alteração (se houver permissão para isso);
- lista dos usuários que estão usando o computador;
- spool de impressão, permitindo a alteração (se houver permissão para isso);
- serviços de acessibilidade;
- sistema de segurança relacionado ao acesso de arquivos/diretórios: permissões de acesso a usuários autorizados;
- controle de acesso através de usuários e senhas;
- etc.

Gabarito: Certo

14. (Quadrix/CREFITO-MG/2021) O objetivo do sistema operacional é distribuir recursos do computador, como, por exemplo, espaço na memória principal e tempo de processador, para torná-lo mais eficiente.

Comentários:

Um dos grandes objetivos do S.O. é gerenciar os recursos, incluindo:

- "espaço na memória principal": faz parte do gerenciamento de memória;
- "tempo de processador": faz parte do gerenciamento de processos;
- entre outros.

Gabarito: Certo

15. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

É função do sistema operacional gerenciar o sistema de segurança de modo que os arquivos sejam acessíveis apenas por usuários autorizados.

Comentários:

Serviços prestados pelo sistema operacional: além dos gerenciamento fundamentais que o sistema operacional realiza (processos, memória, E/S e armazenamento), alguns outros também são realizados, tais como:

- manutenção da data/hora, permitindo inclusive a alteração (se houver permissão para isso);
- lista dos usuários que estão usando o computador;
- spool de impressão, permitindo a alteração (se houver permissão para isso);
- serviços de acessibilidade;
- sistema de segurança relacionado ao acesso de arquivos/diretórios: permissões de acesso a usuários autorizados;
- controle de acesso através de usuários e senhas;
- etc.

Gabarito: Certo

16. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

Nem mesmo os sistemas operacionais mais modernos permitem que múltiplos programas estejam na memória principal ao mesmo tempo.

Comentários:

É característica comum aos sistemas operacionais modernos ser multitarefa, ou seja, permitem que mais de um processo seja executado ao mesmo tempo. Existe uma ilusão de paralelismo, mas na verdade um processador compartilha o tempo com mais de um processo através de um algoritmo de escalonamento.

Gabarito: Errado

17. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

O sistema operacional tem como função, entre outras, ocultar as particularidades dos discos e de outros dispositivos de E/S, com a finalidade de fornecer ao programador um modelo de arquivos agradável e independente de dispositivos.

Comentários:

A característica ressaltada na questão está relacionada aos modos usuário e kernel. As particularidades mais "baixo nível", como acesso a discos e outros dispositivos de E/S são preocupação do modo kernel e o programador só tem que saber usar as chamadas de sistema para ter acesso.

Gabarito: Certo

18. (Quadrix/CREFITO-MG/2021) Os utilitários (programas de sistema), uma modalidade específica de programa, são sempre executados dentro do kernel do sistema operacional.

Comentários:

No kernel (núcleo) estão as funções principais de um sistema operacional. Os utilitários (software de compactação, por exemplo) não ficam no kernel! Eles são instalados no sistema operacional e atuam no modo usuário.

Gabarito: Errado

Lista de Questões - CEBRASPE

1. (CEBRASPE/SEDF/2017) Acerca dos princípios de sistemas operacionais, julgue o item a seguir.

   A interface entre o sistema operacional e os programas de usuário é definida por um conjunto de instruções estendidas denominadas chamadas de sistema.

2. (CEBRASPE/ABIN/2018) Julgue o item a seguir, relativo a sistemas operacionais.

   O início da execução de um programa provocada pelo usuário leva à criação de processos deamons.

3. (CEBRASPE/Min. da Economia/2020) Julgue o próximo item, relativos a sistemas operacionais.

   O sistema operacional atua como alocador e gerenciador dos recursos que um sistema computacional possui, tais como tempo de CPU, espaço de memória e dispositivos de entrada/saída.

GABARITO

GABARITO

1. Certo
2. Errado
3. Certo

Lista de Questões - FGV

1. (FGV/TJ-TO/2022) O journaling é um recurso suportado por alguns sistemas de arquivos e sua função é:

A) replicar os volumes de disco lógico em discos rígidos físicos separados para garantir disponibilidade contínua;

B) segmentar os dados e distribuí-los por diferentes dispositivos de armazenamento físico;

C) registrar metadados acerca das operações feitas nas estruturas de dados e diretórios do sistema de arquivo;

D) fazer cache da deduplicação de dados para reduzir as necessidades de armazenamento;

E) garantir que dados excluídos definitivamente do disco rígido sejam irrecuperáveis.

GABARITO

GABARITO

1. Letra C

Lista de Questões - FCC

1. (FCC/DPE-SP/2010) NÃO é uma função do sistema operacional:

A) Permitir aos programas armazenar e obter informações.

B) Controlar o fluxo de dados entre os componentes do computador.

C) Responder a erros e a pedidos do usuário.

D) Impor escalonamento entre programas que solicitam recursos.

E) Gerenciar apenas a base de dados.

2. (FCC/TRT16/2014) Um Sistema Operacional (SO) realiza o gerenciamento

..I.. , que inclui o fornecimento do sistema de arquivos para a representação de arquivos e diretórios e o gerenciamento do espaço em dispositivos com grande capacidade de armazenamento de dados.

..II.. , que são a unidade básica de trabalho do SO. Isso inclui a sua criação, sua exclusão e o fornecimento de mecanismos para a sua comunicação e sincronização.

..III.. , controlando que partes estão sendo usadas e por quem. Além disso, é responsável pela alocação e liberação dinâmica de seu espaço.

As lacunas I, II e III são, correta e respectivamente, preenchidas por:

A) de armazenamento - de processos - de memória

B) em memória secundária - de serviços - em memória principal

C) de arquivos - de barramentos - de discos

D) de discos - de threads - de cache

E) de I/O - de tempos de CPU - de RAM

3. (FCC/TRF3/2016) Um Técnico Judiciário de TI do TRF3, ao estudar os princípios dos sistemas operacionais, teve sua atenção voltada ao processo que perfaz a interface do usuário com o sistema operacional. Observou que este processo lê o teclado a espera de comandos, interpreta-os e passa seus parâmetros ao sistema operacional. Entendeu, com isto, que serviços como login/logout, manipulação de arquivos e execução de programas são, portanto, solicitados por meio do interpretador de comandos ou

A) Kernel.

B) System Calls.

C) Shell.

D) Cache.

E) Host.

GABARITO

GABARITO

1. Letra E
2. Letra A
3. Letra C

Lista de Questões - VUNESP

1. (VUNESP/PC-BA/2018) As versões modernas do sistema operacional Windows, como a versão 10, aparentam ao seu usuário que várias tarefas são executadas ao mesmo tempo. Essa característica é conhecida como

A) monotarefa preemptiva.

B) monotarefa sem preempção.

C) multitarefa preemptiva.

D) multitarefa sem preempção.

E) time-sharing sem preempção.

GABARITO

GABARITO

1. Letra C

Lista de Questões - MULTIBANCAS

1. (Quadrix/COFECI/2017) O gerenciador de memória é a parte do sistema operacional que gerencia, parcialmente, a hierarquia de memórias.

2. (UPENET-IAUPE/UPE/2017) O software responsável pelo gerenciamento dos recursos do hardware para o usuário, a fim de que os softwares aplicativos não tenham que interagir diretamente com os dispositivos periféricos, é definido como

A) compilador.

B) driver.

C) sistema operacional.

D) drive.

E) controlador.

3. (IESES/IGP-SC/2017) Considere as afirmativas abaixo referentes as funções que são de responsabilidade de um Sistema Operacional Moderno:

I. Controlar os dispositivos de entrada/saída.

II. Efetuar o gerenciamento de programas em execução.

III. Oferecer mecanismos de proteção aos recursos básicos do computador.

Estão corretas as afirmativas:

A) I e III

B) II e III

C) I, II e III

D) I e II

4. (AOCP/PRODEB/2018) A difusão dos primeiros computadores pessoais (PCs), feita pela Apple e IBM, aconteceu em 1981, época em que houve um salto muito grande em termos de tecnologia e utilidade dessas máquinas, caracterizadas, já na década de 1990, pela versatilidade extraordinária de transformar o processamento e o armazenamento de dados centralizados em um sistema compartilhado e interativo de computadores em rede. A partir de então, esse processo, veio só a se desenvolver e se disseminar em praticamente todas as áreas de atuação. Desde aquela época até hoje, um dos pontos principais da computação são os Sistemas Operacionais (SO). Assinale a alternativa correta sobre o que são os SO.

A) Um programa ou conjunto de programas cuja função é gerenciar os recursos do sistema, fornecendo uma interface entre o computador e o usuário.

B) São simples interfaces para a comunicação entre o hardware e o usuário.

C) São dispositivos conectados à placa mãe que têm como função gerenciar os recursos de hardware.

D) Sistemas que têm como única e exclusiva função realizar a ligação e a troca de dados entre computadores pessoais.

E) Sistemas para controle e gerenciamento de recursos, que existem exclusivamente em computadores pessoais.

5. (Quadrix/CRQ 4ª Região-SP/2018) Quanto a sistemas operacionais, julgue o item.

   Em ambiente multiprogramação, é necessário que exista uma proteção, por exemplo, contra o acesso de dispositivos de E/S ou a alocação de memória por mais de um programa ao mesmo tempo. Assim, o sistema operacional e os programas de usuários operam em modo privilegiado, o que garante que os conflitos no uso dos recursos não ocorram.

6. (Quadrix/CRQ 4ª Região-SP/2018) Quanto a sistemas operacionais, julgue o item.

   No emprego da técnica denominada de multitarefa, mesmo que o sistema computacional possua somente um processador, tem-se a ilusão de que vários programas estão sendo executados simultaneamente.

7. (COSEAC/UFF - 2019) Os sistemas operacionais normalmente possuem uma casca, que é a parte visível com a qual o usuário entra em contato, e outra parte interna. Essas duas partes são conhecidas, respectivamente, por:

A) API e shell.

B) GUI e cluster.

C) shell e kernel.

D) kernel e CPU.

E) buffers e spooling.

8. (AOCP/UFFS/2019) A interface entre o sistema operacional e os programas de usuários é definida por um conjunto de instruções estendidas disponibilizadas pelo sistema operacional. Essas instruções são denominadas

A) processos.

B) chamadas de sistema.

C) prompt de comando.

D) shell.

E) vetores de interrupção.

9. (IBADE/Pref. de Itapemirim-ES/2019) Existe uma função em diversos sistemas operacionais que consiste em armazenar em disco os arquivos de impressão produzidos pelos aplicativos e promover sua impressão de forma sequencial, permitindo ao usuário intervir na ordem da fila, cancelar arquivos, suspender a impressão, etc. Chama-se essa função de:

A) Buffer.

B) Spooling.

C) Print File.

D) Defrag.

E) Retain.

10. (UPENET-IAUPE/Prefeitura de Carnaíba-PE/2019) Uma das características para o uso de diferentes programas, sendo executados ao mesmo tempo em um computador, denomina-se

A) Interpretativa.

B) Colegiada.

C) Multitarefa.

D) Conectado.

E) Diretório.

11. (UFMT/UFT/2019) Os sistemas operacionais constituem interfaces de abstração do hardware e permitem aos usuários de diferentes níveis de capacitação utilizarem adequadamente o computador. Independentemente da interface de usuários, os sistemas operacionais são baseados em princípios comuns que permitem a interoperabilidade. Sobre o assunto, assinale a afirmativa INCORRETA.

A) A manipulação de objetos como arquivos é feita pelas system calls, tornando transparente aos usuários as complexas operações.

B) Os programas dos usuários se comunicam com o sistema operacional por meio de system calls, que são procedimentos que devem ser escritos pelo usuário.

C) Um processo é basicamente um programa em execução que usa os recursos de hardware e do sistema operacional, como pilha, ponteiros, registradores e outros.

D) Os sistemas operacionais implementam os conceitos de processos, arquivos, chamadas de sistemas e interface de usuários (comandos ou gráficas).

12. (Consulplan/Pref. de Formiga-MG/2020) Um Sistema Operacional pode ser definido como um programa que gerencia o computador, de modo que trabalhe de maneira correta, gerando um ambiente de comunicação, que faz a interação entre o usuário e a máquina. Dois subsistemas do Sistema Operacional desempenham essas funções; assinale-os.

A) BIOS e Setup.

B) Shell e Kernel.

C) Processador e Memória.

D) Registradores e Memória Cache.

13. (Quadrix/CREFITO-MG/2021) A manutenção de informações a respeito de seu estado (data atual, hora atual, lista dos usuários que estão usando o computador, entre outras) pode ser considerada como um serviço que é oferecido pelo sistema operacional.

14. (Quadrix/CREFITO-MG/2021) O objetivo do sistema operacional é distribuir recursos do computador, como, por exemplo, espaço na memória principal e tempo de processador, para torná-lo mais eficiente.

15. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

    É função do sistema operacional gerenciar o sistema de segurança de modo que os arquivos sejam acessíveis apenas por usuários autorizados.

16. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

    Nem mesmo os sistemas operacionais mais modernos permitem que múltiplos programas estejam na memória principal ao mesmo tempo.

17. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

    O sistema operacional tem como função, entre outras, ocultar as particularidades dos discos e de outros dispositivos de E/S, com a finalidade de fornecer ao programador um modelo de arquivos agradável e independente de dispositivos.

18. (Quadrix/CREFITO-MG/2021) Os utilitários (programas de sistema), uma modalidade específica de programa, são sempre executados dentro do kernel do sistema operacional.

GABARITO

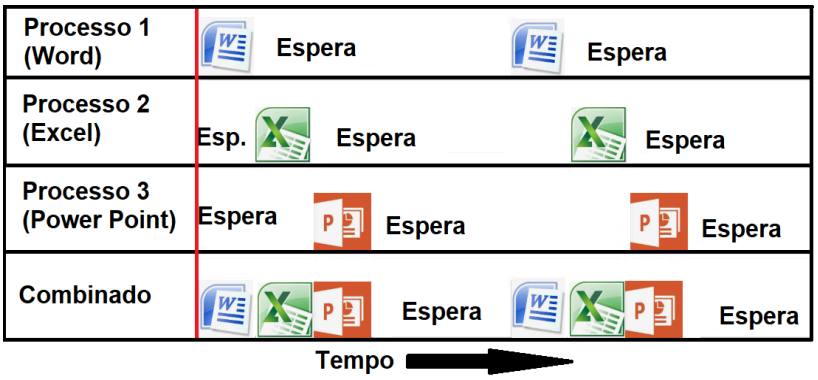
GABARITO

1. Certo
2. Letra C
3. Anulada
4. Letra A
5. Errado
6. Certo
7. Letra C
8. Letra B
9. Letra B
10. Letra C
11. Letra B
12. Letra B
13. Certo
14. Certo
15. Certo
16. Errado
17. Certo
18. Errado

Gerenciamento de Processos

Um sistema operacional moderno permite que diversas atividades sejam realizadas ao mesmo tempo, mesmo que a máquina possua apenas um processador! Mas como é possível fazer com que 2 ou mais programas sejam executados ao mesmo tempo com apenas 1 processador? Aí lembramos da ideia de pseudoparalelismo que existe com o escalonamento de uso do processador, sendo que cada processo recebe uma fatia de tempo, de acordo com alguma política ou algoritmo.

Mas o que é processo? É simplesmente uma instância de um programa em execução, incluindo os valores correntes dos registradores (PC, IR, entre outros) e das variáveis (ex.: soma, total, em um programa que realiza cálculos). Cada processo pensa que está "sozinho no mundo" e executa em um processador (CPU) virtual, mas sabemos que na prática o processador alterna de um processo para outro. Essa possibilidade de alternância entre processos é conhecida como multiprogramação ou multitarefa. Veja a figura abaixo, onde é mostrada a execução de 3 processos em um único processador:



Na figura vemos apenas um processo de cada programa, mas pode haver N processos do Word, do Excel, do Power Point, ou qualquer outro. O limite se resume a recursos (memória, limitação do sistema operacional, entre outros).

Quando o processador muda o processo, ocorre uma troca de contexto (chaveamento ou mudança de contexto). Trata-se de um processo computacional de armazenar e restaurar o estado (contexto) de um processador de forma que múltiplos processos possam compartilhar esse processador (multiprogramação). É garantido que quando o contexto anterior armazenado for restaurado, o ponto de execução volte ao mesmo estado que foi deixado durante o armazenamento.

Imagine que você tenha carregado um documento do Word, em seguida uma planilha do Excel e uma apresentação do Power Point, todos os arquivos muito grandes. Suponha que tenham sido carregadas 50 páginas do Word, 50 do Excel, 100 slides do Power Point, depois o processador volta para o processo do Word e carrega as restantes, e assim por diante. Geralmente tudo ocorre muito rápido e temos a impressão de um paralelismo de verdade, a não ser que seu processador seja muito antigo e você queira carregar dezenas ou centenas de processos que exijam um certo desempenho.

Criação, Término e Hierarquia de Processos

Existem quatro eventos principais que acarretam a criação de processos, a saber:

- Inicialização do sistema;
- Realização de uma chamada de sistema por um processo em execução para a criação de um processo, ex.: fork() no Linux;
- Um pedido de um usuário para a criação de um novo processo, ex.: clicar duas vezes para abrir um documento do Word;
- Início de uma tarefa em lote (computadores de grande porte).

Quando o S.O. é inicializado, geralmente vários processos são criados, alguns em primeiro plano (*foreground*), que interagem com os usuários; e outros em segundo plano (*background*), que não estão associados a algum usuário, mas possuem alguma função específica (ex.: processo que aguarda a solicitação por impressão). Processos em *background* também são chamados de *daemons*, mas um conceito mais aprofundado de *daemon* seria:

> "Um tipo especial de processo executado em segundo plano, geralmente sem interação direta com o usuário. O termo *daemon* tem origem do sistema operacional Unix e representa processos que são iniciados durante o boot do sistema e continuam a ser executados enquanto o sistema está ativo. Alguns exemplos são o *daemon* do sistema de impressão (*cupsd*) e o *daemon* de agendamento de tarefas (*cron*)."

"Dentro" de um processo, além do código do programa em si, podemos encontrar:

- Contexto de software: informações como nome do processo, identificador (PID), proprietário (*owner* - UID), prioridade de execução, entre outros;
- Contexto de hardware: valores de registradores;
- Espaço de endereçamento: espaço reservado para os dados do processo (ex.: texto editado através do Word).

Na figura abaixo podemos ver um esquema de um processo:

Existe uma estrutura de dados no núcleo do sistema operacional que serve para armazenar a informação necessária para tratar um determinado processo. Trata-se do Bloco de Controle do Processo (BCP, ou PCB - *Proccess Control Block*). Como o PCB possui informações críticas do processo ele deve ficar armazenado em uma área da memória protegida do acesso de usuários. Geralmente as informações contidas em um PCB incluem:

- Identificador do processo (PID);
- Registradores da CPU;
- O espaço de endereçamento do processo;
- Prioridade do processo;
- Entre outras.

Tabela de Processo: estrutura de dados responsável por habilitar o sistema operacional a localizar e acessar rapidamente o bloco de controle de processo (PCB) de um processo.

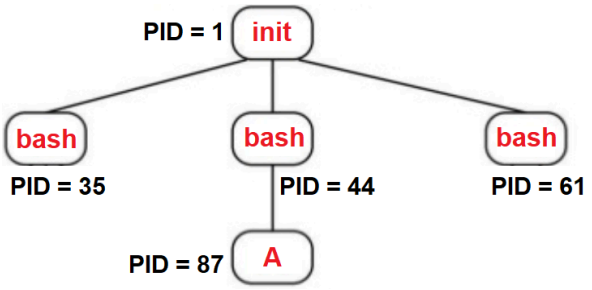
Nada dura para sempre, em algum momento ocorre o término do processo, normalmente devido a alguma das seguintes situações:

- Término normal (voluntário);
- Término por erro (voluntário), ex.: divisão por zero;
- Erro fatal (involuntário), ex.: programa recebe como parâmetro o nome de um arquivo que não existe;
- Eliminado por outro processo (involuntário), ex.: comando kill (Linux).

A maioria dos processos termina porque já cumpriu sua tarefa. Se você executa um programa que recebe 10 valores, realiza uma bateria de cálculos, mostra um resultado e finaliza, neste momento ele envia uma chamada de sistema para avisar ao S.O. que terminou (ex.: exit). Os programas aceitam término voluntário, geralmente com algum menu ou combinação de teclas (ex.: ALT + F4, no Windows).

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* (*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):

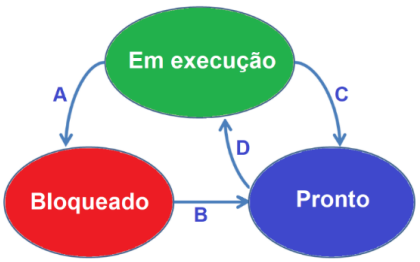


Estados de um Processo

De uma forma resumida, um processo pode estar em um dos seguintes estados:

- Executando: realmente utilizando o processador (em execução);
- Pronto: temporariamente parado, mas pronto (aguardando em "uma fila") para utilizar o processador;
- Bloqueado (ou em espera): incapaz de executar até que algum evento ocorra (ex.: terminar de receber os dados de um arquivo que está sendo lido de um HD).

As quatro trocas possíveis entre os estados são apresentadas na figura abaixo:



Quando um processo está em execução (no processador) e é necessário realizar alguma atividade que não dependa do processador (ex.: aguardar os dados do arquivo "teste.txt" solicitados ao HD), não tem o porquê ele ainda utilizar o processador! Seria um desperdício! Então o processo é bloqueado (A) e um outro processo que estiver pronto (e for a sua vez) deverá ocupar a CPU (D). Depois que o HD realizar a leitura do arquivo e colocar os dados em um buffer, é enviado um aviso para dizer que está tudo pronto, bastando buscar no buffer. Nesse momento o processo deixa de estar bloqueado e fica pronto (entra na fila) para utilizar a CPU novamente (B).

Os processos que estão prontos ficam no aguardo de sua vez para utilizar a CPU (D) e os que estão em execução, não solicitam nenhuma atividade que enseje um bloqueio, mas que utilizam a CPU até um determinado limite de tempo, perdem o uso da CPU e voltam para a "fila" dos prontos (C) . Dessa forma os processos ficam mudando seus estados até finalizarem.

Dependendo do comportamento do processo, ele ficará mais tempo bloqueado ou pronto. Processos que passam a maior parte computando (CPU-bound) tendem a ir poucas vezes para o

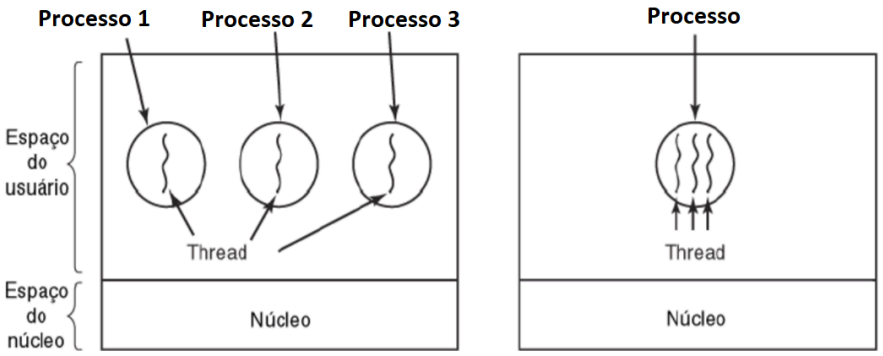
estado bloqueado (ou talvez nunca!), pois utilizam muito o processador e quando a fatia de tempo termina vão para o estado "pronto". Já os processos que esperam muito por E/S (I/O-bound) tendem a ficarem bloqueados seguidamente, indo depois para o estado "pronto" e somente depois poderem utilizar a CPU. Um exemplo de um processo *CPU-bound* é a execução de um filme (fica mais de hora rodando). Um exemplo de processo *I/O-bound* é um chat, pois aguarda a digitação do teclado a todo momento.

Threads

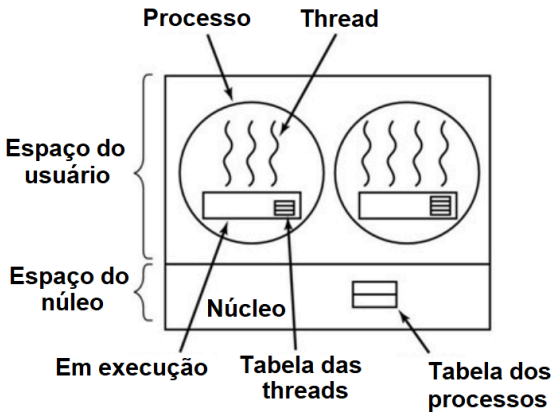
Um processo "tradicional" possui um espaço de endereçamento e um fluxo de controle (execução do código). Porém, há situações em que se deseja ter mais de um fluxo de controle e execução no mesmo processo, executando quase em paralelo. Esses fluxos são chamados *threads* (ou processos leves). Resumindo: *threads* de um mesmo processo compartilham a mesma seção de código na memória. Porém, cada *thread* possui os seus valores nos registradores e na pilha, ou seja, a cada troca de contexto entre as *threads*, esses valores são atualizados.

Imagine um editor de texto, que possui inúmeras funcionalidades: contador de palavras, contador de páginas, correção ortográfica instantânea, entre outras. Cada uma delas geralmente é implementada em uma *thread*, então a cada digitação elas verificam se a quantidade de palavras aumentou/diminuiu (e atualiza essa informação na tela), se a quantidade de páginas foi alterada, se a palavra digitada está correta (após consultar um arquivo de dicionário), e assim por diante.

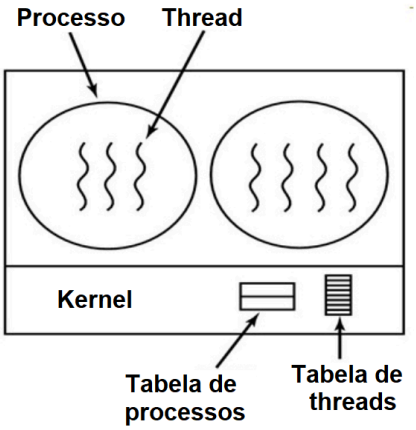
Se não fosse assim, vários processos deveriam ser carregados (o que seria mais "pesado") ou a execução de todo o código na sequência para verificar todas as funcionalidades. Abaixo uma figura para mostrar 3 processos contendo uma única *thread* (esquerda) e um único processo com 3 *threads* (direita).



Geralmente as *threads* são divididas em duas categorias: thread ao nível do usuário e thread ao nível do *kernel* (núcleo). Quando falamos em *thread* ao nível de usuário queremos dizer que o controle das *threads*, o escalonamento entre elas, fica a cargo do espaço do usuário. Na figura a seguir podemos ver que a tabelas dos processos fica no *kernel*, mas as tabelas das *threads* de cada processo ficam nos próprios processos.



Quando falamos em *thread* ao nível do *kernel* queremos dizer que o controle das *threads* fica a cargo do *kernel*, ou seja, tanto a tabela de processos quanto a tabela de *threads* ficam no *kernel*:



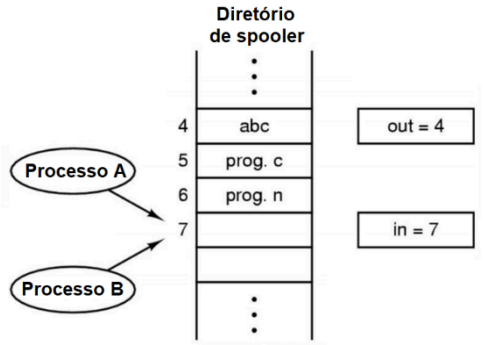
Comunicação entre Processos

Os processos frequentemente necessitam se comunicar. Um exemplo simples de entender é quando um usuário utiliza uma sequência de comandos no *shell* com o *pipe* separando cada um deles. Isso significa que a saída de um é a entrada do seguinte. Ex.:

$ ls | grep x | sort -r | tee arquivo_saida

No exemplo acima, o resultado do "ls" é a entrada para o comando "grep x". O resultado é a entrada para "sort -r" e o resultado deste último é a entrada para "tee arquivo_saida".

Condições de corrida: alguns processos que trabalham juntos podem compartilhar algum armazenamento comum, ou seja, cada processo pode ler e escrever nesse local. Esse armazenamento compartilhado pode estar na memória principal ou pode ser um arquivo. Vamos utilizar como exemplo o diretório de *spooler*, onde os arquivos são colocados para que o *daemon* de impressora verifique de tempos em tempos o que deve ser impresso e mover a fila. Imagine a situação em que dois processos (A e B) quase que simultaneamente decidem enviar para o diretório de *spooler* o arquivo a ser impresso:



Seções críticas: para evitar as condições de corrida ou outras situações que envolvem memória compartilhada ou arquivo compartilhado, deve-se encontrar uma maneira de proibir que mais de um processo leia e modifique dados compartilhados ao mesmo tempo. Ou seja, é necessária uma exclusão mútua. O problema mostrado na figura anterior ocorreu porque o processo B começou a utilizar uma das variáveis compartilhadas antes que o processo A tivesse terminado de trabalhar com ela.

Nem sempre o sistema acessa alguma região compartilhada, mas quando isso ocorre ele está acessando uma região crítica ou seção crítica. São necessárias quatro condições válidas para termos uma boa solução para esse problema:

1. Dois processos não podem estar ao mesmo tempo dentro de uma seção crítica;
2. Nenhuma suposição pode ser realizada sobre as velocidades ou sobre o número de processadores;
3. Nenhum processo que está sendo executado fora de uma seção crítica pode bloquear outros processos;
4. Nenhum processo deve ter que esperar eternamente para entrar em uma seção crítica.

Semáforo: é um tipo de variável que pode ser verificada e alterada em instruções atômicas, ou seja, sem possibilidades de interrupções. Esse tipo de variável é empregado em tarefas como o compartilhamento de recursos entre processos. É uma variável do tipo inteiro que possui o valor 0 quando não tem nenhum sinal a despertar, ou um valor positivo quando um ou mais sinais para despertar estiverem pendentes (usamos o termo "despertar", pois um fluxo de execução é "colocado para dormir" quando tenta entrar em uma região crítica que já está ocupada).

Existem as operações down e up (ou sleep e wakeup). A operação *down* verifica se o valor é maior que 0. Se for, ele decrementa um e continua. Se o valor for 0, o processo (ou a *thread*) é "colocado para dormir" (bloqueado) sem completar a operação *down*. É garantido que iniciada uma operação de semáforo, nenhum outro processo possa acessar o semáforo até que a operação tenha terminado ou sido bloqueada (ação atômica). Isso evita as condições de corrida.

Vamos imaginar a seguinte situação: há um semáforo que está com o valor 0, ou seja, já tem processo(s) ocupando a região crítica. Enquanto isso, outros 3 chegam a essa região e "tentam entrar", ficando bloqueados. Ao sair da região crítica, é aplicada a operação *up* (valor do semáforo passa para 1) e um dos três processos pode entrar (aleatório, fila ou outra maneira de escolher). Quando tal processo entrar é aplicada a operação *down* (semáforo passa a ser 0). Resumindo: o valor do semáforo diz quantos processos podem entrar!

Mutex: versão simplificada do semáforo, quando não for necessário "contar", utilizando-se apenas os estados "livre" ou "ocupado". Consequentemente é necessário apenas um bit para representá-lo, mas na prática geralmente é utilizado um valor inteiro. Quando um processo precisa entrar na região crítica, ele chama *mutex_lock*, que será bem-sucedida se a região estiver livre. Caso contrário, o processo ficará bloqueado até que o processo que estiver na região crítica saia (*mutex_unlock*).

Escalonamento de Processos

O escalonamento de processos é uma atividade de processamento realizada pela CPU de um computador. Essa atividade permite executar de forma mais eficiente os processos considerados prioritários para o sistema operacional. Existem situações em que pode ocorrer o escalonamento, mas existem duas em que ele DEVE ocorrer:

- Quando um processo termina: não tem motivo para ocupar tempo de processamento após concluída sua execução. Ex.: o processo recebe a CPU por 5ms, mas conclui o seu uso em 3ms;

- Quando um processo é bloqueado em uma operação de E/S ou em um semáforo: não tem motivo para ficar aguardando "dentro" da CPU por um retorno do dispositivo ou a saída do semáforo.

Escalonador (*scheduler*): rotina responsável por determinar a ordem de execução dos processos. Ele gerencia a alocação de recursos da CPU entre os vários processos concorrentes em um sistema.

Throughput ("Taxa de Transferência"): critério de escalonamento que representa o número de processos executados em um determinado intervalo de tempo. O *throughput* refere-se à quantidade de trabalho realizado em um sistema durante um período específico. Essa medida é frequentemente expressa em termos de processos ou tarefas completadas por unidade de tempo. Quanto maior o número de processos que podem ser concluídos em um intervalo de tempo, maior é o *throughput*.

Preempção: é o ato do S.O. utilizar as interrupções do relógio para retirar a CPU do processo em execução. Ou seja, o processo não pode monopolizar o processador! Quando o sistema não for preemptivo pode ocorrer uma situação denominada *starvation* (inanição). Traduzindo, *starvation* quer dizer "morrer de fome", ou seja, aquele processo que nunca consegue chegar ao processador, fica eternamente aguardando, sempre tem alguém que "fura a fila". Abaixo podemos ver o conceito de *starvation*.

Starvation (inanição, ou privação): situação na qual um processo ou *thread* é incapaz de prosseguir com a execução devido à não recepção dos recursos necessários para avançar. Isso pode ocorrer em ambientes multitarefa quando um processo fica impedido de obter acesso a recursos cruciais, como processador, memória, ou dispositivos de entrada e saída.

Sistema multitarefa preemptivo: um sistema que possibilita a execução de mais de um processo ao mesmo tempo geralmente é preemptivo, ou seja, em algum momento o S.O. retira o processador do processo e coloca outro no lugar.

Tempo de *turnaround*: tempo de existência de um processo, ou seja, o tempo desde a sua criação até seu término. As políticas de escalonamento buscam minimizar o tempo de turnaround.

As categorias de algoritmos de escalonamento são:

- Lote: geralmente utilizado em computadores de grande porte, sem usuários esperando uma resposta rápida. São aceitáveis algoritmos não-preemptivos ou algoritmos preemptivos com longos períodos de tempo para cada processo. Isso reduz as trocas de processo, o que melhora o desempenho;
- Interativo: em um ambiente em que os usuários interagem, a preempção é fundamental! Assim não ocorre uma monopolização da CPU por um processo, negando o processamento a outros. Uso típico em computadores de propósito geral, ex.: PC para jogos, edição de textos, navegação na Internet etc.;

- Tempo real: por incrível que pareça a preempção pode ser desnecessária, pois os processos sabem que não podem ser executados por longos períodos de tempo. Normalmente os processos realizam suas atividades e são rapidamente bloqueados. Um sistema de tempo real executa apenas o que é necessário, ex.: radar que registra a velocidade do veículo e fotografa se ultrapassar um determinado limite.

Escalonamento em Sistemas de Lote

Alguns algoritmos são convenientes tanto para sistemas de lote como nos interativos, mas vamos estudar agora aqueles mais utilizados apenas para sistemas de lote.

First-Come First-Served (FCFS): o nome do algoritmo já diz tudo ("O primeiro a chegar é o primeiro a ser atendido"), ou seja, os processos recebem tempo de CPU na ordem em que solicitam. Basicamente há uma fila única e os processos vão "entrando" nela, os processos prontos são executados na ordem da fila e quando um processo é bloqueado (aguardando uma entrada, por exemplo), ele retorna para o fim da fila quando estiver pronto novamente. Ok, mas se um processo começar a ser executado, nunca for bloqueado e tiver uma estimativa de execução de 4h, ele monopolizará o processador esse tempo todo? Isso mesmo! Esse algoritmo é do tipo não-preemptivo, uma característica de sistemas de lote.

Shortest-Job First (SJF) ("Tarefa mais curta primeiro"): algoritmo não-preemptivo que presume que os tempos de execução são conhecidos previamente. Imagine uma situação em que os *jobs* (tarefas) de uma empresa são executados há muitos anos e já se sabe que os *jobs* do tipo A levam 2 minutos, do tipo B 6 minutos e do tipo C 4 minutos. Os *jobs* são agendados à tarde para serem executados às 8h do dia seguinte. Vamos ver como ficaria o escalonamento para a execução no dia seguinte dos *jobs* que foram adicionados na ordem B1, A1, C1, A2 e C2:

Shortest Remaining Time Next (SRT) ("Menor tempo de execução restante"): versão preemptiva do algoritmo SJF. Nesse algoritmo, o escalonador sempre escolhe o *job* com tempo de execução mais curto (obviamente o tempo precisa ser conhecido previamente). Quando um novo *job* chega, seu tempo é comparado com o que falta para concluir o *job* corrente. Se o novo precisar de menos tempo, o processo corrente é suspenso e o novo é iniciado. Esse esquema permite que *jobs* novos curtos tenham prioridade.

Escalonamento em Sistemas Interativos

Round-robin: trata-se do algoritmo mais conhecido e cobrado! Com ele é realizado um rodízio entre os processos, sendo que a cada processo é atribuído um intervalo de tempo (quantum), durante o qual ele pode ser executado. Se ao final do *quantum* o processo ainda estiver em execução é realizada a preempção da CPU e esta é alocada a um outro processo. Obviamente que se o processo tiver terminado antes do fim do *quantum* ou se tiver sido bloqueado, a troca da CPU é realizada neste momento.

Um ponto interessante é a definição da duração do *quantum*, pois trocar de um processo para outro exige uma quantidade de tempo para salvar e carregar registradores e mapas de memória, atualizar tabelas e listas etc. Vamos supor que esse chaveamento de contexto demore 1ms e que o *quantum* seja de 9ms. Nesse caso, 10% da CPU seriam desperdiçados em sobrecarga administrativa.

Escalonamento por prioridade: cada processo recebe uma prioridade e o processo na "fila de pronto" com a maior prioridade tem a permissão para executar. Por exemplo, um *daemon* que envia um e-mail em segundo plano deveria ter uma prioridade mais baixa do que a de um processo responsável por uma videoconferência ao vivo.

Para evitar que processos com alta prioridade monopolizem o uso da CPU, o escalonador pode diminuir a prioridade do processo em execução em cada interrupção de relógio. Se sua prioridade ficar abaixo da do próximo processo com maior prioridade, deve ocorrer um chaveamento de processo. Uma alternativa é dar um *quantum* a cada processo e, quando esse *quantum* esgotar, é dada a chance de execução do próximo processo com maior prioridade.

Escalonamento por múltiplas filas: basicamente os processos são agrupados em classes e cada classe possui uma prioridade: 1 *quantum*, 2 *quanta* (esse é o plural de *quantum*), 4, 8, e assim por diante. Na medida em que um processo utiliza todos os *quanta* destinados a ele, em seguida ele é movido para a classe seguinte (a que recebe mais quanta). Por exemplo, se um processo precisa de 50 *quanta*, primeiro ele recebe 1, depois 2, 4, 8, 16, 32. No total o processo teria recebido 63 *quanta*, ou seja, na última classe em que esteve (recebendo 32 *quanta*), só utilizaria 19 e interromperia a execução, liberando o processador.

Escalonamento garantido: a ideia é "fazer promessas realistas aos usuários sobre o desempenho, e cumpri-las!". Como assim? Se houver N usuários trabalhando em uma CPU, cada um recebe cerca de 1/N do poder dessa CPU! Essa é uma promessa realista simples de cumprir. Para que essa promessa realmente seja cumprida, o sistema deve monitorar quanto da CPU cada processo de cada usuário recebeu e dar mais poder de CPU para quem teve menos. Assim ocorre uma compensação e aos poucos todos usuários terão um uso da CPU similar.

Escalonamento por sorteio: o escalonamento garantido parece uma boa, mas é difícil implementar! Uma solução mais simples é o algoritmo de escalonamento por sorteio, o qual fornece "bilhetes de loteria" para os vários recursos do sistema (ex.: tempo de CPU). Quando uma decisão de escalonamento tiver que ser tomada, um "bilhete" é sorteado e o ganhador recebe o recurso. Os processos mais importantes podem receber "bilhetes" extras, aumentando as chances de serem sorteados.

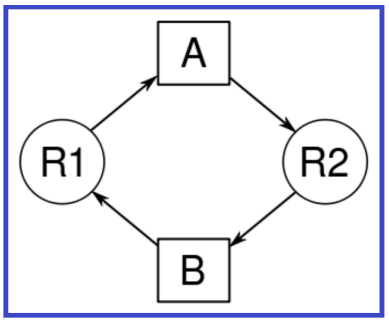
Deadlock (Impasse)

Um *deadlock* (impasse) ocorre quando um conjunto de processos está esperando por um evento que só pode ser causado por outro processo do conjunto. Ou seja, ficam todos "amarrados" em um "abraço da morte", sem poder continuar seus processamentos. Alguns exemplos de recursos que só podem ser utilizados por um processo por vez: unidades de fita e impressoras. Ou você acha que é possível dois processos escrevendo em uma fita ou mandando imprimir ao mesmo tempo? Ficaria uma bagunça!

Por isso os sistemas operacionais possuem a capacidade de garantir (por algum tempo) que um processo tenha o acesso exclusivo a determinados recursos, sejam de hardware ou de software. Vamos supor a seguinte situação: os processos A e B desejam digitalizar uma fotografia através de um *scanner* e em seguida gravar em um CD-R. Digamos que A tenha requisitado primeiro o *scanner* e ao mesmo tempo B tenha requisitado o gravador de CD-R. Em seguida A faz requisição do gravador de CD-R, mas ele está ocupado, pois B não o "largou". Assim fica o processo A aguardando o recurso gravador de CD-R e B aguardando o recurso *scanner*. Assim temos um *deadlock*!

As quatro condições (de Coffman) que devem ser verdadeiras para que ocorra um *deadlock* são:

1. Condição de exclusão mútua: cada recurso ou está correntemente atribuído a exatamente um processo ou está disponível;
2. Condição de posse e espera: os processos que possuem recursos garantidos anteriormente podem solicitar novos recursos (um acumulador de recursos!);
3. Ausência de preempção: os recursos garantidos não podem ser retirados à força de um processo;
4. Condição de espera circular: um encadeamento circular de dois ou mais processos, cada um esperando por um recurso mantido pelo próximo do encadeamento:



Algoritmo do avestruz: estratégia mais simples! O que um avestruz faz? Coloca a cabeça em um buraco... pois é, é isso mesmo! O algoritmo finge que nada aconteceu, simplesmente ignora. E, por incrível que pareça, é a estratégia adotada pela maioria dos sistemas operacionais (Windows,

Linux, entre outros). Por quê? Porque o preço seria alto em realizar muitas restrições ao usuário. Melhor deixar acontecer...e se for o caso, o usuário reinicia a máquina ou mata um processo, caso ocorra algum travamento.

Algoritmo do banqueiro (Banker's algorithm): utilizado para evitar deadlock em sistemas operacionais. Ele foi proposto por Edsger Dijkstra e é utilizado em sistemas que têm um número fixo de recursos e processos concorrentes que solicitam e liberam esses recursos.

Detecção e recuperação: o sistema monitora as requisições e liberações de recursos. Sempre que um recurso é solicitado ou liberado, o grafo de recursos é atualizado e uma verificação para saber se há ciclos é realizada. Se houver um ciclo, um dos processos do ciclo é eliminado. Se não resolver o *deadlock*, outro será eliminado e assim por diante.

Prevenção de impasses: estratégia que impõe restrições sobre os processos a fim de tornar os impasses impossíveis. As quatro condições que já vimos dão um indício de possíveis soluções.

Evitação de impasses: não há a imposição de regras, mas sim uma análise criteriosa de cada pedido de recurso. Para isso, algumas informações devem estar disponíveis antecipadamente.

Questões Comentadas - CEBRASPE

1. (CEBRASPE/ABIN/2010) No contexto de sistemas operacionais, semáforos são tipos de variáveis que podem ser verificadas e alteradas em instruções atômicas, ou seja, sem possibilidades de interrupções. Esse tipo de variável é empregado em tarefas como o compartilhamento de recursos entre processos.

Comentários:

Como vimos há pouco, um semáforo possui as operações atômicas up e down. Esse tipo de variável (semáforo) é ideal para controlar o acesso a regiões críticas, evitando que mais processos (ou threads) que determinado limite consigam acessar tal região ao mesmo tempo. Isso é o ideal no compartilhamento de recursos, evitando problemas como, por exemplo, o compartilhamento de uma impressora (imagine cada processo imprimindo uma página de seu documento ao mesmo tempo!).

Gabarito: Certo

2. (CEBRASPE/STF/2013) Em um algoritmo de escalonamento FIFO, os processos são executados na mesma ordem que chegam à fila. Quando um processo do tipo cpu-bound está na frente da fila, todos os processos devem esperá-lo terminar seu ciclo de processador.

Comentários:

Um processo CPU-bound é aquele que usa muito o processador, ou seja, raramente é bloqueado. Por isso os demais devem esperar acabar seu ciclo de uso do processador (quantum).

Gabarito: Certo

3. (CEBRASPE/TRE-PI/2016) A respeito das características do algoritmo de escalonamento SPF (shortest process first), assinale a opção correta.

A) Os processos são executados na ordem em que chegam à fila de espera e executados até o final, sem nenhum evento preemptivo.

B) No SPF, um processo recém-chegado e em espera, cujo tempo estimado de execução completa seja menor, provoca a preempção de um processo em execução que apresente tempo estimado de execução completa maior.

C) O SPF favorece processos longos em detrimento dos mais curtos. Estes, ao chegarem à fila de espera, são obrigados a aguardar a conclusão dos processos longos que já estiverem em andamento, para, então, entrar em execução.

D) Os processos são despachados na ordem em que são colocados em espera e recebem uma quantidade limitada de tempo do processador para execução; além disso, são interrompidos caso sua execução não se conclua dentro do intervalo de tempo delimitado.

E) O escalonador seleciona o processo que estiver à espera e possuir o menor tempo de execução estimado e o coloca em execução até a sua conclusão.

Comentários:

O nome mudou um pouquinho, mas a ideia é a mesma:

Shortest-Job First (SJF) ("Tarefa mais curta primeiro"): algoritmo não-preemptivo, presume que os tempos de execução sejam conhecidos previamente. Imagine uma situação em que os jobs (tarefas) de uma empresa são executados há muitos anos e já se sabe que os jobs do tipo A levam 2 minutos, do tipo B 6 minutos e do tipo C 4 minutos. Os jobs são agendados à tarde para serem executados às 8h do dia seguinte. Vamos ver como ficaria o escalonamento para a execução no dia seguinte dos jobs (tarefas) que foram adicionados na ordem B1, A1, C1, A2 e C2:

Gabarito: E

4. (CEBRASPE/TRE-TO/2017) Considerando o contexto de gerenciamento de processos dos sistemas operacionais, assinale a opção que apresenta a estrutura de dados responsável por habilitar o sistema operacional a localizar e acessar rapidamente o bloco de controle de processo (PCB) de um processo.

A) árvore de processos.

B) lista de bloqueados.

C) tabela de processo.

D) região de pilha.

E) lista de prontos.

Comentários:

Existe uma estrutura de dados no núcleo do sistema operacional que serve para armazenar a informação necessária para tratar um determinado processo. Trata-se do Bloco de Controle do Processo (PCB - Proccess Control Block). Como o PCB possui informações críticas do processo ele deve ficar armazenado em uma área da memória protegida do acesso de usuários. Geralmente as informações contidas em um PCB incluem:

- Identificador do processo (PID);
- Registradores da CPU;
- O espaço de endereçamento do processo;
- Prioridade do processo;
- Entre outras.

E a estrutura de dados responsável por habilitar o sistema operacional a localizar e acessar rapidamente o bloco de controle de processo (PCB) de um processo é denominada Tabela de Processo.

Gabarito: C

5. (CEBRASPE/TRF1/2017) Na técnica denominada escalonamento de processos, o sistema operacional mantém parte do espaço de endereçamento de um processo na memória principal e parte em dispositivo de armazenamento secundário, realizando trocas de trechos de código e de dados entre eles, de acordo com a necessidade.

Comentários:

Quando falamos em escalonamento de processos, estamos lidando com processos em execução, ou seja, estão na memória RAM. Claro que parte dele pode estar na memória virtual, em disco. Então podemos ver que a questão misturou os conceitos.

Gabarito: Errado

6. (CEBRASPE/STJ/2018) Em relação aos fundamentos de sistema operacional, julgue o item a seguir.

Um processo existente no sistema operacional pode ter um número zero de processos-pai.

Comentários:

O único processo que pode ter número zero de processos pai é o init, do Linux, ou um equivalente, em outro sistema. Como ele é o processo "inicial", ele é pai, mas não possui pai!

Gabarito: Certo

7. (CEBRASPE/MPE-PI/2018) Julgue o item a seguir, acerca de sistemas operacionais.

Uma das causas de deadlocks em sistemas operacionais é a disputa por recursos do sistema que podem ser usados apenas por um processo de cada vez.

Comentários:

Essa é a primeira das quatro condições, a condição de exclusão mútua, a qual define que cada recurso ou está correntemente atribuído a exatamente um processo ou está disponível.

Gabarito: Certo

8. (CEBRASPE/SLU-DF/2019) Em relação aos microcomputadores, julgue o item a seguir.

Uma das características dos sistemas preemptivos é o fato de eles serem monotarefa.

Comentários:

Um sistema preemptivo é aquele que pode tirar o recurso do processo, pode tirar por exemplo, o processador após ter passado o *quantum*. Se isso acontece, é porque outro processo vai utilizar o processador, ou seja, trata-se de um sistema multitarefa!

Gabarito: Errado

9. (CEBRASPE/Min. da Economia/2020) Julgue o próximo item, relativos a sistemas operacionais.

No sistema operacional, o bloco de controle de processo (BCP) representa e guarda informações associadas a um processo, como, por exemplo, o seu estado pronto ou em execução.

Comentários:

Existe uma estrutura de dados no núcleo do sistema operacional que serve para armazenar a informação necessária para tratar um determinado processo. Trata-se do Bloco de Controle do Processo (BCP, ou PCB - *Proccess Control Block*). Como o PCB possui informações críticas do processo ele deve ficar armazenado em uma área da memória protegida do acesso de usuários. Geralmente as informações contidas em um PCB incluem:

- Identificador do processo (PID);
- Registradores da CPU;
- O espaço de endereçamento do processo;
- Prioridade do processo;
- Entre outras.

Gabarito: Certo

10. (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.

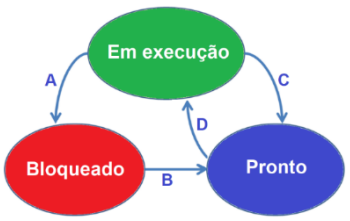
C) executável.

D) parado.

E) zumbi.

Comentários:

Para quem já estudou a aula de Linux fica mais fácil, mas digamos que você não tenha estudado e se depara com uma questão assim. E aí? Vamos ver a figura novamente e tentar por eliminação:



(A) Pronto - podemos ver que pode executar;

(B) Dormente - específico do Linux, não vimos, vamos ver a próxima;

(C) Executável - parece ser o equivalente ao pronto, pois é "executável", pode ser executado;

(D) Parado - parece o equivalente ao "bloqueado", pois está parado esperando algo (ex.: leitura de HD);

(E) Zumbi - específico do Linux, não vimos, vamos ver a próxima.

Com essa análise, mesmo sem ter estudado Linux, daria para marcar a alternativa D, não é?

Gabarito: D

11. (CEBRASPE/SEFAZ-CE/2021) Julgue o próximo item, acerca de conhecimentos de informática.

O gerenciamento de processos é responsável por executar simultaneamente vários processos nos sistemas operacionais Windows e Linux, por meio do compartilhamento de tempo e processadores.

Comentários:

O gerenciamento de processos implementa algum algoritmo ou política, sendo possível executar simultaneamente vários processos (multitarefa) através do uso de *quantum* (fatia de tempo para o uso do processador). O algoritmo mais usado é o *round-robin*.

Gabarito: Certo

12. (CEBRASPE/SEFIN de Fortaleza-CE/2023) Acerca de primitivas de sincronização e deadlocks, julgue o próximo item.

No ambiente de programação, um semáforo é um sinal emitido pelo hardware para que o sistema operacional possa executar um laço com o objetivo de evitar um deadlock.

Comentários:

Semáforo é um tipo de variável que pode ser verificada e alterada em instruções atômicas, ou seja, sem possibilidades de interrupções. Esse tipo de variável é empregado em tarefas como o compartilhamento de recursos entre processos. É uma variável do tipo inteiro que possui o valor 0 quando não tem nenhum sinal a despertar, ou um valor positivo quando um ou mais sinais para despertar estiverem pendentes (usamos o termo "despertar", pois um fluxo de execução é "colocado para dormir" quando tenta entrar em uma região crítica que já está ocupada).

Existem as operações down e up (ou sleep e wakeup). A operação *down* verifica se o valor é maior que 0. Se for, ele decrementa um e continua. Se o valor for 0, o processo (ou a *thread*) é "colocado para dormir" (bloqueado) sem completar a operação *down*. É garantido que iniciada uma operação de semáforo, nenhum outro processo possa acessar o semáforo até que a operação tenha terminado ou sido bloqueada (ação atômica). Isso evita as condições de corrida.

Vamos imaginar a seguinte situação: há um semáforo que está com o valor 0, ou seja, já tem processo(s) ocupando a região crítica. Enquanto isso, outros 3 chegam a essa região e "tentam entrar", ficando bloqueados. Ao sair da região crítica, é aplicada a operação *up* (valor do semáforo passa para 1) e um dos três processos pode entrar (aleatório, fila ou outra maneira de escolher). Quando tal processo entrar é aplicada a operação *down* (semáforo passa a ser 0). Resumindo: o valor do semáforo diz quantos processos podem entrar!

Gabarito: Errado

QUESTÕES COMENTADAS - FGV

1. (FGV/TJ-RO/2015) Considere o trecho a seguir no contexto de sistemas operacionais.

"Um conjunto de processos está num bloqueio perpétuo quando cada processo do conjunto está esperando por um evento que apenas outro processo do conjunto pode causar."

A situação descrita é típica da ocorrência de um:

A) timeout;

B) deadlock;

C) timestamp;

D) system halt;

E) pipeline.

Comentários:

Um *deadlock* (impasse) ocorre quando um conjunto de processos está esperando por um evento que só pode ser causado por outro processo do conjunto. Ou seja, ficam todos "amarrados" em um "abraço da morte", sem poder continuar seus processamentos. Alguns exemplos de recursos que só podem ser utilizados por um processo por vez: unidades de fita e impressoras. Ou você acha que é possível dois processos escrevendo em uma fita ou mandando imprimir ao mesmo tempo? Ficaria uma bagunça!

Por isso os sistemas operacionais possuem a capacidade de garantir (por algum tempo) que um processo tenha o acesso exclusivo a determinados recursos, sejam de hardware ou de software. Vamos supor a seguinte situação: os processos A e B desejam digitalizar uma fotografia através de um *scanner* e em seguida gravar em um CD-R. Digamos que A tenha requisitado primeiro o *scanner* e ao mesmo tempo B tenha requisitado o gravador de CD-R. Em seguida A faz requisição do gravador de CD-R, mas ele está ocupado, pois B não o "largou". Assim fica o processo A aguardando o recurso gravador de CD-R e B aguardando o recurso *scanner*. Assim temos um *deadlock*!

Gabarito: B

2. (FGV/IBGE/2016) Jonas, Analista de Suporte Operacional do IBGE, realizou uma análise minuciosa dos processos e threads do servidor que ele mantém. Durante a análise, Jonas identificou que três processos estavam na lista de espera por um recurso compartilhado. Além disso, Jonas também identificou uma situação inusitada: um desses processos nunca conseguia executar sua região crítica e, por conta disso, nunca acessava o recurso compartilhado.

A situação inusitada encontrada por Jonas é a de:

A) lock;

B) starvation;

C) sincronização condicional;

D) threads;

E) stack

Comentários:

Preempção: é o ato do S.O. utilizar as interrupções do relógio para retirar a CPU do processo em execução. Ou seja, o processo não pode monopolizar o processador! Quando o sistema não for preemptivo pode ocorrer uma situação denominada *starvation* (inanição). Traduzindo, *starvation* quer dizer "morrer de fome", ou seja, aquele processo que nunca consegue chegar ao processador, fica eternamente aguardando, sempre tem alguém que "fura a fila".

Gabarito: B

3. (FGV/FUNSAÚDE-CE/2021) Considere um sistema operacional onde o processo P1 precisa dos recursos R1 e R2 para prosseguir com seu processamento. Concomitantemente, há um processo P2, que também precisa dos recursos R1 e R2. A situação em que, simultaneamente, o processo P1 detém o recurso R1 e espera pelo recurso R2 e o processo P2 detém o recurso R2 e espera pelo recurso R1, é denominada

A) Checkout.

B) Deadlock.

C) Livelock.

D) Lockdown.

E) Shutdown.

Comentários:

Um *deadlock* (impasse) ocorre quando um conjunto de processos está esperando por um evento que só pode ser causado por outro processo do conjunto. Ou seja, ficam todos "amarrados" em um "abraço da morte", sem poder continuar seus processamentos. Alguns exemplos de recursos que só podem ser utilizados por um processo por vez: unidades de fita e impressoras. Ou você acha que é possível dois processos escrevendo em uma fita ou mandando imprimir ao mesmo tempo? Ficaria uma bagunça!

Por isso os sistemas operacionais possuem a capacidade de garantir (por algum tempo) que um processo tenha o acesso exclusivo a determinados recursos, sejam de hardware ou de software. Vamos supor a seguinte situação: os processos A e B desejam digitalizar uma fotografia através de um *scanner* e em seguida gravar em um CD-R. Digamos que A tenha requisitado primeiro o *scanner* e ao mesmo tempo B tenha requisitado o gravador de CD-R. Em seguida A faz requisição do gravador de CD-R, mas ele está ocupado, pois B não o "largou". Assim fica o processo A aguardando o recurso gravador de CD-R e B aguardando o recurso *scanner*. Assim temos um *deadlock*!

Gabarito: B

4. (FGV/Prefeitura de Manaus-AM/2022) Sobre o escalonamento de processos, analise as afirmativas abaixo:

I. O algoritmo Round Robin é preemptivo e baseado em quantum de tempo para cada processo, de forma rotativa.

II. O escalonamento não preemptivo é o mais indicado para ambientes de tempo compartilhado.

III. Em escalonadores não preemptivos, um processo permanece em execução tanto quanto possível, só saindo no caso de término de execução, ou quando executa uma instrução que ocasione uma mudança para um estado de espera.

Está correto apenas o que se afirma em

A) I.

B) II.

C) III.

D) I e II.

E) I e III.

Comentários:

I. CORRETA - O algoritmo Round Robin é preemptivo, ou seja, o processo pode ser retirado do processador. É baseado em quantum de tempo para cada processo, ou seja, há uma fatia de tempo para cada processo, de forma rotativa.

II. ERRADA - O correto seria: "O escalonamento não preemptivo é o mais indicado para ambientes de processamento em lote."

III. CORRETA - "Em escalonadores não preemptivos, um processo permanece em execução tanto quanto possível, só saindo no caso de término de execução, ou quando executa uma instrução que ocasione uma mudança para um estado de espera.". Ou seja, monopoliza o processador e só sai nas duas condições mostradas.

Gabarito: E

5. (FGV/TJ-SE/2023) A rotina do Sistema Operacional (SO) que tem como principal função implementar os critérios da política de escalonamento é:

A) dispatcher;

B) preempção;

C) tempo de resposta;

D) processamento batch;

E) escalonador (scheduler).

Comentários:

Escalonador (*scheduler*): rotina responsável por determinar a ordem de execução dos processos. Ele gerencia a alocação de recursos da CPU entre os vários processos concorrentes em um sistema.

Gabarito: E

6. (FGV/TJ-SE/2023) A política de escalonamento é a base da gerência do processador. Ryu é analista de sistemas e sabe que as características de cada Sistema Operacional (SO) determinam quais são os principais aspectos para a implementação de uma política de escalonamento adequada. Brevemente Ryu adotará, em seu projeto de SO, o critério de escalonamento que representa o número de processos executados em um determinado intervalo de tempo.

   Em seu projeto de SO, Ryu deve utilizar o critério de escalonamento:

A) throughput;

B) tempo de espera;

C) tempo de turnaround;

D) tempo de processador;

E) utilização do processador.

Comentários:

Throughput ("Taxa de Transferência"): critério de escalonamento que representa o número de processos executados em um determinado intervalo de tempo. O *throughput* refere-se à quantidade de trabalho realizado em um sistema durante um período específico. Essa medida é frequentemente expressa em termos de processos ou tarefas completadas por unidade de tempo. Quanto maior o número de processos que podem ser concluídos em um intervalo de tempo, maior é o *throughput*.

Gabarito: A

7. (FGV/TJ-SE/2023) As propriedades de cada sistema operacional determinam quais são os principais aspectos para a implementação de uma política de escalonamento adequada.

   A política de escalonamento classificada como preemptiva tem a seguinte característica:

A) sistemas operacionais que implementam escalonamento com preempção são mais simples, contudo, possibilitam políticas de escalonamento menos flexíveis;

B) foi a primeira política de escalonamento implementada nos sistemas monoprogramáveis, onde predominava tipicamente o processamento batch;

C) quando um processo está em execução, nenhum evento externo pode ocasionar a perda do uso do processador;

D) o processo sai do estado de execução caso termine seu processamento ou execute instruções do próprio código que ocasionem uma mudança para o estado de espera;

E) o sistema operacional pode interromper um processo em execução e passá-lo para o estado de pronto, com o objetivo de alocar outro processo na UCP.

Comentários:

Preempção: é o ato do S.O. utilizar as interrupções do relógio para retirar a CPU do processo em execução. Ou seja, o processo não pode monopolizar o processador! Quando o sistema não for preemptivo pode ocorrer uma situação denominada *starvation* (inanição). Traduzindo, *starvation* quer dizer "morrer de fome", ou seja, aquele processo que nunca consegue chegar ao processador, fica eternamente aguardando, sempre tem alguém que "fura a fila".

Gabarito: E

8. (FGV/TJ-SE/2023) Para um usuário interagir com um computador sem o Sistema Operacional (SO), ele deve conhecer profundamente diversos detalhes sobre o hardware do equipamento. Um técnico de programação de sistemas está testando um SO e verificou que se trata de uma arquitetura multiprogramada na qual vários processos são executados de forma concorrente.

   Nesse contexto, o algoritmo de escalonamento que seleciona o processo que tiver o menor tempo de processador ainda por executar é o:

A) não preemptivo;

B) por prioridade;

C) circular;

D) First-In-First-Out (FIFO);

E) Shortest-Job-First (SJF).

Comentários:

Shortest-Job First (SJF) ("Tarefa mais curta primeiro"): algoritmo não-preemptivo que presume que os tempos de execução são conhecidos previamente. Imagine uma situação em que os *jobs* (tarefas) de uma empresa são executados há muitos anos e já se sabe que os *jobs* do tipo A levam 2 minutos, do tipo B 6 minutos e do tipo C 4 minutos. Os *jobs* são agendados à tarde para serem executados às 8h do dia seguinte. Vamos ver como ficaria o escalonamento para a execução no dia seguinte dos *jobs* que foram adicionados na ordem B1, A1, C1, A2 e C2:

Não concordo muito com a resposta, mas é a melhor das opções. Na minha opinião o correto seria o Shortest Remaining Time Next (SRT) ("Menor tempo de execução restante"), versão

preemptiva do algoritmo SJF. Nesse algoritmo, o escalonador sempre escolhe o *job* com tempo de execução mais curto (obviamente o tempo precisa ser conhecido previamente). Quando um novo *job* chega, seu tempo é comparado com o que falta para concluir o *job* corrente. Se o novo precisar de menos tempo, o processo corrente é suspenso e o novo é iniciado. Esse esquema permite que *jobs* novos curtos tenham prioridade.

Gabarito: E

9. (FGV/DPE-RS/2023) Ana trabalha na empresa X, que presta serviço de Tecnologia da Informação à DPE/RS. Durante o funcionamento do site da Defensoria, notou a ocorrência de deadlock. Ana iniciou uma investigação para identificar a tarefa que ocasionou o deadlock. A atividade executada pela aplicação que ocasionou o deadlock foi:

A) um processo de update da aplicação que corrompeu um dado no banco de dados, impedindo o acesso ao banco;

B) o excesso de processos enviados para a execução em memória principal, causando lentidão;

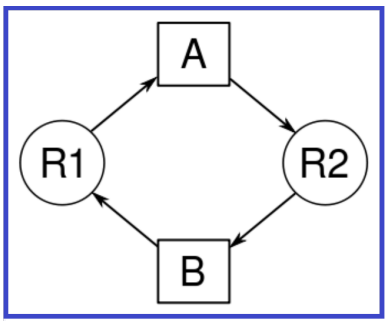
C) a execução de alguns processos, em background, que não liberavam recursos para a inicialização de outras tarefas;

D) o uso compartilhado dos recursos causando lentidão ao sistema para o controle de alterações;

E) o bloqueio de um usuário em virtude de ter esgotado a quantidade de erro de senha durante o processo de autenticação.

Comentários:

O deadlock ocorre quando processos ficam esperando um ao outro para liberar recursos. Uma figura que mostra a espera circular é mostrada abaixo.



Gabarito: C

QUESTÕES COMENTADAS - FCC

1. (FCC/Câmara Municipal-SP/2014) No escalonamento usando o algoritmo Round-Robin,

A) o escalonador seleciona o processo à espera com o menor tempo de execução estimado até a conclusão, reduzindo o tempo médio de espera, mas aumentando a variância dos tempos de resposta.

B) processos são despachados na ordem FIFO (First-in-First-Out), mas recebem uma quantidade limitada de tempo de processador denominada quantum.

C) a prioridade de cada processo é uma função não apenas do seu tempo de serviço, mas também do tempo que passou esperando pelo serviço.

D) o escalonador ajusta dinamicamente o comportamento do processo, de tal forma que o próximo processo a obter o processador seja aquele que chegar à frente da fila de nível mais alto, que não estiver vazia, na rede de filas.

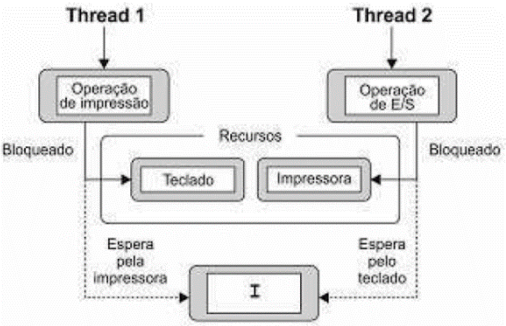
E) o processo que tem o prazo de execução mais curto é favorecido, medindo a diferença entre o tempo que um processo requer para finalizar e o tempo restante até atingir o seu prazo final.

Comentários:

Algoritmo Round-robin: é realizado um rodízio entre os processos, sendo que a cada processo é atribuído um intervalo de tempo (quantum), durante o qual ele pode ser executado. Se ao final do quantum o processo ainda estiver em execução é realizada a preempção da CPU e esta é alocada a um outro processo. Obviamente que se o processo tiver terminado antes do quantum ter expirado ou se tiver sido bloqueado, a troca da CPU é realizada neste momento.

Gabarito: B

2. (FCC/DPE-RS/2017) Considere a figura abaixo.



Do ponto de vista do sistema operacional, a situação indica que a caixa I deve ser preenchida com?

A) starvation.

B) multithreading.

C) superthreading.

D) deadlock.

E) hyperthreading.

Comentários:

Note que a thread 1 está realizando uma operação de impressão, bloqueando o teclado e está à espera da impressora. A thread 2 está realizando uma operação de E/S, bloqueando a impressora e à espera do teclado. Ou seja, nenhuma thread libera o recurso que está usando e cada uma quer um recurso que a outra possui. As duas ficarão "trancadas", esperando... isso é um deadlock!

Gabarito: D

3. (FCC/DPE-RS/2017) Dentre as políticas de escalonamento de processos a seguir, a que apresenta maior probabilidade de ocasionar o starvation é a

A) Round Robin.

B) de tempo compartilhado.

C) First In First Out.

D) preemptiva.

E) não preemptiva.

Comentários:

Uma política de escalonamento não-preemptiva é aquela que um recurso não pode ser retirado de um processo, a não ser que "ele queira". Imagine um processo que comece a utilizar a CPU e fique utilizando por 1h. Além disso digamos que há uma certa prioridade entre os processos e existe uma demanda muito grande pelo uso da CPU. Pode ocorrer que um processo com prioridade baixa jamais seja executado! Ou seja, starvation!

Gabarito: E

4. (FCC/DPE-AM/2018) Em um sistema operacional típico, os estados de um processo são:

A) Ativo, Desocupado, Finalizado e Executando.

B) Bloqueado, Desbloqueado, Ativo e Suspenso.

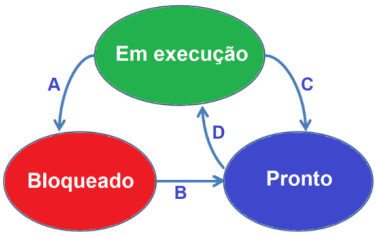
C) Executando, Bloqueado e Pronto.

D) Parado, Ocupado, em Execução e Finalizado.

E) Pronto, Terminado, Ativo e Processando.

Comentários:

Você tem que saber fazer esse desenho na sua prova:



Gabarito: C

5. (FCC/DPE-AM/2018) Em um sistema operacional de computador, três processos estão na seguinte situação:

– o processo P1 tem a posse do recurso R1.

– o processo P2 tem a posse do recurso R2.

– o processo P3 tem a posse do recurso R3.

O processo P1 solicita o recurso R2, o processo P2 solicita o recurso R3, e o processo P3 solicita o recurso R1. Sobre essa situação, é correto afirmar que

A) não haverá deadlock, pois o processo P1 não solicitou o recurso R3.

B) tem-se uma condição de deadlock.

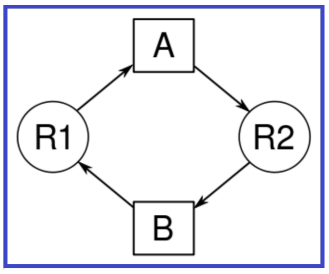
C) não haverá deadlock, pois o processo P3 não solicitou o recurso R2

D) só ocorrerá deadlock caso P1 solicite o recurso R3, P2 solicite o recurso R1 e P3 solicite o recurso R2.

E) não haverá deadlock, pois o processo P2 não solicitou o recurso R1.

Comentários:

Podemos ver que nenhum processo "larga" o recurso que possui. Além disso, cada processo solicita um recurso que está alocado por outro processo, de forma circular. Isso caracteriza um deadlock! Na figura abaixo vemos com 2 processos:



Gabarito: B

6. (FCC/DPE-AM/2018) Em um sistema operacional típico de um computador, três processos se encontram na seguinte situação:

– o processo P1 envia uma mensagem ao Processo P2.

– o processo P2, ao receber a mensagem de P1, responde essa mensagem a P1.

– o processo P1, ao receber a mensagem de resposta de P2, responde a P2 com uma nova mensagem, e assim sucessivamente.

– o processo P3 se encontra bloqueado, situação da qual sairá apenas quando receber uma mensagem do processo P1.

Considerando que a prioridade do processo P3 é menor do que as prioridades dos processos P1 e P2, tem-se que

A) a cada troca de mensagens entre P1 e P2 as respectivas prioridades automaticamente serão reduzidas, e quando elas forem inferior à do processo P3, esse será executado.

B) após passar 1 segundo bloqueado, o relógio de tempo real do sistema operacional automaticamente dará oportunidade de o processo P3 ser desbloqueado.

C) o processo P3 sairá dessa situação assim que uma interrupção qualquer ocorra.

D) ocorrerá uma situação conhecida como impasse (deadlock).

E) ocorrerá uma situação conhecida como inanição (starvation).

Comentários:

Podemos ver que as mensagens ficam apenas entre P1 e P2, enquanto P3 está no estado bloqueado, esperando uma mensagem de P1, que não ocorrerá!

Ou seja, ele vai ficar eternamente esperando, vai "morrer de fome" (*starvation*)! Vamos ver o conceito de *starvation* abaixo.

Starvation (inanição, ou privação): situação na qual um processo ou *thread* é incapaz de prosseguir com a execução devido à não recepção dos recursos necessários para avançar. Isso pode ocorrer em ambientes multitarefa quando um processo fica impedido de obter acesso a recursos cruciais, como processador, memória, ou dispositivos de entrada e saída.

Gabarito: E

QUESTÕES COMENTADAS - VUNESP

1. (VUNESP/Pref. de Caieiras-SP/2015) Em um sistema operacional típico, vários processos podem se encontrar no estado "pronto" em um dado instante. A gerência do processador efetua a escolha de qual desses processos receberá o processador. Essa escolha atende a critérios previamente definidos, que fazem parte da política de

A) alocação de memória.

B) escalonamento de processos.

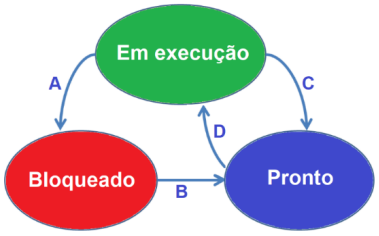
C) minimização do throughput do sistema

D) particionamento da Unidade Central de Processamento.

E) virtualização da memória principal.

Comentários:

Na figura abaixo podemos ver que a partir do estado pronto o processo vai para a execução, no processador.



Para saber qual dos processos utilizará o processador, existe uma política de escalonamento de processos, que pode ser a round-robin (mais cobrada em provas de concurso), SJF (*Shortest Job First*), entre outras.

Gabarito: B

2. (VUNESP/Pref. de P. Prudente-SP/2016) Os sistemas operacionais modernos empregam elementos que são fluxos independentes de execução que pertencem a um mesmo processo e que exigem menos recursos de controle do sistema operacional. Esses elementos são denominados

A) Bash.

B) Buffers.

C) Kernel.

D) Semáforos.

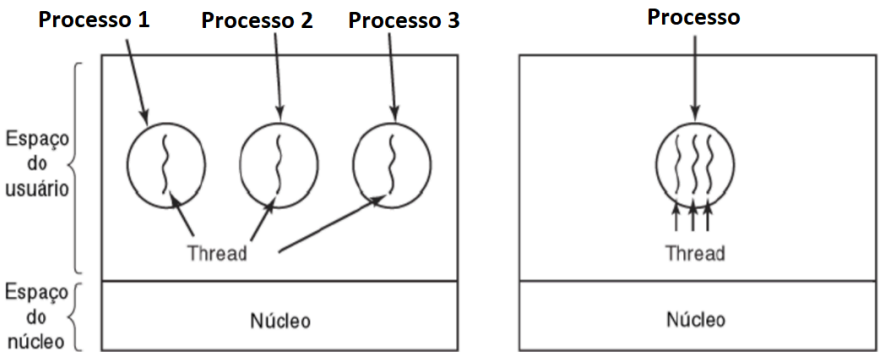
E) Threads.

Comentários:

Um processo "tradicional" possui um espaço de endereçamento e um fluxo de controle (execução do código). Porém, há situações em que se deseja ter mais de um fluxo de controle e execução no mesmo processo, executando quase em paralelo. Esses fluxos são chamados *threads* (ou processos leves). Resumindo: *threads* de um mesmo processo compartilham a mesma seção de código na memória.

Imagine um editor de texto, que possui inúmeras funcionalidades: contador de palavras, contador de páginas, correção ortográfica instantânea, entre outras. Cada uma delas geralmente é implementada em uma *thread*, então a cada digitação elas verificam se a quantidade de palavras aumentou/diminuiu (e atualiza essa informação na tela), se a quantidade de páginas foi alterada, se a palavra digitada está correta (após consultar um arquivo de dicionário), e assim por diante.

Abaixo podemos ver as *threads* dentro de um processo.



Gabarito: E

3. (VUNESP/FUNDUNESP/2016) Considere a seguinte situação ocorrendo no ambiente de um sistema operacional no qual um conjunto de processos está sendo executado: "em determinado momento, cada processo está esperando um evento que somente outro processo poderá fazer acontecer". Essa descrição refere-se a

A) deadlock.

B) interrupts.

C) multitasking.

D) throughput.

E) time-sharing.

Comentários:

Um *deadlock* (impasse) ocorre quando um conjunto de processos está esperando por um evento que só pode ser causado por outro processo do conjunto. Ou seja, ficam todos "amarrados" em um "abraço da morte", sem poder continuar seus processamentos. Alguns exemplos de recursos que só podem ser utilizados por um processo por vez: unidades de fita e impressoras. Ou você

acha que é possível dois processos escrevendo em uma fita ou mandando imprimir ao mesmo tempo? Ficaria uma bagunça!

Por isso os sistemas operacionais possuem a capacidade de garantir (por algum tempo) que um processo tenha o acesso exclusivo a determinados recursos, sejam de hardware ou de software. Vamos supor a seguinte situação: os processos A e B desejam digitalizar uma fotografia através de um *scanner* e em seguida gravar em um CD-R. Digamos que A tenha requisitado primeiro o *scanner* e ao mesmo tempo B tenha requisitado o gravador de CD-R. Em seguida A faz requisição do gravador de CD-R, mas ele está ocupado, pois B não o "largou". Assim fica o processo A aguardando o recurso gravador de CD-R e B aguardando o recurso *scanner*. Assim temos um *deadlock*!

Gabarito: A

4. (VUNESP/PC-BA/2018) As versões modernas do sistema operacional Windows, como a versão 10, aparentam ao seu usuário que várias tarefas são executadas ao mesmo tempo. Essa característica é conhecida como

A) monotarefa preemptiva.

B) monotarefa sem preempção.

C) multitarefa preemptiva.

D) multitarefa sem preempção.

E) time-sharing sem preempção.

Comentários:

Sistema multitarefa preemptivo: um sistema que possibilita a execução de mais de um processo ao mesmo tempo geralmente é preemptivo, ou seja, em algum momento o S.O. retira o processador do processo e coloca outro no lugar.

Gabarito: C

5. (VUNESP/Câm. Piracicaba-SP/2019) No contexto de gerenciamento de processos e fluxos de execução (threads) em sistemas operacionais, assinale a alternativa em que todos os itens listados sejam privados para cada thread, isto é, não são compartilhados com outros threads no mesmo processo.

A) Arquivos abertos e registradores.

B) Arquivos abertos e pilha.

C) Registradores e pilha.

D) Espaço de endereçamento e registradores.

E) Espaço de endereçamento e arquivos abertos.

Comentários:

Um processo "tradicional" possui um espaço de endereçamento e um fluxo de controle (execução do código). Porém, há situações em que se deseja ter mais de um fluxo de controle e execução no mesmo processo, executando quase em paralelo. Esses fluxos são chamados *threads* (ou processos leves). Resumindo: *threads* de um mesmo processo compartilham a mesma seção de código na memória. Porém, cada *thread* possui os seus valores nos registradores e na pilha, ou seja, a cada troca de contexto entre as *threads*, esses valores são atualizados.

Gabarito: C

6. (VUNESP/Câm. de Piracicaba-SP/2019) Deadlock é um problema potencial em qualquer sistema operacional. Sejam dois processos PA e PB, e dois recursos RX e RY.

Assinale a alternativa que reflete a ocorrência de um deadlock.

A) PA possui acesso exclusivo a RX e RY, enquanto PB aguarda por acesso exclusivo a RX.

B) PA possui acesso exclusivo a RX, enquanto PB solicita acesso exclusivo a RY.

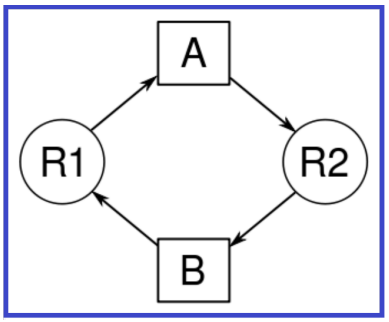
C) PA possui acesso exclusivo a RX e aguarda por acesso exclusivo a RY, enquanto PB possui acesso exclusivo a RY e aguarda por acesso exclusivo a RX.

D) PA possui acesso exclusivo a RX, enquanto PB possui acesso exclusivo a RY. PA libera RX e então PB requisita acesso exclusivo a RX.

E) PA possui acesso exclusivo a RX, enquanto PB possui acesso exclusivo a RY. PA então libera RX ao mesmo tempo que PB libera RY.

Comentários:

Vimos em aula a figura abaixo como exemplo de ocorrência de *deadlock*. Das alternativas propostas, a que se encaixa é "PA possui acesso exclusivo a RX e aguarda por acesso exclusivo a RY, enquanto PB possui acesso exclusivo a RY e aguarda por acesso exclusivo a RX". Podemos substituir PA por A, RX por R2, PB por B e RY por R1.



Gabarito: C

7. (VUNESP/Câmara de Tatuí-SP/2019) Em um sistema operacional, existem diversas situações que poderiam acarretar o deadlock. Algumas estratégias podem ser utilizadas no seu tratamento, a exemplo daquela conhecida como "Algoritmo do Avestruz", que

A) ignora a sua existência.

B) detecta o deadlock e recupera o sistema.

C) detecta a possibilidade de sua ocorrência e evita que ele ocorra.

D) elimina os processos causadores do deadlock, sem afetar os demais.

E) reinicia o sistema se o travamento ultrapassar um tempo predefinido.

Comentários:

O algoritmo do avestruz age igual uma ave avestruz, mesmo! A ave costuma colocar o cabeça em um buraco, certo? Então é isso! O algoritmo "coloca a cabeça em um buraco" e ignora a existência do *deadlock*.

Gabarito: A

8. (VUNESP/Prefeitura de Pindamonhangaba-SP/2023) O deadlock é um fenômeno que pode ocorrer em sistemas operacionais. Assinale a alternativa correta relacionada a ele.

A) A sua ocorrência se manifesta em computadores com um único processador, mas não ocorre em computadores com mais de um processador.

B) Ele pode ser evitado dinamicamente, pela alocação cuidadosa de recursos entre os processos.

C) Não existe forma de detectá-lo, e quando ele ocorrer, o computador deve ser reinicializado.

D) O deadlock torna a execução dos processos mais lenta devido à atribuição inadequada de prioridades aos processos, mas não interrompe a execução de nenhum deles.

E) Um algoritmo utilizado no seu tratamento é o Algoritmo do Avestruz, que identifica a sua ocorrência e reinicia os processos envolvidos.

Comentários:

A) ERRADA - O *deadlock* pode ocorrer com ou mais processadores, depende da alocação de recursos e a dependência entre eles.

B) CORRETA - Não é uma tarefa fácil, mas o *deadlock* pode ser evitado dinamicamente, se houver uma alocação cuidadosa de recursos entre os processos, para evitar um impasse, quando um recurso está alocado por um processo, não o libera e precisa de outro recurso, também alocado, e assim por diante.

C) ERRADA - Existem formas de detecção e pode-se "matar" algum processo, por exemplo. Ou reiniciar a máquina, também, para "zerar" tudo.

D) ERRADA - O deadlock "trava" alguns processos, fica um dependendo do outro.

E) ERRADA - O Algoritmo do Avestruz ignora a existência de *deadlock*, igual uma ave avestruz que coloca a cabeça em um buraco.

Gabarito: B

QUESTÕES COMENTADAS - CESGRANRIO

1. (CESGRANRIO/TRANSPETRO/2011) Os Sistemas Operacionais estão sujeitos a um fenômeno denominado deadlock. Para que uma situação de deadlock seja criada, as seguintes condições devem acontecer simultaneamente

A) exclusão mútua (mutual exclusion), monopolização de recursos (hold and wait), não preempção (no preemption) e espera circular (circular wait).

B) exclusão mútua (mutual exclusion), transferência excessiva de páginas (thrashing), superposição de processos (process overlapping) e espera circular (circular wait).

C) transferência excessiva de páginas (thrashing), superposição de processos (process overlapping), monopolização de recursos (hold and wait) e não preempção (no preemption).

D) exclusão mútua (mutual exclusion), monopolização de recursos (hold and wait), superposição de processos (process overlapping) e falha de escalonamento (scheduling fail).

E) transferência excessiva de páginas (thrashing), não preempção (no preemption), espera circular (circular wait) e falha de escalonamento (scheduling fail).

Comentários:

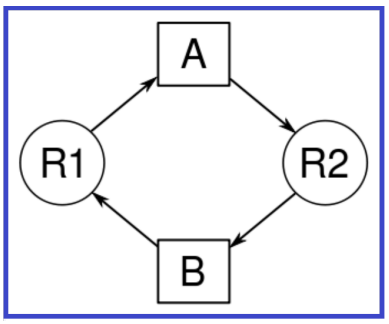
As quatro condições que devem ser verdadeiras para que ocorra um deadlock são:

1. Condição de exclusão mútua: cada recurso ou está correntemente atribuído a exatamente um processo ou está disponível;

2. Condição de posse e espera: os processos que possuem recursos garantidos anteriormente podem solicitar novos recursos (um acumulador de recursos!);

3. Ausência de preempção: os recursos garantidos não podem ser retirados à força de um processo;

4. Condição de espera circular: um encadeamento circular de dois ou mais processos, cada um esperando por um recurso mantido pelo próximo do encadeamento:



Gabarito: A

2. (CESGRANRIO/Petrobras/2014) Sejam quatro processos criados no instante de tempo 0 (P1 , P2 , P3 e P4) e que possuam as características descritas na Tabela a seguir.

Processo	Tempo de UCP (u.t.)	Prioridade
P1	11	2
P2	25	1
P3	12	4
P4	8	3

Considerando-se um tempo de troca de contexto igual a 4 u.t., qual é o tempo de turnaround de cada processo caso a política de execução de processos seja a SJF?

A) P1-11/ P2- 40/P3 -56/P4-68

B) P1-19/ P2-56/P3 -31/P4-8

C) P1-23/ P2-68 / P3-39/ P4-8

D) P1-27/ P2 -68/P3 - 43/P4-12

E) P1-40/ P2 - 25/P3-35/ P4-52

Comentários:

O tempo de *turnaround* é o tempo de existência de um processo, ou seja, o tempo desde a sua criação até seu término. As políticas de escalonamento buscam minimizar o tempo de *turnaround*. A política SJF (*Shortest Job First*) executa todo o processo menor, depois o segundo menor e assim por diante. Pegadinha!!! A prioridade não importa para a política SJF, então aquela coluna na tabela só serve pra atrapalhar!!!

Sabendo que o tempo de troca de contexto é de 4 u.t. (unidades de tempo), temos:

- P4 (8 u.t.): terminou a execução em 8 u.t.;
- troca de contexto (4 u.t.);
- P1 (11 u.t.): terminou a execução em 23 u.t. (8 + 4 + 11);
- troca de contexto (4 u.t.);
- P3 (12 u.t.): terminou a execução em 39 u.t. (8 + 4 + 11 + 4 + 12);
- troca de contexto (4 u.t.);
- P2 (25 u.t.): terminou a execução em 68 u.t. (8 + 4 + 11 + 4 + 12 + 4 + 25).

Gabarito: C

3. (CESGRANRIO/UNIRIO/2016) Dentre as funções realizadas por Sistemas Operacionais está o escalonamento de processos, que é responsável pela decisão de qual processo deve receber atenção do processador a cada instante. Uma certa política de escalonamento coloca todos os processos em uma fila circular para o atendimento, especificando uma fatia de tempo (chamada quantum) após a qual o processo em execução é, preemptivamente, suspenso, e o próximo da fila passa a executar. Que nome se dá a essa política de escalonamento?

A) Round-Robin

B) Shortest-Job-First

C) Priority Scheduling

D) First-Come-First-Served

E) Multilevel Queue Scheduling

Comentários:

Round-robin: trata-se do algoritmo mais conhecido e cobrado! Com ele é realizado um rodízio entre os processos, sendo que a cada processo é atribuído um intervalo de tempo (*quantum*), durante o qual ele pode ser executado. Se ao final do *quantum* o processo ainda estiver em execução é realizada a preempção da CPU e esta é alocada a um outro processo. Obviamente que se o processo tiver terminado antes do fim do *quantum* ou se tiver sido bloqueado, a troca da CPU é realizada neste momento.

Um ponto interessante é a definição da duração do *quantum*, pois trocar de um processo para outro exige uma quantidade de tempo para salvar e carregar registradores e mapas de memória, atualizar tabelas e listas etc. Vamos supor que esse chaveamento de contexto demore 1ms e que o *quantum* seja de 9ms. Nesse caso, 10% da CPU seriam desperdiçados em sobrecarga administrativa.

Gabarito: A

4. (CESGRANRIO/IBGE/2016) Processos de sistemas operacionais podem se encontrar em um dentre três estados.

   O estado que caracteriza que um processo está adormecido, aguardando a ocorrência de um evento externo, caracterizando a entrega voluntária do uso do processador ao sistema operacional é chamado de

A) bloqueado

B) despacho

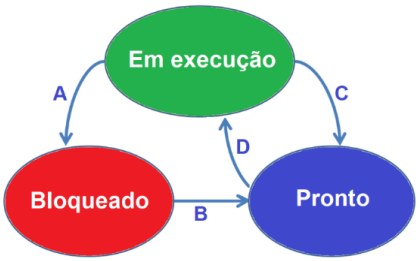
C) em execução

D) preemptivo

E) pronto

Comentários:

O processo, quando em execução, e antes de terminar a "fatia de tempo dele" (quantum), pode entregar o processador para ficar esperando alguma coisa, como por exemplo a leitura de um arquivo do disco rígido. Nesse momento ele fica bloqueado, esperando pela leitura do arquivo e não precisaria ficar usando o processador.



Gabarito: A

5. (CESGRANRIO/IBGE/2016) Seja um sistema operacional que implementa multiprogramação e mantém cinco processos simultaneamente na memória. Considere-se ainda que os processos são independentes, com tempo de overhead desprezível (igual a zero). Sabe-se que todos os processos demandam exatos 40% do tempo aguardando a conclusão de operação de entrada e saída.

   Nesse caso, qual é o percentual de utilização do processador?

A) 1,02%

B) 6,25%

C) 60,00%

D) 93,75%

E) 98,98%

Comentários:

Questão de probabilidade e não de sistemas operacionais!!! 🙂

Mas vamos lá... pense no seguinte: são 5 processos e mesmo que 4 estejam bloqueados (aguardando a conclusão de uma operação de entrada e saída), 1 estaria usando o processador. Então a única situação em que o processador fica ocioso é quando os 5 processos estiverem aguardando a conclusão de uma operação de entrada e saída. Cada processo demanda 40% do tempo aguardando a conclusão de uma operação de entrada e saída, como está no enunciado.

Lembrando que a multiplicação significa "E", então a probabilidade de P1 E P2 E P3 E P4 E P5 estarem aguardando E/S é 0,4 x 0,4 x 0,4 x 0,4 x 0,4 = 0,01024 (ou 1,024%).

Logo, 100% - 1,024% = 98,976% de probabilidade do processador estar em uso. Arredondando, fica 98,98%.

Gabarito: E

6. (CESGRANRIO/Petrobras/2018) A política de escalonamento de processos de um sistema operacional é uma das partes que mais pode influenciar no seu desempenho.

   A estratégia que associa, a cada processo, um valor baseado no tempo em que ele deverá ocupar a CPU e escolhe o de menor valor para a execução é denominada

A) First-Come-First-serve

B) Last-Come-First-Serve

C) Longest-job-first

D) Round Robin

E) Shortest-job-first

Comentários:

Shortest-Job First (SJF) ("Tarefa mais curta primeiro"): algoritmo não-preemptivo que presume que os tempos de execução são conhecidos previamente. Imagine uma situação em que os *jobs* (tarefas) de uma empresa são executados há muitos anos e já se sabe que os *jobs* do tipo A levam 2 minutos, do tipo B 6 minutos e do tipo C 4 minutos. Os *jobs* são agendados à tarde para serem executados às 8h do dia seguinte. Vamos ver como ficaria o escalonamento para a execução no dia seguinte dos *jobs* que foram adicionados na ordem B1, A1, C1, A2 e C2:

Gabarito: E

7. (CESGRANRIO/Transpetro/2018) A política de escalonamento estabelece os critérios utilizados para selecionar o processo que fará uso do processador.

   No escalonamento não preemptivo, quando um processo está em execução,

A) apenas o sistema operacional pode ocasionar a perda do uso do processador.

B) qualquer processo em espera pode ocasionar a perda do uso do processador.

C) qualquer processo pronto pode ocasionar a perda do uso do processador.

D) nenhum evento externo pode ocasionar a perda do uso do processador.

E) nem mesmo o próprio processo pode ocasionar a perda do uso do processador.

Comentários:

Preempção é o ato do S.O. utilizar as interrupções do relógio para retirar a CPU do processo em execução. Ou seja, o processo não pode monopolizar o processador! Quando o sistema não for preemptivo pode ocorrer uma situação denominada *starvation* (inanição). Traduzindo, *starvation* quer dizer "morrer de fome", ou seja, aquele processo que nunca consegue chegar ao processador, fica eternamente aguardando, sempre tem alguém que "fura a fila".

Como a questão busca o conceito de escalonamento não preemptivo, então "nenhum evento externo pode ocasionar a perda do uso do processador", ou seja, o processo começa a usar o processador e só deixa de usar quando terminar por completo.

Gabarito: D

8. (CESGRANRIO/Transpetro/2018) Em sistemas operacionais multitarefas e interativos é comum utilizar um algoritmo de escalonamento de processos. Um algoritmo amplamente usado é conhecido como round-robin.

Nesse algoritmo, cada processo

A) vai para uma fila de acordo com sua prioridade. As filas de maior prioridade são executadas primeiro. Cada fila é executada em ordem de chegada. Quando todas as filas são executadas, inicia-se de novo pela fila de maior prioridade.

B) recebe um número. A CPU seleciona aleatoriamente um dos processos em espera e o executa por um tempo fixo. Se esse tempo se esgota, a CPU seleciona aleatoriamente outro processo para substituí-lo.

C) possui um grau de importância que define sua ordem em uma lista de espera. Na sua vez, o processo é executado por um tempo fixo. Se esse tempo é ultrapassado, a CPU dá o controle para o próximo da lista, e o processo que estava sendo executado volta para uma posição à frente de todos os outros processos de menor importância

D) na lista de espera recebe um intervalo de tempo fixo em que é permitido executar. Na sua vez, se esse intervalo é ultrapassado, a CPU dá o controle para o próximo processo da lista, e o processo que estava sendo executado vai para o fim da lista.

E) na fila de espera possui uma prioridade. As prioridades são usadas para calcular o intervalo de tempo que o processo deve ficar na CPU, quanto maior a prioridade, maior o tempo. Quando esse tempo se esgota, a CPU dá o controle para o próximo processo da lista, e o processo que estava sendo executado vai para o fim da lista.

Comentários:

O round-robin realiza um rodízio entre os processos, sendo que a cada processo é atribuído um intervalo de tempo (*quantum*), durante o qual ele pode ser executado. Se ao final do *quantum* o processo ainda estiver em execução é realizada a preempção da CPU e esta é alocada a um outro processo. Obviamente que se o processo tiver terminado antes do fim do *quantum* ou se tiver sido bloqueado, a troca da CPU é realizada neste momento.

Um ponto interessante é a definição da duração do *quantum*, pois trocar de um processo para outro exige uma quantidade de tempo para salvar e carregar registradores e mapas de memória, atualizar tabelas e listas etc. Vamos supor que esse chaveamento de contexto demore 1ms e que o *quantum* seja de 9ms. Nesse caso, 10% da CPU seriam desperdiçados em sobrecarga administrativa.

Gabarito: D

9. (CESGRANRIO/LIQUIGÁS/2018) A gerência do processador visa a otimizar o seu uso a partir do emprego de técnicas de escalonamento de processos. Dentre os critérios adotados para interromper o processo que está em execução, o término da fatia de tempo é amplamente utilizado pelos sistemas operacionais.

Esse critério é adotado no escalonamento

A) Shortest Job First

B) First In First Out

C) First Come First Served

D) Round Robin (Circular)

E) por Prioridade

Comentários:

O round-robin realiza um rodízio entre os processos, sendo que a cada processo é atribuído um intervalo de tempo (*quantum*), durante o qual ele pode ser executado. Se ao final do *quantum* o processo ainda estiver em execução é realizada a preempção da CPU e esta é alocada a um outro processo. Obviamente que se o processo tiver terminado antes do fim do *quantum* ou se tiver sido bloqueado, a troca da CPU é realizada neste momento.

Gabarito: D

10. (CESGRANRIO/UNIRIO/2019) Um sistema operacional deve ter o completo domínio sobre os recursos da máquina. O escalonamento de recursos, o controle de entrada e saída (E/S), a gerência da memória, a gerência do processador, o escalonamento de processos e a segurança são funções que o sistema operacional deve exercer.

Um conceito fundamental em todos os sistemas operacionais é o processo, que significa

A) basicamente um programa em execução

B) atividades para gerenciar a memória

C) chamadas ao sistema (system calls)

D) um serviço oferecido pelo sistema operacional

E) um controlador das entradas e saídas

Comentários:

Um processo é simplesmente uma instância de um programa em execução, incluindo os valores correntes dos registradores (PC, IR, entre outros) e das variáveis (ex.: soma, total, em um programa que realiza cálculos). Cada processo pensa que está "sozinho no mundo" e executa em um processador (CPU) virtual, mas sabemos que na prática o processador alterna de um processo para outro.

Gabarito: A

11. (CESGRANRIO/UNIRIO/2019) Deadlock, no contexto de um sistema operacional, é uma

A) ação ou função implementada como uma sequência de uma ou mais instruções que são indivisíveis, ou seja, nenhum outro processo pode interromper a operação.

B) situação em que dois ou mais processos não podem continuar suas execuções, visto que cada um deles espera pelos recursos alocados entre eles.

C) situação em que um processo executável é esquecido indefinidamente pelo escalonador; embora esteja apto a executar, nunca é escolhido.

D) seção de código dentro do processo que requisita acesso a recursos compartilhados, e que não deve ser executado se outro processo está na mesma seção de código.

E) condição na qual múltiplos threads ou processos leem e gravam em uma área de dados compartilhada, e o resultado final depende do tempo relativo a cada operação executada.

Comentários:

Um *deadlock* (impasse) ocorre quando um conjunto de processos está esperando por um evento que só pode ser causado por outro processo do conjunto. Ou seja, ficam todos "amarrados" em um "abraço da morte", sem poder continuar seus processamentos. Alguns exemplos de recursos que só podem ser utilizados por um processo por vez: unidades de fita e impressoras. Ou você acha que é possível dois processos escrevendo em uma fita ou mandando imprimir ao mesmo tempo? Ficaria uma bagunça!

Gabarito: B

12. (CESGRANRIO/Caixa/2021) A sincronização entre processos concorrentes é fundamental para garantir a confiabilidade dos sistemas multiprogramáveis. Um mecanismo de sincronização simples, que permite implementar a exclusão mútua sem a deficiência da espera ocupada (busy wait), é o

A) deadlock

B) mutual lock

C) escalonamento binário

D) buffer contador

E) semáforo mutex

Comentários:

Mutex: versão simplificada do semáforo, quando não for necessário "contar", utilizando-se apenas os estados "livre" ou "ocupado". Consequentemente é necessário apenas um bit para representá-lo, mas na prática geralmente é utilizado um valor inteiro. Quando um processo precisa entrar na região crítica, ele chama *mutex_lock*, que será bem-sucedida se a região estiver livre. Caso contrário, o processo ficará bloqueado até que o processo que estiver na região crítica saia (*mutex_unlock*).

Gabarito: E

13. (CESGRANRIO/Transpetro/2023) Em um sistema operacional moderno, vários processos devem ser atendidos simultaneamente, dando ao usuário a impressão de estarem rodando simultaneamente. Para isso, é necessário gerenciar o processador por meio de algoritmos de escalonamento, que definem que processo executa e quais ficam esperando, de acordo com diferentes parâmetros.

Um dos critérios que podem ser usados para comparar esses algoritmos é o tempo de turnaround, que conta o tempo

A) entre a submissão de um pedido e a primeira resposta a ele.

B) que o processo realmente passa sendo executado pelo processador.

C) que um processo passa em estado de bloqueio, geralmente aguardando uma operação de E/S ser concluída.

D) total, desde a submissão do processo até a sua conclusão.

E) total que um processo passa na fila de pronto, aguardando para ser executado pelo processador.

Comentários:

Tempo de *turnaround*: tempo de existência de um processo, ou seja, o tempo desde a sua criação até seu término. As políticas de escalonamento buscam minimizar o tempo de turnaround.

Gabarito: D

QUESTÕES COMENTADAS - MULTIBANCAS

1. (ESAF/CGU/2008) Analise as seguintes afirmações, levando em conta as chamadas de sistemas usadas com semáforos, e assinale a opção verdadeira.

I. A chamada de sistema UP adiciona uma unidade ao valor corrente de um semáforo.

II. Se o valor do semáforo é zero, uma chamada de sistema DOWN não será completada e o processo será suspenso.

III. Quando um processo inicia a execução de uma chamada de sistema UP ou DOWN, nenhum outro processo terá acesso ao semáforo até que o processo complete a execução ou seja suspenso.

A) Apenas I e II são verdadeiras.

B) Apenas I e III são verdadeiras.

C) Apenas II e III são verdadeiras.

D) I, II e III são verdadeiras.

E) I, II e III são falsas.

Comentários:

Em um semáforo existem as operações down e up (ou sleep e wakeup). A operação down verifica se o valor é maior que 0. Se for, ele decrementa um e continua. Se o valor for 0, o processo (ou a thread) é "colocado para dormir" (bloqueado) sem completar a operação down. É garantido que iniciada uma operação de semáforo, nenhum outro processo pode acessar o semáforo até que a operação tenha terminado ou sido bloqueada (ação atômica). Isso evita as condições de corrida.

Gabarito: D

2. (IADES/PG-DF/2011) O escalonamento de tarefas é uma atividade de processamento realizada pela CPU de um computador. Esta atividade permite executar de forma mais eficiente os processos considerados prioritários para o sistema operacional. Assinale a alternativa que apresenta o escalonamento de tarefas em um computador, utilizado como servidor de arquivos de uma rede.

A) O escalonamento garantido busca atender a demanda da rede, priorizando ações de leitura

B) O algoritmo de escalonamento FIFO (First In, First Out) atua na gravação de arquivos em disco, implementando o conceito de pilha de escalonamento.

C) Os algoritmos de escalonamento preemptivos devem permitir que um processo seja interrompido durante sua execução.

D) O algoritmo de escalonamento de múltiplas filas permite o acesso simultâneo a arquivos e banco de dados disponibilizados na rede.

E) O escalonador de longo prazo seleciona os processos na interface de rede, dando prioridade às ações de I/O (Input/Output).

Comentários:

(A) O escalonamento garantido é aquele que busca fazer promessas realistas aos usuários sobre o desempenho; (B) O algoritmo de escalonamento FIFO (First In, First Out) é uma fila, o primeiro a entrar é o primeiro a sair; (C) Exato! Os preemptivos permitem que um processo seja interrompido durante sua execução, evitando uma monopolização do uso da CPU; (D) O algoritmo de escalonamento de múltiplas filas atua com classes de prioridade; (E) Não chegamos a ver na aula. Trata-se de um escalonador que seleciona os processos que estão na memória secundária e que serão levados para a memória principal.

Gabarito: C

3. (FUNRIO/IF-PA/2016) Sistemas operacionais compartilham recursos, havendo a possibilidade de deadlocks. A literatura especializada indica quatro condições necessárias para que um deadlock ocorra. O algoritmo de Avestruz utiliza uma estratégia para lidar com deadlocks conhecida como

A) detectar.

B) detectar e recuperar.

C) evitar.

D) ignorar.

E) prevenir.

Comentários:

Algoritmo do avestruz: estratégia mais simples! O que um avestruz faz? Coloca a cabeça em um buraco... pois é, é isso mesmo! O algoritmo finge que nada aconteceu, simplesmente ignora. E, por incrível que pareça, é a estratégia adotada pela maioria dos sistemas operacionais (Windows, Linux, entre outros). Por quê? Porque o preço seria alto em realizar muitas restrições ao usuário. Melhor deixar acontecer...e se for o caso, o usuário reinicia a máquina ou mata um processo, caso ocorra algum travamento.

Gabarito: D

4. (IESES/IGP-SC/2017) Acerca da gerência de processos dos sistemas operacionais, assinale a alternativa correta:

A) Um conjunto de processos está em estado de deadlock quando todos os processos no conjunto estão esperando por um evento que só pode ser causado por outro processo do conjunto.

B) Em um escalonamento preemptivo, um processo só perde o processador se terminar ou entrar em estado de espera.

C) No algoritmo de escalonamento de processos Round Robin, o escalonador sempre escolhe para execução o processo com menor expectativa de tempo de processamento. Esse algoritmo baseia-se no fato de que privilegiando processos pequenos o tempo médio de espera decresce.

D) Starvation é uma situação que não pode ocorrer quando um sistema operacional provê prioridades a processos.

Comentários:

(A) Isso aí... fica um aguardando o outro liberar um recurso, que por sua vez aguarda um terceiro e por aí vai... (B) Em um escalonamento preemptivo um processo perde o processador se terminar, ficar bloqueado ou se terminar seu quantum; (C) No algoritmo de escalonamento de processos Round Robin ocorre um "rodízio", sendo que cada processo recebe a mesma quantidade de quantum em uma ordem FIFO; (D) Starvation justamente ocorre quando um sistema operacional provê prioridades a processos. Imagine o uso do recurso processador, sendo que um processo tenha a prioridade máxima e um outro tenha a mínima. O primeiro leve 4 dias para terminar a execução e antes disso outros processos "surgiram" com uma prioridade maior do que aquele que tem a mínima...quando ele terá acesso ao processador?

Gabarito: A

5. (CONSULPLAN/TRE-RJ/2017) Quando um processo aguarda por um recurso que nunca estará disponível ou mesmo um evento que não ocorrerá, acontece uma situação denominada deadlock (ou como alguns autores denominam: impasse ou adiamento indefinido). Para que um deadlock ocorra, quatro condições são necessárias. Uma delas tem a seguinte definição: "cada recurso só pode estar alocado a um único processo em um determinado instante". Assinale a alternativa que apresenta tal condição.

A) Espera circular.

B) Exclusão mútua.

C) Não-preempção.

D) Espera por recurso.

Comentários:

Ou um processo ou outro pode utilizar determinado recurso. O nome é bem intuitivo: exclusão mútua. Ou seja, se um processo A está utilizando um recurso R, um processo B não pode utilizar o recurso R ao mesmo tempo!

Gabarito: B

6. (COMPERVE/UFRN/2018) Sistemas operacionais modernos têm uma gerência de processos e de threads bem definida. Nesse contexto, é correto afirmar:

A) threads de um mesmo processo compartilham a mesma seção de código na memória.

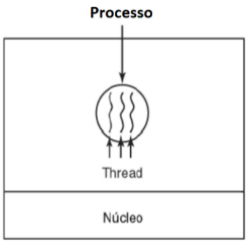
B) threads de um mesmo processo compartilham a mesma seção da pilha na memória.

C) todas as variáveis de uma thread são compartilhadas com as outras threads do mesmo processo.

D) todos os contextos de uma thread são compartilhados com as outras threads do mesmo processo.

Comentários:

A figura abaixo ajuda a entender. Note que existem 3 threads em um único processo, ou seja, 3 fluxos de execução do mesmo código. Por exemplo: um editor de textos com 3 threads é carregado na memória, uma responsável pela correção gramatical, uma pela contagem de palavras e a outra pelo resto (na "vida real" existem bem mais, claro). Então o processo é carregado na memória, no endereçamento X a Y, os fluxos de execução das 3 threads ocorrem "quase em paralelo" e todos eles estarão entre X e Y!



Gabarito: A

7. (FUNDEP/CODEMIG/2018) O escalonamento de processos permite que um computador possa executar diversos programas em pseudoparalelismo, o que viabiliza aspectos como a multiprogramação. Qual entre os algoritmos de escalonamento a seguir seria mais adequado para sistemas de processamento em lote?

A) Primeiro a chegar, primeiro a ser servido.

B) Round Robin.

C) Escalonamento em duas fases.

D) Escalonamento por loteria.

Comentários:

Processamento em lote não possui interação, os processos são colocados lá e podem ser executados na sequência. Quando a natureza das tarefas for adequada a esse tipo de processamento, a grande vantagem é que não há perda de desempenho com chaveamentos de contexto! Desse modo, os algoritmos apresentados o mais adequado seria o "Primeiro a chegar, primeiro a ser servido", pois segue a ordem de chegada e não tem chaveamento de processos com a designação de um quantum a cada um.

Gabarito: A

8. (FUNDEP/CODEMIG/2018) Um dos problemas relacionados ao gerenciamento de um sistema operacional diz respeito ao deadlock, o qual também pode ocorrer em banco de dados. Uma vez que gerenciar o deadlock pode ser uma tarefa que exija muito tempo do processador, a maior parte dos sistemas operacionais não trata desse problema. Em alguns sistemas críticos, entretanto, tratar os deadlocks é uma tarefa importante.

Qual entre as formas de tratamento a seguir se baseia em retirar o recurso do processo?

A) Através de preempção.

B) Revertendo o estado do processo.

C) Matando o processo.

D) Verificando a trajetória do processo.

Comentários:

O exemplo mais comum é retirar um processo do processador após um determinado tempo (quantum) e essa possibilidade de retirar um recurso do processo tem o nome de preempção.

Gabarito: A

9. (COPESE/Câmara de Palmas-TO/2018) Os sistemas operacionais modernos possuem diversos mecanismos para detecção e tratamento de situações de deadlock. Assinale a alternativa que NÃO apresenta um destes mecanismos.

A) O sistema irá escolher criteriosamente um processo e o terminará. Se a situação de deadlock não for resolvida, outros processos serão eliminados até que tudo esteja resolvido.

B) Os recursos são retirados dos processos e entregue aos outros até que o deadlock seja eliminado.

C) Os processos podem ser capazes de detectar um deadlock e voltar ao estado de execução anterior antes de pedir um recurso.

D) Um processo que detém um recurso fica esperando pela liberação de outro recurso, eliminando assim o deadlock.

Comentários:

A única alternativa que não fala em preempção (retirar algum recurso) ou voltar a um estado anterior (que não havia deadlock) é a letra D. Pelo contrário, ainda afirma que detém um recurso e FICA ESPERANDO pela liberação de outro recurso, o que não ajuda em nada a liminar a situação de impasse.

Gabarito: D

10. (FUNDEP/CODEMIG/2018) Um dos problemas relacionados ao gerenciamento de um sistema operacional diz respeito ao deadlock, o qual também pode ocorrer em banco de dados. Uma vez que gerenciar o deadlock pode ser uma tarefa que exija muito tempo do processador, a maior parte dos sistemas operacionais não trata desse problema. Em alguns sistemas críticos, entretanto, tratar os deadlocks é uma tarefa importante.

Qual entre as formas de tratamento a seguir se baseia em retirar o recurso do processo?

A) Através de preempção.

B) Revertendo o estado do processo.

C) Matando o processo.

D) Verificando a trajetória do processo.

Comentários:

Preempção é o ato do S.O. utilizar as interrupções do relógio para retirar a CPU do processo em execução. Ou seja, o processo não pode monopolizar o processador! Quando o sistema não for preemptivo pode ocorrer uma situação denominada *starvation* (inanição). Traduzindo, *starvation* quer dizer "morrer de fome", ou seja, aquele processo que nunca consegue chegar ao processador, fica eternamente aguardando, sempre tem alguém que "fura a fila". Abaixo podemos ver o conceito de *starvation*.

Gabarito: A

11. (COPESE/Câmara de Palmas-TO/2018) Os sistemas operacionais modernos possuem diversos mecanismos para detecção e tratamento de situações de deadlock. Assinale a alternativa que NÃO apresenta um destes mecanismos.

A) O sistema irá escolher criteriosamente um processo e o terminará. Se a situação de deadlock não for resolvida, outros processos serão eliminados até que tudo esteja resolvido.

B) Os recursos são retirados dos processos e entregue aos outros até que o deadlock seja eliminado.

C) Os processos podem ser capazes de detectar um deadlock e voltar ao estado de execução anterior antes de pedir um recurso.

D) Um processo que detém um recurso fica esperando pela liberação de outro recurso, eliminando assim o deadlock.

Comentários:

A) APRESENTA - Após uma escolha criteriosa, matar um processo que está "trancando a rua" pode resolver a situação!

B) APRESENTA - Se os recursos que estavam sendo usados e não estavam sendo "largados" por um processo, forem retirados, pode ser que o *deadlock* seja eliminado.

C) APRESENTA - Se os processos conseguirem detectar um deadlock e voltar ao estado de execução anterior antes de pedir um recurso, é possível eliminar o *deadlock*.

D) NÃO APRESENTA - O simples fato de ficar esperando esperando não resolve nada! Aí vai depender da sorte!

Gabarito: D

12. (UFCG/UFCG/2019) Deadlocks (impasses) podem ocorrer em sistemas operacionais, bancos de dados e outros sistemas concorrentes. Leia as assertivas abaixo e marque a alternativa correta.

I- Um conjunto de processos estão em condição de deadlock se cada processo no conjunto estiver aguardando um evento que apenas outro processo no conjunto cause.

II- Um deadlock ocorre se e somente se as quatro condições de Coffman forem satisfeitas.

III- Um deadlock pode ocorrer ao utilizar somente recursos não-preemptivos.

IV- Uma forma de evitar deadlocks é garantir que a condição de posse-e-espera não ocorra.

V- Uma forma de evitar deadlocks é garantir que a condição de preempção não ocorra.

A) Somente I está correta.

B) Somente I e II estão corretas.

C) Somente I e III estão corretas.

D) Somente I, II e IV estão corretas.

E) I, II, III, IV e V estão corretas.

Comentários:

I- CORRETA - Fica um dependendo do outro, um "abraço da morte".

II- CORRETA - Exato, são elas: condição de exclusão mútua, condição de posse e espera, ausência de preempção e condição de espera circular.

III- INCORRETA - Além de recursos não preemptivos (ausência de preempção), existem as outras três condições de Coffman: condição de exclusão mútua, condição de posse e espera e condição de espera circular, para que ocorra um *deadlock*.

IV- CORRETA - Condição de posse-e-espera é uma das condições de Coffman. Basta que uma delas não ocorra para não ocorrer o *deadlock*.

V- INCORRETA - Na verdade deve-se garantir que a condição de preempção ocorra!

Gabarito: D

13. (Quadrix/CRA-PR/2019) A respeito do gerenciamento de processos e do gerenciamento de memória nos sistemas operacionais, julgue o item.

Embora os sistemas operacionais executem diversas operações de processo, como, por exemplo, criar e suspender um processo, eles não são capazes de alterar a prioridade de um processo.

Comentários:

É possível alterar a prioridade de um processo sim! Inclusive um comando bastante utilizado no Linux para isso é o "renice" (alteração da prioridade pelo usuário). Se o usuário pode, é claro que o S.O. também pode!

Gabarito: Errado

14. (IF-PA/IF-PA/2019) Em relação à gerência de processo, marque a alternativa CORRETA:

A) o processo é um programa em no estado de pronto.

B) os estados do processo são (execução, pronto, bloqueado ou espera).

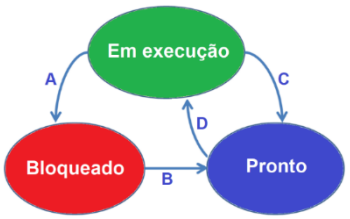
C) a thread permite que apenas uma execução ocorra no mesmo ambiente do processo.

D) os sinais são mecanismos que permitem notificar o sistema operacional de eventos gerados pelo processador.

E) escalonamento é a escolha do processo, em estado de execução.

Comentários:

(A) Processo é um programa em execução; (B) Melhor ver na figura abaixo; (C) Thread permite que fluxos de execução em um mesmo processo ocorram quase que em paralelo; (D) Interrupções são mecanismos que permitem notificar o sistema operacional de eventos gerados pelo processador; (E) escalonamento é a escolha do processo, em estado de pronto.



Obs.: Bloqueado ou em espera é a mesma coisa!

Gabarito: B

15. (UFCG/UFCG/2019) Com relação a inanição (starvation) em sistemas multitarefa, escolha a alternativa INCORRETA.

A) Starvation ocorre quando um processo nunca é executado porque outros processos (de maior prioridade, por exemplo) sempre são executados.

B) Starvation pode ocorrer por falhas no algoritmo de agendamento.

C) Uma forma de evitar starvation é utilizar uma política de alocação first-come, first-served, de forma que os primeiros que chegarem, serão os primeiros a serem atendidos.

D) Algoritmos modernos de agendamento, que utilizam prioridades em processos, não possuem estratégias para impedir starvation.

E) Starvation pode ocorrer em um algoritmo de agendamento que sempre executa os trabalhos com menor tempo de execução primeiro.

Comentários:

A) CORRETA - Starvation é "morrer de fome", quando um processo fica esperando "eternamente" e nunca é executado.

B) CORRETA - Falhas no algoritmo de agendamento podem fazer com que um processo nunca seja executado.

C) CORRETA - A política de alocação FCFS não dá prioridades diferentes, assim, em algum momento o processo será executado (na ordem da fila).

D) INCORRETA - Algoritmos modernos de agendamento que utilizam prioridades em processos possuem estratégias para impedir starvation, pois é um problema bem conhecido e normalmente é tratado!

E) CORRETA - Starvation pode ocorrer em um algoritmo de agendamento que sempre executa os trabalhos com menor tempo de execução primeiro, pois sempre pode chegar um processo "pequeno" e vai "furando a fila".

Gabarito: D

16. (FURB/Pref. de Porto Belo-SC/2019) Sobre gerência de processos, verifique as afirmativas a seguir:

I- Para que dois processos sejam considerados em deadlock, devem acatar de forma simultânea a pelo menos 3 das 4 condições: posse e espera, não preempção, exclusão mútua e espera circular.

II- FCFS é considerada a forma de escalonamento mais elementar e se caracteriza por simplesmente acatar as tarefas na sequência em que surgem, sendo um algoritmo não preemptivo.

III- O Round-Robin (RR) é considerado um algoritmo bem adequado para sistemas de tempo compartilhado.

IV- A JVM (Java Virtual Machine) utiliza um algoritmo de escalonamento de threads não preemptivo e baseado em prioridade que seleciona para execução sempre as threads mais antigas.

V- O algoritmo SJF é um caso especial do algoritmo geral de escalonamento por prioridade e sua maior dificuldade em seu uso é a estimativa, a priori, da duração de cada tarefa.

Assinale a alternativa correta:

A) Apenas as afirmativas I, IV e V estão corretas.

B) Apenas as afirmativas I, II e III estão corretas.

C) Apenas as afirmativas II, III, e V estão corretas.

D) Apenas as afirmativas II, IV e V estão corretas.

E) Apenas as afirmativas I, II, III e V estão corretas.

Comentários:

I- INCORRETA - As quatro condições de Coffman devem ser atendidas para ocorrer o deadlock!

II- CORRETA - FCFS é uma fila comum, sem direito a furar a fila! É um algoritmo não preemptivo, ou seja, o processo não é tirado do processador.

III- CORRETA - É considerado bem adequado porque faz um "rodízio", dividindo o uso do processador.

IV- INCORRETA - A JVM emprega um modelo de escalonamento de threads que é influenciado pelas prioridades atribuídas às threads, mas a implementação exata pode variar entre diferentes máquinas virtuais Java.

V- CORRETA - A prioridade é para os processos "menores" (shortest), mas como estimar o tempo de uso do processador de cada processo? É difícil!

Gabarito: C

17. (IF-PA/IF-PA/2019) Em relação à gerência de processo, marque a alternativa CORRETA:

A) o processo é um programa no estado de pronto.

B) os estados do processo são (execução, pronto, bloqueado ou espera).

C) a thread permite que apenas uma execução ocorra no mesmo ambiente do processo.

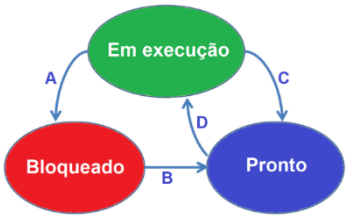
D) os sinais são mecanismos que permitem notificar o sistema operacional de eventos gerados pelo processador.

E) escalonamento é a escolha do processo, em estado de execução.

Comentários:

A) INCORRETA - o processo é um programa em execução. Dependendo do momento pode estar no estado de pronto, em execução ou bloqueado (em espera).

B) CORRETA - Veja a figura abaixo.



C) INCORRETA - a thread é um fluxo de execução dentro do processo. Pode haver várias threads em um único processo.

D) INCORRETA - sinais são mecanismos que permitem que processos ou o kernel de um sistema operacional sejam notificados sobre eventos específicos que ocorrem no sistema. São usados para lidar com eventos assíncronos, como interrupções de hardware, erros de execução ou solicitações de terminação de processo.

E) INCORRETA - escalonamento é a escolha do processo, em estado de pronto, para ir para o estado “em execução”.

Gabarito: B

18. (IDECAN/IF-RR/2020) Os processos inicializados em um sistema operacional podem possuir três estados quanto ao processamento na CPU:

i) pronto;

ii) em execução; e

iii) bloqueado.

Assinale a alternativa que contém o responsável pelo gerenciamento e o controle dos estados de cada processo.

A) Thread

B) Escalonador

C) Memória

D) Arquivos

E) Dispositivos de Entrada e Saída

Comentários:

Escalonador (*scheduler*): rotina responsável por determinar a ordem de execução dos processos. Ele gerencia a alocação de recursos da CPU entre os vários processos concorrentes em um sistema.

Gabarito: B

19. (IDECAN/IF-RR/2020) Em sistemas operacionais, há o conceito "É uma instância de um programa em execução, incluindo as variáveis". Assinale a alternativa ao que ele se refere:

A) Thread

B) Sistema operacional

C) Multiprogramação

D) Escalonador

E) Processo

Comentários:

Processo é simplesmente uma instância de um programa em execução, incluindo os valores correntes dos registradores (PC, IR, entre outros) e das variáveis (ex.: soma, total, em um programa que realiza cálculos). Cada processo pensa que está "sozinho no mundo" e executa em um processador (CPU) virtual, mas sabemos que na prática o processador alterna de um processo para outro. Essa possibilidade de alternância entre processos é conhecida como multiprogramação ou multitarefa.

Gabarito: E

20. (IBADE/Pref. de Vila Velha-ES/2020) O Sistema Operacional XYZ utiliza o algoritmo de alocação circular (Round-Robin) para alocação de processos. O quantum de tempo é de 10 ms. Considere que os processos P1, P2 e P3 entram na fila de processos prontos em t = 0. A tabela mostra a ordem de chegada e a duração de cada processo:

Ordem de chegada: 1° Processo: P1 Duração(ms): 20

Ordem de chegada: 2° Processo: P2 Duração(ms): 4

Ordem de chegada: 3° Processo: P3 Duração(ms): 6

Desprezando-se o tempo necessário para a troca de contexto, determine o tempo médio de espera:

A) 10 ms.

B) 20 ms.

C) 11,3 ms.

D) 13,4 ms.

E) 15,1 ms.

Comentários:

Como a troca de contexto não será considerada, vamos analisar o quantum = 10ms e a ordem de execução abaixo.

Processo	Executou / Esperou				Total da espera
P1	Executou 10ms	Esperou 4ms	Esperou 6ms	Executou 10ms	10ms
P2	Esperou 10ms	Executou 4ms			10ms
P3	Esperou 10ms	Esperou 4ms	Executou 6ms		14ms

Podemos ver que a cada momento (coluna na planilha) apenas um processo utiliza o processador, enquanto os outros esperam. Depois foi só somar o total de espera de cada processo, sendo que a soma das esperas deu 34ms. 34/3 = 11,3ms de média.

Gabarito: C

21. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

O processo é um conjunto de instruções originário de uma chamada. Logo, ele não pode criar um ou mais processos.

Comentários:

Um processo é um programa em execução. É possível criar um processo a partir de outro. Ex.: a partir de um shell (processo) é possível executar um script (outro processo).

Gabarito: Errado

22. (SELECON/EMGEPRON/2021) Os atuais sistemas operacionais empregam um recurso por meio do qual as aplicações são executadas em áreas independentes, possibilitando, no caso de um funcionamento anormal de uma delas, que esta possa ser finalizada, mantendo as demais em processamento normal. Esse recurso é denominado multitarefa:

A) compartilhada

B) distributiva

C) preemptiva

D) otimizada

Comentários:

Sistema multitarefa preemptivo é um sistema que possibilita a execução de mais de um processo ao mesmo tempo geralmente é preemptivo, ou seja, em algum momento o S.O. retira o processador do processo e coloca outro no lugar.

Gabarito: C

23. (CESGRANRIO/Caixa/2021) A sincronização entre processos concorrentes é fundamental para garantir a confiabilidade dos sistemas multiprogramáveis. Um mecanismo de sincronização simples, que permite implementar a exclusão mútua sem a deficiência da espera ocupada (busy wait), é o

A) deadlock

B) mutual lock

C) escalonamento binário

D) buffer contador

E) semáforo mutex

Comentários:

Mutex: versão simplificada do semáforo, quando não for necessário "contar", utilizando-se apenas os estados "livre" ou "ocupado". Consequentemente é necessário apenas um bit para representá-lo, mas na prática geralmente é utilizado um valor inteiro. Quando um processo precisa entrar na região crítica, ele chama *mutex_lock*, que será bem-sucedida se a região estiver livre. Caso contrário, o processo ficará bloqueado até que o processo que estiver na região crítica saia (*mutex_unlock*).

Gabarito: E

24. (CETAP/SEPLAD-PA/2021) Em um sistema operacional, quais das transições de estado listadas nas alternativas a seguir não é uma transição possível?

A) Do estado de pronto para executando.

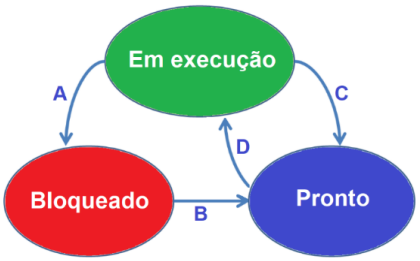
B) Do estado de pronto para bloqueado.

C) Do estado de executando para pronto.

D) Do estado de executando para bloqueado.

Comentários:

Abaixo podemos ver todas as transições possíveis. Não são possíveis do estado pronto para bloqueado, nem do estado bloqueado para executando ("em execução").



Gabarito: B

25. (AOCP/ITEP-RN/2021) Alguns problemas podem ocorrer durante o funcionamento de um sistema operacional. Quando se trata de processos, o deadlock é um dos problemas mais conhecidos. Qual das alternativas a seguir caracteriza corretamente um deadlock?

A) O usuário do sistema tem a sua conta bloqueada após esgotar o limite das tentativas de login no processo de autenticação.

B) Uma falha no funcionamento de um processo dentro do espaço de usuário acaba corrompendo um arquivo em edição.

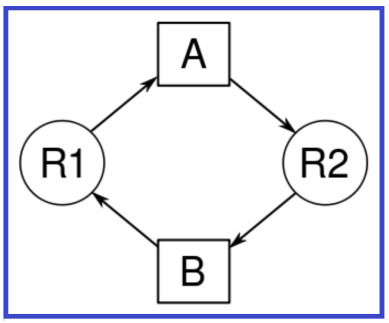
C) Um funcionamento anômalo de um dos componentes de energia causa a interrupção do processador, resultando na falha de processos vitais do sistema operacional e, consequentemente, no seu travamento.

D) A quantidade de processos em execução, devido ao grande número de aplicações executadas pelo usuário, causa esgotamento da memória RAM, gerando lentidão no sistema.

E) A execução dos processos nunca termina, ocupando os recursos do sistema a ponto de impedir a inicialização de outras tarefas.

Comentários:

Abaixo vemos uma representação de deadlock. Agora imagine em uma escala de centenas de processos/recursos em uma espera circular. Isso poderia acarretar em uma situação em que a execução dos processos nunca termina, com a ocupação dos recursos do sistema a ponto de impedir a inicialização de outras tarefas.



Gabarito: E

26. (IDECAN/SEFAZ-RR/2023) As interrupções do sistema operacional atuam como auxiliares na interação entre camadas de software de entrada e saída. Selecione a alternativa que ocorre

quando várias tarefas concorrem para a utilização de um mesmo recurso, em um sistema operacional.

A) Thread

B) Segmentação

C) DeadLock

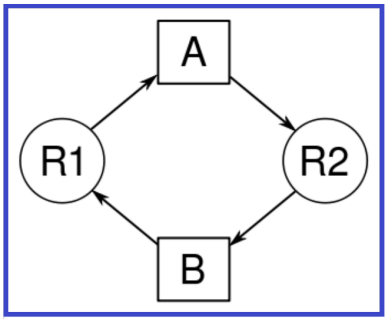
D) Multithreading

E) Swapping

Comentários:

As quatro condições (de Coffman) que devem ser verdadeiras para que ocorra um *deadlock* são:

1. Condição de exclusão mútua: cada recurso ou está correntemente atribuído a exatamente um processo ou está disponível;
2. Condição de posse e espera: os processos que possuem recursos garantidos anteriormente podem solicitar novos recursos (um acumulador de recursos!);
3. Ausência de preempção: os recursos garantidos não podem ser retirados à força de um processo;
4. Condição de espera circular: um encadeamento circular de dois ou mais processos, cada um esperando por um recurso mantido pelo próximo do encadeamento:



Gabarito: C

27. (UFMA/UFMA/2023) Em sistemas operacionais, o "algoritmo do Banqueiro", desenvolvido por Edsger Dijkstra é utilizado para:

A) encerrar deadlocks

B) resolver deadlocks

C) recuperar-se de deadlocks

D) iniciar deadlocks

E) evitar deadlocks

Comentários:

Algoritmo do banqueiro (Banker's algorithm): utilizado para evitar deadlock em sistemas operacionais. Ele foi proposto por Edsger Dijkstra e é utilizado em sistemas que têm um número fixo de recursos e processos concorrentes que solicitam e liberam esses recursos.

Gabarito: E

Lista de Questões - CEBRASPE

1. (CEBRASPE/ABIN/2010) No contexto de sistemas operacionais, semáforos são tipos de variáveis que podem ser verificadas e alteradas em instruções atômicas, ou seja, sem possibilidades de interrupções. Esse tipo de variável é empregado em tarefas como o compartilhamento de recursos entre processos.

2. (CEBRASPE/STF/2013) Em um algoritmo de escalonamento FIFO, os processos são executados na mesma ordem que chegam à fila. Quando um processo do tipo cpu-bound está na frente da fila, todos os processos devem esperá-lo terminar seu ciclo de processador.

3. (CEBRASPE/TRE-PI/2016) A respeito das características do algoritmo de escalonamento SPF (shortest process first), assinale a opção correta.

A) Os processos são executados na ordem em que chegam à fila de espera e executados até o final, sem nenhum evento preemptivo.

B) No SPF, um processo recém-chegado e em espera, cujo tempo estimado de execução completa seja menor, provoca a preempção de um processo em execução que apresente tempo estimado de execução completa maior.

C) O SPF favorece processos longos em detrimento dos mais curtos. Estes, ao chegarem à fila de espera, são obrigados a aguardar a conclusão dos processos longos que já estiverem em andamento, para, então, entrar em execução.

D) Os processos são despachados na ordem em que são colocados em espera e recebem uma quantidade limitada de tempo do processador para execução; além disso, são interrompidos caso sua execução não se conclua dentro do intervalo de tempo delimitado.

E) O escalonador seleciona o processo que estiver à espera e possuir o menor tempo de execução estimado e o coloca em execução até a sua conclusão.

4. (CEBRASPE/TRE-TO/2017) Considerando o contexto de gerenciamento de processos dos sistemas operacionais, assinale a opção que apresenta a estrutura de dados responsável por habilitar o sistema operacional a localizar e acessar rapidamente o bloco de controle de processo (PCB) de um processo.

A) árvore de processos.

B) lista de bloqueados.

C) tabela de processo.

D) região de pilha.

E) lista de prontos.

5. (CEBRASPE/TRF1/2017) Na técnica denominada escalonamento de processos, o sistema operacional mantém parte do espaço de endereçamento de um processo na memória

principal e parte em dispositivo de armazenamento secundário, realizando trocas de trechos de código e de dados entre eles, de acordo com a necessidade.

6. (CEBRASPE/STJ/2018) Em relação aos fundamentos de sistema operacional, julgue o item a seguir.

   Um processo existente no sistema operacional pode ter um número zero de processos-pai.

7. (CEBRASPE/MPE-PI/2018) Julgue o item a seguir, acerca de sistemas operacionais.

   Uma das causas de deadlocks em sistemas operacionais é a disputa por recursos do sistema que podem ser usados apenas por um processo de cada vez.

8. (CEBRASPE/SLU-DF/2019) Em relação aos microcomputadores, julgue o item a seguir.

   Uma das características dos sistemas preemptivos é o fato de eles serem monotarefa.

9. (CEBRASPE/Min. da Economia/2020) Julgue o próximo item, relativos a sistemas operacionais.

   No sistema operacional, o bloco de controle de processo (BCP) representa e guarda informações associadas a um processo, como, por exemplo, o seu estado pronto ou em execução.

10. (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.

11. (CEBRASPE/SEFAZ-CE/2021) Julgue o próximo item, acerca de conhecimentos de informática.

    O gerenciamento de processos é responsável por executar simultaneamente vários processos nos sistemas operacionais Windows e Linux, por meio do compartilhamento de tempo e processadores.

12. (CEBRASPE/SEFIN de Fortaleza-CE/2023) Acerca de primitivas de sincronização e deadlocks, julgue o próximo item.

    No ambiente de programação, um semáforo é um sinal emitido pelo hardware para que o sistema operacional possa executar um laço com o objetivo de evitar um deadlock.

GABARITO

GABARITO

1. Certo
2. Certo
3. Letra E
4. Letra C
5. Errado
6. Certo
7. Certo
8. Errado
9. Certo
10. Letra D
11. Certo
12. Errado

Lista de Questões - FGV

1. (FGV/TJ-RO/2015) Considere o trecho a seguir no contexto de sistemas operacionais.

   "Um conjunto de processos está num bloqueio perpétuo quando cada processo do conjunto está esperando por um evento que apenas outro processo do conjunto pode causar."

   A situação descrita é típica da ocorrência de um:

A) timeout;

B) deadlock;

C) timestamp;

D) system halt;

E) pipeline.

2. (FGV/IBGE/2016) Jonas, Analista de Suporte Operacional do IBGE, realizou uma análise minuciosa dos processos e threads do servidor que ele mantém. Durante a análise, Jonas identificou que três processos estavam na lista de espera por um recurso compartilhado. Além disso, Jonas também identificou uma situação inusitada: um desses processos nunca conseguia executar sua região crítica e, por conta disso, nunca acessava o recurso compartilhado.

   A situação inusitada encontrada por Jonas é a de:

A) lock;

B) starvation;

C) sincronização condicional;

D) threads;

E) stack

3. (FGV/FUNSAÚDE-CE/2021) Considere um sistema operacional onde o processo P1 precisa dos recursos R1 e R2 para prosseguir com seu processamento. Concomitantemente, há um processo P2, que também precisa dos recursos R1 e R2. A situação em que, simultaneamente, o processo P1 detém o recurso R1 e espera pelo recurso R2 e o processo P2 detém o recurso R2 e espera pelo recurso R1, é denominada

A) Checkout.

B) Deadlock.

C) Livelock.

D) Lockdown.

E) Shutdown.

4. (FGV/Prefeitura de Manaus-AM/2022) Sobre o escalonamento de processos, analise as afirmativas abaixo:

I. O algoritmo Round Robin é preemptivo e baseado em quantum de tempo para cada processo, de forma rotativa.

II. O escalonamento não preemptivo é o mais indicado para ambientes de tempo compartilhado.

III. Em escalonadores não preemptivos, um processo permanece em execução tanto quanto possível, só saindo no caso de término de execução, ou quando executa uma instrução que ocasione uma mudança para um estado de espera.

Está correto apenas o que se afirma em

A) I.

B) II.

C) III.

D) I e II.

E) I e III.

5. (FGV/TJ-SE/2023) A rotina do Sistema Operacional (SO) que tem como principal função implementar os critérios da política de escalonamento é:

A) dispatcher;

B) preempção;

C) tempo de resposta;

D) processamento batch;

E) escalonador (scheduler).

6. (FGV/TJ-SE/2023) A política de escalonamento é a base da gerência do processador. Ryu é analista de sistemas e sabe que as características de cada Sistema Operacional (SO) determinam quais são os principais aspectos para a implementação de uma política de escalonamento adequada. Brevemente Ryu adotará, em seu projeto de SO, o critério de escalonamento que representa o número de processos executados em um determinado intervalo de tempo.

Em seu projeto de SO, Ryu deve utilizar o critério de escalonamento:

A) throughput;

B) tempo de espera;

C) tempo de turnaround;

D) tempo de processador;

E) utilização do processador.

7. (FGV/TJ-SE/2023) As propriedades de cada sistema operacional determinam quais são os principais aspectos para a implementação de uma política de escalonamento adequada.

   A política de escalonamento classificada como preemptiva tem a seguinte característica:

A) sistemas operacionais que implementam escalonamento com preempção são mais simples, contudo, possibilitam políticas de escalonamento menos flexíveis;

B) foi a primeira política de escalonamento implementada nos sistemas monoprogramáveis, onde predominava tipicamente o processamento batch;

C) quando um processo está em execução, nenhum evento externo pode ocasionar a perda do uso do processador;

D) o processo sai do estado de execução caso termine seu processamento ou execute instruções do próprio código que ocasionem uma mudança para o estado de espera;

E) o sistema operacional pode interromper um processo em execução e passá-lo para o estado de pronto, com o objetivo de alocar outro processo na UCP.

8. (FGV/TJ-SE/2023) Para um usuário interagir com um computador sem o Sistema Operacional (SO), ele deve conhecer profundamente diversos detalhes sobre o hardware do equipamento. Um técnico de programação de sistemas está testando um SO e verificou que se trata de uma arquitetura multiprogramada na qual vários processos são executados de forma concorrente.

   Nesse contexto, o algoritmo de escalonamento que seleciona o processo que tiver o menor tempo de processador ainda por executar é o:

A) não preemptivo;

B) por prioridade;

C) circular;

D) First-In-First-Out (FIFO);

E) Shortest-Job-First (SJF).

9. (FGV/DPE-RS/2023) Ana trabalha na empresa X, que presta serviço de Tecnologia da Informação à DPE/RS. Durante o funcionamento do site da Defensoria, notou a ocorrência de deadlock. Ana iniciou uma investigação para identificar a tarefa que ocasionou o deadlock. A atividade executada pela aplicação que ocasionou o deadlock foi:

A) um processo de update da aplicação que corrompeu um dado no banco de dados, impedindo o acesso ao banco;

B) o excesso de processos enviados para a execução em memória principal, causando lentidão;

C) a execução de alguns processos, em background, que não liberavam recursos para a inicialização de outras tarefas;

D) o uso compartilhado dos recursos causando lentidão ao sistema para o controle de alterações;

E) o bloqueio de um usuário em virtude de ter esgotado a quantidade de erro de senha durante o processo de autenticação.

GABARITO

GABARITO

1. Letra B
2. Letra B
3. Letra B
4. Letra E
5. Letra E
6. Letra A
7. Letra E
8. Letra E
9. Letra C

Lista de Questões - FCC

1. (FCC/Câmara Municipal-SP/2014) No escalonamento usando o algoritmo Round-Robin,

A) o escalonador seleciona o processo à espera com o menor tempo de execução estimado até a conclusão, reduzindo o tempo médio de espera, mas aumentando a variância dos tempos de resposta.

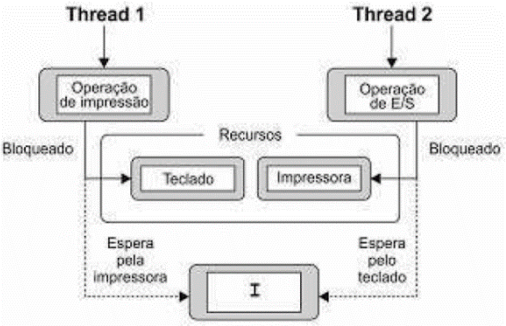
B) processos são despachados na ordem FIFO (First-in-First-Out), mas recebem uma quantidade limitada de tempo de processador denominada quantum.

C) a prioridade de cada processo é uma função não apenas do seu tempo de serviço, mas também do tempo que passou esperando pelo serviço.

D) o escalonador ajusta dinamicamente o comportamento do processo, de tal forma que o próximo processo a obter o processador seja aquele que chegar à frente da fila de nível mais alto, que não estiver vazia, na rede de filas.

E) o processo que tem o prazo de execução mais curto é favorecido, medindo a diferença entre o tempo que um processo requer para finalizar e o tempo restante até atingir o seu prazo final.

2. (FCC/DPE-RS/2017) Considere a figura abaixo.



Do ponto de vista do sistema operacional, a situação indica que a caixa I deve ser preenchida com?

A) starvation.

B) multithreading.

C) superthreading.

D) deadlock.

E) hyperthreading.

3. (FCC/DPE-RS/2017) Dentre as políticas de escalonamento de processos a seguir, a que apresenta maior probabilidade de ocasionar o starvation é a

A) Round Robin.

B) de tempo compartilhado.

C) First In First Out.

D) preemptiva.

E) não preemptiva.

4. (FCC/DPE-AM/2018) Em um sistema operacional típico, os estados de um processo são:

A) Ativo, Desocupado, Finalizado e Executando.

B) Bloqueado, Desbloqueado, Ativo e Suspenso.

C) Executando, Bloqueado e Pronto.

D) Parado, Ocupado, em Execução e Finalizado.

E) Pronto, Terminado, Ativo e Processando.

5. (FCC/DPE-AM/2018) Em um sistema operacional de computador, três processos estão na seguinte situação:

   – o processo P1 tem a posse do recurso R1.

   – o processo P2 tem a posse do recurso R2.

   – o processo P3 tem a posse do recurso R3.

   O processo P1 solicita o recurso R2, o processo P2 solicita o recurso R3, e o processo P3 solicita o recurso R1. Sobre essa situação, é correto afirmar que

A) não haverá deadlock, pois o processo P1 não solicitou o recurso R3.

B) tem-se uma condição de deadlock.

C) não haverá deadlock, pois o processo P3 não solicitou o recurso R2

D) só ocorrerá deadlock caso P1 solicite o recurso R3, P2 solicite o recurso R1 e P3 solicite o recurso R2.

E) não haverá deadlock, pois o processo P2 não solicitou o recurso R1.

6. (FCC/DPE-AM/2018) Em um sistema operacional típico de um computador, três processos se encontram na seguinte situação:

   – o processo P1 envia uma mensagem ao Processo P2.

   – o processo P2, ao receber a mensagem de P1, responde essa mensagem a P1.

   – o processo P1, ao receber a mensagem de resposta de P2, responde a P2 com uma nova mensagem, e assim sucessivamente.

– o processo P3 se encontra bloqueado, situação da qual sairá apenas quando receber uma mensagem do processo P1.

Considerando que a prioridade do processo P3 é menor do que as prioridades dos processos P1 e P2, tem-se que

A) a cada troca de mensagens entre P1 e P2 as respectivas prioridades automaticamente serão reduzidas, e quando elas forem inferior à do processo P3, esse será executado.

B) após passar 1 segundo bloqueado, o relógio de tempo real do sistema operacional automaticamente dará oportunidade de o processo P3 ser desbloqueado.

C) o processo P3 sairá dessa situação assim que uma interrupção qualquer ocorra.

D) ocorrerá uma situação conhecida como impasse (deadlock).

E) ocorrerá uma situação conhecida como inanição (starvation).

GABARITO

GABARITO

1. Letra B
2. Letra D
3. Letra E
4. Letra C
5. Letra B
6. Letra E

Lista de Questões - VUNESP

1. (VUNESP/Pref. de Caieiras-SP/2015) Em um sistema operacional típico, vários processos podem se encontrar no estado "pronto" em um dado instante. A gerência do processador efetua a escolha de qual desses processos receberá o processador. Essa escolha atende a critérios previamente definidos, que fazem parte da política de

A) alocação de memória.

B) escalonamento de processos.

C) minimização do throughput do sistema

D) particionamento da Unidade Central de Processamento.

E) virtualização da memória principal.

2. (VUNESP/Pref. de P. Prudente-SP/2016) Os sistemas operacionais modernos empregam elementos que são fluxos independentes de execução que pertencem a um mesmo processo e que exigem menos recursos de controle do sistema operacional. Esses elementos são denominados

A) Bash.

B) Buffers.

C) Kernel.

D) Semáforos.

E) Threads.

3. (VUNESP/FUNDUNESP/2016) Considere a seguinte situação ocorrendo no ambiente de um sistema operacional no qual um conjunto de processos está sendo executado: "em determinado momento, cada processo está esperando um evento que somente outro processo poderá fazer acontecer". Essa descrição refere-se a

A) deadlock.

B) interrupts.

C) multitasking.

D) throughput.

E) time-sharing.

4. (VUNESP/PC-BA/2018) As versões modernas do sistema operacional Windows, como a versão 10, aparentam ao seu usuário que várias tarefas são executadas ao mesmo tempo. Essa característica é conhecida como

A) monotarefa preemptiva.

B) monotarefa sem preempção.

C) multitarefa preemptiva.

D) multitarefa sem preempção.

E) time-sharing sem preempção.

5. (VUNESP/Câm. Piracicaba-SP/2019) No contexto de gerenciamento de processos e fluxos de execução (threads) em sistemas operacionais, assinale a alternativa em que todos os itens listados sejam privados para cada thread, isto é, não são compartilhados com outros threads no mesmo processo.

A) Arquivos abertos e registradores.

B) Arquivos abertos e pilha.

C) Registradores e pilha.

D) Espaço de endereçamento e registradores.

E) Espaço de endereçamento e arquivos abertos.

6. (VUNESP/Câm. de Piracicaba-SP/2019) Deadlock é um problema potencial em qualquer sistema operacional. Sejam dois processos PA e PB, e dois recursos RX e RY.

   Assinale a alternativa que reflete a ocorrência de um deadlock.

A) PA possui acesso exclusivo a RX e RY, enquanto PB aguarda por acesso exclusivo a RX.

B) PA possui acesso exclusivo a RX, enquanto PB solicita acesso exclusivo a RY.

C) PA possui acesso exclusivo a RX e aguarda por acesso exclusivo a RY, enquanto PB possui acesso exclusivo a RY e aguarda por acesso exclusivo a RX.

D) PA possui acesso exclusivo a RX, enquanto PB possui acesso exclusivo a RY. PA libera RX e então PB requisita acesso exclusivo a RX.

E) PA possui acesso exclusivo a RX, enquanto PB possui acesso exclusivo a RY. PA então libera RX ao mesmo tempo que PB libera RY.

7. (VUNESP/Câmara de Tatuí-SP/2019) Em um sistema operacional, existem diversas situações que poderiam acarretar o deadlock. Algumas estratégias podem ser utilizadas no seu tratamento, a exemplo daquela conhecida como “Algoritmo do Avestruz”, que

A) ignora a sua existência.

B) detecta o deadlock e recupera o sistema.

C) detecta a possibilidade de sua ocorrência e evita que ele ocorra.

D) elimina os processos causadores do deadlock, sem afetar os demais.

E) reinicia o sistema se o travamento ultrapassar um tempo predefinido.

8. (VUNESP/Prefeitura de Pindamonhangaba-SP/2023) O deadlock é um fenômeno que pode ocorrer em sistemas operacionais. Assinale a alternativa correta relacionada a ele.

A) A sua ocorrência se manifesta em computadores com um único processador, mas não ocorre em computadores com mais de um processador.

B) Ele pode ser evitado dinamicamente, pela alocação cuidadosa de recursos entre os processos.

C) Não existe forma de detectá-lo, e quando ele ocorrer, o computador deve ser reinicializado.

D) O deadlock torna a execução dos processos mais lenta devido à atribuição inadequada de prioridades aos processos, mas não interrompe a execução de nenhum deles.

E) Um algoritmo utilizado no seu tratamento é o Algoritmo do Avestruz, que identifica a sua ocorrência e reinicia os processos envolvidos.

GABARITO

GABARITO

1. Letra B
2. Letra E
3. Letra A
4. Letra C
5. Letra C
6. Letra C
7. Letra A
8. Letra B

Lista de Questões - CESGRANRIO

1. (CESGRANRIO/TRANSPETRO/2011) Os Sistemas Operacionais estão sujeitos a um fenômeno denominado deadlock. Para que uma situação de deadlock seja criada, as seguintes condições devem acontecer simultaneamente

A) exclusão mútua (mutual exclusion), monopolização de recursos (hold and wait), não preempção (no preemption) e espera circular (circular wait).

B) exclusão mútua (mutual exclusion), transferência excessiva de páginas (thrashing), superposição de processos (process overlapping) e espera circular (circular wait).

C) transferência excessiva de páginas (thrashing), superposição de processos (process overlapping), monopolização de recursos (hold and wait) e não preempção (no preemption).

D) exclusão mútua (mutual exclusion), monopolização de recursos (hold and wait), superposição de processos (process overlapping) e falha de escalonamento (scheduling fail).

E) transferência excessiva de páginas (thrashing), não preempção (no preemption), espera circular (circular wait) e falha de escalonamento (scheduling fail).

2. (CESGRANRIO/Petrobras/2014) Sejam quatro processos criados no instante de tempo 0 (P1 , P2 , P3 e P4) e que possuam as características descritas na Tabela a seguir.

Processo	Tempo de UCP (u.t.)	Prioridade
P1	11	2
P2	25	1
P3	12	4
P4	8	3

Considerando-se um tempo de troca de contexto igual a 4 u.t., qual é o tempo de turnaround de cada processo caso a política de execução de processos seja a SJF?

A) P1-11/ P2- 40/P3 -56/P4-68

B) P1-19/ P2-56/P3 -31/P4-8

C) P1-23/ P2-68 / P3-39/ P4-8

D) P1-27/ P2 -68/P3 - 43/P4-12

E) P1-40/ P2 - 25/P3-35/ P4-52

3. (CESGRANRIO/UNIRIO/2016) Dentre as funções realizadas por Sistemas Operacionais está o escalonamento de processos, que é responsável pela decisão de qual processo deve receber atenção do processador a cada instante. Uma certa política de escalonamento coloca todos os processos em uma fila circular para o atendimento, especificando uma fatia de tempo

(chamada quantum) após a qual o processo em execução é, preemptivamente, suspenso, e o próximo da fila passa a executar. Que nome se dá a essa política de escalonamento?

A) Round-Robin

B) Shortest-Job-First

C) Priority Scheduling

D) First-Come-First-Served

E) Multilevel Queue Scheduling

4. (CESGRANRIO/IBGE/2016) Processos de sistemas operacionais podem se encontrar em um dentre três estados.

   O estado que caracteriza que um processo está adormecido, aguardando a ocorrência de um evento externo, caracterizando a entrega voluntária do uso do processador ao sistema operacional é chamado de

A) bloqueado

B) despacho

C) em execução

D) preemptivo

E) pronto

5. (CESGRANRIO/IBGE/2016) Seja um sistema operacional que implementa multiprogramação e mantém cinco processos simultaneamente na memória. Considere-se ainda que os processos são independentes, com tempo de overhead desprezível (igual a zero). Sabe-se que todos os processos demandam exatos 40% do tempo aguardando a conclusão de operação de entrada e saída.

   Nesse caso, qual é o percentual de utilização do processador?

A) 1,02%

B) 6,25%

C) 60,00%

D) 93,75%

E) 98,98%

6. (CESGRANRIO/Petrobras/2018) A política de escalonamento de processos de um sistema operacional é uma das partes que mais pode influenciar no seu desempenho.

   A estratégia que associa, a cada processo, um valor baseado no tempo em que ele deverá ocupar a CPU e escolhe o de menor valor para a execução é denominada

A) First-Come-First-serve

B) Last-Come-First-Serve

C) Longest-job-first

D) Round Robin

E) Shortest-job-first

7. (CESGRANRIO/Transpetro/2018) A política de escalonamento estabelece os critérios utilizados para selecionar o processo que fará uso do processador.

No escalonamento não preemptivo, quando um processo está em execução,

A) apenas o sistema operacional pode ocasionar a perda do uso do processador.

B) qualquer processo em espera pode ocasionar a perda do uso do processador.

C) qualquer processo pronto pode ocasionar a perda do uso do processador.

D) nenhum evento externo pode ocasionar a perda do uso do processador.

E) nem mesmo o próprio processo pode ocasionar a perda do uso do processador.

8. (CESGRANRIO/Transpetro/2018) Em sistemas operacionais multitarefas e interativos é comum utilizar um algoritmo de escalonamento de processos. Um algoritmo amplamente usado é conhecido como round-robin.

Nesse algoritmo, cada processo

A) vai para uma fila de acordo com sua prioridade. As filas de maior prioridade são executadas primeiro. Cada fila é executada em ordem de chegada. Quando todas as filas são executadas, inicia-se de novo pela fila de maior prioridade.

B) recebe um número. A CPU seleciona aleatoriamente um dos processos em espera e o executa por um tempo fixo. Se esse tempo se esgota, a CPU seleciona aleatoriamente outro processo para substituí-lo.

C) possui um grau de importância que define sua ordem em uma lista de espera. Na sua vez, o processo é executado por um tempo fixo. Se esse tempo é ultrapassado, a CPU dá o controle para o próximo da lista, e o processo que estava sendo executado volta para uma posição à frente de todos os outros processos de menor importância

D) na lista de espera recebe um intervalo de tempo fixo em que é permitido executar. Na sua vez, se esse intervalo é ultrapassado, a CPU dá o controle para o próximo processo da lista, e o processo que estava sendo executado vai para o fim da lista.

E) na fila de espera possui uma prioridade. As prioridades são usadas para calcular o intervalo de tempo que o processo deve ficar na CPU, quanto maior a prioridade, maior o tempo. Quando esse tempo se esgota, a CPU dá o controle para o próximo processo da lista, e o processo que estava sendo executado vai para o fim da lista.

9. (CESGRANRIO/LIQUIGÁS/2018) A gerência do processador visa a otimizar o seu uso a partir do emprego de técnicas de escalonamento de processos. Dentre os critérios adotados para interromper o processo que está em execução, o término da fatia de tempo é amplamente utilizado pelos sistemas operacionais.

   Esse critério é adotado no escalonamento

A) Shortest Job First

B) First In First Out

C) First Come First Served

D) Round Robin (Circular)

E) por Prioridade

10. (CESGRANRIO/UNIRIO/2019) Um sistema operacional deve ter o completo domínio sobre os recursos da máquina. O escalonamento de recursos, o controle de entrada e saída (E/S), a gerência da memória, a gerência do processador, o escalonamento de processos e a segurança são funções que o sistema operacional deve exercer.

   Um conceito fundamental em todos os sistemas operacionais é o processo, que significa

A) basicamente um programa em execução

B) atividades para gerenciar a memória

C) chamadas ao sistema (system calls)

D) um serviço oferecido pelo sistema operacional

E) um controlador das entradas e saídas

11. (CESGRANRIO/UNIRIO/2019) Deadlock, no contexto de um sistema operacional, é uma

A) ação ou função implementada como uma sequência de uma ou mais instruções que são indivisíveis, ou seja, nenhum outro processo pode interromper a operação.

B) situação em que dois ou mais processos não podem continuar suas execuções, visto que cada um deles espera pelos recursos alocados entre eles.

C) situação em que um processo executável é esquecido indefinidamente pelo escalonador; embora esteja apto a executar, nunca é escolhido.

D) seção de código dentro do processo que requisita acesso a recursos compartilhados, e que não deve ser executado se outro processo está na mesma seção de código.

E) condição na qual múltiplos threads ou processos leem e gravam em uma área de dados compartilhada, e o resultado final depende do tempo relativo a cada operação executada.

12. (CESGRANRIO/Caixa/2021) A sincronização entre processos concorrentes é fundamental para garantir a confiabilidade dos sistemas multiprogramáveis. Um mecanismo de sincronização

simples, que permite implementar a exclusão mútua sem a deficiência da espera ocupada (busy wait), é o

A) deadlock

B) mutual lock

C) escalonamento binário

D) buffer contador

E) semáforo mutex

13. (CESGRANRIO/Transpetro/2023) Em um sistema operacional moderno, vários processos devem ser atendidos simultaneamente, dando ao usuário a impressão de estarem rodando simultaneamente. Para isso, é necessário gerenciar o processador por meio de algoritmos de escalonamento, que definem que processo executa e quais ficam esperando, de acordo com diferentes parâmetros.

Um dos critérios que podem ser usados para comparar esses algoritmos é o tempo de turnaround, que conta o tempo

A) entre a submissão de um pedido e a primeira resposta a ele.

B) que o processo realmente passa sendo executado pelo processador.

C) que um processo passa em estado de bloqueio, geralmente aguardando uma operação de E/S ser concluída.

D) total, desde a submissão do processo até a sua conclusão.

E) total que um processo passa na fila de pronto, aguardando para ser executado pelo processador.

GABARITO

GABARITO

1. Letra A
2. Letra C
3. Letra A
4. Letra A
5. Letra E
6. Letra E
7. Letra D
8. Letra D
9. Letra D
10. Letra A
11. Letra B
12. Letra E
13. Letra D

LISTA DE QUESTÕES - MULTIBANCAS

1. (ESAF/CGU/2008) Analise as seguintes afirmações, levando em conta as chamadas de sistemas usadas com semáforos, e assinale a opção verdadeira.

I. A chamada de sistema UP adiciona uma unidade ao valor corrente de um semáforo.

II. Se o valor do semáforo é zero, uma chamada de sistema DOWN não será completada e o processo será suspenso.

III. Quando um processo inicia a execução de uma chamada de sistema UP ou DOWN, nenhum outro processo terá acesso ao semáforo até que o processo complete a execução ou seja suspenso.

A) Apenas I e II são verdadeiras.

B) Apenas I e III são verdadeiras.

C) Apenas II e III são verdadeiras.

D) I, II e III são verdadeiras.

E) I, II e III são falsas.

2. (IADES/PG-DF/2011) O escalonamento de tarefas é uma atividade de processamento realizada pela CPU de um computador. Esta atividade permite executar de forma mais eficiente os processos considerados prioritários para o sistema operacional. Assinale a alternativa que apresenta o escalonamento de tarefas em um computador, utilizado como servidor de arquivos de uma rede.

A) O escalonamento garantido busca atender a demanda da rede, priorizando ações de leitura

B) O algoritmo de escalonamento FIFO (First In, First Out) atua na gravação de arquivos em disco, implementando o conceito de pilha de escalonamento.

C) Os algoritmos de escalonamento preemptivos devem permitir que um processo seja interrompido durante sua execução.

D) O algoritmo de escalonamento de múltiplas filas permite o acesso simultâneo a arquivos e banco de dados disponibilizados na rede.

E) O escalonador de longo prazo seleciona os processos na interface de rede, dando prioridade às ações de I/O (Input/Output).

3. (FUNRIO/IF-PA/2016) Sistemas operacionais compartilham recursos, havendo a possibilidade de deadlocks. A literatura especializada indica quatro condições necessárias para que um deadlock ocorra. O algoritmo de Avestruz utiliza uma estratégia para lidar com deadlocks conhecida como

A) detectar.

B) detectar e recuperar.

C) evitar.

D) ignorar.

E) prevenir.

4. (IESES/IGP-SC/2017) Acerca da gerência de processos dos sistemas operacionais, assinale a alternativa correta:

A) Um conjunto de processos está em estado de deadlock quando todos os processos no conjunto estão esperando por um evento que só pode ser causado por outro processo do conjunto.

B) Em um escalonamento preemptivo, um processo só perde o processador se terminar ou entrar em estado de espera.

C) No algoritmo de escalonamento de processos Round Robin, o escalonador sempre escolhe para execução o processo com menor expectativa de tempo de processamento. Esse algoritmo baseia-se no fato de que privilegiando processos pequenos o tempo médio de espera decresce.

D) Starvation é uma situação que não pode ocorrer quando um sistema operacional provê prioridades a processos.

5. (CONSULPLAN/TRE-RJ/2017) Quando um processo aguarda por um recurso que nunca estará disponível ou mesmo um evento que não ocorrerá, acontece uma situação denominada deadlock (ou como alguns autores denominam: impasse ou adiamento indefinido). Para que um deadlock ocorra, quatro condições são necessárias. Uma delas tem a seguinte definição: "cada recurso só pode estar alocado a um único processo em um determinado instante". Assinale a alternativa que apresenta tal condição.

A) Espera circular.

B) Exclusão mútua.

C) Não-preempção.

D) Espera por recurso.

6. (COMPERVE/UFRN/2018) Sistemas operacionais modernos têm uma gerência de processos e de threads bem definida. Nesse contexto, é correto afirmar:

A) threads de um mesmo processo compartilham a mesma seção de código na memória.

B) threads de um mesmo processo compartilham a mesma seção da pilha na memória.

C) todas as variáveis de uma thread são compartilhadas com as outras threads do mesmo processo.

D) todos os contextos de uma thread são compartilhados com as outras threads do mesmo processo.

7. (FUNDEP/CODEMIG/2018) O escalonamento de processos permite que um computador possa executar diversos programas em pseudoparalelismo, o que viabiliza aspectos como a multiprogramação. Qual entre os algoritmos de escalonamento a seguir seria mais adequado para sistemas de processamento em lote?

A) Primeiro a chegar, primeiro a ser servido.

B) Round Robin.

C) Escalonamento em duas fases.

D) Escalonamento por loteria.

8. (FUNDEP/CODEMIG/2018) Um dos problemas relacionados ao gerenciamento de um sistema operacional diz respeito ao deadlock, o qual também pode ocorrer em banco de dados. Uma vez que gerenciar o deadlock pode ser uma tarefa que exija muito tempo do processador, a maior parte dos sistemas operacionais não trata desse problema. Em alguns sistemas críticos, entretanto, tratar os deadlocks é uma tarefa importante.

Qual entre as formas de tratamento a seguir se baseia em retirar o recurso do processo?

A) Através de preempção.

B) Revertendo o estado do processo.

C) Matando o processo.

D) Verificando a trajetória do processo.

9. (COPESE/Câmara de Palmas-TO/2018) Os sistemas operacionais modernos possuem diversos mecanismos para detecção e tratamento de situações de deadlock. Assinale a alternativa que NÃO apresenta um destes mecanismos.

A) O sistema irá escolher criteriosamente um processo e o terminará. Se a situação de deadlock não for resolvida, outros processos serão eliminados até que tudo esteja resolvido.

B) Os recursos são retirados dos processos e entregue aos outros até que o deadlock seja eliminado.

C) Os processos podem ser capazes de detectar um deadlock e voltar ao estado de execução anterior antes de pedir um recurso.

D) Um processo que detém um recurso fica esperando pela liberação de outro recurso, eliminando assim o deadlock.

10. (FUNDEP/CODEMIG/2018) Um dos problemas relacionados ao gerenciamento de um sistema operacional diz respeito ao deadlock, o qual também pode ocorrer em banco de dados. Uma vez que gerenciar o deadlock pode ser uma tarefa que exija muito tempo do processador, a maior parte dos sistemas operacionais não trata desse problema. Em alguns sistemas críticos, entretanto, tratar os deadlocks é uma tarefa importante.

    Qual entre as formas de tratamento a seguir se baseia em retirar o recurso do processo?

A) Através de preempção.

B) Revertendo o estado do processo.

C) Matando o processo.

D) Verificando a trajetória do processo.

11. (COPESE/Câmara de Palmas-TO/2018) Os sistemas operacionais modernos possuem diversos mecanismos para detecção e tratamento de situações de deadlock. Assinale a alternativa que NÃO apresenta um destes mecanismos.

A) O sistema irá escolher criteriosamente um processo e o terminará. Se a situação de deadlock não for resolvida, outros processos serão eliminados até que tudo esteja resolvido.

B) Os recursos são retirados dos processos e entregue aos outros até que o deadlock seja eliminado.

C) Os processos podem ser capazes de detectar um deadlock e voltar ao estado de execução anterior antes de pedir um recurso.

D) Um processo que detém um recurso fica esperando pela liberação de outro recurso, eliminando assim o deadlock.

12. (UFCG/UFCG/2019) Deadlocks (impasses) podem ocorrer em sistemas operacionais, bancos de dados e outros sistemas concorrentes. Leia as assertivas abaixo e marque a alternativa correta.

I- Um conjunto de processos estão em condição de deadlock se cada processo no conjunto estiver aguardando um evento que apenas outro processo no conjunto cause.

II- Um deadlock ocorre se e somente se as quatro condições de Coffman forem satisfeitas.

III- Um deadlock pode ocorrer ao utilizar somente recursos não-preemptivos.

IV- Uma forma de evitar deadlocks é garantir que a condição de posse-e-espera não ocorra.

V- Uma forma de evitar deadlocks é garantir que a condição de preempção não ocorra.

A) Somente I está correta.

B) Somente I e II estão corretas.

C) Somente I e III estão corretas.

D) Somente I, II e IV estão corretas.

E) I, II, III, IV e V estão corretas.

13. (Quadrix/CRA-PR/2019) A respeito do gerenciamento de processos e do gerenciamento de memória nos sistemas operacionais, julgue o item.

Embora os sistemas operacionais executem diversas operações de processo, como, por exemplo, criar e suspender um processo, eles não são capazes de alterar a prioridade de um processo.

14. (IF-PA/IF-PA/2019) Em relação à gerência de processo, marque a alternativa CORRETA:

A) o processo é um programa em no estado de pronto.

B) os estados do processo são (execução, pronto, bloqueado ou espera).

C) a thread permite que apenas uma execução ocorra no mesmo ambiente do processo.

D) os sinais são mecanismos que permitem notificar o sistema operacional de eventos gerados pelo processador.

E) escalonamento é a escolha do processo, em estado de execução.

15. (UFCG/UFCG/2019) Com relação a inanição (starvation) em sistemas multitarefa, escolha a alternativa INCORRETA.

A) Starvation ocorre quando um processo nunca é executado porque outros processos (de maior prioridade, por exemplo) sempre são executados.

B) Starvation pode ocorrer por falhas no algoritmo de agendamento.

C) Uma forma de evitar starvation é utilizar uma política de alocação first-come, first-served, de forma que os primeiros que chegarem, serão os primeiros a serem atendidos.

D) Algoritmos modernos de agendamento, que utilizam prioridades em processos, não possuem estratégias para impedir starvation.

E) Starvation pode ocorrer em um algoritmo de agendamento que sempre executa os trabalhos com menor tempo de execução primeiro.

16. (FURB/Pref. de Porto Belo-SC/2019) Sobre gerência de processos, verifique as afirmativas a seguir:

I- Para que dois processos sejam considerados em deadlock, devem acatar de forma simultânea a pelo menos 3 das 4 condições: posse e espera, não preempção, exclusão mútua e espera circular.

II- FCFS é considerada a forma de escalonamento mais elementar e se caracteriza por simplesmente acatar as tarefas na sequência em que surgem, sendo um algoritmo não preemptivo.

III- O Round-Robin (RR) é considerado um algoritmo bem adequado para sistemas de tempo compartilhado.

IV- A JVM (Java Virtual Machine) utiliza um algoritmo de escalonamento de threads não preemptivo e baseado em prioridade que seleciona para execução sempre as threads mais antigas.

V- O algoritmo SJF é um caso especial do algoritmo geral de escalonamento por prioridade e sua maior dificuldade em seu uso é a estimativa, a priori, da duração de cada tarefa.

Assinale a alternativa correta:

A) Apenas as afirmativas I, IV e V estão corretas.

B) Apenas as afirmativas I, II e III estão corretas.

C) Apenas as afirmativas II, III, e V estão corretas.

D) Apenas as afirmativas II, IV e V estão corretas.

E) Apenas as afirmativas I, II, III e V estão corretas.

17. (IF-PA/IF-PA/2019) Em relação à gerência de processo, marque a alternativa CORRETA:

A) o processo é um programa no estado de pronto.

B) os estados do processo são (execução, pronto, bloqueado ou espera).

C) a thread permite que apenas uma execução ocorra no mesmo ambiente do processo.

D) os sinais são mecanismos que permitem notificar o sistema operacional de eventos gerados pelo processador.

E) escalonamento é a escolha do processo, em estado de execução.

18. (IDECAN/IF-RR/2020) Os processos inicializados em um sistema operacional podem possuir três estados quanto ao processamento na CPU:

i) pronto;

ii) em execução; e

iii) bloqueado.

Assinale a alternativa que contém o responsável pelo gerenciamento e o controle dos estados de cada processo.

A) Thread

B) Escalonador

C) Memória

D) Arquivos

E) Dispositivos de Entrada e Saída

19. (IDECAN/IF-RR/2020) Em sistemas operacionais, há o conceito "É uma instância de um programa em execução, incluindo as variáveis". Assinale a alternativa ao que ele se refere:

A) Thread

B) Sistema operacional

C) Multiprogramação

D) Escalonador

E) Processo

20. (IBADE/Pref. de Vila Velha-ES/2020) O Sistema Operacional XYZ utiliza o algoritmo de alocação circular (Round-Robin) para alocação de processos. O quantum de tempo é de 10 ms. Considere que os processos P1, P2 e P3 entram na fila de processos prontos em t = 0. A tabela mostra a ordem de chegada e a duração de cada processo:

    Ordem de chegada: 1° Processo: P1 Duração(ms): 20

    Ordem de chegada: 2° Processo: P2 Duração(ms): 4

    Ordem de chegada: 3° Processo: P3 Duração(ms): 6

    Desprezando-se o tempo necessário para a troca de contexto, determine o tempo médio de espera:

A) 10 ms.

B) 20 ms.

C) 11,3 ms.

D) 13,4 ms.

E) 15,1 ms.

21. (Quadrix/CRECI-MS/2021) Com relação aos fundamentos dos sistemas operacionais, julgue o item.

    O processo é um conjunto de instruções originário de uma chamada. Logo, ele não pode criar um ou mais processos.

22. (SELECON/EMGEPRON/2021) Os atuais sistemas operacionais empregam um recurso por meio do qual as aplicações são executadas em áreas independentes, possibilitando, no caso de um funcionamento anormal de uma delas, que esta possa ser finalizada, mantendo as demais em processamento normal. Esse recurso é denominado multitarefa:

A) compartilhada

B) distributiva

C) preemptiva

D) otimizada

23. (CESGRANRIO/Caixa/2021) A sincronização entre processos concorrentes é fundamental para garantir a confiabilidade dos sistemas multiprogramáveis. Um mecanismo de sincronização

simples, que permite implementar a exclusão mútua sem a deficiência da espera ocupada (busy wait), é o

A) deadlock

B) mutual lock

C) escalonamento binário

D) buffer contador

E) semáforo mutex

24. (CETAP/SEPLAD-PA/2021) Em um sistema operacional, quais das transições de estado listadas nas alternativas a seguir não é uma transição possível?

A) Do estado de pronto para executando.

B) Do estado de pronto para bloqueado.

C) Do estado de executando para pronto.

D) Do estado de executando para bloqueado.

25. (AOCP/ITEP-RN/2021) Alguns problemas podem ocorrer durante o funcionamento de um sistema operacional. Quando se trata de processos, o deadlock é um dos problemas mais conhecidos. Qual das alternativas a seguir caracteriza corretamente um deadlock?

A) O usuário do sistema tem a sua conta bloqueada após esgotar o limite das tentativas de login no processo de autenticação.

B) Uma falha no funcionamento de um processo dentro do espaço de usuário acaba corrompendo um arquivo em edição.

C) Um funcionamento anômalo de um dos componentes de energia causa a interrupção do processador, resultando na falha de processos vitais do sistema operacional e, consequentemente, no seu travamento.

D) A quantidade de processos em execução, devido ao grande número de aplicações executadas pelo usuário, causa esgotamento da memória RAM, gerando lentidão no sistema.

E) A execução dos processos nunca termina, ocupando os recursos do sistema a ponto de impedir a inicialização de outras tarefas.

26. (IDECAN/SEFAZ-RR/2023) As interrupções do sistema operacional atuam como auxiliares na interação entre camadas de software de entrada e saída. Selecione a alternativa que ocorre quando várias tarefas concorrem para a utilização de um mesmo recurso, em um sistema operacional.

A) Thread

B) Segmentação

C) DeadLock

D) Multithreading

E) Swapping

27. (UFMA/UFMA/2023) Em sistemas operacionais, o "algoritmo do Banqueiro", desenvolvido por Edsger Dijkstra é utilizado para:

A) encerrar deadlocks

B) resolver deadlocks

C) recuperar-se de deadlocks

D) iniciar deadlocks

E) evitar deadlocks

GABARITO

GABARITO

1. Letra D
2. Letra C
3. Letra D
4. Letra A
5. Letra B
6. Letra A
7. Letra A
8. Letra A
9. Letra D
10. Letra A
11. Letra D
12. Letra D
13. Errado
14. Letra B
15. Letra D
16. Letra C
17. Letra B
18. Letra B
19. Letra E
20. Letra C
21. Errado
22. Letra C
23. Letra E
24. Letra B
25. Letra E
26. Letra C
27. Letra E

Gerenciamento de Memória

O componente central de um sistema operacional que determina o local da memória onde deverá ser colocado o código de um novo processo, lido de um arquivo previamente armazenado em um dispositivo de entrada e saída (HD, por exemplo) é denominado gerenciador de memória.

Antes de avançar no assunto é importante conhecer o termo *memory leak* (vazamento de memória), que pode ocorrer em duas situações:

1. Blocos de memória estão alocados e disponíveis para serem utilizados pelo programa, mas não são acessíveis porque a aplicação não tem nenhum ponteiro apontando para essa área de memória. Ou seja, tais blocos de memória não podem ser utilizados nem pelo programa nem por outros processos (ficam alocados e sem acesso!);

2. Blocos de memória possuem dados que poderiam ser liberados por estarem inacessíveis e que, por algum "esquecimento", ainda são referenciados no código. Ou seja, mesmo sem estarem sendo usados, não podem ser liberados.

Os gerenciadores de memória podem ser divididos em duas classes: os que alternam os processos entre a memória principal e o disco durante a execução (*swapping*) e os que não alternam. Nosso foco será na primeira classe, visto que é basicamente o que é utilizado há um bom tempo e o que é cobrado em concurso!

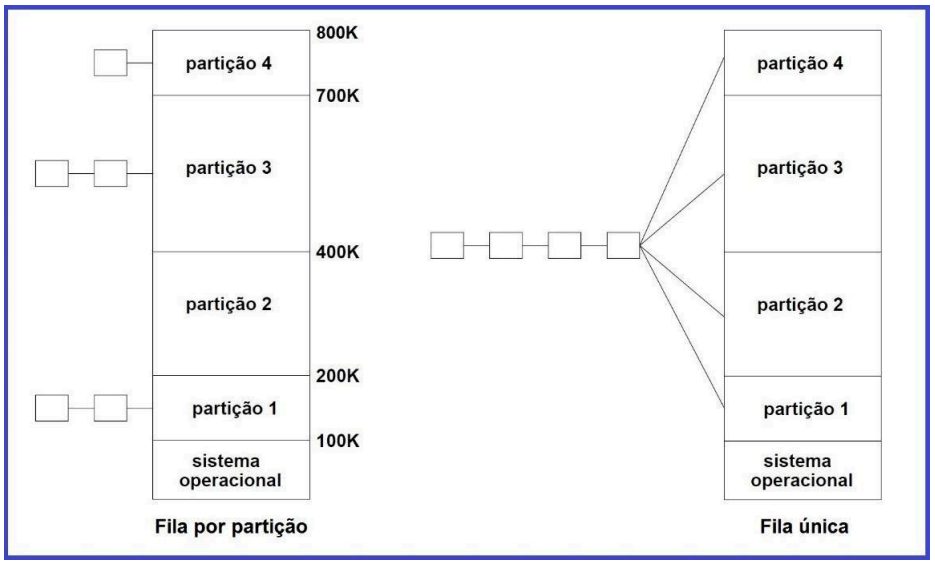
Multiprogramação com Partições Fixas

Quando existe mais de um processo na memória por vez (sistemas multitarefa), alguma forma de organização da memória deve ser estabelecida. A mais simples é a divisão da memória em N partições (possivelmente com tamanhos diferentes, porém de tamanho fixo). Tais partições podem ser estabelecidas na configuração do sistema operacional, por exemplo.

Ao ser inicializado, um processo pode ser colocado em uma fila de entrada para ocupar a menor partição de tamanho suficiente para carregá-lo. Como as partições são fixas, qualquer espaço em uma partição não utilizado pelo processo é desperdiçado, afinal de contas apenas um processo pode ocupar uma partição.

Basicamente pode-se adotar a estratégia de ter uma fila para cada partição (de acordo com o tamanho do processo, para evitar tanto desperdício) ou uma fila única (mais fácil de implementar, mas não evita desperdício. A figura a seguir ajuda a esclarecer (detalhe: se o processo tiver tamanho menor ou igual a 100 KB, ele pode ir para a fila da partição 1 ou 4, se for adotada a estratégia de fila por partição).

Essa estratégia com partições fixas, definidas pelo operador, foi utilizada pelo sistema operacional OS/360 (*mainframes* da IBM) e era chamado de MFT (*Multiprograming with a Fixed number of Task*). Como vimos, ele é simples de se entender e de implementar, e, embora seja considerado obsoleto, cai em prova de concurso!



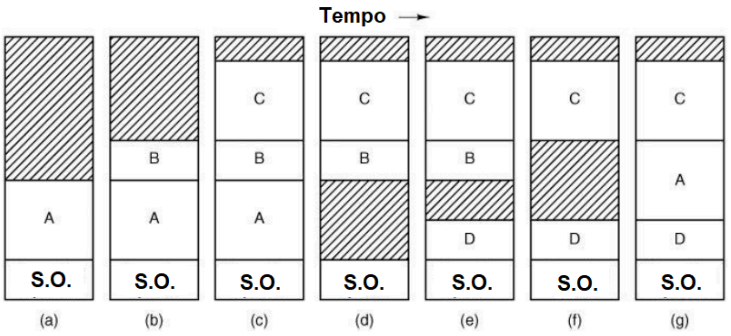
Swapping

Há momentos em que não há memória principal suficiente para todos os processos ativos, de maneira que os processos excedentes devem ser mantidos em disco e trazidos para a memória dinamicamente. Para isso pode ser usada uma das seguintes estratégias:

- Swapping: traz cada processo em sua totalidade, executa-o por algum tempo e o coloca novamente no disco;

- Memória virtual: os programas podem ser executados mesmo estando apenas parcialmente na memória principal.

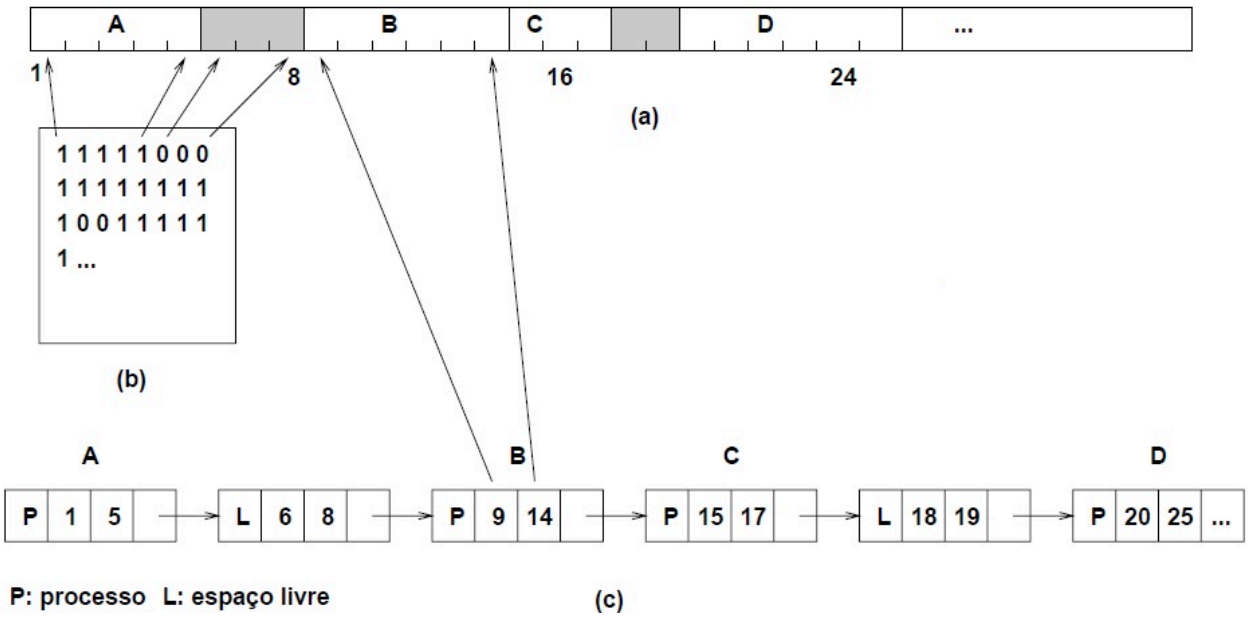
Vamos focar agora no *swapping*...analise a figura abaixo e em seguida vamos comentá-la.



Podemos ver a memória em diferentes instantes de tempo, sendo que o S.O. ocupa uma pequena parte (debaixo) e o restante é destinado para processos do usuário. Inicialmente apenas o processo A foi carregado, em seguida o processo B e depois o C. Quando falamos "carregado" pode ser porque acabou de ser criado ou foi recuperado do disco. Digamos que o processo D precisa ser carregado, porém não há espaço para ele. Então A foi colocado em disco e D foi colocado em seu lugar (na verdade, sobrou espaço). Na sequência foi necessário carregar A, e para isso B teve que ser retirado da memória e colocado em disco.

Na figura podemos ver que fragmentos são criados na medida em que processos são carregados ou retirados da memória. Para evitar isso é possível utilizar a técnica de compactação de memória, que move os processos "para baixo" quando há algum espaço livre, deixando a parte "de cima" sem fragmentos. Por exemplo, na parte (d) da figura os processos B e C seriam deslocados para baixo, "encostando" no S.O. Porém, essa técnica exige muito tempo de CPU e geralmente não é utilizada.

Gerenciamento de memória com mapa de bits: a memória é dividida em unidades de alocação de tamanho fixo, que podem ser desde um pequeno número de palavras até alguns Kilobytes (ex.: 4KB). Para cada unidade de alocação existe um bit no mapa de bits, que é 0 se a unidade estiver livre ou 1 caso esteja ocupada (ou vice-versa, dependendo da implementação). Fica mais fácil entender olhando a figura abaixo:



Na figura vemos:

(a) Um pedaço da memória contendo 4 processos (A, B, C e D). Observamos também quantas unidades de alocação cada processo ocupa: A = 5, B = 6, C = 3 e D = 6. Se a unidade de alocação tiver 4KB, por exemplo, o processo A teria o tamanho entre (16KB + 1 byte) e 20KB, pois não há garantia que o processo ocupará toda a última unidade de alocação (seria muita "sorte");

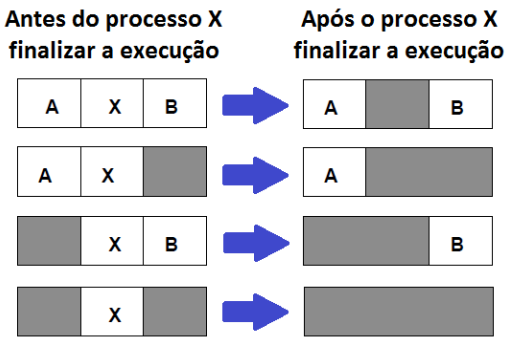
(b) Um mapa de bits correspondente ao pedaço de memória mostrado em (a), ou seja, para cada unidade de alocação ocupada há um bit 1 e para cada unidade de alocação livre existe um bit 0. Você pode contar cinco bits 0 apenas, pois são correspondentes àqueles "quadrados pintados" e ainda pode ver que eles aparecem nas posições 6, 7, 8, 18 e 19. O resto das unidades de alocação está ocupado (bit 1);

(c) Uma lista encadeada representando as mesmas informações mostradas no item (b), da seguinte forma: "P" representa que está ocupado por um processo, o campo seguinte informa a posição inicial, o seguinte a posição final e o último campo é um ponteiro para o próximo nodo da lista. Se estiver livre, no lugar "P" é colocado um "L".

O tamanho de cada unidade de alocação é uma importante característica do projeto. Para unidades de alocação menores tem-se um mapa de bits maior. Entretanto, mesmo com uma unidade de alocação tão pequena como com 4 bytes, 32 bits de memória irão requerer somente 1 bit no mapa (1/32 = 3,1% da memória). Se a unidade de alocação for grande, o mapa de bits será pequeno, mas uma porção considerável da memória pode ser desperdiçada (lembre daquele finalzinho do processo na última unidade de alocação, afinal de contas é difícil ter um processo com um tamanho múltiplo exato da unidade de alocação).

O problema principal do mapa de bits é que quando se decide trazer um processo de N unidades de alocação para a memória, o gerenciador de memória deve pesquisar no mapa de bits uma sequência de N bits 0 consecutivos no mapa. Tal operação é lenta, motivo pelo qual essa estratégia é evitada na prática, porém é cobrada em concurso!

Gerenciamento de memória com listas encadeadas: uma outra maneira de gerenciar a memória é manter uma lista de alocações e segmentos de memória livre, onde um segmento é um processo ou um espaço livre entre dois processos. Podemos voltar à figura mostrada no mapa de bits e verificar que no item (c) há uma lista encadeada de segmentos e em cada nodo podemos ver se aquele "pedaço" está ocupado por um processo (P) ou livre (L). Quando um processo é terminado, ao seu lado podemos ter outros processos ou espaços livres. Quatro situações podem ocorrer ("X" é o processo que vai finalizar sua execução):



Memória Virtual

Um fato notório na evolução da informática é que o tamanho dos programas vem superando a quantidade de memória disponível para carregá-los. Uma solução que geralmente era adotada no passado era a divisão do programa em partes (*overlays*). O *overlay* 0 começa a ser executado primeiro. Quando finaliza, o próximo *overlay* é executado, e assim sucessivamente. Os *overlays* eram mantidos em disco e permutados para a memória pelo sistema operacional.

O grande problema é que a divisão do programa era deixada a cargo do programador, então dependia da habilidade desse profissional. Surgiu então um método que deixa esse particionamento do programa a cargo do sistema operacional: a memória virtual. A ideia básica é que a combinação do tamanho do programa, dados e pilha, pode exceder a quantidade de memória física disponível. Por exemplo, você pode ter uma máquina com 8 GB de memória RAM e ter diversos processos em execução, totalizando 10 GB (2 GB a mais que a memória física).

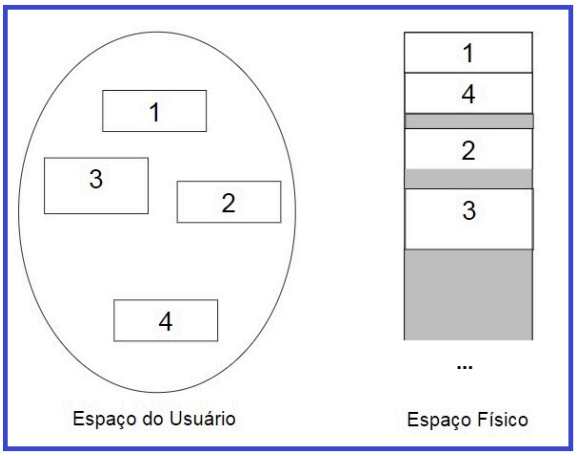
Qual a mágica? O S.O. mantém aquelas partes do programa correntemente em uso na memória principal e o resto no disco. Digamos que um programa de tamanho 50MB esteja executando em

uma máquina que aloque 5 MB de memória por processo, escolhendo-se cuidadosamente quais dos 5 MB será mantido na memória a cada instante, com segmentos sendo permutados entre disco e memória assim que forem necessários. Vamos ver a seguir duas formas de implementar a memória virtual: a segmentação e a paginação.

Segmentação: os programas são divididos em sub-rotinas e estruturas de dados, em segmentos de tamanhos diversos, cada um com o seu próprio espaço de endereçamento. A alocação da memória ocorre de maneira não fixa, ou seja, cada segmento pode ter um tamanho diferente (a alocação depende da lógica do programa). O mapeamento é feito através das tabelas de mapeamento de segmentos e os endereços são compostos pelo número do segmento e um deslocamento dentro do segmento (ex.: segmento 0, deslocamento = 500 bytes).

Cada entrada na tabela mantém o endereço físico do segmento, o tamanho do segmento, se ele está ou não na memória e sua proteção. Para isso, o S.O. mantém uma tabela com as áreas livres e ocupadas da memória e somente segmentos referenciados são transferidos para a memória principal. Nesse modelo ocorre fragmentação externa, pois internamente não "sobra espaço", mas externamente sempre pode ficar um "pedaço" pequeno sem uso, entre dois segmentos.

Imagine a seguinte situação: o usuário executa diversos programas, encerra alguns, abre outros, e assim por diante. Em algum momento existem os processos 1, 2, 3 e 4 no "espaço do usuário", os primeiros três com tamanho semelhante (3,9 KB, 4 KB e 4,2 KB) e o quarto com tamanho 6 KB, além de outros processos. Esses outros processos foram terminados e apenas os quatro ficaram na memória (fora os processos do S.O.). Uma configuração possível seria a seguinte:

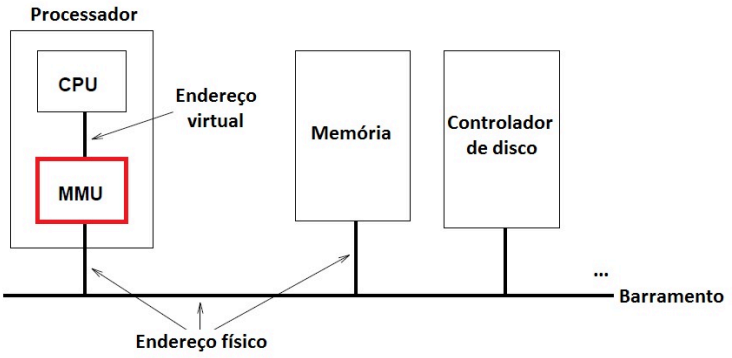


É possível verificar que os processos ocupam segmentos de tamanhos variáveis e existe uma fragmentação externa (fora do segmento), digamos que uma com cerca de 1KB (entre os processos 4 e 2) e uma com cerca de 1,5 KB (entre os processos 2 e 3).

Até pode ser utilizado algum algoritmo para compactação, para unir o 2 com o 4 e o 3 com o 2, eliminando os fragmentos, mas é um custo computacional elevado!

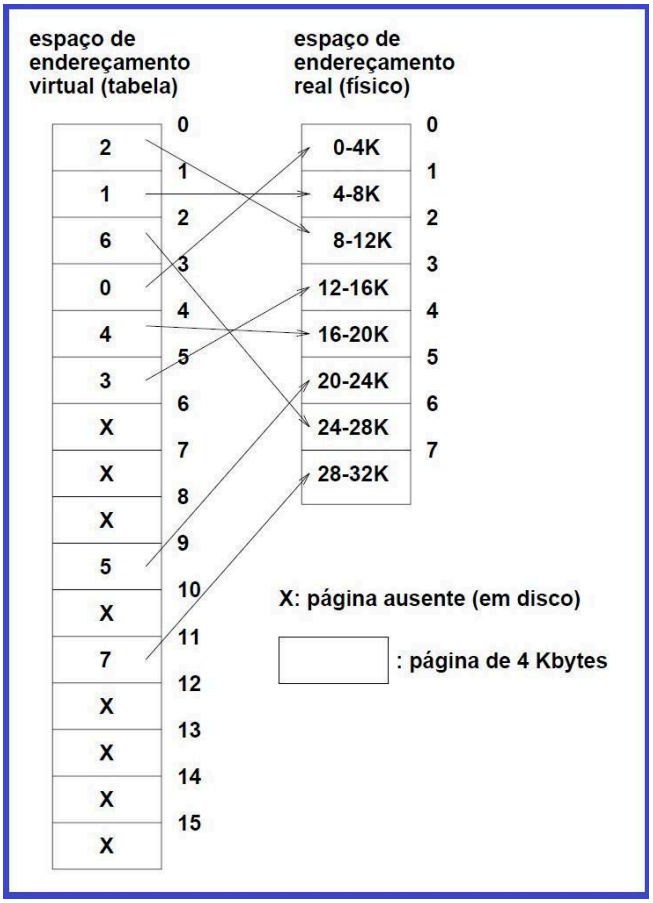
Paginação: técnica utilizada pela maioria dos sistemas com memória virtual. Em qualquer computador existe um determinado conjunto de endereços de memória que programas podem referenciar. Trata-se de endereços virtuais que formam o espaço virtual de endereçamento do processo. Em computadores sem memória virtual, o endereço virtual é colocado diretamente no barramento de memória, uma palavra da memória física com o mesmo endereço é lida ou escrita.

Com o uso da memória virtual, os endereços de memória não vão diretamente para o barramento de memória, eles vão à unidade de gerenciamento de memória (MMU - *Memory Management Unit*), um hardware específico que mapeia os endereços virtuais em endereços da memória física:



Vamos ver um exemplo de um computador que pode gerar endereços de 16 bits (2^{16} = 65536 = 64KB). Esses são os endereços virtuais. Agora imagine que tal computador tenha somente 32KB de memória física, então embora programas de 64KB possam ser escritos, eles não podem ser carregados totalmente para a memória para serem executados. Uma cópia completa do programa deve estar presente no disco e segmentos desse programa podem ser carregados para a memória pelo sistema na medida em que se tornem necessários.

O espaço de endereçamento virtual é dividido em unidades chamadas páginas. As unidades correspondentes na memória física são chamadas quadros (*page frames*). Para não ter erro pense no seguinte: considere uma página como uma tela de pintura que deve ser colocada em um quadro (uma moldura). Assim fica mais fácil para não se confundir, certo? 🙂



As páginas e os quadros são sempre do mesmo tamanho. No exemplo ao lado elas são de 4 KB. Com 64 KB de espaço de endereço virtual e 32 KB de memória física, temos 16 páginas e 8 quadros. As transferências entre a memória e o disco sempre são realizadas em unidades de páginas (4 KB, no nosso exemplo).

Quando um programa tenta acessar o endereço 5000, por exemplo, o endereço virtual 9000 é enviado para a MMU. Como temos páginas de 4 KB (4096 bytes), sabemos que a página 0 possui os endereços 0-4095, a página 1 os endereços 4096-8191, a página 2 os endereços 8192-12287, e assim por diante. A tabela de páginas tem a função de mapear as páginas virtuais em molduras de página física.

A MMU reconhece que o endereço 9000 fica na página 2, que é mapeada para o quadro 6 (veja a figura). A transformação do endereço é realizada e colocada no barramento. A tabela de memória nem sabe da existência da MMU! Apenas sabe de uma requisição para leitura ou escrita

no endereço 9000. Então, a MMU mapeia todo endereço virtual entre 8192 e 12287 (página 2) em endereço físico de 24576 a 28671 (quadro 6).

Agora, digamos que o programa queira acessar o endereço 24576 (exatamente o 1° byte da página 6). Olhando na figura, verificamos que se trata de uma página ausente! A MMU verifica que a página não está mapeada e força a CPU a causar uma interrupção (*trap*) no sistema operacional. Essa interrupção é chamada falta de página (*page fault*). O S.O. remove um quadro, de acordo com um algoritmo (o quadro menos usado, por exemplo), e escreve seu conteúdo de volta no disco. Ele então busca a página referenciada para o quadro liberado, atualiza o mapa, e retorna à instrução interrompida.

Como as páginas são do mesmo tamanho (ex.: 4 KB), é muito provável que a última página de um processo tenha um espaço não utilizado. Imagine um processo com tamanho 9 KB, ele precisa de 3 páginas (2 "completas" + 1 usando apenas 1 KB dos 4 KB). Isso é conhecido como fragmentação interna, pois existem as páginas de tamanho fixo e algumas ficarão sem ser preenchidas (nenhum outro processo pode utilizá-las).

TLB (*Translation Lookaside Buffer*): é uma memória cache especializada que armazena as traduções de endereços virtuais para endereços físicos. Quando um programa faz referência a um endereço virtual, a TLB é consultada para encontrar a correspondência do endereço virtual com o endereço físico correspondente na memória RAM. A TLB acelera o acesso à memória, traduzindo endereços virtuais em endereços físicos de forma eficiente. Sem a TLB, o sistema precisaria consultar a tabela de páginas toda vez que uma referência de memória fosse feita, o que seria muito mais lento do que o acesso direto à TLB.

Quando um programa faz referência a um endereço virtual, a TLB é a primeira a ser consultada. Se houver uma correspondência na TLB ("hit"), o endereço físico é recuperado. Se não houver uma correspondência ("miss" - uma falta), o sistema realiza uma consulta à tabela de páginas, atualiza a TLB com a nova tradução e, em seguida, acessa a memória RAM.

Dirty bit ("bit de sujeira"): trata-se de um marcador que indica se uma página de memória foi alterada desde que foi carregada na memória. É frequentemente utilizado em ambientes de memória virtual, onde partes do programa ou dados são movidos entre a memória RAM e o armazenamento secundário (geralmente disco).

Funcionamento do gerenciamento de memória por paginação:

- Criação do Processo: Quando um processo é criado, o sistema operacional aloca espaço na memória para suas páginas;

- Execução do Processo: Durante a execução do processo, o sistema operacional pode mover páginas entre a memória RAM e a memória secundária (ex.: disco rígido). Isso é realizado para otimizar o uso da memória, mantendo na memória RAM apenas as páginas necessárias para o processo em execução;

- Término do Processo: Quando um processo é finalizado, o sistema operacional libera a memória associada a esse processo, o que pode incluir a liberação de todas as páginas alocadas para o processo.

Thrashing: é um fenômeno que ocorre quando um sistema operacional gasta a maior parte do seu tempo trocando páginas entre a memória RAM e a memória secundária (ex.: disco rígido), em vez de executar efetivamente as instruções dos programas. Ou seja, o sistema fica sobrecarregado com a execução de vários processos, cada um exigindo acesso frequente a páginas de memória que não estão atualmente na memória RAM. O *thrashing* faz com que o desempenho fique extremamente baixo.

Para evitar o *thrashing*, os sistemas operacionais utilizam algoritmos de substituição de página eficientes, como por exemplo o algoritmo de segunda chance, para garantir que as páginas mais relevantes para a execução dos processos estejam sempre presentes na memória RAM. A otimização do tamanho da carga de trabalho e a gestão eficiente da memória virtual são essenciais para prevenir o *thrashing* e manter um desempenho satisfatório do sistema.

Armazenamento em disco: quando o sistema de memória virtual está habilitado e a memória RAM fica toda ocupada, necessitando de mais memória, páginas de memória são colocadas em disco para que algum espaço seja disponibilizado na memória RAM. Há a opção de armazenamento da memória virtual:

- em uma partição do disco, geralmente utilizada no Linux (partição "swap", que possui um sistema de arquivos próprio);
- em um arquivo de swap (no Windows o nome do arquivo é Pagefile.sys).

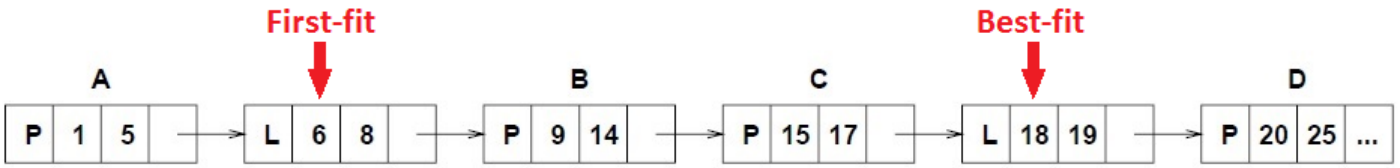
Algoritmos de Substituição em Segmentação

A partir do momento em que processos (P) e espaços livres (L) são mantidos na lista ordenada por endereços, algoritmos podem ser utilizados para alocar memória, com o objetivo de criar ou permutar processos. Esses algoritmos são chamados quando o gerenciador de memória precisa de um segmento de memória de N bytes. Vamos analisar esses algoritmos a seguir.

First-fit: procura ao longo da lista de segmentos até encontrar um espaço livre de tamanho maior ou igual a N. Caso o espaço livre seja maior que N, o espaço livre é quebrado em dois segmentos: um para o processo (N bytes) e o outro para a memória que não foi utilizada. É um algoritmo rápido, pois termina a busca o mais cedo possível.

Next-fit: funciona da mesma forma que o *First-fit*, com uma exceção: guarda a posição da lista onde o último espaço livre foi alocado. Quando for chamado novamente, o algoritmo começa a procurar a partir desse ponto.

Best-fit: procura pela lista inteira e "pega" o espaço livre de tamanho mais próximo de N. É lento e cria na memória espaços livres pequenos que dificilmente serão alocados. Porém, para N grande, esse algoritmo aumenta as chances de se encontrar na lista um espaço livre de tamanho adequado, visto que minimiza o uso de espaços livres grandes para atender requisições pequenas. Imagine que um bloco de tamanho 2 seja solicitado, o algoritmo *First-fit* alocaria o espaço livre 6-7, e o *Best-fit* o espaço livre 18-19:



Quick-fit: mantém listas separadas para tamanhos comumente solicitados, como por exemplo, uma lista para espaços livres de tamanho 4 KB, outra para espaços de 8 KB, e assim sucessivamente. Com este algoritmo, encontra-se um espaço livre de tamanho requerido muito rapidamente, mas com a desvantagem de todos os esquemas de classificar os espaços livres por tamanho. Ou seja, quando um processo termina ou é permutado para o disco, determinar seus vizinhos para uma possível fusão é uma operação custosa. Se fusões não forem realizadas, a memória rapidamente ficará fragmentada em um grande número de pequenos espaços livres não utilizáveis.

Circular-fit: utiliza uma lista circular de blocos de memória livres. A lista circular é percorrida para encontrar um bloco de tamanho adequado para atender a uma solicitação de alocação de memória. Funciona assim:

- A memória livre é organizada em uma lista circular;
- Quando uma solicitação de alocação de memória é recebida, a lista circular é percorrida em busca do primeiro bloco de memória livre que atenda aos requisitos da solicitação;
- Se um bloco adequado for encontrado, ele é alocado para o processo. Caso contrário, a busca continua no próximo bloco da lista circular;
- A lista é percorrida até que um bloco adequado seja encontrado ou até que a lista completa seja verificada.

Algoritmos de Substituição de Página

Ótimo: quando ocorre uma falta de página, um determinado conjunto de páginas está na memória. Uma dessas páginas será referenciada em muitas das próximas instruções. Outras páginas não serão referenciadas antes de 10, 100, ou quem sabe 1000 instruções. Cada página pode ser "marcada" com o número de instruções que serão executadas antes que a página seja inicialmente referenciada.

O algoritmo simplesmente diz que a página com o maior rótulo deve ser removida, adiando-se o máximo possível a próxima falta de página. O único problema é que ele não é realizável. No momento da falta de página, o sistema operacional não tem como saber quando cada página será referenciada.

NRU (*Not Recently Used*) "Não Recentemente Usada": para que o S.O. saiba quais páginas foram e quais não foram utilizadas recentemente, muitos computadores com memória virtual possuem 2 bits associados a cada página. Um bit R (Referência) é ativado pelo hardware em qualquer leitura ou escrita de página, ou seja, se houver uma referência, marca com o bit 1. O outro bit, M (Modificação), é ativado pelo hardware quando uma página é escrita (modificada). É importante que esses bits sejam atualizados em qualquer referência de memória, assim, é essencial que eles sejam ativados pelo hardware. Uma vez que um bit for ativado, ele permanece ativado até que o sistema operacional o desative (por software).

Os bits R e M podem ser usados para construir o algoritmo NRU da seguinte forma: quando um processo é iniciado, ambos os bits de página (para todas as páginas) são declarados 0 pelo sistema operacional. A cada interrupção de relógio o bit R é zerado, para distinguir páginas que não foram referenciadas recentemente daquelas que tenham sido.

Quando uma falta de página ocorre, o S.O. verifica todas as páginas e as classifica em 4 categorias baseado nos valores correntes de seus bits R e M:

- Classe 0: não referenciada, não modificada;
- Classe 1: não referenciada, modificada;
- Classe 2: referenciada, não modificada;
- Classe 3: referenciada, modificada.

Interrupções de relógio não zeram o bit M porque essa informação é necessária para determinar se uma página terá que ser reescrita no disco ou não. O algoritmo NRU remove uma página aleatória da classe não vazia de numeração mais baixa (classe 0). As características principais do NRU é que ele é fácil de entender, eficiente e gera um desempenho que é tido como adequado.

LRU (*Least Recently Used*) "Menos Usada Recentemente": quando ocorre uma falta de página, retira-se a página que não tem sido usada por um tempo longo. É um algoritmo teoricamente realizável, porém seu custo é alto. Para sua implementação completa, é necessário manter uma lista encadeada de todas as páginas em memória, com a página mais recentemente usada no início e a menos recentemente usada no fim.

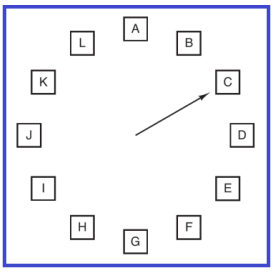
A dificuldade é que a lista deve ser atualizada em toda referência (leitura ou escrita) de memória. Encontrar a página na lista, removê-la de sua posição atual e movê-la para o início representa um grande esforço computacional!

FIFO (*First In, First Out*): o S.O. mantém uma lista de todas as páginas correntes na memória, sendo a página do início da lista a mais antiga e a página do fim a carregada mais recentemente. Em uma falta de página, a página do início é removida e a nova página carregada é acrescentada no fim da lista. É a tradicional fila que estamos acostumados em diversas situações na vida!

Segunda chance: no algoritmo FIFO pode ocorrer de uma página muito utilizada ser retirada, pois sua vez chegou no andamento da fila! Para dar uma "segunda chance" existe uma variação em relação ao FIFO. A ideia é que primeiro seja examinada a página mais antiga como uma candidata a ser removida. Se seu bit R (Referência) for 0, a página é trocada imediatamente. Se o bit R for 1, o bit é zerado e a página é colocada no fim da lista de páginas (essa é a segunda chance da página!). Dessa forma a pesquisa continua. Resumindo: O algoritmo segunda chance busca por uma página antiga que não tenha sido referenciada na interrupção de relógio anterior.

Relógio: utiliza uma lista ordenada de forma circular tal como um relógio. O ponteiro do relógio aponta para a página mais antiga e assim que ocorrer uma falta a página mais antiga é verificada. Se o bit R desta página for 0 ela é substituída, caso contrário esse bit é zerado e o ponteiro aponta para a próxima página mais antiga. Esse processo é então repetido até a próxima página com o bit R=0 ser encontrada.

Vamos ver um exemplo:



Imagine a situação mostrada ao lado. Se a página C tiver R=0, então C será removida e o ponteiro apontará para D. Se C tiver R=1, R receberá o valor 0 e o ponteiro apontará para D (neste último caso, a página C não será removida).

Outros Conceitos

Abaixo veremos alguns conceitos já cobrados em provas de concurso e que não foram "encaixados" nos títulos anteriores.

Garbage Collector (Coletor de lixo): é um coletor de "sujeira" do sistema, tendo como principal função a de descartar os espaços de memória alocados que não são mais usados pelo sistema operacional. Tempos atrás, em linguagens mais antigas, como por exemplo a linguagem C, esse processo era feito manualmente pelo desenvolvedor. Esse conceito é mais comumente explorado por linguagens de programação, mas também pode ser em sistemas operacionais.

QUESTÕES COMENTADAS - CEBRASPE

1. (CEBRASPE/TRE-PI/2016) O componente central de um sistema operacional, que determina o local da memória onde deverá ser colocado o código de um novo processo chamado para ser executado por um processo pai, lido de um arquivo previamente armazenado em um dispositivo de entrada e saída, que, por sua vez, está conectado à rede local, é denominado

A) gerenciador de sistema de arquivos.

B) gerenciador de comunicação interprocessos.

C) gerenciador de memória.

D) escalonador de processos.

E) gerenciador de entrada e saída.

Comentários:

Como vimos no 1° parágrafo sobre esse assunto:

O componente central de um sistema operacional que determina o local da memória onde deverá ser colocado o código de um novo processo, lido de um arquivo previamente armazenado em um dispositivo de entrada e saída (HD, por exemplo) é denominado gerenciador de memória.

Gabarito: C

2. (CEBRASPE/TRF1/2017) Na técnica denominada escalonamento de processos, o sistema operacional mantém parte do espaço de endereçamento de um processo na memória principal e parte em dispositivo de armazenamento secundário, realizando trocas de trechos de código e de dados entre eles, de acordo com a necessidade.

Comentários:

A questão está falando do conceito de memória virtual! Não tem nada a ver com escalonamento de processos!

Gabarito: Errado

3. (CEBRASPE/Polícia Federal/2018) A técnica de swapping consiste em transferir temporariamente um processo da memória para o disco do computador e depois carregá-lo novamente em memória.

Comentários:

De forma detalhada podemos afirmar que "Há momentos em que não há memória principal suficiente para todos os processos ativos, de maneira que os processos excedentes devem ser mantidos em disco e trazidos para a memória dinamicamente. Para isso pode ser usado uma das seguintes estratégias:"

- Swapping: traz cada processo em sua totalidade, executa-o por algum tempo e o coloca novamente no disco;
- Memória virtual: os programas podem ser executados mesmo estando apenas parcialmente na memória principal.

Note que a estratégia swapping transfere o processo por completo, enquanto a memória virtual permite partes.

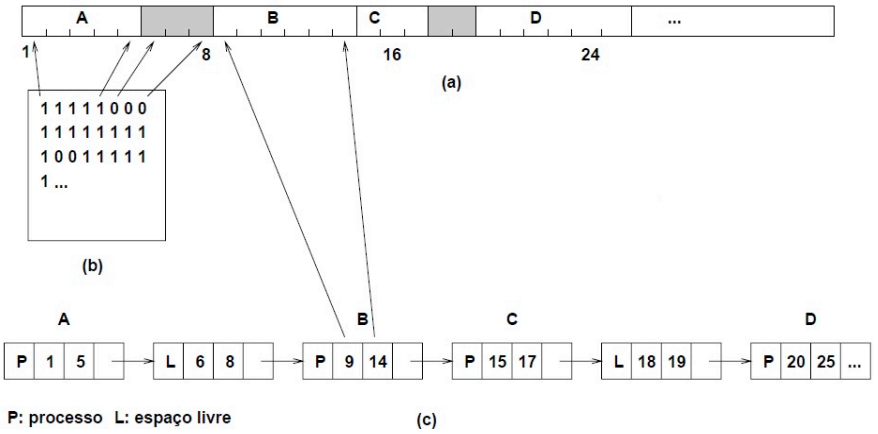
Gabarito: Certo

4. (CEBRASPE/EBSERH/2018) Uma das técnicas mais complexas para o gerenciamento do uso de memória é o mapa de bites, que consiste em manter uma lista encadeada de segmentos de memória alocados e disponíveis.

Comentários:

Como vemos na figura ao lado, o gerenciamento baseado em bits é relativamente simples, basta controlar as unidades de alocação que estão ocupadas ou livres (bits 1 ou 0).

Pode ser utilizada uma lista encadeada ou não, depende de como é realizada a implementação.



Gabarito: Errado

5. (CEBRASPE/IFF/2018) Em sistemas operacionais que usam paginação para gerenciamento de memória, os algoritmos de troca de página escolhem uma página a ser removida da memória para que uma nova seja alocada. Em um desses algoritmos, uma página modificada que não tenha sido referenciada pelo menos no último tique de relógio (tipicamente em 20 milissegundos) é removida em vez de uma página não modificada que tenha sido muito usada. Trata-se do algoritmo denominado

A) troca ótima de página.

B) troca de página não recentemente usada (NRU).

C) troca de página FIFO.

D) segunda chance.

E) troca de página menos recentemente usada (LRU).

Comentários:

NRU (*Not Recently Used*) "Não Recentemente Usada": para que o S.O. saiba quais páginas foram e quais não foram utilizadas recentemente, muitos computadores com memória virtual possuem 2

bits associados a cada página. Um bit R (Referência) é ativado pelo hardware em qualquer leitura ou escrita de página, ou seja, se houver uma referência, marca com o bit 1. O outro bit, M (Modificação), é ativado pelo hardware quando uma página é escrita (modificada). É importante que esses bits sejam atualizados em qualquer referência de memória, assim, é essencial que eles sejam ativados pelo hardware. Uma vez que um bit for ativado, ele permanece ativado até que o sistema operacional o desative (por software).

Os bits R e M podem ser usados para construir o algoritmo NRU da seguinte forma: quando um processo é iniciado, ambos os bits de página (para todas as páginas) são declarados 0 pelo sistema operacional. A cada interrupção de relógio o bit R é zerado, para distinguir páginas que não foram referenciadas recentemente daquelas que tenham sido.

Quando uma falta de página ocorre, o S.O. verifica todas as páginas e as classifica em 4 categorias baseado nos valores correntes de seus bits R e M:

- Classe 0: não referenciada, não modificada;
- Classe 1: não referenciada, modificada;
- Classe 2: referenciada, não modificada;
- Classe 3: referenciada, modificada.

Interrupções de relógio não zeram o bit M porque essa informação é necessária para determinar se uma página terá que ser reescrita no disco ou não. O algoritmo NRU remove uma página aleatória da classe não vazia de numeração mais baixa (classe 0). As características principais do NRU é que ele é fácil de entender, eficiente e gera um desempenho que é tido como adequado.

Gabarito: B

6. (CEBRASPE/Min. da Economia/2020) Julgue o próximo item, relativos a sistemas operacionais.

   O gerenciamento de memória pode ocorrer por meio do método básico, no qual um processo que está para ser executado tem suas páginas carregadas em quaisquer quadros de memória disponíveis a partir de sua origem, por exemplo, de um sistema de arquivos.

Comentários:

O que a questão afirma é que há um método básico em que um programa, ao ser executado (processo), tem suas páginas (então utiliza memória virtual por paginação) carregadas em quadros de memória (os quadros são em relação à memória física) disponíveis (marcados como "livre" pelo S.O.), a partir de sua origem (sistema de arquivos em um disco rígido, por exemplo).

Não tem pegadinha nenhuma! É isso, mesmo! E como saber quais quadros estão livres? Aí vai da gerência de memória do sistema operacional!

Gabarito: Certo

7. (CEBRASPE/Prefeitura de Barra dos Coqueiros-SE/2020) Assinale a opção correta, a respeito de sistemas operacionais (SO).

A) Swapping é uma técnica que permite subdividir a memória endereçável do SO.

B) Um processo pode ser alocado ou na quantidade exata de memória exigida (exemplo de particionamento variável, mais eficiente) ou na menor partição possível (exemplo de particionamento fixo, mais simples).

C) Cabe ao SO prover e gerenciar acesso controlado aos arquivos e programas, enquanto, por questão de desempenho, cabe somente ao hardware o acesso aos dispositivos de E/S.

D) Escalonador é a parte que contém as funções básicas de um SO e as que são usadas com mais frequência.

E) Comparado ao RAID 0, o RAID 1 possui como vantagem a necessidade de usar apenas a metade de espaço em disco, porém tem como desvantagem a indisponibilidade dos dados em caso de falha.

Comentários:

A) ERRADA - Swapping é uma que pega o processo por inteiro e coloca na memória e no disco, fazendo essa "troca" (swap).

B) CORRETA - São formas de particionamento, cada uma com suas vantagens e desvantagens!

C) ERRADA - Cabe ao S.O. também o acesso aos dispositivos de E/S, como por exemplo um disco rígido. São realizadas chamadas de sistema para o HD, por exemplo.

D) ERRADA - Escalonador é a parte que escala, que seleciona "quem" vai usar algum recurso.

E) ERRADA - Não abordamos nesta aula, mas o RAID1 é um espelhamento, então se um disco falhar o outro tem uma cópia exata.

Gabarito: B

8. (CEBRASPE/PG-DF/2021) No que se refere a sistemas operacionais, julgue o próximo item.

Quando a memória é gerenciada por mapa de bites, o valor 1 indica que a unidade referenciada está ocupada e o valor 0 indica que ela está livre.

Comentários:

Por padrão é assim que funciona. Até pode ser invertido, se algum implementador quiser fazer!

Gabarito: Certo

9. (CEBRASPE/PG-DF/2021) No que se refere a sistemas operacionais, julgue o próximo item.

A paginação de memória é executada pelo sistema operacional em três etapas: criação do processo, término do processo e limpeza do buffer residual.

Comentários:

Um resumo de como funciona o gerenciamento de memória por paginação é:

- Criação do Processo: Quando um processo é criado, o sistema operacional aloca espaço na memória para suas páginas;

- Execução do Processo: Durante a execução do processo, o sistema operacional pode mover páginas entre a memória RAM e a memória secundária (ex.: disco rígido). Isso é realizado para otimizar o uso da memória, mantendo na memória RAM apenas as páginas necessárias para o processo em execução;

- Término do Processo: Quando um processo é finalizado, o sistema operacional libera a memória associada a esse processo, o que pode incluir a liberação de todas as páginas alocadas para o processo.

A questão não mencionou a execução do processo e inventou uma "limpeza do buffer residual".

Gabarito: Errado

QUESTÕES COMENTADAS - FGV

1. (FGV/IBGE/2017) A estratégia de alocação de memória que busca o menor espaço livre suficiente para satisfazer cada requisição denomina-se:

A) minor fit;

B) save fit;

C) best fit;

D) first fit;

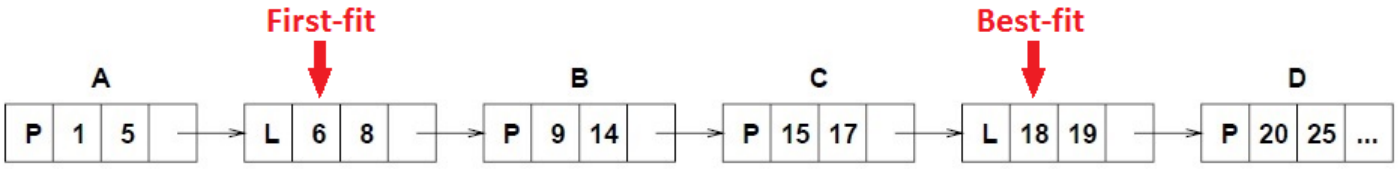
E) worst fit.

Comentários:

First-fit: procura ao longo da lista de segmentos até encontrar um espaço livre de tamanho maior ou igual a N. Caso o espaço livre seja maior que N, o espaço livre é quebrado em dois segmentos: um para o processo (N bytes) e o outro para a memória que não foi utilizada. É um algoritmo rápido, pois termina a busca o mais cedo possível.

Next-fit: funciona da mesma forma que o *First-fit*, com uma exceção: guarda a posição da lista onde o último espaço livre foi alocado. Quando for chamado novamente, o algoritmo começa a procurar a partir desse ponto.

Best-fit: procura pela lista inteira e "pega" o espaço livre de tamanho mais próximo de N. É lento e cria na memória espaços livres pequenos que dificilmente serão alocados. Porém, para N grande, esse algoritmo aumenta as chances de se encontrar na lista um espaço livre de tamanho adequado, visto que minimiza o uso de espaços livres grandes para atender requisições pequenas. Imagine que um bloco de tamanho 2 seja solicitado, o algoritmo *First-fit* alocaria o espaço livre 6-7, e o *Best-fit* o espaço livre 18-19:



Quick-fit: mantém listas separadas para tamanhos comumente solicitados, como por exemplo, uma lista para espaços livres de tamanho 4 KB, outra para espaços de 8 KB, e assim sucessivamente. Com este algoritmo, encontra-se um espaço livre de tamanho requerido muito rapidamente, mas com a desvantagem de todos os esquemas de classificar os espaços livres por tamanho. Ou seja, quando um processo termina ou é permutado para o disco, determinar seus vizinhos para uma possível fusão é uma operação custosa. Se fusões não forem realizadas, a memória rapidamente ficará fragmentada em um grande número de pequenos espaços livres não utilizáveis.

Gabarito: C

2. (FGV/AL-RO/2018) A função dos computadores é executar processos mas para que os processos sejam executados, eles devem estar na memória do computador. O gerenciamento da memória é uma das funções dos sistemas operacionais.

   Uma técnica empregada pelo sistema operacional para permitir que processos não carregados na memória principal sejam executados, a partir da emulação de uma memória significativamente maior, é

A) o swapping.

B) a memória física.

C) a memória virtual.

D) a paginação.

E) o cache.

Comentários:

A) ERRADA - Veja o swapping na explicação da alternativa C.

B) ERRADA - A memória física (RAM) é onde os processos (ou partes deles) devem estar para executar as instruções. A memória RAM é menor que a memória virtual (ver explicação na alternativa C).

C) CORRETA - A memória virtual emula um determinado tamanho (maior que a memória RAM. Um dos papeis da gerência de memória é fazer o controle dos processos que vão para a memória RAM ou para o disco, além de estabelecer se vai o processo inteiro (swapping) ou partes dele (paginação).

D) ERRADA - Veja a paginação na explicação da alternativa C.

E) ERRADA - A memória cache fica mais próxima ao processador do que a memória RAM e armazena as instruções e os dados que mais foram carregados ou escritos na memória RAM.

Gabarito: C

3. (FGV/MPE-AL/2018) A implementação de linguagens de programação em geral depara-se com a questão do gerenciamento de memória. Nesse contexto, assinale a opção que melhor descreve o que é conhecido como "memory leak".

A) A liberação de trechos de memória ainda em uso por um ou mais objetos.

B) A impossibilidade de liberar a memória ocupada por objetos que se tornaram inalcançáveis.

C) A incapacidade de recuperar trechos de memória virtualizados.

D) A invasão de trechos de memória por objetos não autorizados.

E) O compartilhamento indesejado de trechos de memória por dois ou mais objetos.

Comentários:

Memory leak (vazamento de memória) que pode ocorrer em duas situações:

1. Blocos de memória estão alocados e disponíveis para serem utilizados pelo programa, mas não são acessíveis porque a aplicação não tem nenhum ponteiro apontando para essa área de memória. Ou seja, tais blocos de memória não podem ser utilizados nem pelo programa nem por outros processos (ficam alocados e sem acesso!);

2. Blocos de memória possuem dados que poderiam ser liberados por estarem inacessíveis e que, por algum "esquecimento", ainda são referenciados no código. Ou seja, mesmo sem estarem sendo usados, não podem ser liberados.

Gabarito: B

4. (FGV/TJ-TO/2022) A estratégia de swapping utilizada em sistemas operacionais para gerenciamento de memória consiste em:

A) dividir a memória em unidades de alocação em kbytes endereçáveis por um mapa de bit;

B) trazer para a memória principal cada processo, executá-lo por um tempo e então colocá-lo de volta na memória secundária;

C) segmentar os processos maiores que a memória principal em subprocessos menores executáveis sob demanda;

D) fracionar a memória principal e secundária em unidades básicas de mesmo tamanho para evitar fragmentação;

E) espalhar os processos sobre áreas não contínuas da memória principal para permitir extrapolarem o espaço inicial alocado.

Comentários:

Vamos analisar as seguintes estratégias:

- *Swapping*: traz cada processo em sua totalidade, executa-o por algum tempo e o coloca novamente no disco;
- Memória virtual: os programas podem ser executados mesmo estando apenas parcialmente na memória principal.

Note que a estratégia swapping transfere o processo por completo, enquanto a memória virtual permite partes.

Gabarito: B

5. (FGV/MPE-GO/2022) Sistemas operacionais têm um importante papel no gerenciamento de memória de um computador, especialmente na recuperação de nacos de memória que foram descartados, seja intencionalmente ou pela ocorrência de erros.

Essa recuperação é usualmente realizada por um processo denominado

A) Garbage collector.

B) Kernel.

C) Memory Launcher.

D) Memory Retrieval.

E) Scheduler.

Comentários:

Garbage Collector (Coletor de lixo): é um coletor de "sujeira" do sistema, tendo como principal função a de descartar os espaços de memória alocados que não são mais usados pelo sistema operacional. Tempos atrás, em linguagens mais antigas, como por exemplo a linguagem C, esse processo era feito manualmente pelo desenvolvedor. Esse conceito é mais comumente explorado por linguagens de programação, mas também pode ser em sistemas operacionais.

Gabarito: A

6. (FGV/TRT13/2022) A estratégia de gerenciamento adotada em sistemas operacionais modernos que consiste em trazer para memória principal cada processo em sua totalidade, executá-lo por um tempo e então colocá-lo de volta no disco é denominada

A) buffering.

B) defragging.

C) offset.

D) relocation.

E) swapping.

Comentários:

Vamos analisar as seguintes estratégias:

- *Swapping*: traz cada processo em sua totalidade, executa-o por algum tempo e o coloca novamente no disco;
- Memória virtual: os programas podem ser executados mesmo estando apenas parcialmente na memória principal.

Note que a estratégia swapping transfere o processo por completo, enquanto a memória virtual permite partes.

Gabarito: E

7. (FGV/Banco do Brasil/2023) No contexto dos sistemas operacionais modernos, swapping consiste em

A) subdividir a memória física em pequenas partições para permitir uma utilização mais eficiente.

B) reordenar o espaço de armazenamento, fazendo com que todo arquivo esteja armazenado de forma contígua.

C) mover dados para uma área de trabalho temporária para um programa externo acessar para processá-lo em tempo futuro.

D) forçar o núcleo do processador a operar numa frequência mais alta do que a especificada pelo fabricante.

E) trazer cada processo em sua totalidade para memória RAM, executá-lo por um tempo e então colocá-lo de volta no disco.

Comentários:

Vamos analisar as seguintes estratégias:

- *Swapping*: traz cada processo em sua totalidade, executa-o por algum tempo e o coloca novamente no disco;
- Memória virtual: os programas podem ser executados mesmo estando apenas parcialmente na memória principal.

Note que a estratégia swapping transfere o processo por completo, enquanto a memória virtual permite partes.

Gabarito: E

8. (FGV/FHEMIG/2023) No contexto do gerenciamento de memória, programas e processos, assinale a afirmativa incorreta.

A) Paginação é uma técnica de gerenciamento de memória que divide o espaço de endereçamento dos processos em blocos fixos de tamanho uniforme chamados "páginas".

B) Swapping é um mecanismo de gerenciamento de memória que permite ao sistema operacional transferir, temporariamente, processos ou partes deles da memória RAM para o armazenamento secundário (como o disco rígido) quando a memória principal está escassa, liberando espaço para outros processos em execução.

C) Segmentação é uma técnica de gerenciamento de memória que divide os processos em blocos lógicos de tamanhos variáveis, para melhor aproveitamento do espaço e organização das informações em sistemas operacionais.

D) Caching é uma técnica de gerenciamento de memória que armazena permanentemente os dados mais usados na memória secundária, reduzindo a necessidade de acesso à memória RAM e melhorando o desempenho geral do sistema.

E) Thrashing é um fenômeno no gerenciamento de memória virtual em que o sistema operacional gasta a maior parte do tempo trocando páginas entre a memória principal e a memória secundária, resultando em um desempenho significativamente reduzido.

Comentários:

A) CORRETA - As páginas podem ser carregadas aos poucos, ao contrário do mecanismo de swapping, que carrega o processo todo!

B) CORRETA - O mecanismo de swapping carrega o processo todo, diferentemente da paginação. Poderíamos marcar como incorreta, mas veremos que tem alternativa pior do que essa!

C) CORRETA - Dependendo dos tamanhos dos processos carregados e descarregados, aos poucos podem ocorrer fragmentações externas.

D) INCORRETA - A memória cache armazena as instruções e dados mais acessados da memória RAM, mas não os armazena permanentemente!

E) CORRETA - Ou seja, o sistema fica sobrecarregado com a execução de vários processos, cada um exigindo acesso frequente a páginas de memória que não estão atualmente na memória RAM. O *thrashing* faz com que o desempenho fique extremamente baixo.

Gabarito: D

9. (FGV/SMPOG de Belo Horizonte-MG/2023) A paginação é uma técnica de gerenciamento de memória comumente implementada em sistemas operacionais modernos. Sobre a paginação, assinale a afirmativa correta.

A) O comprimento das páginas do sistema operacional pode mudar durante a execução de um programa.

B) É necessário adicionar memória física na máquina para ampliar o espaço de endereçamento linear dos programas.

C) Uma página lógica pode ser carregada em qualquer página física que esteja livre.

D) Os programas que ocupam múltiplas páginas físicas causam a fragmentação externa da memória.

Comentários:

A) ERRADA - As páginas são de tamanho fixo, ex.: 4 KB.

B) ERRADA - A paginação usa memória virtual, que é maior que a memória RAM. Não é necessário adicionar memória RAM para ampliar o espaço de endereçamento "linear" dos programas.

C) CORRETA - Existe um algoritmo para definir em qual quadro de página será carregada.

D) ERRADA - Causam a fragmentação interna (dentro da última página). Ex.: página de tamanho 4 KB, processo com tamanho 9 KB carregado por completo, a terceira página terá uma fragmentação interna de 3 KB.

Gabarito: C

QUESTÕES COMENTADAS - FCC

1. (FCC/CREMESP/2016) Em um computador, quando o sistema de memória virtual está habilitado, a memória RAM é dividida em blocos chamados de páginas. Quando a memória RAM está toda ocupada e o Sistema Operacional ou algum software necessita de mais memória, para liberar espaço, o processador armazena o conteúdo de páginas de memória que não estão sendo usadas naquele exato momento

A) no swap file.

B) na memória cache.

C) nos registradores de dados.

D) na Unidade Lógica de Armazenamento – ULA.

E) na virtual machine.

Comentários:

Quando o sistema de memória virtual está habilitado e a memória RAM fica toda ocupada, necessitando de mais memória, páginas de memória são colocadas em disco para que algum espaço seja disponibilizado na memória RAM. Há a opção de armazenamento da memória virtual:

- em uma partição do disco, geralmente utilizada no Linux (partição "swap", que possui um sistema de arquivos próprio);
- em um arquivo de swap (no Windows o nome do arquivo é Pagefile.sys).

Gabarito: A

2. (FCC/ARTESP/2017) No gerenciamento de memória, o mecanismo de paginação utiliza algoritmos de substituição de páginas, que são políticas definidas para escolher qual página da memória será removida para dar lugar a uma página que foi solicitada e que precisa ser carregada. Dentre estes encontra-se o algoritmo

A) FIFO – First In First Out que consiste em substituir a última página que foi carregada na memória. Esta escolha não leva em consideração se a página está sendo muito utilizada ou não, assim a quantidade de falta de páginas tende a diminuir quando o tamanho da memória também diminui.

B) NRU – Not Recently Used que procura por páginas que não foram referenciadas nos últimos acessos para serem substituídas e também verifica, através de um bit de modificação, se a página teve seu conteúdo alterado durante sua permanência em memória.

C) LIFO – Last In First Out, que consiste em substituir a página que foi carregada há mais tempo na memória. Na ocorrência de uma falta de página, a primeira página da lista será substituída e a nova será acrescentada ao final da lista.

D) CLOCK, que faz a substituição da última página acessada. Este escolhe a última página acessada para ser substituída. Dessa forma, é possível explorar com mais eficiência o princípio de localidade temporal apresentada pelos acessos.

E) MRR – Most Recently Removed, que mantém todas as páginas em uma lista circular. A ordem mantida segue a sequência em que elas foram carregadas em memória. Além disso, é adicionado um byte de uso que indica se a página foi referenciada novamente depois de ter sido carregada.

Comentários:

A) ERRADA - FIFO consiste em substituir a PRIMEIRA página que foi carregada na memória.

B) CORRETA - NRU (Not Recently Used) procura por páginas que não foram referenciadas recentemente (o nome do algoritmo já deixa claro isso). Esse controle ocorre através de um bit de modificação.

C) ERRADA - LIFO não aparece como política de substituição de páginas, mas o conceito de LIFO, o mesmo funcionamento de uma pilha, retiraria o que foi colocado por último (topo da pilha).

D) ERRADA - CLOCK utiliza uma lista ordenada de forma circular tal como um relógio. O ponteiro do relógio aponta para a página mais antiga e assim que ocorrer uma falta a página mais antiga é verificada. Se o bit R desta página for 0 ela é substituída, caso contrário esse bit é zerado e o ponteiro aponta para a próxima página mais antiga.

E) ERRADA - MRR (Most Recently Removed) não existe!

Gabarito: B

3. (FCC/DPE-AM/2018) Em um computador com um sistema operacional típico nem sempre é possível manter na memória todos os processos por falta de espaço. Uma solução comumente adotada nessas situações é a utilização de uma área no disco para a colocação de processos que estejam momentaneamente bloqueados. Esse mecanismo é conhecido como

A) swapping.

B) paginação.

C) particionamento.

D) segmentação.

E) compactação.

Comentários:

O foco da questão é essa parte: "utilização de uma área no disco para a colocação de processos que estejam momentaneamente bloqueados". Analisando as alternativas, poderíamos ficar em dúvida entre swapping e paginação. Lendo novamente a parte destacada, podemos entender que o examinador quer saber sobre a "colocação de processos", então podemos deduzir que são os processos por inteiro, pois a questão não fala em dividi-los (em páginas), então só sobra a alternativa swapping. Preste muita atenção nessa sutil diferença e releia o enunciado, se tiver dúvidas. Não se afobe na hora de marcar a resposta!

Gabarito: A

4. (FCC/TRF4/2019) No sistema de memória virtual por paginação, determinado problema ocorre em dois níveis:

– No próprio processo: elevado número de page faults; processo passa mais tempo esperando por páginas do que executando.

– No sistema: existem mais processos competindo por memória principal do que espaço disponível.

A solução seria reduzir o número de páginas de cada processo na memória.

O problema se refere

A) à excessiva transferência de blocos entre a memória principal e a secundária, conhecida como thrashing.

B) à excessiva seleção de processos para saírem da memória, denominada swapping.

C) ao overflow da tabela que controla o esquema de mapeamento associativo.

D) à inexistência do Translation Lookside Buffer (TLB).

E) à política de paginação antecipada, conhecida como cache hit.

Comentários:

Thrashing é um fenômeno que ocorre quando um sistema operacional gasta a maior parte do seu tempo trocando páginas entre a memória RAM e a memória secundária (ex.: disco rígido), em vez de executar efetivamente as instruções dos programas. Ou seja, o sistema fica sobrecarregado com a execução de vários processos, cada um exigindo acesso frequente a páginas de memória que não estão atualmente na memória RAM. O *thrashing* faz com que o desempenho fique extremamente baixo.

Para evitar o *thrashing*, os sistemas operacionais utilizam algoritmos de substituição de página eficientes, como por exemplo o algoritmo de segunda chance, para garantir que as páginas mais relevantes para a execução dos processos estejam sempre presentes na memória RAM. A otimização do tamanho da carga de trabalho e a gestão eficiente da memória virtual são essenciais para prevenir o *thrashing* e manter um desempenho satisfatório do sistema.

Gabarito: A

5. (FCC/AL-AP/2020) Quando um processo do sistema operacional tem mais espaço de endereçamento do que o computador tem de memória principal e o processo deseja utilizá-lo inteiramente, isso

A) incorrerá em erro no espaço de endereçamento, travando a máquina.

B) incorrerá em erro no espaço de endereçamento, todavia, o ciclo de processamento se completará solicitando intervenção externa.

C) pode ser resolvido pelo uso de memória virtual.

D) evidencia erro de planejamento, obrigando o administrador a adquirir mais recursos de memória.

E) não tem como ocorrer, pois todos os processos que são alocados em um computador já são estabelecidos dentro dos recursos computacionais existentes.

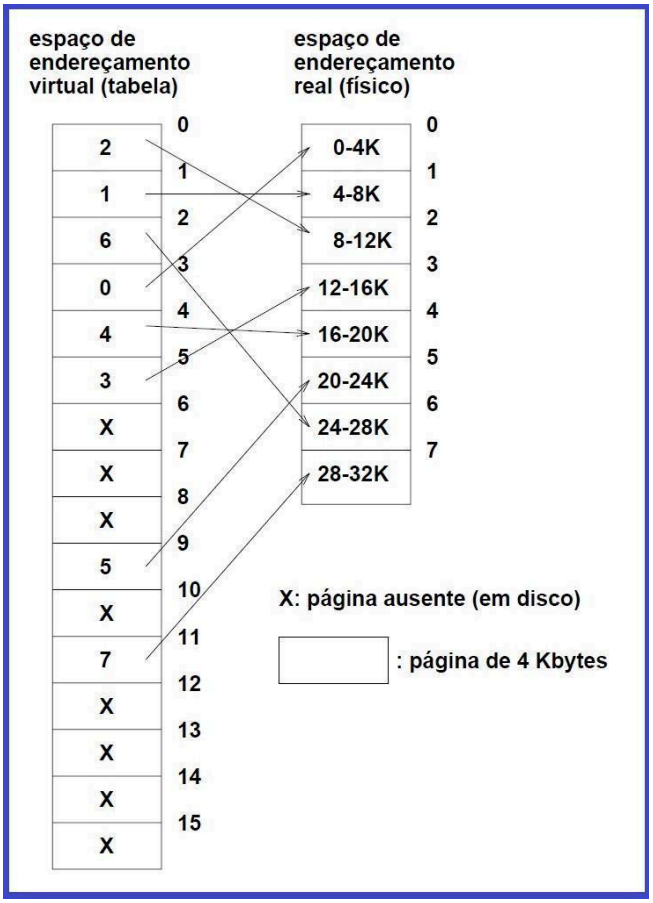
Comentários:

Para expandir o uso de memória sem ter memória física, uma solução é o uso de memória virtual, com o uso da memória secundária para o armazenamento temporário de processos ou parte deles que não estão em uso em determinado momento. Abaixo veremos como isso funciona para o esquema de memória virtual com o uso de paginação.

O espaço de endereçamento virtual é dividido em unidades chamadas páginas. As unidades correspondentes na memória física são chamadas quadros (*page frames*). Para não ter erro pense no seguinte: considere uma página como uma tela de pintura que deve ser colocada em um quadro (uma moldura).

As páginas e os quadros são sempre do mesmo tamanho. No exemplo ao lado elas são de 4 KB. Com 64 KB de espaço de endereço virtual e 32 KB de memória física, temos 16 páginas e 8 quadros. As transferências entre a memória e o disco sempre são realizadas em unidades de páginas (4 KB, no nosso exemplo).

Quando um programa tenta acessar o endereço 5000, por exemplo, o endereço virtual 9000 é enviado para a MMU. Como temos páginas de 4 KB (4096 bytes), sabemos que a página 0 possui os endereços 0-4095, a página 1 os endereços 4096-8191, a página 2 os endereços 8192-12287, e assim por diante. A tabela de páginas tem a função de mapear as páginas virtuais em molduras de página física.



Gabarito: C

Questões Comentadas - VUNESP

1. (VUNESP/Pref. de Itapevi-SP/2019) Memória Virtual é uma técnica de gerenciamento de memória de computadores muito utilizada, que pode ser implementada com o emprego de dois mecanismos:

A) a paginação e a fragmentação.

B) a paginação e a segmentação.

C) a segmentação e o particionamento.

D) a virtualização e o swapping.

E) o swapping e o particionamento.

Comentários:

Segmentação: os programas são divididos em segmentos de tamanhos diversos, cada um com o seu próprio espaço de endereçamento. A alocação da memória ocorre de maneira não fixa, ou seja, cada segmento pode ter um tamanho diferente (a alocação depende da lógica do programa). O mapeamento é feito através das tabelas de mapeamento de segmentos e os endereços são compostos pelo número do segmento e um deslocamento dentro do segmento (ex.: segmento 0, deslocamento = 500 bytes).

Paginação: técnica utilizada pela maioria dos sistemas com memória virtual. Em qualquer computador existe um determinado conjunto de endereços de memória que programas podem referenciar. Trata-se de endereços virtuais que formam o espaço virtual de endereçamento do processo. Em computadores sem memória virtual, o endereço virtual é colocado diretamente no barramento de memória, uma palavra da memória física com o mesmo endereço é lida ou escrita.

Gabarito: B

2. (VUNESP/Pref. de Ilhabela-SP/2020) Em sistemas operacionais, a técnica conhecida como swapping

A) traz totalmente cada processo para a memória, executa-o por algum tempo e retorna-o para o disco posteriormente.

B) permite que um processo seja executado apenas com uma parte do código na memória, podendo a outra parte ficar permanentemente em disco.

C) permite que um processo possa ser executado diretamente no disco no qual ele esteja armazenado, sem precisar ser carregado na memória.

D) é uma boa alternativa para os computadores modernos, por ser econômica e de maior rapidez na execução dos processos, possibilitando que se instale menos memória RAM no computador.

E) apresenta como vantagem a redução do uso de operações de entrada/saída.

Comentários:

Há momentos em que não há memória principal suficiente para todos os processos ativos, de maneira que os processos excedentes devem ser mantidos em disco e trazidos para a memória dinamicamente. Para isso pode ser usada uma das seguintes estratégias:

- Swapping: traz cada processo em sua totalidade, executa-o por algum tempo e o coloca novamente no disco;

- Memória virtual: os programas podem ser executados mesmo estando apenas parcialmente na memória principal.

Gabarito: A

QUESTÕES COMENTADAS - CESGRANRIO

1. (CESGRANRIO/LIQUIGÁS/2018) A gerência de memória efetuada pelos sistemas operacionais inclui a definição de uma política para a substituição de páginas de memória, que trata da escolha de uma página da memória principal que deve ser substituída quando uma nova página precisa ser carregada.

   Dentre as políticas básicas mais utilizadas, há uma que opta por substituir a página que está há mais tempo sem ser referenciada, sendo conhecida como

A) FIFO.

B) LIFO.

C) Optimal.

D) LRU.

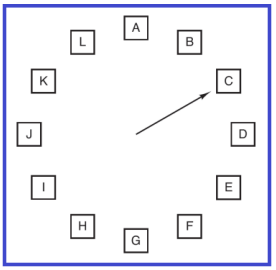
E) Clock Policy.

Comentários:

Vamos prestar atenção nesta parte: "substituir a página que está há mais tempo sem ser referenciada". Ou seja, a que recentemente está sendo pouco usada. Vamos analisar cada sigla:

(A) FIFO = First In, First Out (uma fila); (B) LIFO = Last In, First Out (uma pilha, nem vimos nesta aula); (C) O algoritmo ótimo é aquele que "prevê" o futuro, não é implementável; (D) Least Recently Used = Menos Utilizado Recentemente (Opa! Só pela sigla já matamos a xarada!); (E) O relógio é aquele aprimoramento do algoritmo FIFO que considera também o bit R. Melhor relembrar o exemplo:

Imagine a situação mostrada ao lado. Se a página C tiver R=0, então C será removida e o ponteiro apontará para D. Se C tiver R=1, R receberá o valor 0 e o ponteiro apontará para D (neste último caso, a página C não será removida).



Gabarito: D

2. (CESGRANRIO/Petrobras/2018) Na implementação da memória virtual em sistemas operacionais, é necessário definir, dentre outros elementos, a política de substituição de páginas. Esses algoritmos escolhem qual página deverá deixar a memória real (chamada página vítima) e voltar ao meio secundário de armazenamento, a fim de ceder espaço à outra página requisitada pelo processador e atender à requisição de falha de página. Uma das formas de implementação desses mecanismos de substituição de páginas prevê a utilização de bits auxiliares associados às páginas.

   Nesse contexto, qual a função do bit de sujeira (dirty bit)?

A) Evitar a utilização de um algoritmo de substituição de páginas da classe local.

B) Evitar que uma página seja eleita como "vítima", ou seja, tenha de deixar a memória.

C) Indicar quando uma página foi alterada durante a execução de um processo.

D) Indicar se a página é candidata à ocorrência de trashing.

E) Indicar se a página foi referenciada dentro de um intervalo de tempo.

Comentários:

Dirty bit ("bit de sujeira"): trata-se de um marcador que indica se uma página de memória foi alterada desde que foi carregada na memória. É frequentemente utilizado em ambientes de memória virtual, onde partes do programa ou dados são movidos entre a memória RAM e o armazenamento secundário (geralmente disco).

Gabarito: C

3. (CESGRANRIO/Petrobras/2018) A gerência de memória de um sistema operacional, dentre outras funções, define a política que será utilizada para escolher a região da memória que um processo ocupará ao ser carregado para execução.

   O processo de alocação e liberação das regiões da memória, dependendo da política escolhida, pode ocasionar situações em que pequenas regiões livres nos espaços entre regiões alocadas a outros processos se tornem difíceis de ser utilizadas, pois seu tamanho não comporta facilmente outros processos, configurando um fenômeno conhecido como

A) fragmentação

B) interrupção

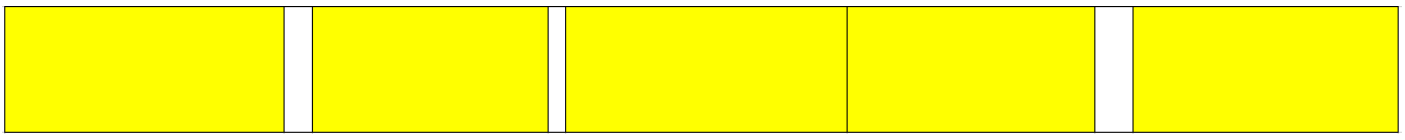
C) deadlock

D) starvation

E) troca de contextos

Comentários:

Imagine a situação abaixo, representando a memória RAM. Em amarelo são mostrados os segmentos alocados e em branco os que estão livres. Neste caso, existem fragmentações externas, ou seja, fragmentos fora dos segmentos alocados. Se fossem segmentos de tamanho fixo e houvesse espaço dentro deles, então haveria fragmentação interna. Dentro das alternativas só há "fragmentação", então essa é a resposta.



Gabarito: A

4. (CESGRANRIO/Transpetro/2018) A gerência de memória visa a maximizar o número de usuários e aplicações, utilizando de forma eficiente o espaço da memória principal. A política

de substituição de páginas visa a selecionar, dentre as páginas alocadas, qual deverá ser liberada.

O algoritmo de substituição de página que seleciona a página na memória principal que está há mais tempo sem ser referenciada é o

A) Clock

B) LFU

C) LRU

D) FIFO

E) FIFO com buffer

Comentários:

LRU (*Least Recently Used*) "Menos Usada Recentemente": quando ocorre uma falta de página, retira-se a página que não tem sido usada por um tempo longo. É um algoritmo teoricamente realizável, porém seu custo é alto. Para sua implementação completa, é necessário manter uma lista encadeada de todas as páginas em memória, com a página mais recentemente usada no início e a menos recentemente usada no fim.

A dificuldade é que a lista deve ser atualizada em toda referência (leitura ou escrita) de memória. Encontrar a página na lista, removê-la de sua posição atual e movê-la para o início representa um grande esforço computacional!

Gabarito: C

5. (CESGRANRIO/LIQUIGÁS/2018) Uma vantagem da memória virtual é permitir um número maior de processos compartilhando a memória principal. Há uma técnica de memória virtual na qual o espaço de endereçamento virtual e o espaço de endereçamento real são divididos em blocos de mesmo tamanho.

Essa técnica é chamada de

A) paginação

B) segmentação

C) swapping

D) mirrowing

E) striping

Comentários:

Paginação: técnica utilizada pela maioria dos sistemas com memória virtual. Em qualquer computador existe um determinado conjunto de endereços de memória que programas podem referenciar. Trata-se de endereços virtuais que formam o espaço virtual de endereçamento do

processo. Em computadores sem memória virtual, o endereço virtual é colocado diretamente no barramento de memória, uma palavra da memória física com o mesmo endereço é lida ou escrita.

Com o uso da memória virtual, os endereços de memória não vão diretamente para o barramento de memória, eles vão à unidade de gerenciamento de memória (MMU - *Memory Management Unit*), um hardware específico que mapeia os endereços virtuais em endereços da memória física

Gabarito: A

6. (CESGRANRIO/Caixa/2021) Na técnica de memória virtual, a tarefa de tradução dos endereços virtuais é realizada por hardware, juntamente com o sistema operacional, para não comprometer o desempenho. O dispositivo de hardware responsável por essa tradução é a

A) FPU

B) ALU

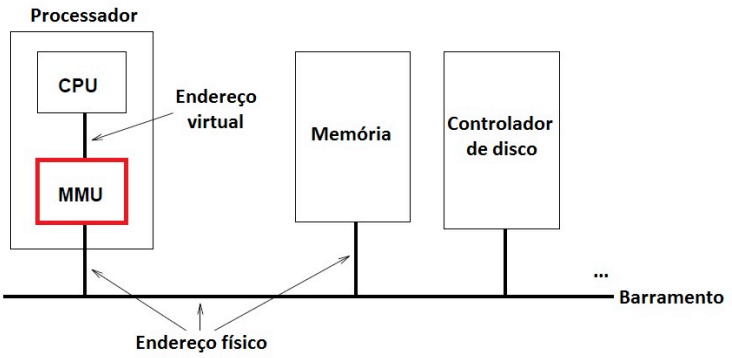
C) MMU

D) VU

E) Cache

Comentários:

Com o uso da memória virtual, os endereços de memória não vão diretamente para o barramento de memória, eles vão à unidade de gerenciamento de memória (MMU - *Memory Management Unit*), um hardware específico que mapeia os endereços virtuais em endereços da memória física:



Gabarito: C

QUESTÕES COMENTADAS - MULTIBANCAS

1. (Cetro/Dataprev/2016) Sobre as técnicas de gerenciamento de memória, assinale a alternativa que apresenta uma característica do gerenciamento de memória virtual por paginação

A) O armazenamento é realizado por meio de endereçamento sequencial denominado "segmento".

B) Gera fragmentação externa.

C) Divide o espaço de endereçamento em blocos de tamanhos variados.

D) Gera a fragmentação interna.

E) Gera a fragmentação interna e a fragmentação externa.

Comentários:

(A) Os segmentos são utilizados pela técnica de segmentação, não o de paginação! (B) "Quem" gera a fragmentação externa é a técnica de segmentação; (C) Paginação divide o espaço de endereçamento em páginas do mesmo tamanho (ex.: 4KB); (D) Gera fragmentação interna sim, na última página. Pode ocorrer de não ter a fragmentação, se o processo tiver o tamanho múltiplo do tamanho da página, ex.: página = 4KB e processo = 12KB exatos! (E) Apenas a interna, pois todas páginas possuem o mesmo tamanho e ela pode estar ocupada (processo) ou livre.

Gabarito: D

2. (COMPERVE/UFRN/2018) Considere as afirmativas abaixo referentes à ocorrência de uma falta de página no gerenciamento de memória de um computador.

I É responsabilidade do sistema operacional detectar uma falta de página.

II O sistema operacional é acionado através de uma interrupção de hardware.

III O processo que sofre a falta de página passa para o estado de espera até que a página seja carregada na memória cache.

IV A falta de página é corrigida com a carga da página em falta, da memória virtual para a memória física.

Estão corretas as afirmações

A) II e III.

B) I e III.

C) II e IV.

D) I e IV.

Comentários:

(I) É responsabilidade do hardware (gera uma interrupção ao S.O.); (II) Isso mesmo, conforme acabamos de ver; (III) Nada disso, leia a IV; (IV) Agora sim! Faltou página? Tem que buscar no disco e carregar na memória!

Gabarito: C

3. (FAURGS/TJ-RS/2018) Em relação à paginação, assinale a alternativa correta.

A) É uma forma de alocação não contígua de memória para o espaço de endereçamento de um processo.

B) Apresenta como vantagem gerar fragmentação externa apenas na última página do processo.

C) Divide o espaço de endereçamento do processo em quatro páginas: código, dados, heap (monte) e pilha.

D) Elimina a fragmentação interna e reduz a fragmentação externa.

E) É um método de gerência de memória usado apenas em sistemas que empregam memória real.

Comentários:

Paginação: técnica utilizada pela maioria dos sistemas com memória virtual. Em qualquer computador existe um determinado conjunto de endereços de memória que programas podem referenciar. Trata-se de endereços virtuais que formam o espaço virtual de endereçamento do processo. Em computadores sem memória virtual, o endereço virtual é colocado diretamente no barramento de memória, uma palavra da memória física com o mesmo endereço é lida ou escrita.

A paginação normalmente gera fragmentação interna na última página, pois raramente um processo tem tamanho múltiplo da página! Um processo pode ter inúmeras páginas (depende do tamanho do processo e da página!). É utilizado em sistemas que utilizam memória virtual.

Gabarito: A

4. (FUNDATEC/AL-RS/2018) Para a gerência de memória de um sistema operacional, existem algoritmos de substituição de página. Um deles, de baixa sobrecarga, possui o seguinte modo de operação: (1) a primeira página a entrar é a primeira a sair; (2) pode ser implementado através de uma lista de todas as páginas correntemente na memória, sendo que a página mais antiga ocupa o início dessa lista e a mais recente ocupa o fim; e (3) quando falta uma página, a mais antiga é retirada e a nova é colocada no fim da lista. Trata-se do algoritmo:

A) FIFO (First In, First Out).

B) LRU (Least Recently Used).

C) NRU (Not Recently Used).

D) Ótimo.

E) Segunda chance.

Comentários:

Esse algoritmo é o mais simples de entender. É uma fila que estamos acostumados no dia a dia. O primeiro a entrar é o primeiro a sair, sem prioridades ou coisas do tipo. Pode ser implementado com vetor, lista (claro que lista é melhor). Estamos falando do FIFO e o examinador ajudou colocando o significado da sigla entre parênteses.

Gabarito: A

5. (COMPERVE/UFRN/2018) Suponha uma memória composta por 5 partições fixas, sendo elas de 500MB (reservado e totalmente ocupado pelo sistema operacional), 200MB, 100MB, 74MB e 300MB, exatamente nesta ordem. O usuário lançou 4 processos A, B, C e D de tamanhos 99MB, 70MB, 250MB e 190MB, respectivamente. Logo em seguida, ele lança o processo E de 87 MB. Sabendo que a alocação da partição visa minimizar a fragmentação interna e que o sistema operacional utiliza memória virtual, o valor que corresponde à área da fragmentação interna da memória após a inserção do processo E é de

A) 87 MB.

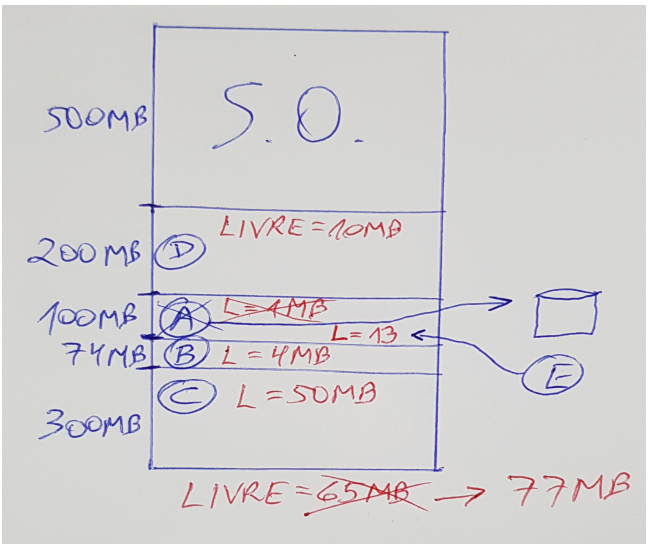
B) 70 MB.

C) 77 MB.

D) 91 MB.

Comentários:

A questão não está muito fácil de ler. Ela quer dizer o seguinte: há 5 partições fixas, sendo que a primeira é destinada ao S.O. (500 MB). As outras 4 possuem os tamanhos 200 MB, 100 MB, 74 MB e 300 MB, nessa ordem. Atente-se à parte que diz "Sabendo que a alocação da partição visa minimizar a fragmentação interna e que o sistema operacional utiliza memória virtual".



Então temos que "encaixar" cada processo na partição que sobre menos espaço "desperdiçado". Acompanhando no desenho ao lado, colocamos os processos na ordem D, A, B, C. Quando surge um processo E (87MB), surge a pergunta: em que partição colocar?
Como não há mais partição livre e há memória virtual, temos que eleger um para tirar da memória (colocar no disco). A questão não fala em algoritmo para isso, mas sabemos que queremos evitar a fragmentação interna. Então devemos escolher a partição em que melhor se "encaixa" o processo E: a partição de tamanho 100 MB. Depois de tirar o processo A da memória e colocar o processo E em "seu lugar", é só fazer o cálculo do espaço livre 🙂.

Gabarito: C

6. (IF-PA/IF-PA/2019) Em relação à estratégia de alocação de partição best-fit, marque a alternativa ERRADA:

A) a melhor partição é escolhida.

B) procura deixar o menor espaço de livre em uma partição.

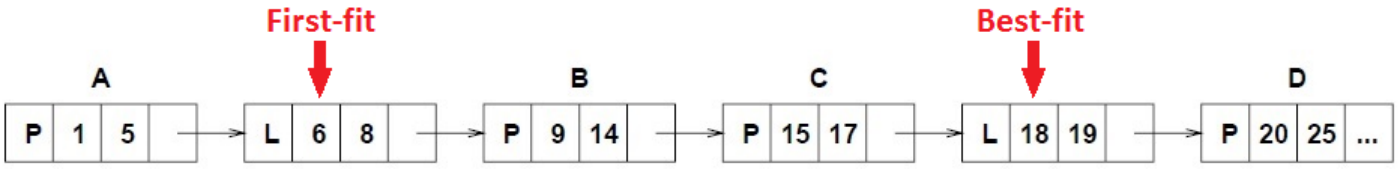
C) a pior partição de tamanho suficiente é escolhida.

D) a lista de partições livres é ordenada por tamanho.

E) aumenta o problema de fragmentação nas partições.

Comentários:

Best-fit: procura pela lista inteira e "pega" o espaço livre de tamanho mais próximo de N. É lento e cria na memória espaços livres pequenos que dificilmente serão alocados. Porém, para N grande, esse algoritmo aumenta as chances de se encontrar na lista um espaço livre de tamanho adequado, visto que minimiza o uso espaços livres grandes para atender requisições pequenas. Imagine que um bloco de tamanho 2 seja solicitado, o algoritmo First-fit alocaria o espaço livre 6-7, e o Best-fit o espaço livre 18-19:



Vamos ver qual está ERRADA:

(A) A melhor partição é escolhida, considerando que "melhor" quer dizer a que mais se "encaixa", sobrando menos espaço de desperdício (que no fim causa fragmentações); (B) Exatamente o que vimos na explicação da A; (C) Essa está errada, é o oposto! A "melhor" é escolhida, conforme vimos na explicação da A; (D) No desenho que vemos acima temos uma única lista de processos e "partes" livres, mas é possível a implementação com listas de partições livres ordenadas por tamanho (inclusive melhora o desempenho); (E) Como o best-fit procura a partição em que o processo melhor se "encaixa", sobram espaços pequenos (o que aumenta a fragmentação nas partições).

Gabarito: C

7. (IF-PA/IF-PA/2019) Em relação à gerência de memória, marque a alternativa CORRETA:

A) a técnica de overlay divide o programa em módulos para ser carregado em memória.

B) a alocação continua simples é utilizada em sistemas multiprogramáveis.

C) a alocação particionada estática é utilizada em sistemas monoprogramados.

D) a técnica de memória virtual combina as memórias cachê e memória principal.

E) a técnica de swapping é utilizada para resolver o problema de velocidade na memória secundária.

Comentários:

Um fato notório na evolução da informática é que o tamanho dos programas vem superando a quantidade de memória disponível para carregá-los. Uma solução que geralmente era adotada no passado era a divisão do programa em partes (overlays). O overlay 0 começa a ser executado primeiro. Quando finaliza, o próximo overlay é executado, e assim sucessivamente. Os overlays eram mantidos em disco e permutados para a memória pelo sistema operacional.

Gabarito: A

8. (IF-PA/IF-PA/2019) Em relação à estratégia de alocação de partição first-fit, marque a alternativa ERRADA:

A) a primeira partição livre de tamanho suficiente é escolhida.

B) realiza vários cálculos consumindo um alto poder de processamento.

C) as listas de áreas livres estão ordenadas crescente e por endereços.

D) quando comparada com as estratégias best-fit e worst-fit é mais rápida.

E) consome menos recurso do sistema.

Comentários:

First-fit poderia ser traduzido para algo como "primeiro a se encaixar". E é isso que ele faz! A primeira partição livre de tamanho suficiente é escolhida e a busca pára nesse momento. Não tem cálculos com alto poder de processamento, só compara o tamanho livre com o tamanho do processo. É mais rápida que best-fit e worst-fit, pois essas duas precisam percorrer toda a lista! Por isso tudo, consome menos recursos! Podem ser implementadas listas com as áreas livres para tornar mais rápida a busca (mas não é obrigatório), ordenadas pelo tamanho livre e endereço. A errada é a alternativa B porque não existem cálculos complexos! Apenas escolhe a primeira partição livre!

Gabarito: B

9. (AOCP/IBGE/2019) Com base nos conceitos de gerenciamento de memória, analise as assertivas e assinale a alternativa que aponta as corretas.

I. Assegurar que dentro da memória principal fique o menor número de processos residentes para processamento.

II. Realizar a permissão e execução de programas que sejam maiores que a memória física disponível para processamento.

III. Realizar a proteção das áreas de memória ocupadas por cada processo.

IV. Uma técnica de gerenciamento de memória é o Swapping que realiza a transferência de processos residentes na memória principal para a memória secundária.

A) Apenas I, II e III.

B) Apenas II, III e IV.

C) Apenas I e III.

D) Apenas II e IV.

E) I, II, III e IV.

Comentários:

I. ERRADA - O gerenciamento de memória se preocupa com a alocação de memória (parte ocupada x livre) e não com a quantidade de processos.

II. CORRETA - Quando um programa for maior que a memória RAM há a possibilidade de uso da memória virtual, que geralmente é implementada nos sistemas operacionais modernos.

III. CORRETA - As áreas de memória ocupadas por processos não podem ser usadas por outros processos.

IV. CORRETA - O Swapping realiza a transferência de processos inteiros na memória principal para a memória secundária, diferente da paginação, que faz a transferência de partes do processo (páginas).

Gabarito: B

10. (AOCP/EMPREL/2019) Qual é o procedimento da gerência de memória em que os programas são divididos em sub-rotinas e estruturas de dados, e depois são colocados em blocos de informações na memória que possuem tamanhos diferentes com seu próprio espaço de endereçamento?

A) Cache.

B) Paginação.

C) Segmentação

D) Hash.

E) Pretty Good Privacy.

Comentários:

Segmentação: os programas são divididos em sub-rotinas e estruturas de dados, em segmentos de tamanhos diversos, cada um com o seu próprio espaço de endereçamento. A alocação da memória ocorre de maneira não fixa, ou seja, cada segmento pode ter um tamanho diferente (a alocação depende da lógica do programa). O mapeamento é feito através das tabelas de mapeamento de segmentos e os endereços são compostos pelo número do segmento e um deslocamento dentro do segmento (ex.: segmento 0, deslocamento = 500 bytes).

Gabarito: C

11. (FURB/Pref. de Porto Belo-SC/2019) O sistema X utiliza paginação com tabela de página armazenada em memória e TLBs. Cada referência à memória leva 200 nanossegundos e 75% de todas as referências à tabela de página são encontradas nas TLBs. Considere que o sistema X leva tempo zero para encontrar uma entrada de página nas TLBs, caso a entrada exista. Qual o tempo efetivo de referência à memória do Sistema X?

A) 250 nanossegundos.

B) 50 nanossegundos.

C) 150 nanossegundos.

D) 350 nanossegundos.

E) 200 nanossegundos.

Comentários:

A TLB (*Translation Lookaside Buffer*) é uma memória cache especializada que armazena as traduções de endereços virtuais para endereços físicos. Quando um programa faz referência a um endereço virtual, a TLB é consultada para encontrar a correspondência do endereço virtual com o endereço físico correspondente na memória RAM. A TLB acelera o acesso à memória, traduzindo endereços virtuais em endereços físicos de forma eficiente. Sem a TLB, o sistema precisaria consultar a tabela de páginas toda vez que uma referência de memória fosse feita, o que seria muito mais lento do que o acesso direto à TLB.

O enunciado nos diz que:

- Cada referência à memória leva 200 nanossegundos;
- 75% de todas as referências à tabela de página são encontradas na TLB;
- Então, 25% das referências não são encontradas na TLB, ou seja, precisam ser consultadas na tabela de páginas, que está na memória RAM, e depois um novo acesso à memória com o endereço consultado na tabela. Resumindo, temos dois acessos = 400 nanossegundos.

Agora vamos ao cálculo:

(0,75 * 200) + (0,25 * 400) = 150 + 100 = 250 nanossegundos.

Gabarito: A

12. (FURB/Pref. de Porto Belo-SC/2019) Em se tratando de gerência de memória, dadas cinco partições de memória e quatro processos, com seus respectivos tamanhos:

Partições	Processos
Partição 1: 200 KB	Processo A: 420 KB
Partição 2: 1000 KB	Processo B: 880 KB
Partição 3: 400 KB	Processo C: 240 KB
Partição 4: 600 KB	Processo D: 900 KB
Partição 5: 1200 KB	

Um algoritmo de alocação foi utilizado e, como resultado, obteve-se a seguinte sequência de alocações:

A foi alocado na partição 2. B foi alocado na partição 5. C foi alocado na partição 5. D espera a próxima partição disponível.

Qual dos seguintes algoritmos foi utilizado?

A) Best-fit.

B) Worst-fit.

C) First-fit.

D) Circular-fit.

E) Last-fit.

Comentários:

Circular-fit: utiliza uma lista circular de blocos de memória livres. A lista circular é percorrida para encontrar um bloco de tamanho adequado para atender a uma solicitação de alocação de memória. Funciona assim:

- A memória livre é organizada em uma lista circular;
- Quando uma solicitação de alocação de memória é recebida, a lista circular é percorrida em busca do primeiro bloco de memória livre que atenda aos requisitos da solicitação;
- Se um bloco adequado for encontrado, ele é alocado para o processo. Caso contrário, a busca continua no próximo bloco da lista circular;
- A lista é percorrida até que um bloco adequado seja encontrado ou até que a lista completa seja verificada.

Vamos aplicar o algoritmo circular-fit na situação proposta e veremos na figura como fica:

- A (420 KB) não cabe na partição 1, mas cabe na 2;
- B (880 KB) não cabe na partição 1, não cabe no que restou da partição 2, não cabe nas partições 3 e 4, mas cabe na 5;
- C (240 KB) cabe no que restou da partição 5;
- D (900 KB) não cabe em nenhuma partição e em nenhum resto de partição, então fica na espera.

Tamanho:	200 KB	1000 KB		400 KB	600 KB	1200 KB	
Segmento:	1	2		3	4	5	
Processo:		A = 420 KB				B = 880 KB	C = 240 KB

Gabarito: D

13. (IBFC/TRE-PA/2020) Caso, em um Sistema Operacional tradicional, comece a ocorrer o uso da capacidade total da memória RAM, o Sistema Operacional passará a utilizar:

A) os registradores

B) a memória virtual

C) a memória ROM

D) a memória cache

Comentários:

Uma solução para a falta de memória RAM é o uso de uma memória virtual, como se fosse uma extensão da memória física. Essa "extensão" existe com o uso da memória secundária (ex.: HD) e o gerenciamento de memória do sistema operacional fica responsável em carregar/descarregar os processos ou partes dele (páginas) na memória RAM e/ou na memória secundária.

Gabarito: B

14. (IBADE/Pref. de Vila Velha-ES/2020) Seja uma memória virtual com 3 blocos e que use o algoritmo LRU (Least Recently Used) como seu algoritmo de substituição de páginas. Admitindo-se que ocorra a seguinte sequência de referência às páginas de memórias: 1,2,3,4,2,3,4,2. Assumindo que inicialmente todos os blocos estão vazios, quantas interrupções de páginas ausentes (page faults) ocorrerão?

A) 5

B) 4

C) 6

D) 7

E) 3

Comentários:

Vamos aos dados:

- Algoritmo: LRU (Menos Recentemente Usado);
- Memória virtual com 3 blocos;
- Blocos vazios no começo;
- Sequência de referência às páginas de memória: 1, 2, 3, 4, 2, 3, 4, 2.

Vamos lá:

Início = Vazio (x - x - x)

1? Page fault! Como está vazio, preenche com 1: 1 - x - x.

2? Page fault! Como tem espaço, preenche com 2: 1 - 2 - x.

3? Page fault! Como tem espaço, preenche com 3: 1 - 2 - 3.

4? Page fault! Substitui o 1 (menos recentemente usado) por 4: 4 - 2 - 3

A partir desse ponto os acessos (2, 3, 4, 2) não geram page faults, pois sempre estão na memória!

Total = 4 page faults.

Gabarito: B

15. (FAPEC/UFMS/2020) Visando a melhorar a tradução de endereços da memória virtual, existe uma memória cache cuja função é diminuir os acessos à tabela de páginas. Assinale a alternativa que representa tal memória.

A) Cache Write-trough.

B) Cache Multi-nível.

C) Translation Lookaside Buffer (TLB).

D) Registrador de tabela de páginas (RTP).

E) Cache Write-back.

Comentários:

TLB (*Translation Lookaside Buffer*): é uma memória cache especializada que armazena as traduções de endereços virtuais para endereços físicos. Quando um programa faz referência a um endereço virtual, a TLB é consultada para encontrar a correspondência do endereço virtual com o endereço físico correspondente na memória RAM. A TLB acelera o acesso à memória, traduzindo endereços virtuais em endereços físicos de forma eficiente. Sem a TLB, o sistema precisaria consultar a tabela de páginas toda vez que uma referência de memória fosse feita, o que seria muito mais lento do que o acesso direto à TLB.

Quando um programa faz referência a um endereço virtual, a TLB é a primeira a ser consultada. Se houver uma correspondência na TLB ("hit"), o endereço físico é recuperado. Se não houver uma correspondência ("miss" - uma falta), o sistema realiza uma consulta à tabela de páginas, atualiza a TLB com a nova tradução e, em seguida, acessa a memória RAM.

Gabarito: C

16. (SELECON/EMGEPRON/2021) No gerenciamento de memória, uma técnica nos sistemas operacionais permite que um programa possa ser espalhado por áreas não contíguas de memória com as características listadas a seguir.

• A memória física é dividida em páginas com tamanho fixo igual ao da página lógica, sendo cada programa carregado página a página, com cada página lógica ocupando uma página física, não necessariamente contíguas.

• O endereço lógico é inicialmente dividido em duas partes: um número de página lógica e um deslocamento dentro da página. O número da página lógica é usado como índice no acesso à tabela de páginas, de forma a obter o número da página física correspondente.

• Não existe fragmentação externa, existindo a interna.

• Além da localização, a tabela de páginas armazena também o bit de validade, sendo (1) se a página está na memória e (0) se a página não está.

• As páginas de processo podem ser transferidas para a memória por demanda, carregando apenas o que é necessário para a execução do programa, e o sistema tenta prever as páginas que serão necessárias à execução do programa.

- Páginas constantemente referenciadas em um processo devem permanecer na memória.

A técnica descrita é conhecida por:

A) alocação

B) fragmentação

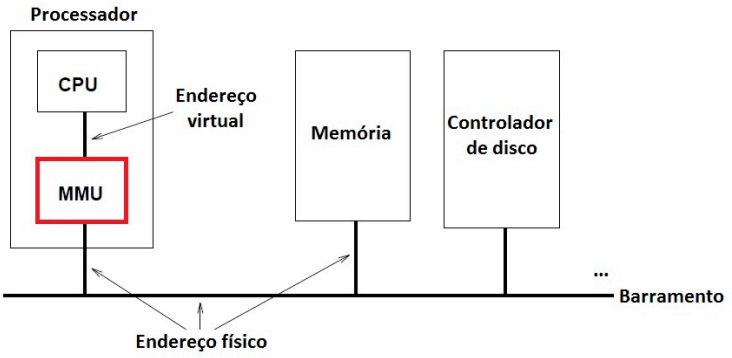
C) segmentação

D) paginação

Comentários:

Paginação: técnica utilizada pela maioria dos sistemas com memória virtual. Em qualquer computador existe um determinado conjunto de endereços de memória que programas podem referenciar. Trata-se de endereços virtuais que formam o espaço virtual de endereçamento do processo. Em computadores sem memória virtual, o endereço virtual é colocado diretamente no barramento de memória, uma palavra da memória física com o mesmo endereço é lida ou escrita.

Com o uso da memória virtual, os endereços de memória não vão diretamente para o barramento de memória, eles vão à unidade de gerenciamento de memória (MMU - *Memory Management Unit*), um hardware específico que mapeia os endereços virtuais em endereços da memória física:

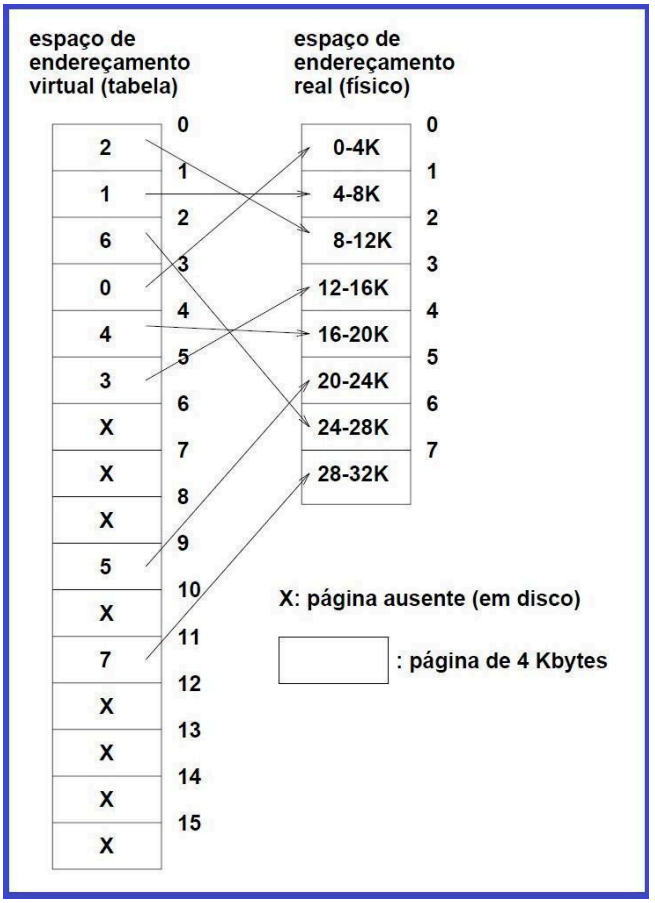


Vamos ver um exemplo de um computador que pode gerar endereços de 16 bits (2^{16} = 65536 = 64KB). Esses são os endereços virtuais. Agora imagine que tal computador tenha somente 32KB de memória física, então embora programas de 64KB possam ser escritos, eles não podem ser carregados totalmente para a memória para serem executados. Uma cópia completa do programa deve estar presente no disco e segmentos desse programa podem ser carregados para a memória pelo sistema na medida em que se tornem necessários.

O espaço de endereçamento virtual é dividido em unidades chamadas páginas. As unidades correspondentes na memória física são chamadas quadros (*page frames*). Para não ter erro pense no seguinte: considere uma página como uma tela de pintura que deve ser colocada em um quadro (uma moldura).

As páginas e os quadros são sempre do mesmo tamanho. No exemplo ao lado elas são de 4 KB. Com 64 KB de espaço de endereço virtual e 32 KB de memória física, temos 16 páginas e 8 quadros. As transferências entre a memória e o disco sempre são realizadas em unidades de páginas (4 KB, no nosso exemplo).

Quando um programa tenta acessar o endereço 5000, por exemplo, o endereço virtual 9000 é enviado para a MMU. Como temos páginas de 4 KB (4096 bytes), sabemos que a página 0 possui os endereços 0-4095, a página 1 os endereços 4096-8191, a página 2 os endereços 8192-12287, e assim por diante. A tabela de páginas tem a função de mapear as páginas virtuais em molduras de página física.



Gabarito: D

17. (AOCP/Câmara de Teresina-PI/2021) Sobre os conceitos de gerenciamento de memória em sistemas operacionais, assinale a alternativa correta.

A) Em sistemas sem memória virtual, o gerenciamento de memória não é necessário.

B) Memória virtual é incompatível com o conceito de Paginação.

C) Técnicas podem ser aplicadas para permitir que a memória física seja estendida por outros espaços de armazenamento, como o disco.

D) Sem memória virtual não se pode realizar troca de páginas.

E) Alocação de memória dinamicamente (heap) não é permitida sem memória virtual.

Comentários:

A) ERRADA - O gerenciamento de memória RAM é necessário. Pode ter a memória virtual ou não, sendo que na prática os sistemas operacionais modernos possuem memória virtual.

B) ERRADA - Paginação é uma das técnicas (a mais usada) de memória virtual.

C) CORRETA - Trata-se da memória virtual!

D) ERRADA - Poder realizar a troca de página até pode, mas não teria muito sentido! Alternativa chatinha essa.

E) ERRADA - A alocação de memória dinâmica realizada pelo programador através de funções apropriadas não depende de memória virtual. O programador nem sabe se o sistema terá ou não memória virtual!

Gabarito: C

18. (IDECAN/IF-CE/2021) No que diz respeito aos sistemas operacionais, analise as afirmativas a seguir:

I. O recurso de memória virtual consiste em criar tabelas que relacionam posições virtuais e reais da RAM para um mesmo aplicativo. Os programas ficam integradas como uma só unidade na memória física e não de forma distribuída, passando uma ideia de que os programas utilizam a memória de forma distribuída e compartilhada.

II. O recurso da paginação é uma técnica de gerência de memória que permite a um programa ser espalhado por áreas não contíguas de memória. Com isso, o espaço de endereçamento lógico de um processo é dividido em unidades lógicas de tamanho fixo e a memória física é dividida em páginas com tamanho fixo, com tamanho igual ao da página lógica.

III.O recurso da segmentação é uma técnica de gerência de memória em que os programas são divididos em segmentos de tamanhos variados cada um com seu próprio espaço de endereçamento.

Assinale

A) se apenas as afirmativas I e II estiverem corretas.

B) se apenas as afirmativas I e III estiverem corretas.

C) se apenas as afirmativas II e III estiverem corretas.

D) se todas as afirmativas estiverem corretas.

Comentários:

I. ERRADA - Não existem posições virtuais da memória RAM. Uma memória secundária (ex.: HD) é utilizada para armazenamento dos processos (ou parte deles) que não estão em uso e um algoritmo é utilizado para gerenciar isso.

II. CORRETA - São as páginas e os quadros de página. Dessa forma partes do processo podem estar na memória física e o restante na memória secundária, de acordo com a necessidade.

III. CORRETA - Os segmentos são criados de acordo com os tamanhos dos programas (processos), o que pode criar fragmentos externos (entre os segmentos).

Gabarito: C

19. (IDECAN/IF-CE/2021) No que tange ao funcionamento dos sistemas operacionais, existem situações em que não é possível manter todos os processos na memória.

O sistema operacional utiliza uma técnica aplicada à gerência de memória, para processos que esperam por memória livre para serem processados. Essa técnica visa melhorar o problema da insuficiência de memória durante a execução de alguns processos em ambientes que operam em multiprogramação; consiste em transferir automaticamente todo o processo da memória principal para o disco e vice-versa. Essa técnica é denominada

A) sweeping.

B) swapping.

C) scanning.

D) spoofing.

Comentários:

Há momentos em que não há memória principal suficiente para todos os processos ativos, de maneira que os processos excedentes devem ser mantidos em disco e trazidos para a memória dinamicamente. Para isso pode ser usada uma das seguintes estratégias:

- Swapping: traz cada processo em sua totalidade, executa-o por algum tempo e o coloca novamente no disco;
- Memória virtual: os programas podem ser executados mesmo estando apenas parcialmente na memória principal.

Gabarito: B

20. (Quadrix/CFO-DF/2022) Em relação à estrutura do sistema operacional, ao gerenciamento de processos e à memória virtual, julgue o item.

Na memória virtual, quando o programa faz referência a uma parte de seu espaço de endereçamento que não está em sua memória física, o sistema operacional é alertado para obter a parte que falta e reexecutar a instrução que falhou.

Comentários:

Quando um programa quiser acessar um endereço relativo a uma página que não esteja na memória RAM, a MMU verifica que a página não está mapeada e força a CPU a causar uma interrupção (*trap*) no sistema operacional. Essa interrupção é chamada falta de página (*page fault*). O S.O. remove um quadro, de acordo com um algoritmo (o quadro menos usado, por exemplo), e escreve seu conteúdo de volta no disco. Ele então busca a página referenciada para o quadro liberado, atualiza o mapa, e retorna à instrução interrompida.

Gabarito: Certo

Lista de Questões - CEBRASPE

1. (CEBRASPE/TRE-PI/2016) O componente central de um sistema operacional, que determina o local da memória onde deverá ser colocado o código de um novo processo chamado para ser executado por um processo pai, lido de um arquivo previamente armazenado em um dispositivo de entrada e saída, que, por sua vez, está conectado à rede local, é denominado

A) gerenciador de sistema de arquivos.

B) gerenciador de comunicação interprocessos.

C) gerenciador de memória.

D) escalonador de processos.

E) gerenciador de entrada e saída.

2. (CEBRASPE/TRF1/2017) Na técnica denominada escalonamento de processos, o sistema operacional mantém parte do espaço de endereçamento de um processo na memória principal e parte em dispositivo de armazenamento secundário, realizando trocas de trechos de código e de dados entre eles, de acordo com a necessidade.

3. (CEBRASPE/Polícia Federal/2018) A técnica de swapping consiste em transferir temporariamente um processo da memória para o disco do computador e depois carregá-lo novamente em memória.

4. (CEBRASPE/EBSERH/2018) Uma das técnicas mais complexas para o gerenciamento do uso de memória é o mapa de bites, que consiste em manter uma lista encadeada de segmentos de memória alocados e disponíveis.

5. (CEBRASPE/IFF/2018) Em sistemas operacionais que usam paginação para gerenciamento de memória, os algoritmos de troca de página escolhem uma página a ser removida da memória para que uma nova seja alocada. Em um desses algoritmos, uma página modificada que não tenha sido referenciada pelo menos no último tique de relógio (tipicamente em 20 milissegundos) é removida em vez de uma página não modificada que tenha sido muito usada. Trata-se do algoritmo denominado

A) troca ótima de página.

B) troca de página não recentemente usada (NRU).

C) troca de página FIFO.

D) segunda chance.

E) troca de página menos recentemente usada (LRU).

6. (CEBRASPE/Min. da Economia/2020) Julgue o próximo item, relativos a sistemas operacionais.

O gerenciamento de memória pode ocorrer por meio do método básico, no qual um processo que está para ser executado tem suas páginas carregadas em quaisquer quadros de memória disponíveis a partir de sua origem, por exemplo, de um sistema de arquivos.

7. (CEBRASPE/Prefeitura de Barra dos Coqueiros-SE/2020) Assinale a opção correta, a respeito de sistemas operacionais (SO).

A) Swapping é uma técnica que permite subdividir a memória endereçável do SO.

B) Um processo pode ser alocado ou na quantidade exata de memória exigida (exemplo de particionamento variável, mais eficiente) ou na menor partição possível (exemplo de particionamento fixo, mais simples).

C) Cabe ao SO prover e gerenciar acesso controlado aos arquivos e programas, enquanto, por questão de desempenho, cabe somente ao hardware o acesso aos dispositivos de E/S.

D) Escalonador é a parte que contém as funções básicas de um SO e as que são usadas com mais frequência.

E) Comparado ao RAID 0, o RAID 1 possui como vantagem a necessidade de usar apenas a metade de espaço em disco, porém tem como desvantagem a indisponibilidade dos dados em caso de falha.

8. (CEBRASPE/PG-DF/2021) No que se refere a sistemas operacionais, julgue o próximo item.

Quando a memória é gerenciada por mapa de bites, o valor 1 indica que a unidade referenciada está ocupada e o valor 0 indica que ela está livre.

9. (CEBRASPE/PG-DF/2021) No que se refere a sistemas operacionais, julgue o próximo item.

A paginação de memória é executada pelo sistema operacional em três etapas: criação do processo, término do processo e limpeza do buffer residual.

GABARITO

GABARITO

1. Letra C
2. Errado
3. Certo
4. Errado
5. Letra B
6. Certo
7. Letra B
8. Certo
9. Errado

Lista de Questões - FGV

1. (FGV/IBGE/2017) A estratégia de alocação de memória que busca o menor espaço livre suficiente para satisfazer cada requisição denomina-se:

A) minor fit;

B) save fit;

C) best fit;

D) first fit;

E) worst fit.

2. (FGV/AL-RO/2018) A função dos computadores é executar processos mas para que os processos sejam executados, eles devem estar na memória do computador. O gerenciamento da memória é uma das funções dos sistemas operacionais.

   Uma técnica empregada pelo sistema operacional para permitir que processos não carregados na memória principal sejam executados, a partir da emulação de uma memória significativamente maior, é

A) o swapping.

B) a memória física.

C) a memória virtual.

D) a paginação.

E) o cache.

3. (FGV/MPE-AL/2018) A implementação de linguagens de programação em geral depara-se com a questão do gerenciamento de memória. Nesse contexto, assinale a opção que melhor descreve o que é conhecido como "memory leak".

A) A liberação de trechos de memória ainda em uso por um ou mais objetos.

B) A impossibilidade de liberar a memória ocupada por objetos que se tornaram inalcançáveis.

C) A incapacidade de recuperar trechos de memória virtualizados.

D) A invasão de trechos de memória por objetos não autorizados.

E) O compartilhamento indesejado de trechos de memória por dois ou mais objetos.

4. (FGV/TJ-TO/2022) A estratégia de swapping utilizada em sistemas operacionais para gerenciamento de memória consiste em:

A) dividir a memória em unidades de alocação em kbytes endereçáveis por um mapa de bit;

B) trazer para a memória principal cada processo, executá-lo por um tempo e então colocá-lo de volta na memória secundária;

C) segmentar os processos maiores que a memória principal em subprocessos menores executáveis sob demanda;

D) fracionar a memória principal e secundária em unidades básicas de mesmo tamanho para evitar fragmentação;

E) espalhar os processos sobre áreas não contínuas da memória principal para permitir extrapolarem o espaço inicial alocado.

5. (FGV/MPE-GO/2022) Sistemas operacionais têm um importante papel no gerenciamento de memória de um computador, especialmente na recuperação de nacos de memória que foram descartados, seja intencionalmente ou pela ocorrência de erros.

   Essa recuperação é usualmente realizada por um processo denominado

A) Garbage collector.

B) Kernel.

C) Memory Launcher.

D) Memory Retrieval.

E) Scheduler.

6. (FGV/TRT13/2022) A estratégia de gerenciamento adotada em sistemas operacionais modernos que consiste em trazer para memória principal cada processo em sua totalidade, executá-lo por um tempo e então colocá-lo de volta no disco é denominada

A) buffering.

B) defragging.

C) offset.

D) relocation.

E) swapping.

7. (FGV/Banco do Brasil/2023) No contexto dos sistemas operacionais modernos, swapping consiste em

A) subdividir a memória física em pequenas partições para permitir uma utilização mais eficiente.

B) reordenar o espaço de armazenamento, fazendo com que todo arquivo esteja armazenado de forma contígua.

C) mover dados para uma área de trabalho temporária para um programa externo acessar para processá-lo em tempo futuro.

D) forçar o núcleo do processador a operar numa frequência mais alta do que a especificada pelo fabricante.

E) trazer cada processo em sua totalidade para memória RAM, executá-lo por um tempo e então colocá-lo de volta no disco.

8. (FGV/FHEMIG/2023) No contexto do gerenciamento de memória, programas e processos, assinale a afirmativa incorreta.

A) Paginação é uma técnica de gerenciamento de memória que divide o espaço de endereçamento dos processos em blocos fixos de tamanho uniforme chamados "páginas".

B) Swapping é um mecanismo de gerenciamento de memória que permite ao sistema operacional transferir, temporariamente, processos ou partes deles da memória RAM para o armazenamento secundário (como o disco rígido) quando a memória principal está escassa, liberando espaço para outros processos em execução.

C) Segmentação é uma técnica de gerenciamento de memória que divide os processos em blocos lógicos de tamanhos variáveis, para melhor aproveitamento do espaço e organização das informações em sistemas operacionais.

D) Caching é uma técnica de gerenciamento de memória que armazena permanentemente os dados mais usados na memória secundária, reduzindo a necessidade de acesso à memória RAM e melhorando o desempenho geral do sistema.

E) Thrashing é um fenômeno no gerenciamento de memória virtual em que o sistema operacional gasta a maior parte do tempo trocando páginas entre a memória principal e a memória secundária, resultando em um desempenho significativamente reduzido.

9. (FGV/SMPOG de Belo Horizonte-MG/2023) A paginação é uma técnica de gerenciamento de memória comumente implementada em sistemas operacionais modernos. Sobre a paginação, assinale a afirmativa correta.

A) O comprimento das páginas do sistema operacional pode mudar durante a execução de um programa.

B) É necessário adicionar memória física na máquina para ampliar o espaço de endereçamento linear dos programas.

C) Uma página lógica pode ser carregada em qualquer página física que esteja livre.

D) Os programas que ocupam múltiplas páginas físicas causam a fragmentação externa da memória.

GABARITO

GABARITO

1. Letra C
2. Letra C
3. Letra B
4. Letra B
5. Letra A
6. Letra E
7. Letra E
8. Letra D
9. Letra C

Lista de Questões - FCC

1. (FCC/CREMESP/2016) Em um computador, quando o sistema de memória virtual está habilitado, a memória RAM é dividida em blocos chamados de páginas. Quando a memória RAM está toda ocupada e o Sistema Operacional ou algum software necessita de mais memória, para liberar espaço, o processador armazena o conteúdo de páginas de memória que não estão sendo usadas naquele exato momento

A) no swap file.

B) na memória cache.

C) nos registradores de dados.

D) na Unidade Lógica de Armazenamento – ULA.

E) na virtual machine.

2. (FCC/ARTESP/2017) No gerenciamento de memória, o mecanismo de paginação utiliza algoritmos de substituição de páginas, que são políticas definidas para escolher qual página da memória será removida para dar lugar a uma página que foi solicitada e que precisa ser carregada. Dentre estes encontra-se o algoritmo

A) FIFO – First In First Out que consiste em substituir a última página que foi carregada na memória. Esta escolha não leva em consideração se a página está sendo muito utilizada ou não, assim a quantidade de falta de páginas tende a diminuir quando o tamanho da memória também diminui.

B) NRU – Not Recently Used que procura por páginas que não foram referenciadas nos últimos acessos para serem substituídas e também verifica, através de um bit de modificação, se a página teve seu conteúdo alterado durante sua permanência em memória.

C) LIFO – Last In First Out, que consiste em substituir a página que foi carregada há mais tempo na memória. Na ocorrência de uma falta de página, a primeira página da lista será substituída e a nova será acrescentada ao final da lista.

D) CLOCK, que faz a substituição da última página acessada. Este escolhe a última página acessada para ser substituída. Dessa forma, é possível explorar com mais eficiência o princípio de localidade temporal apresentada pelos acessos.

E) MRR – Most Recently Removed, que mantém todas as páginas em uma lista circular. A ordem mantida segue a sequência em que elas foram carregadas em memória. Além disso, é adicionado um byte de uso que indica se a página foi referenciada novamente depois de ter sido carregada.

3. (FCC/DPE-AM/2018) Em um computador com um sistema operacional típico nem sempre é possível manter na memória todos os processos por falta de espaço. Uma solução comumente adotada nessas situações é a utilização de uma área no disco para a colocação de processos que estejam momentaneamente bloqueados. Esse mecanismo é conhecido como

A) swapping.

B) paginação.

C) particionamento.

D) segmentação.

E) compactação.

4. (FCC/TRF4/2019) No sistema de memória virtual por paginação, determinado problema ocorre em dois níveis:

   – No próprio processo: elevado número de page faults; processo passa mais tempo esperando por páginas do que executando.

   – No sistema: existem mais processos competindo por memória principal do que espaço disponível.

   A solução seria reduzir o número de páginas de cada processo na memória.

   O problema se refere

A) à excessiva transferência de blocos entre a memória principal e a secundária, conhecida como thrashing.

B) à excessiva seleção de processos para saírem da memória, denominada swapping.

C) ao overflow da tabela que controla o esquema de mapeamento associativo.

D) à inexistência do Translation Lookside Buffer (TLB).

E) à política de paginação antecipada, conhecida como cache hit.

5. (FCC/AL-AP/2020) Quando um processo do sistema operacional tem mais espaço de endereçamento do que o computador tem de memória principal e o processo deseja utilizá-lo inteiramente, isso

A) incorrerá em erro no espaço de endereçamento, travando a máquina.

B) incorrerá em erro no espaço de endereçamento, todavia, o ciclo de processamento se completará solicitando intervenção externa.

C) pode ser resolvido pelo uso de memória virtual.

D) evidencia erro de planejamento, obrigando o administrador a adquirir mais recursos de memória.

E) não tem como ocorrer, pois todos os processos que são alocados em um computador já são estabelecidos dentro dos recursos computacionais existentes.

GABARITO

GABARITO

1. Letra A
2. Letra B
3. Letra A
4. Letra A
5. Letra C

Lista de Questões - VUNESP

1. (VUNESP/Pref. de Itapevi-SP/2019) Memória Virtual é uma técnica de gerenciamento de memória de computadores muito utilizada, que pode ser implementada com o emprego de dois mecanismos:

A) a paginação e a fragmentação.

B) a paginação e a segmentação.

C) a segmentação e o particionamento.

D) a virtualização e o swapping.

E) o swapping e o particionamento.

2. (VUNESP/Pref. de Ilhabela-SP/2020) Em sistemas operacionais, a técnica conhecida como swapping

A) traz totalmente cada processo para a memória, executa-o por algum tempo e retorna-o para o disco posteriormente.

B) permite que um processo seja executado apenas com uma parte do código na memória, podendo a outra parte ficar permanentemente em disco.

C) permite que um processo possa ser executado diretamente no disco no qual ele esteja armazenado, sem precisar ser carregado na memória.

D) é uma boa alternativa para os computadores modernos, por ser econômica e de maior rapidez na execução dos processos, possibilitando que se instale menos memória RAM no computador.

E) apresenta como vantagem a redução do uso de operações de entrada/saída.

GABARITO

GABARITO

1. Letra B
2. Letra A

Lista de Questões - CESGRANRIO

1. (CESGRANRIO/LIQUIGÁS/2018) A gerência de memória efetuada pelos sistemas operacionais inclui a definição de uma política para a substituição de páginas de memória, que trata da escolha de uma página da memória principal que deve ser substituída quando uma nova página precisa ser carregada.

   Dentre as políticas básicas mais utilizadas, há uma que opta por substituir a página que está há mais tempo sem ser referenciada, sendo conhecida como

A) FIFO.

B) LIFO.

C) Optimal.

D) LRU.

E) Clock Policy.

2. (CESGRANRIO/Petrobras/2018) Na implementação da memória virtual em sistemas operacionais, é necessário definir, dentre outros elementos, a política de substituição de páginas. Esses algoritmos escolhem qual página deverá deixar a memória real (chamada página vítima) e voltar ao meio secundário de armazenamento, a fim de ceder espaço à outra página requisitada pelo processador e atender à requisição de falha de página. Uma das formas de implementação desses mecanismos de substituição de páginas prevê a utilização de bits auxiliares associados às páginas.

   Nesse contexto, qual a função do bit de sujeira (dirty bit)?

A) Evitar a utilização de um algoritmo de substituição de páginas da classe local.

B) Evitar que uma página seja eleita como "vítima", ou seja, tenha de deixar a memória.

C) Indicar quando uma página foi alterada durante a execução de um processo.

D) Indicar se a página é candidata à ocorrência de trashing.

E) Indicar se a página foi referenciada dentro de um intervalo de tempo.

3. (CESGRANRIO/Petrobras/2018) A gerência de memória de um sistema operacional, dentre outras funções, define a política que será utilizada para escolher a região da memória que um processo ocupará ao ser carregado para execução.

   O processo de alocação e liberação das regiões da memória, dependendo da política escolhida, pode ocasionar situações em que pequenas regiões livres nos espaços entre regiões alocadas a outros processos se tornem difíceis de ser utilizadas, pois seu tamanho não comporta facilmente outros processos, configurando um fenômeno conhecido como

A) fragmentação

B) interrupção

C) deadlock

D) starvation

E) troca de contextos

4. (CESGRANRIO/Transpetro/2018) A gerência de memória visa a maximizar o número de usuários e aplicações, utilizando de forma eficiente o espaço da memória principal. A política de substituição de páginas visa a selecionar, dentre as páginas alocadas, qual deverá ser liberada.

   O algoritmo de substituição de página que seleciona a página na memória principal que está há mais tempo sem ser referenciada é o

A) Clock

B) LFU

C) LRU

D) FIFO

E) FIFO com buffer

5. (CESGRANRIO/LIQUIGÁS/2018) Uma vantagem da memória virtual é permitir um número maior de processos compartilhando a memória principal. Há uma técnica de memória virtual na qual o espaço de endereçamento virtual e o espaço de endereçamento real são divididos em blocos de mesmo tamanho.

   Essa técnica é chamada de

A) paginação

B) segmentação

C) swapping

D) mirrowing

E) striping

6. (CESGRANRIO/Caixa/2021) Na técnica de memória virtual, a tarefa de tradução dos endereços virtuais é realizada por hardware, juntamente com o sistema operacional, para não comprometer o desempenho. O dispositivo de hardware responsável por essa tradução é a

A) FPU

B) ALU

C) MMU

D) VU

E) Cache

GABARITO

GABARITO

1. Letra D
2. Letra C
3. Letra A
4. Letra C
5. Letra A
6. Letra C

Lista de Questões - MULTIBANCAS

1. (Cetro/Dataprev/2016) Sobre as técnicas de gerenciamento de memória, assinale a alternativa que apresenta uma característica do gerenciamento de memória virtual por paginação

A) O armazenamento é realizado por meio de endereçamento sequencial denominado "segmento".

B) Gera fragmentação externa.

C) Divide o espaço de endereçamento em blocos de tamanhos variados.

D) Gera a fragmentação interna.

E) Gera a fragmentação interna e a fragmentação externa.

2. (COMPERVE/UFRN/2018) Considere as afirmativas abaixo referentes à ocorrência de uma falta de página no gerenciamento de memória de um computador.

I É responsabilidade do sistema operacional detectar uma falta de página.

II O sistema operacional é acionado através de uma interrupção de hardware.

III O processo que sofre a falta de página passa para o estado de espera até que a página seja carregada na memória cache.

IV A falta de página é corrigida com a carga da página em falta, da memória virtual para a memória física.

Estão corretas as afirmações

A) II e III.

B) I e III.

C) II e IV.

D) I e IV.

3. (FAURGS/TJ-RS/2018) Em relação à paginação, assinale a alternativa correta.

A) É uma forma de alocação não contígua de memória para o espaço de endereçamento de um processo.

B) Apresenta como vantagem gerar fragmentação externa apenas na última página do processo.

C) Divide o espaço de endereçamento do processo em quatro páginas: código, dados, heap (monte) e pilha.

D) Elimina a fragmentação interna e reduz a fragmentação externa.

E) É um método de gerência de memória usado apenas em sistemas que empregam memória real.

4. (FUNDATEC/AL-RS/2018) Para a gerência de memória de um sistema operacional, existem algoritmos de substituição de página. Um deles, de baixa sobrecarga, possui o seguinte modo de operação: (1) a primeira página a entrar é a primeira a sair; (2) pode ser implementado através de uma lista de todas as páginas correntemente na memória, sendo que a página mais antiga ocupa o início dessa lista e a mais recente ocupa o fim; e (3) quando falta uma página, a mais antiga é retirada e a nova é colocada no fim da lista. Trata-se do algoritmo:

A) FIFO (First In, First Out).

B) LRU (Least Recently Used).

C) NRU (Not Recently Used).

D) Ótimo.

E) Segunda chance.

5. (COMPERVE/UFRN/2018) Suponha uma memória composta por 5 partições fixas, sendo elas de 500MB (reservado e totalmente ocupado pelo sistema operacional), 200MB, 100MB, 74MB e 300MB, exatamente nesta ordem. O usuário lançou 4 processos A, B, C e D de tamanhos 99MB, 70MB, 250MB e 190MB, respectivamente. Logo em seguida, ele lança o processo E de 87 MB. Sabendo que a alocação da partição visa minimizar a fragmentação interna e que o sistema operacional utiliza memória virtual, o valor que corresponde à área da fragmentação interna da memória após a inserção do processo E é de

A) 87 MB.

B) 70 MB.

C) 77 MB.

D) 91 MB.

6. (IF-PA/IF-PA/2019) Em relação à estratégia de alocação de partição best-fit, marque a alternativa ERRADA:

A) a melhor partição é escolhida.

B) procura deixar o menor espaço de livre em uma partição.

C) a pior partição de tamanho suficiente é escolhida.

D) a lista de partições livres é ordenada por tamanho.

E) aumenta o problema de fragmentação nas partições.

7. (IF-PA/IF-PA/2019) Em relação à gerência de memória, marque a alternativa CORRETA:

A) a técnica de overlay divide o programa em módulos para ser carregado em memória.

B) a alocação continua simples é utilizada em sistemas multiprogramáveis.

C) a alocação particionada estática é utilizada em sistemas monoprogramados.

D) a técnica de memória virtual combina as memórias cachê e memória principal.

E) a técnica de swapping é utilizada para resolver o problema de velocidade na memória secundária.

8. (IF-PA/IF-PA/2019) Em relação à estratégia de alocação de partição first-fit, marque a alternativa ERRADA:

A) a primeira partição livre de tamanho suficiente é escolhida.

B) realiza vários cálculos consumindo um alto poder de processamento.

C) as listas de áreas livres estão ordenadas crescente e por endereços.

D) quando comparada com as estratégias best-fit e worst-fit é mais rápida.

E) consome menos recurso do sistema.

9. (AOCP/IBGE/2019) Com base nos conceitos de gerenciamento de memória, analise as assertivas e assinale a alternativa que aponta as corretas.

I. Assegurar que dentro da memória principal fique o menor número de processos residentes para processamento.

II. Realizar a permissão e execução de programas que sejam maiores que a memória física disponível para processamento.

III. Realizar a proteção das áreas de memória ocupadas por cada processo.

IV. Uma técnica de gerenciamento de memória é o Swapping que realiza a transferência de processos residentes na memória principal para a memória secundária.

A) Apenas I, II e III.

B) Apenas II, III e IV.

C) Apenas I e III.

D) Apenas II e IV.

E) I, II, III e IV.

10. (AOCP/EMPREL/2019) Qual é o procedimento da gerência de memória em que os programas são divididos em sub-rotinas e estruturas de dados, e depois são colocados em blocos de informações na memória que possuem tamanhos diferentes com seu próprio espaço de endereçamento?

A) Cache.

B) Paginação.

C) Segmentação

D) Hash.

E) Pretty Good Privacy.

11. (FURB/Pref. de Porto Belo-SC/2019) O sistema X utiliza paginação com tabela de página armazenada em memória e TLBs. Cada referência à memória leva 200 nanossegundos e 75% de todas as referências à tabela de página são encontradas nas TLBs. Considere que o sistema X leva tempo zero para encontrar uma entrada de página nas TLBs, caso a entrada exista. Qual o tempo efetivo de referência à memória do Sistema X?

A) 250 nanossegundos.

B) 50 nanossegundos.

C) 150 nanossegundos.

D) 350 nanossegundos.

E) 200 nanossegundos.

12. (FURB/Pref. de Porto Belo-SC/2019) Em se tratando de gerência de memória, dadas cinco partições de memória e quatro processos, com seus respectivos tamanhos:

Partições	Processos
Partição 1: 200 KB	Processo A: 420 KB
Partição 2: 1000 KB	Processo B: 880 KB
Partição 3: 400 KB	Processo C: 240 KB
Partição 4: 600 KB	Processo D: 900 KB
Partição 5: 1200 KB	

Um algoritmo de alocação foi utilizado e, como resultado, obteve-se a seguinte sequência de alocações:

A foi alocado na partição 2. B foi alocado na partição 5. C foi alocado na partição 5. D espera a próxima partição disponível.

Qual dos seguintes algoritmos foi utilizado?

A) Best-fit.

B) Worst-fit.

C) First-fit.

D) Circular-fit.

E) Last-fit.

13. (IBFC/TRE-PA/2020) Caso, em um Sistema Operacional tradicional, comece a ocorrer o uso da capacidade total da memória RAM, o Sistema Operacional passará a utilizar:

A) os registradores

B) a memória virtual

C) a memória ROM

D) a memória cache

14. (IBADE/Pref. de Vila Velha-ES/2020) Seja uma memória virtual com 3 blocos e que use o algoritmo LRU (Least Recently Used) como seu algoritmo de substituição de páginas. Admitindo-se que ocorra a seguinte sequência de referência às páginas de memórias: 1,2,3,4,2,3,4,2. Assumindo que inicialmente todos os blocos estão vazios, quantas interrupções de páginas ausentes (page faults) ocorrerão?

A) 5

B) 4

C) 6

D) 7

E) 3

15. (FAPEC/UFMS/2020) Visando a melhorar a tradução de endereços da memória virtual, existe uma memória cache cuja função é diminuir os acessos à tabela de páginas. Assinale a alternativa que representa tal memória.

A) Cache Write-trough.

B) Cache Multi-nível.

C) Translation Lookaside Buffer (TLB).

D) Registrador de tabela de páginas (RTP).

E) Cache Write-back.

16. (SELECON/EMGEPRON/2021) No gerenciamento de memória, uma técnica nos sistemas operacionais permite que um programa possa ser espalhado por áreas não contíguas de memória com as características listadas a seguir.

• A memória física é dividida em páginas com tamanho fixo igual ao da página lógica, sendo cada programa carregado página a página, com cada página lógica ocupando uma página física, não necessariamente contíguas.

• O endereço lógico é inicialmente dividido em duas partes: um número de página lógica e um deslocamento dentro da página. O número da página lógica é usado como índice no acesso à tabela de páginas, de forma a obter o número da página física correspondente.

• Não existe fragmentação externa, existindo a interna.

• Além da localização, a tabela de páginas armazena também o bit de validade, sendo (1) se a página está na memória e (0) se a página não está.

• As páginas de processo podem ser transferidas para a memória por demanda, carregando apenas o que é necessário para a execução do programa, e o sistema tenta prever as páginas que serão necessárias à execução do programa.

• Páginas constantemente referenciadas em um processo devem permanecer na memória.

A técnica descrita é conhecida por:

A) alocação

B) fragmentação

C) segmentação

D) paginação

17. (AOCP/Câmara de Teresina-PI/2021) Sobre os conceitos de gerenciamento de memória em sistemas operacionais, assinale a alternativa correta.

A) Em sistemas sem memória virtual, o gerenciamento de memória não é necessário.

B) Memória virtual é incompatível com o conceito de Paginação.

C) Técnicas podem ser aplicadas para permitir que a memória física seja estendida por outros espaços de armazenamento, como o disco.

D) Sem memória virtual não se pode realizar troca de páginas.

E) Alocação de memória dinamicamente (heap) não é permitida sem memória virtual.

18. (IDECAN/IF-CE/2021) No que diz respeito aos sistemas operacionais, analise as afirmativas a seguir:

I. O recurso de memória virtual consiste em criar tabelas que relacionam posições virtuais e reais da RAM para um mesmo aplicativo. Os programas ficam integradas como uma só unidade na memória física e não de forma distribuída, passando uma ideia de que os programas utilizam a memória de forma distribuída e compartilhada.

II. O recurso da paginação é uma técnica de gerência de memória que permite a um programa ser espalhado por áreas não contíguas de memória. Com isso, o espaço de endereçamento lógico de um processo é dividido em unidades lógicas de tamanho fixo e a memória física é dividida em páginas com tamanho fixo, com tamanho igual ao da página lógica.

III.O recurso da segmentação é uma técnica de gerência de memória em que os programas são divididos em segmentos de tamanhos variados cada um com seu próprio espaço de endereçamento.

Assinale

A) se apenas as afirmativas I e II estiverem corretas.

B) se apenas as afirmativas I e III estiverem corretas.

C) se apenas as afirmativas II e III estiverem corretas.

D) se todas as afirmativas estiverem corretas.

19. (IDECAN/IF-CE/2021) No que tange ao funcionamento dos sistemas operacionais, existem situações em que não é possível manter todos os processos na memória.

O sistema operacional utiliza uma técnica aplicada à gerência de memória, para processos que esperam por memória livre para serem processados. Essa técnica visa melhorar o problema da insuficiência de memória durante a execução de alguns processos em ambientes que operam em multiprogramação; consiste em transferir automaticamente todo o processo da memória principal para o disco e vice-versa. Essa técnica é denominada

A) sweeping.

B) swapping.

C) scanning.

D) spoofing.

20. (Quadrix/CFO-DF/2022) Em relação à estrutura do sistema operacional, ao gerenciamento de processos e à memória virtual, julgue o item.

Na memória virtual, quando o programa faz referência a uma parte de seu espaço de endereçamento que não está em sua memória física, o sistema operacional é alertado para obter a parte que falta e reexecutar a instrução que falhou.

GABARITO

GABARITO

1. Letra D
2. Letra C
3. Letra A
4. Letra A
5. Letra C
6. Letra C
7. Letra A
8. Letra B
9. Letra B
10. Letra C
11. Letra A
12. Letra D
13. Letra B
14. Letra B
15. Letra C
16. Letra D
17. Letra C
18. Letra C
19. Letra B
20. Certo

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.