



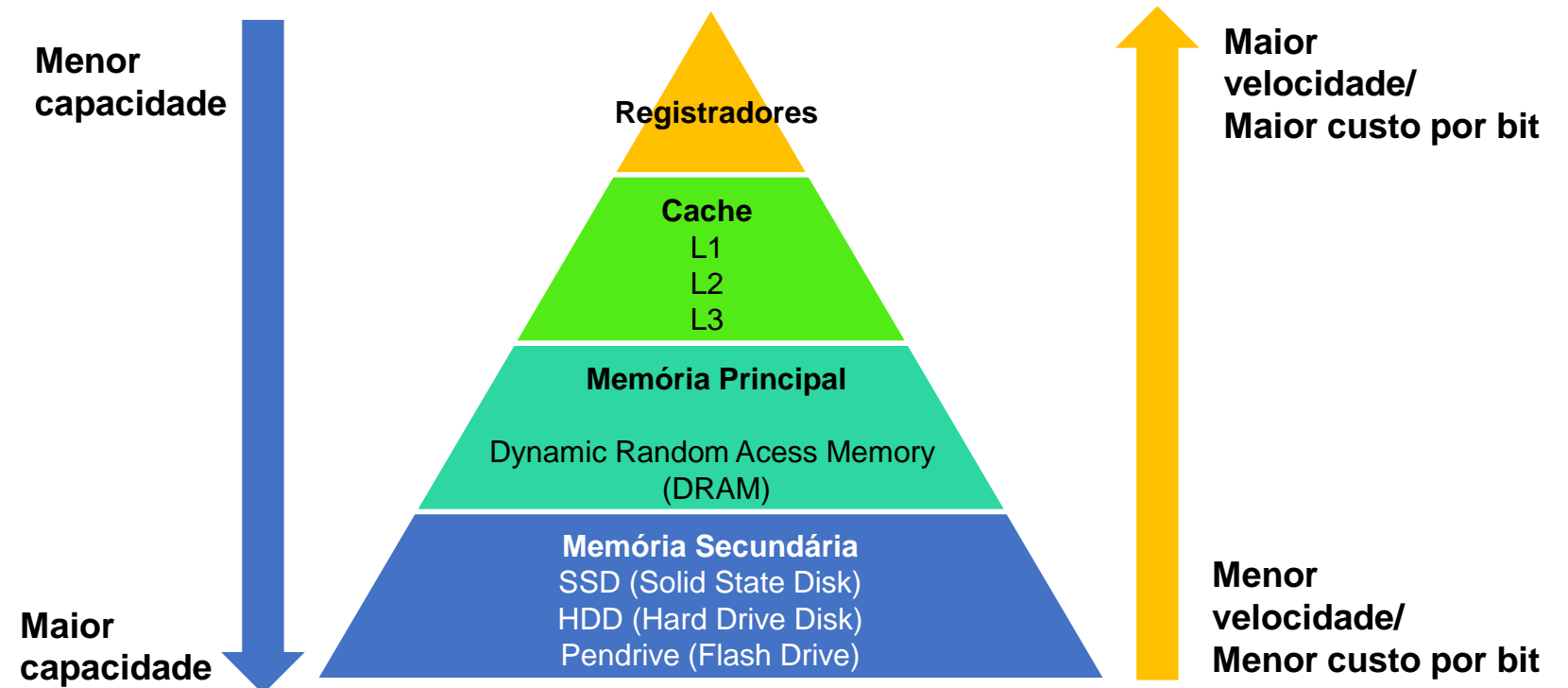
UNIDADE II

Sistemas Operacionais Abertos e Mobile

Prof. Me. Michel Fernandes

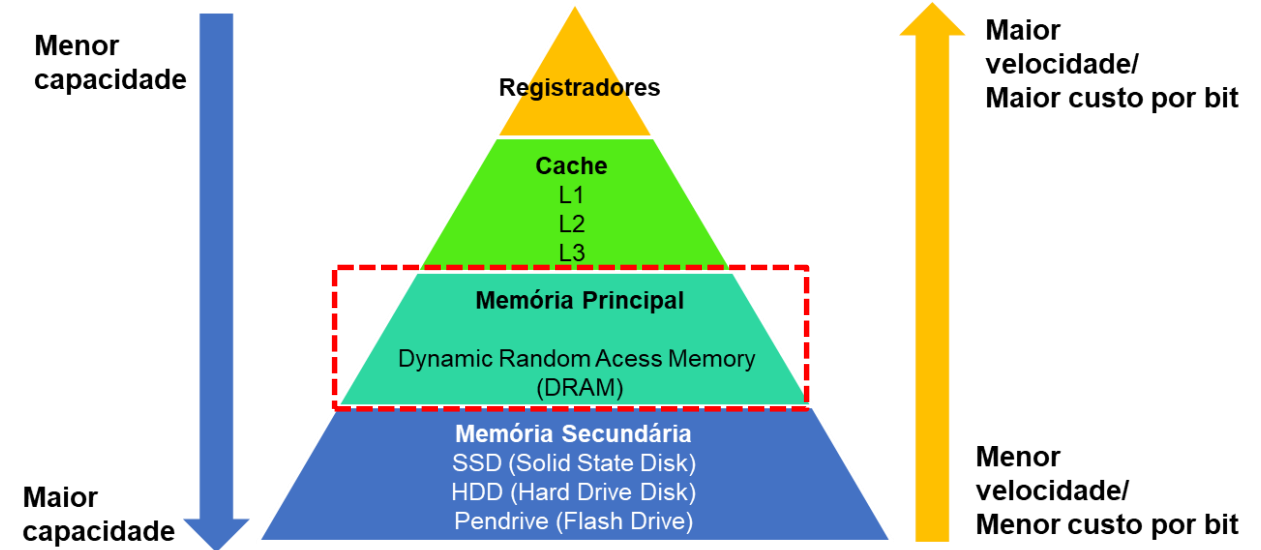
Hierarquia de memória

- Subsistema de memória, projetado de modo que componentes sejam organizados hierarquicamente.
- As características analisadas são: tempo de acesso, capacidade, volatilidade, tecnologia de fabricação, temporariedade e custo.



Memória principal

- Memória principal, também chamada de memória RAM – Random Access Memory.
- Armazenamento de dados dos programas e os próprios programas.
- Memórias que o processador pode endereçar diretamente.
- Acesso aleatório significa que consegue acessar qualquer célula. Pode ser alcançada com 1 única instrução.



RAM module SDRAM 8GB DDR4.

Fonte: [https://commons.wikimedia.org/wiki/File:Two_8_GB_DDR4-2133_ECC_1.2_V_RDIMMs_\(straightened\).jpg](https://commons.wikimedia.org/wiki/File:Two_8_GB_DDR4-2133_ECC_1.2_V_RDIMMs_(straightened).jpg).

Parâmetros da memória principal

- **Tempo de acesso:** da ordem de nano segundos.
- **Capacidade:** com endereçamento de 32 bits, a maior quantidade de memória disponível era 4 Gb.
 - Com 64 bits aumentou significativamente.
- **Volatilidade:** É volátil.
- **Temporariedade:** para que um programa seja executado, é necessário que ele esteja armazenado na memória principal.

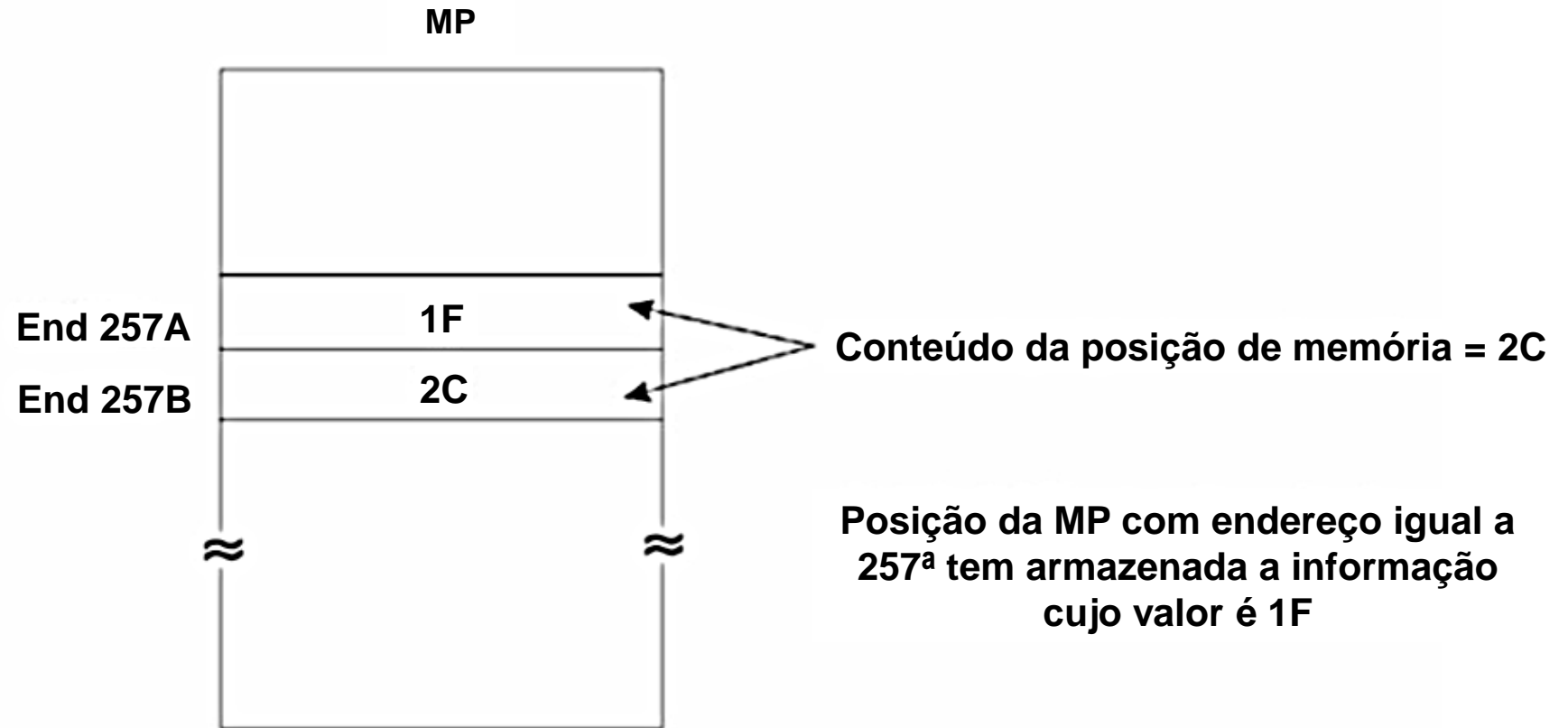
Tipos de memória

- Principais tipos de memória de semicondutor.
- Memória RAM precisa receber uma fonte de alimentação constante.
- Sem energia, os dados são perdidos.

| Tipo de memória | Categoria | Forma de apagar | Mecanismo de escrito | Volatilidade |
|--------------------------------------|-------------------------------|---------------------------------|----------------------|--------------|
| Memória de acesso aleatório (RAM) | Memória de Escrita e Leitura | Eletricamente, em nível de byte | Eletricamente | Volátil |
| Memória somente de leitura (ROM) | Memória somente de leitura | Não é possível | Máscara | Não volátil |
| ROM programável (PROM) | | | Eletricamente | |
| PROM apagável (EPROM) | Luz UV, nível de chip | | | |
| PROM eletricamente apagável (EEPROM) | Eletricamente, nível de byte | | | |
| Memória Flash | Eletricamente, nível de bloco | | | |

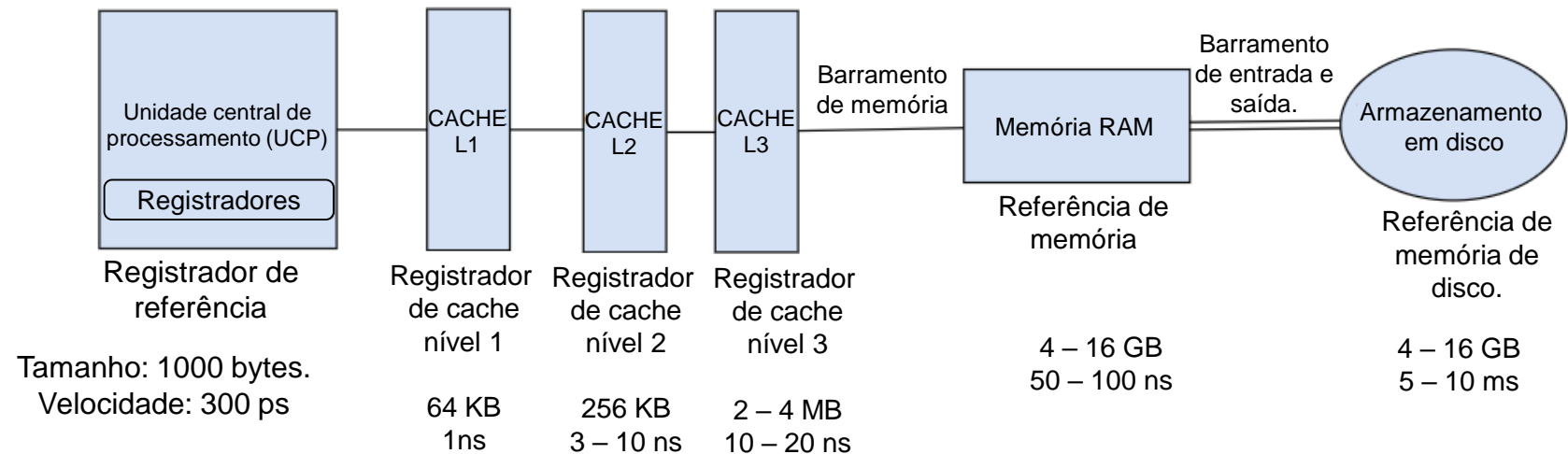
Organização da memória principal

- Palavra: unidade de informação do sistema processador/MP que deve representar o valor de um dado ou uma instrução de máquina.
- Endereço.
- Conteúdo.



Memória cache

- A velocidade dos processadores é em geral muito maior do que as velocidades das memórias RAM.
- Memória cache soluciona essa limitação entre as velocidades de processamento e a velocidade da memória principal.
- A memória cache é uma memória com menor capacidade que a RAM, porém, possui velocidades mais altas.



Fonte: adaptado de Hennessy, 2017.

Conceito de gerenciamento de memória

- Programas são armazenados em memórias secundárias, como discos ou fitas, por ser um meio não volátil, abundante e de baixo custo.
- Como o processador somente executa instruções localizadas na memória principal, o sistema operacional deve sempre transferir programas da memória secundária para a memória principal antes de eles serem executados.
- Como o tempo de acesso à memória secundária é muito superior ao tempo de acesso à memória principal, o sistema operacional deve buscar reduzir o número de operações de E/S à memória secundária, para evitar problemas de desempenho do sistema.

Funções de gerenciamento de memória

- Deve tentar manter na memória principal o maior número possível de processos residentes.
- Permitir e maximizar o compartilhamento do processador e demais recursos computacionais.
- Mesmo na ausência de espaço livre, o sistema deve permitir que novos processos sejam aceitos e executados.
- Permitir a execução de programas que sejam maiores que a memória física disponível.

Alocação de memória

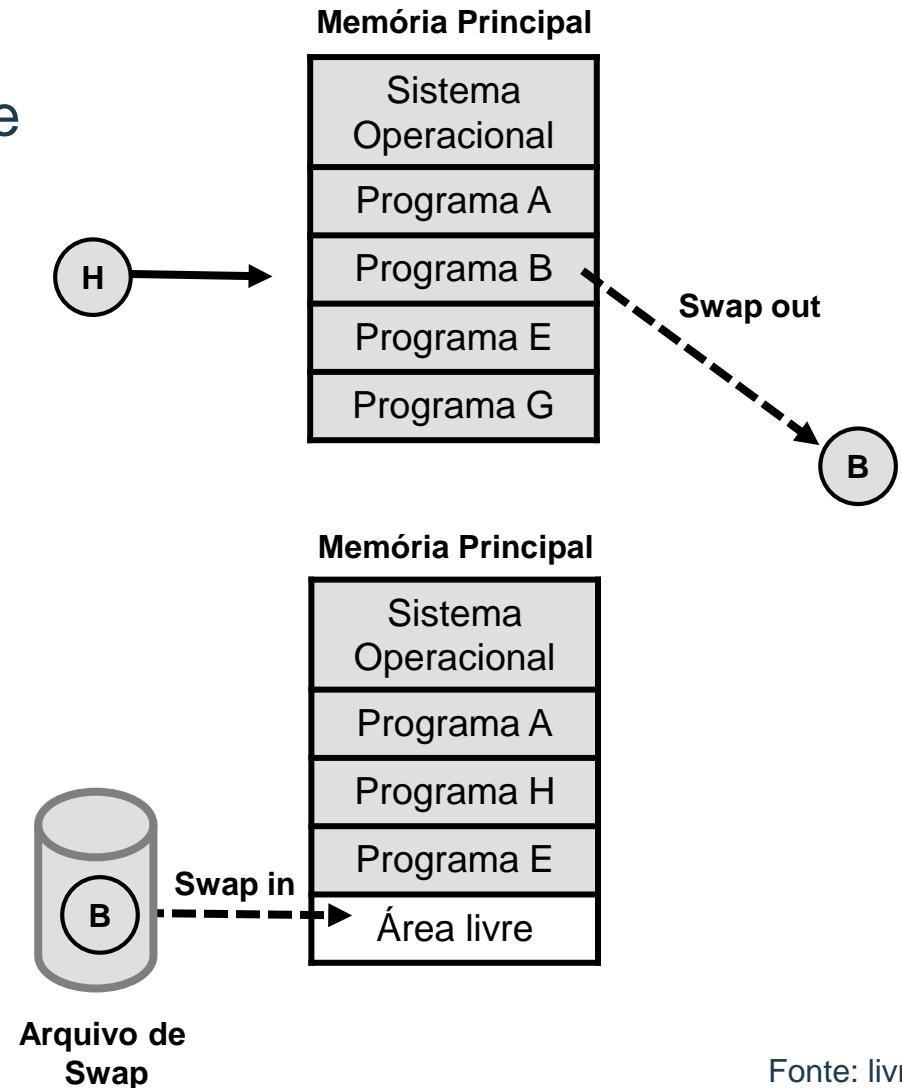
- Reservar áreas de memória para uso do SO e dos processos.

O alocador de memória:

- Atende solicitações do núcleo ou de processos.
- Aloca e libera áreas de memória.
- Gerencia quais áreas estão livres ou ocupadas.
- Deve ser rápido e eficiente (baixo desperdício de memória).

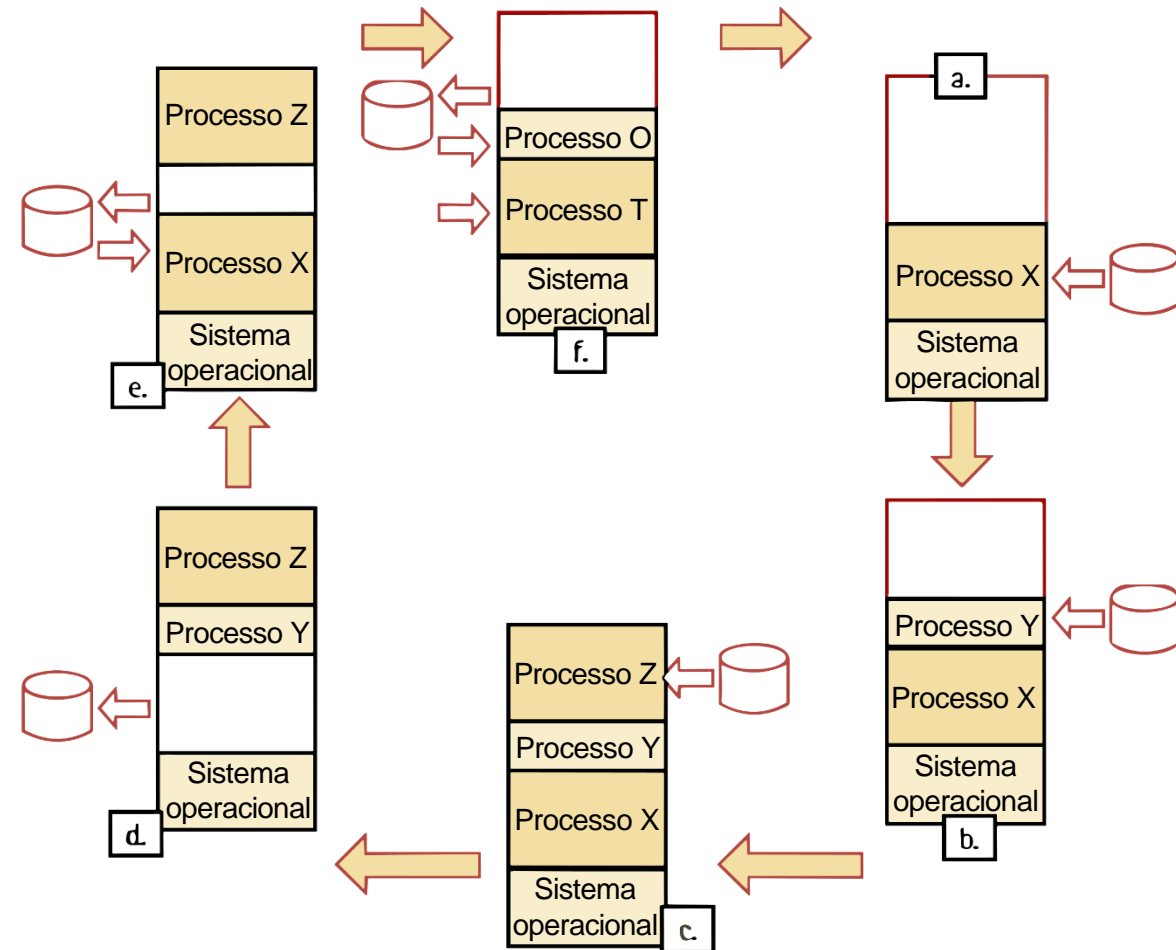
Swapping

- Mesmo na ausência de espaço livre, o sistema deve permitir que novos processos sejam aceitos e executados.
- Isso é possível por meio da transferência temporária de processos residentes na memória principal para a memória secundária, liberando espaço para novos processos.
- Esse mecanismo é conhecido como Swapping ou Permuta de Memória.



Swapping

- Em a., SO ocupa a parte mais baixa da memória e, logo em seguida, temos o processo X ocupando uma parte da memória disponível.
- Em b., um novo processo Y é criado ou trazido do disco para a memória.
- Em c., um novo processo Z é adicionado.
- Em d., o processo X fica ocioso, então, é enviado para o disco rígido.
- Em e/f, outros processos sendo trocados e o ciclo vai sendo executado até que novos processos entrem e disputem o tempo de CPU e memória.

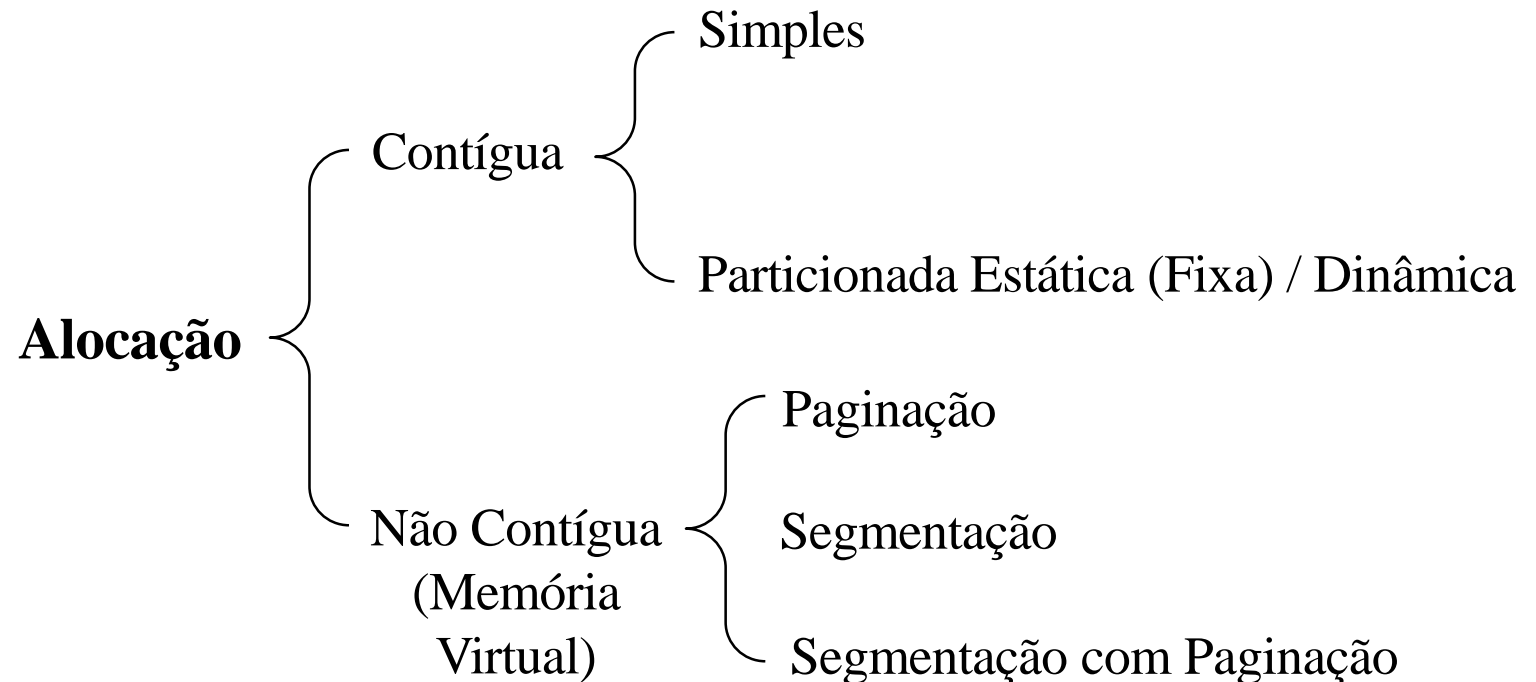


Alocação de memória

- Formas de alocar memória.

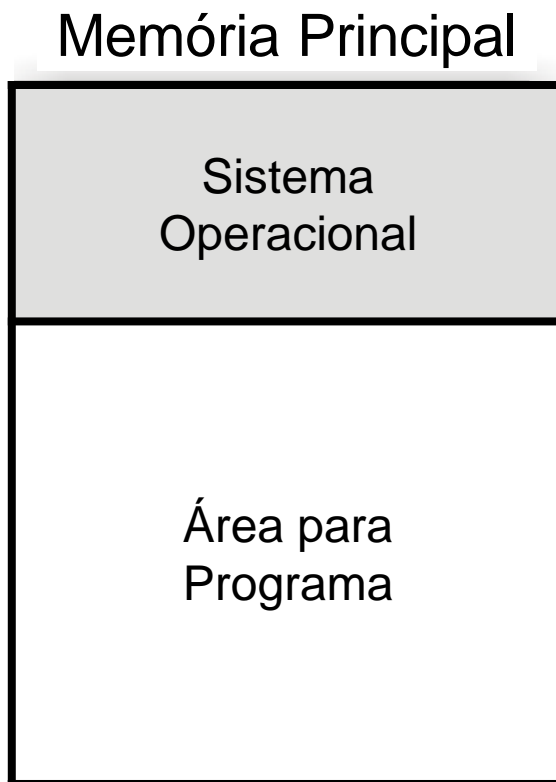
Problema:

- Gerenciar uma ou mais áreas de memória.
- Atender pedidos de alocação e liberação de blocos.
- Otimizar o uso da memória.
- Evitar/minimizar a fragmentação.



Alocação contígua simples

- Implementada nos primeiros sistemas operacionais, e para sistemas monoprogramáveis.
- A memória principal é subdividida em duas áreas:
 - Uma para o sistema operacional e outra para o programa do usuário.
 - Dessa forma, o programador deve desenvolver suas aplicações preocupado em não ultrapassar o espaço de memória disponível.



Alocação contígua simples

- Esquema em que o usuário tem controle sobre toda a memória principal, inclusive a área do sistema operacional.
- Implementa controle de proteção do sistema por meio de registrador que delimita a área do sistema operacional.
- Fácil implementação e código reduzido, porém não utiliza os recursos computacionais de forma eficiente, pois apenas um usuário/aplicação pode dispor desse recurso.
 - Entretanto, somente é aplicável para sistemas monoprogramáveis.

Interatividade

Qual o nome da técnica da gerência de memória que consiste em transferir temporariamente processos residentes na memória principal para a memória secundária, liberando espaço para novos processos?

- a) Swapping.
- b) Memória cache.
- c) Escalonamento.
- d) Alocação dinâmica.
- e) Fragmentação externa.

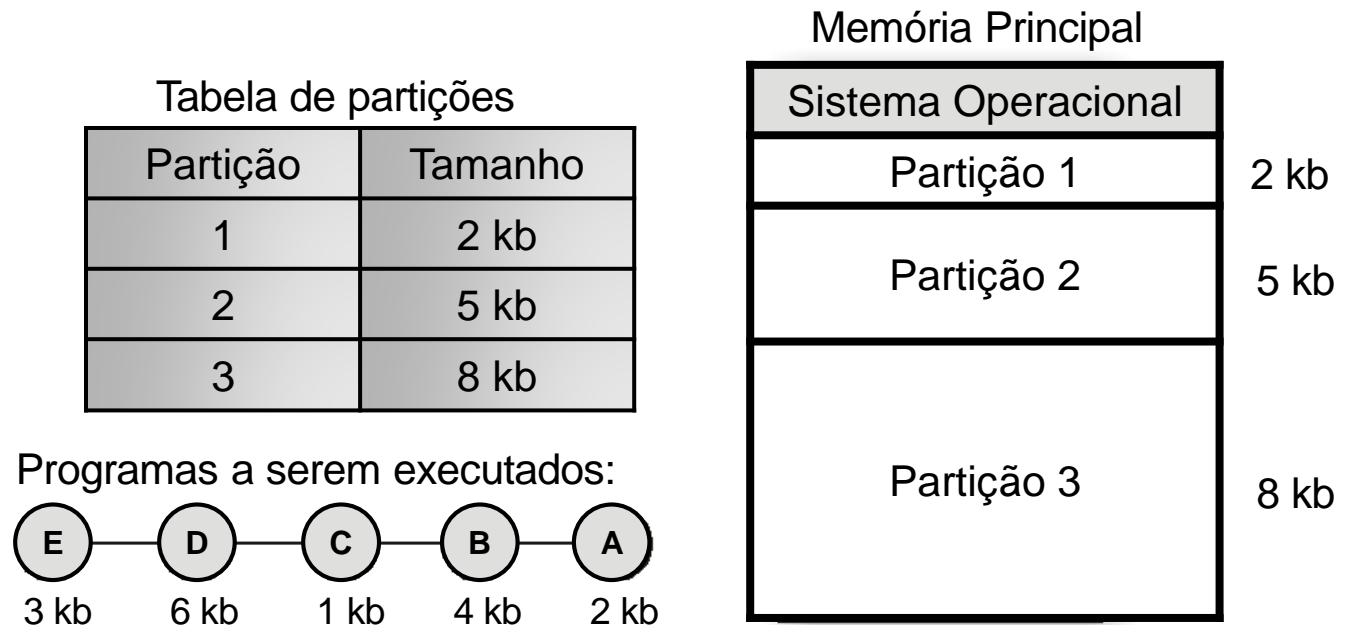
Resposta

Qual o nome da técnica da gerência de memória que consiste em transferir temporariamente processos residentes na memória principal para a memória secundária, liberando espaço para novos processos?

- a) **Swapping.**
- b) Memória cache.
- c) Escalonamento.
- d) Alocação dinâmica.
- e) Fragmentação externa.

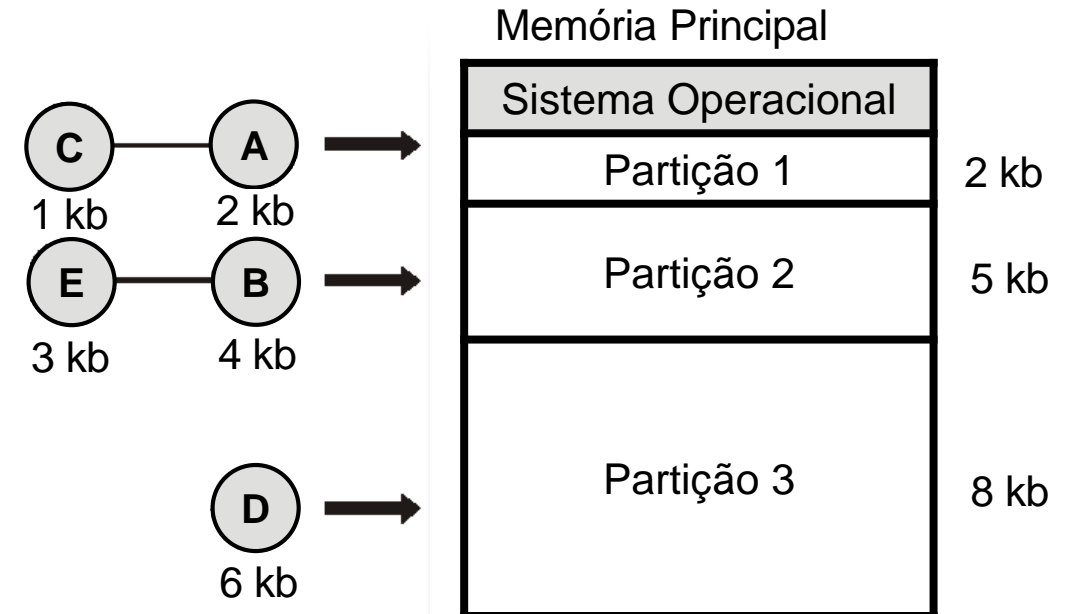
Alocação contígua particionada estática

- Os sistemas multiprogramáveis são muito mais eficientes no uso do processador, necessitando, assim, que diversos programas estejam na memória principal ao mesmo tempo e que novas formas de gerência da memória sejam implementadas.
- Inicialmente, os programas só podiam ser carregados e executados em apenas uma partição específica, mesmo se outras estivessem disponíveis.
- No exemplo acima, supondo que os programas A e B estivessem sendo executados, os programas C e E não poderiam ser processados na terceira partição, mesmo esta estando livre.



Alocação contígua particionada fixa relocável

- Com a evolução dos compiladores, montadores, ligadores e carregadores, o código gerado deixou de ser absoluto e passou a ser relocável.
- No código relocável, todas as referências a endereços no programa são relativas ao início do código e não a endereços físicos de memória.
- Dessa forma, os programas puderam ser executados a partir de qualquer partição.



Alocação contígua particionada fixa relocável

- Para manter o controle sobre as partições alocadas, a gerência de memória mantém uma tabela com o endereço inicial de cada partição, seu tamanho, e se está em uso ou não.

| Partição | Tamanho | Livre |
|----------|---------|-------|
| 1 | 2 kb | Não |
| 2 | 5 kb | Sim |
| 3 | 8 kb | Não |



Fragmentação interna

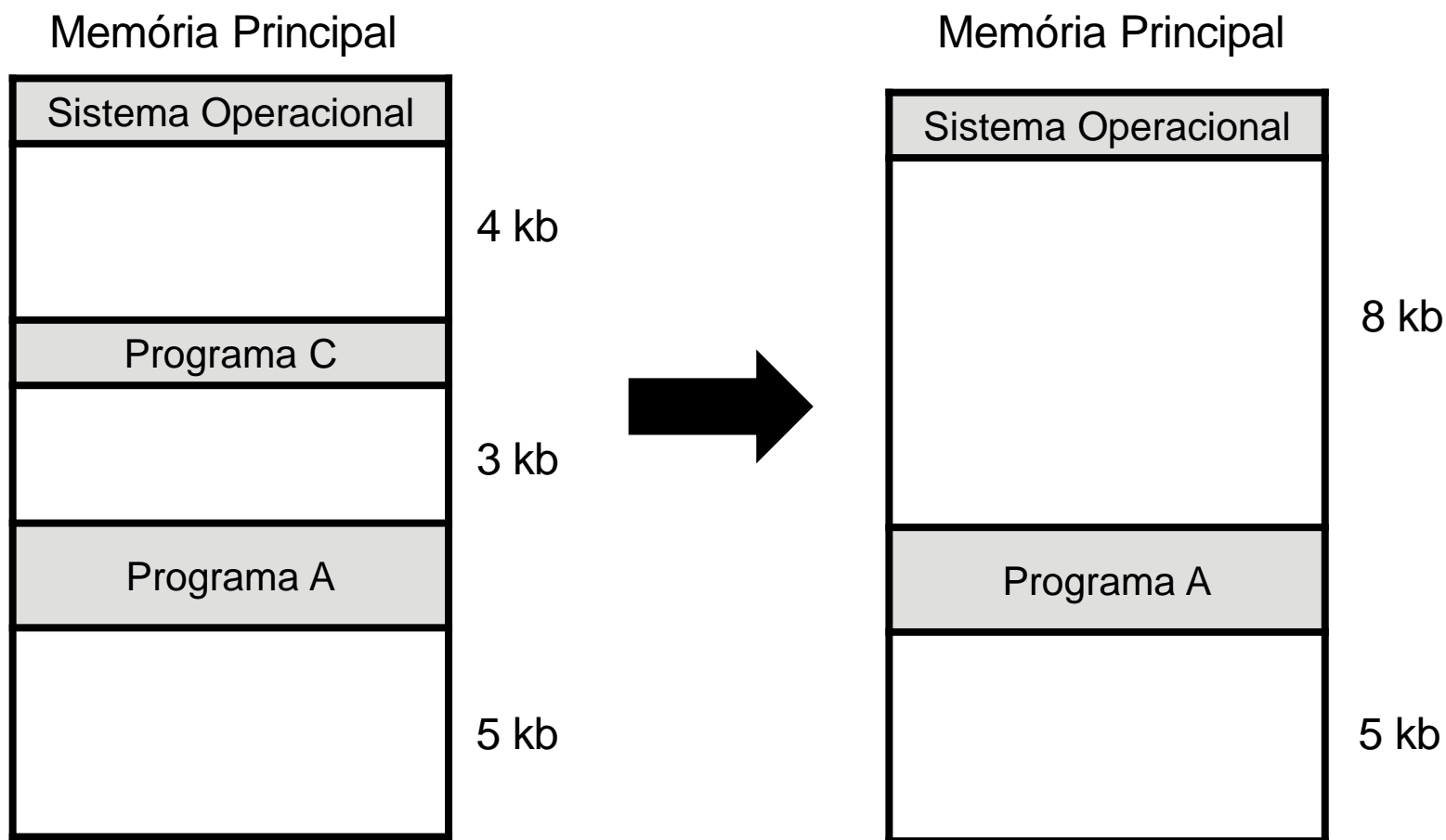
- Tanto nos sistemas de alocação fixa quanto nos de alocação relocável, os programas, normalmente, não preenchem totalmente as partições onde são carregados, deixando área de memória livre.
- Esse problema é conhecido como fragmentação interna.

Alocação contígua particionada dinâmica

- Na alocação particionada dinâmica ou variável, foi eliminado o conceito de partições de tamanho fixo. Nesse esquema, cada programa utilizaria o espaço necessário, tornando essa área sua partição.
- Como cada programa utiliza apenas o espaço que necessita, o programa de fragmentação interna não ocorre.

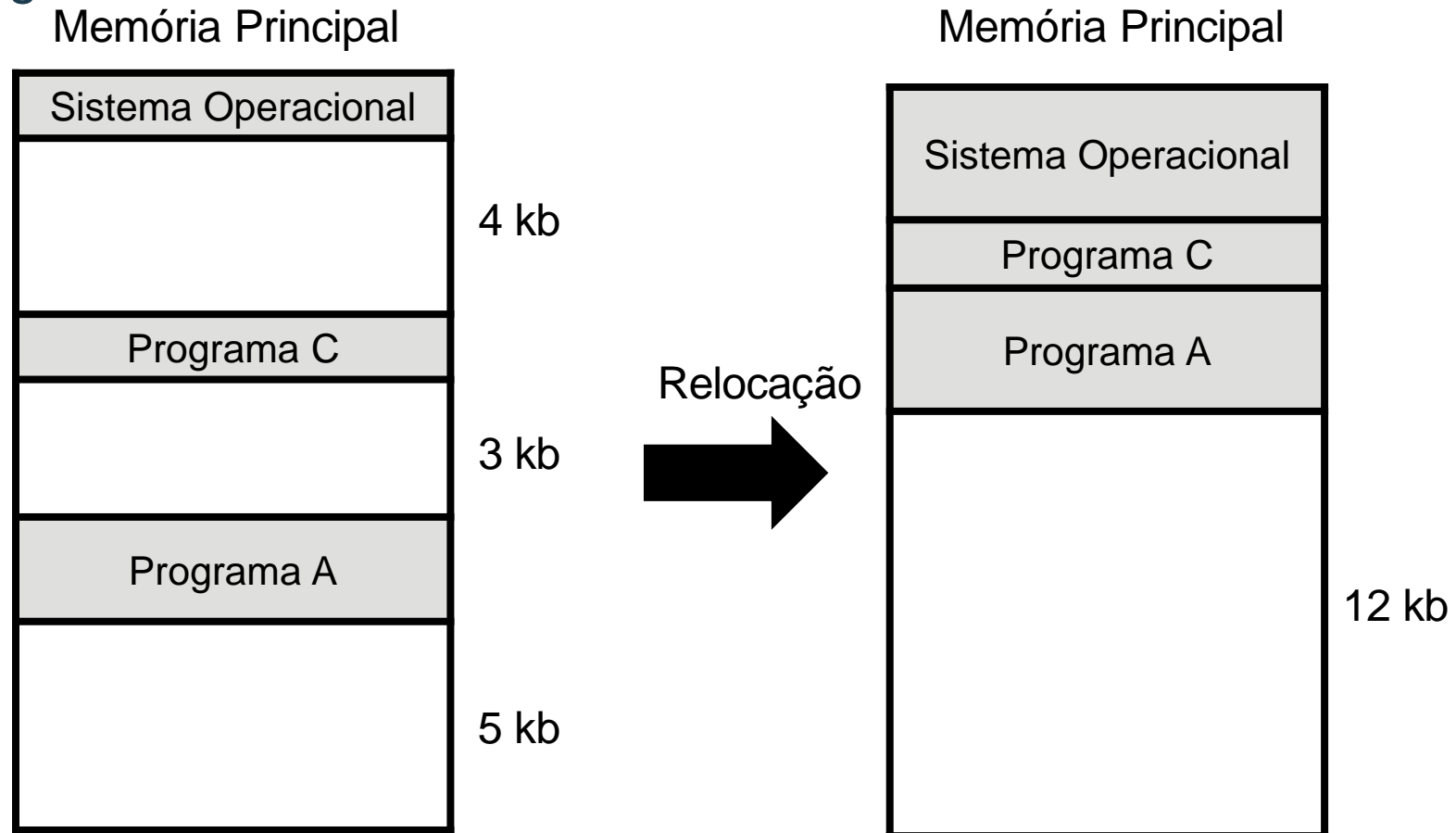
Fragmentação externa

- Os programas foram terminando sua execução e saindo da memória principal, mas deixavam espaços cada vez menores na memória, não permitindo o ingresso de novos programas que fossem maiores que o bloco deixado.



Soluções para o problema de fragmentação externa

- Conforme os programas terminam, apenas os espaços livres adjacentes são reunidos, produzindo áreas livres de tamanho maior.
- Relocação de todas as partições ocupadas, eliminando todos os espaços entre elas e criando uma única área livre contígua.
- Para que essa solução possa ser implementada, é necessário que o sistema tenha a capacidade de mover os diversos programas na memória principal, ou seja, realizar a relocação dinâmica.



Alocação contígua particionada dinâmica

- Vantagens:
 - reduz em muito o problema da fragmentação.
- Desvantagem:
 - aumenta a complexidade do algoritmo e o consumo de recursos do sistema (processador e área de disco).

Algoritmos de alocação

- Sistema operacional deve decidir qual bloco livre será associado a um processo.
- Algoritmos de alocação:
 - best-fit.
 - first-fit.
 - next-fit.
 - worst-fit.

Algoritmos de alocação – First fit

- “O que primeiro couber”.
- Busca por espaço livre desde o início da memória.
- Escolhe o primeiro bloco disponível que seja grande o suficiente.
- Método tenta primeiro utilizar as áreas livres de endereços mais baixos.
- Boa chance de se obter uma grande partição livre nos endereços mais altos.
- Algoritmo mais rápido.
- A lista de áreas livres está ordenada por endereços crescentemente.
- Consome menos recursos para a busca.

Algoritmos de alocação – Best fit

- “O que melhor couber”.
- Escolhe-se a partição onde o processo deixa o menor espaço sem utilização.
- Objetivo é garantir a melhor escolha de partição livre.
- Desvantagem do algoritmo:
 - escolha da partição mais aproximada resulta em pequenas partições livres.
 - tendência é ter grande quantidade de pequenas áreas livres não contíguas.
 - aumentando o problema da fragmentação.
- Solução pode ser o emprego de compactação de memória.

Algoritmos de alocação – Worst fit

- Escolhe-se a partição onde o processo deixa o maior espaço sem utilização.
- Escolhe o maior espaço livre na memória.
- Nesse algoritmo, a lista de áreas livres deve estar ordenada por tamanho para otimizar a busca.
- Comparado ao best-fit, reduz (não elimina) o problema da fragmentação.

Algoritmos de alocação – Next Fit

- Similar ao first-fit.
- Diferença está na busca, que ocorre a partir do endereço da última posição alocada e não a primeira posição de memória.

Gerenciamento de memória no Linux

Linux divide a memória física em quatro zonas ou regiões:

- ZONE_DMA
- ZONE_DMA32
- ZONE_NORMAL
- ZONE_HIGHMEM

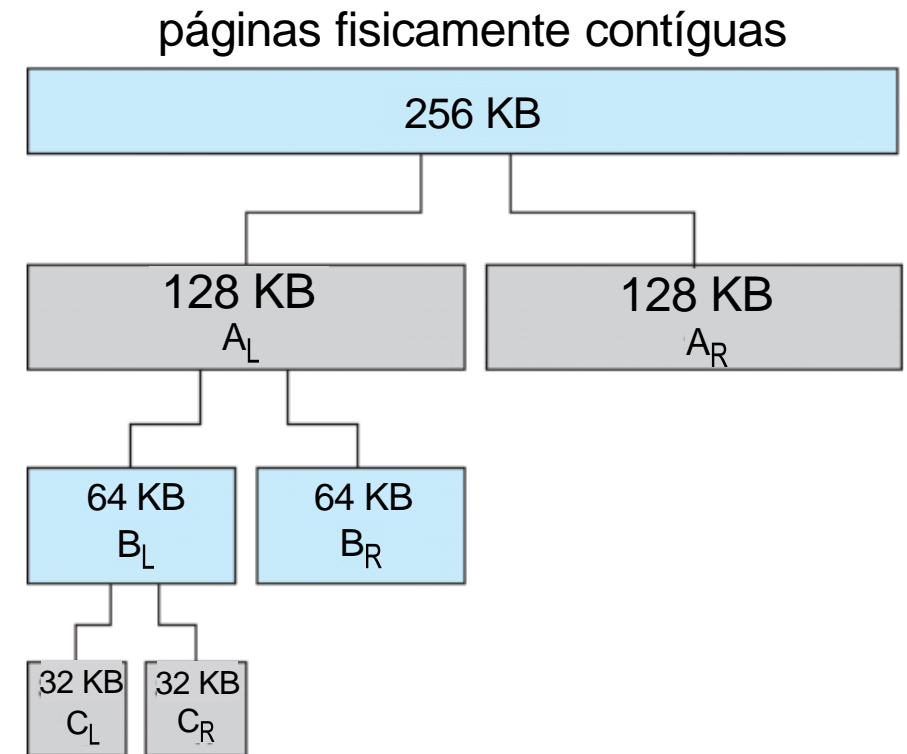
| zona | memória física |
|--------------|----------------|
| ZONE_DMA | < 16 MB |
| ZONE_NORMAL | 16..896 MB |
| ZONE_HIGHMEM | > 896 MB |

Fonte: Silberschatz. *et al*, 2015, p. 432

- Divisão da memória em cada uma dessas regiões depende da arquitetura.
 - Zona DMA: acesso direto à memória por dispositivos E/S.
 - Zona NORMAL: páginas normais mapeadas regularmente.
 - Zona de MEMÓRIA ALTA: Memória física que não é mapeada para o espaço de endereçamento do kernel.

Linux: alocação em sistema de pares

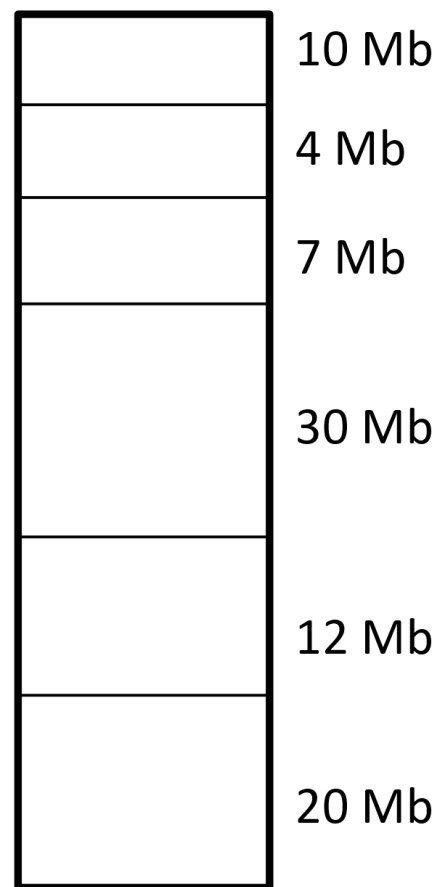
- Principal gerenciador de memória física no kernel do Linux é o alocador de páginas.
- O alocador usa um sistema de pares, ou buddy system, para rastrear páginas físicas disponíveis.
- Sistema de pares, ou buddy system, aloca a memória a partir de um segmento de tamanho fixo composto por páginas fisicamente contíguas.
- Quando ocorre a liberação de duas regiões parceiras alocadas, estas são combinadas para formar uma região maior, denominada heap de pares ou buddy heap.



Interatividade

Considere os espaços de memória livre da figura. O próximo processo a ser alocado irá ocupar 15 Mb de memória. Em qual lacuna esse processo será alocado?

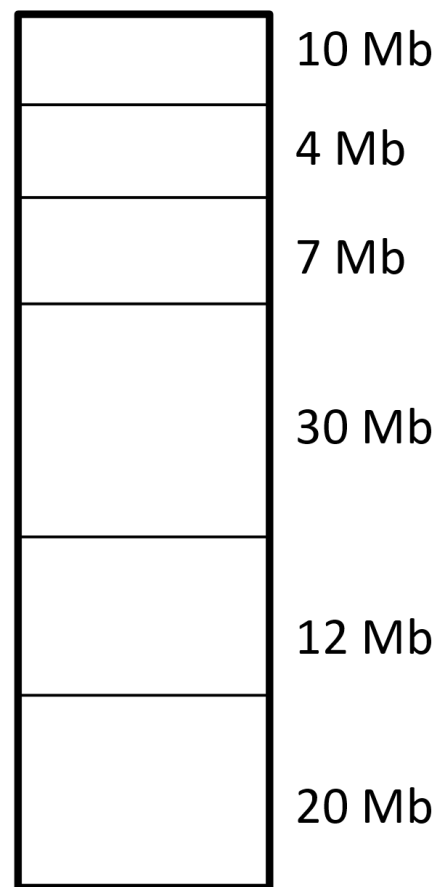
- a) 10 Mb.
- b) 7 Mb.
- c) 30 Mb.
- d) 12 Mb.
- e) 20 Mb.



Resposta

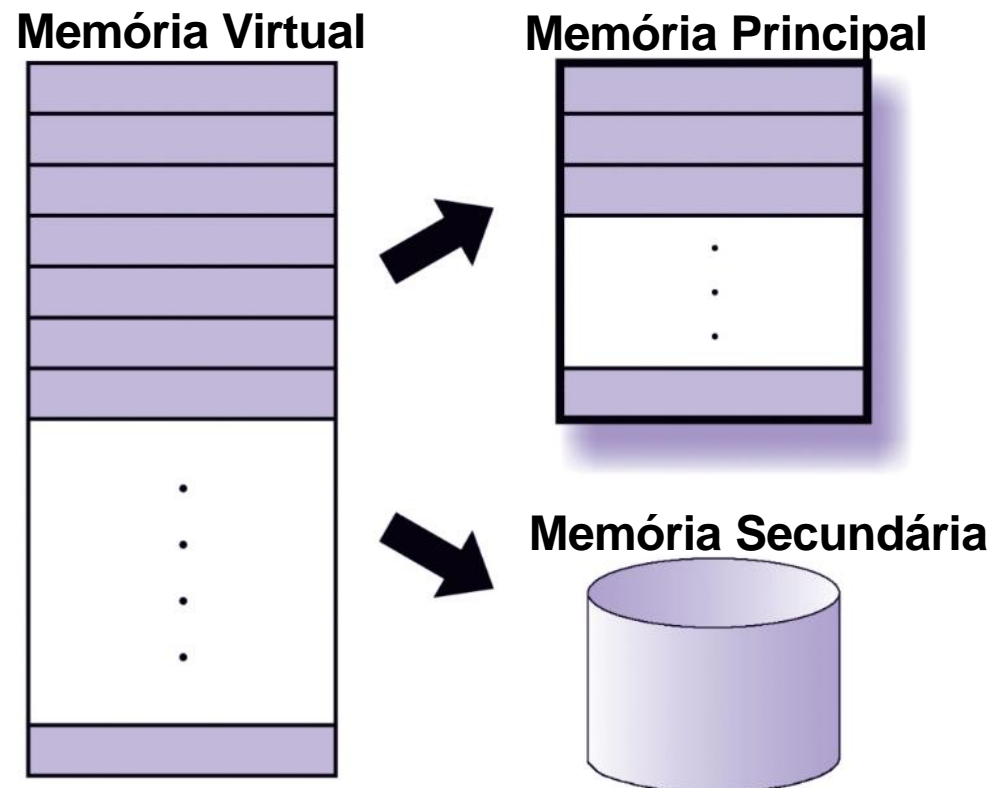
Considere os espaços de memória livre da figura. O próximo processo a ser alocado irá ocupar 15 Mb de memória. Em qual lacuna esse processo será alocado?

- a) 10 Mb.
- b) 7 Mb.
- c) 30 Mb.
- d) 12 Mb.
- e) 20 Mb.



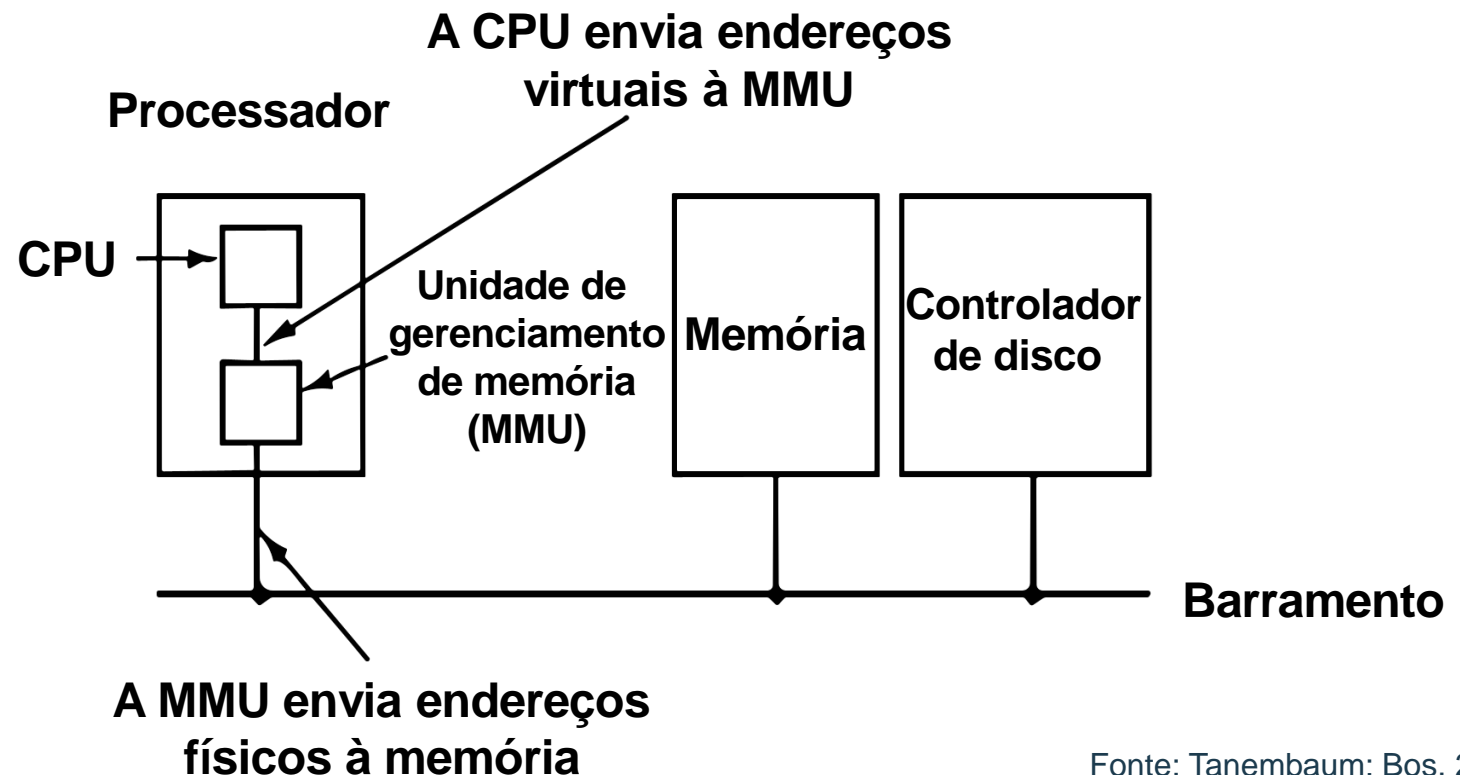
Memória virtual

- Espaço de endereçamento virtual e real.
- Memória virtual aumenta a memória total disponível em um sistema, adicionando à memória real um espaço em disco, por meio de uma memória secundária.
- É utilizada, pois o tamanho de memória ocupada pelo software é crescente, para permitir que programas maiores que a memória de acesso aleatório (RAM) sejam executados.



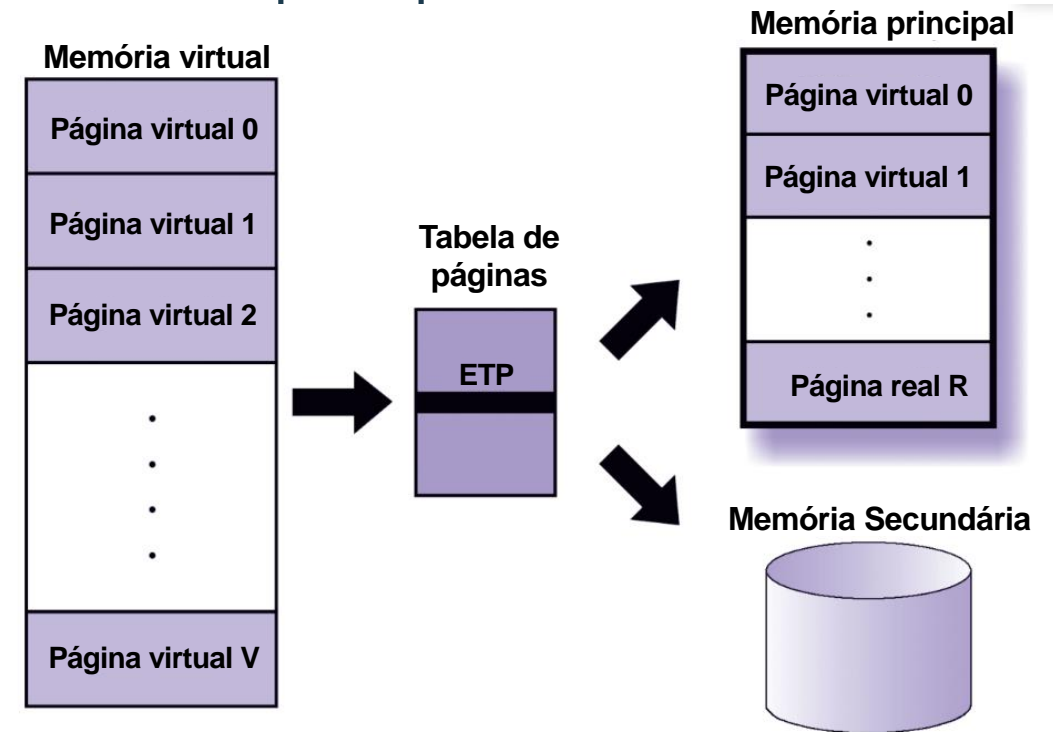
Memória virtual

- MMU: Unidade de gerenciamento de memória.
 - Conversão de endereços virtuais em endereços reais.
- A MMU suporta o sistema operacional na realização do mapeamento dos endereços da memória física e endereços da memória virtual



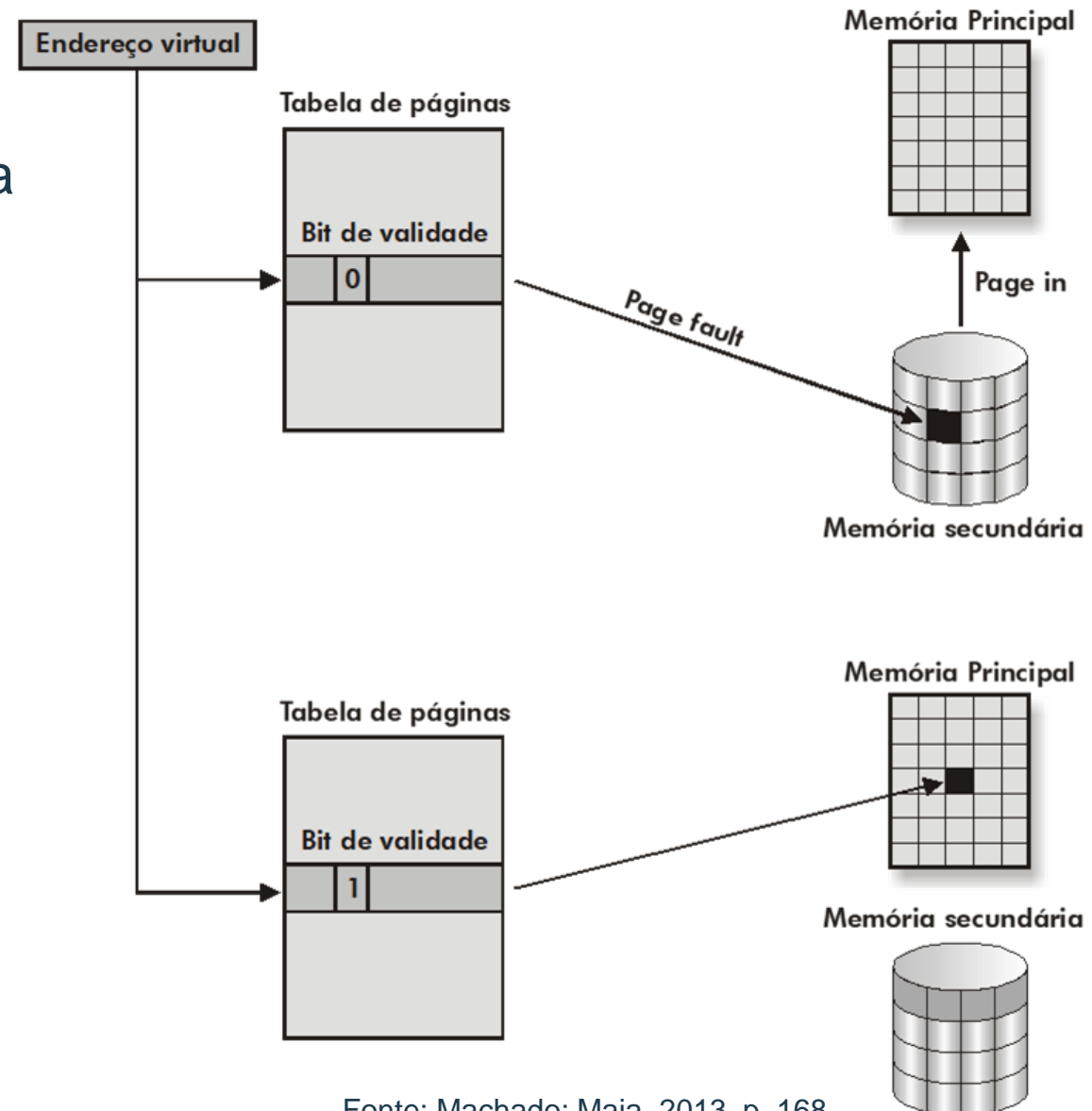
Memória virtual por paginação

- Espaço de endereçamento virtual e real são divididos em blocos de tamanho fixo que são denominados páginas.
- As páginas no espaço real são denominadas páginas reais ou frames.
- Cada página virtual do processo gera uma entrada na tabela de páginas (ETP).
- Bit de validade: indica se uma página está ou não na memória principal.
 - Se o bit for 0, não está carregada (page fault).



Mecanismo de tradução

- Bit de validade, ou valid bit, é que indica se uma página está ou não na memória principal.
- Se o bit for 0, não está carregada e é gerada uma interrupção de falta de página, ou page fault, que desvia a execução para o sistema operacional. Nesse caso o sistema transfere a página da memória secundária para a memória principal, realizando page in, ou paginação.
- Se o bit for igual a 1, a página está já carregada na memória principal.



Memória virtual por paginação

Informações de controle na tabela de páginas:

- Bit de validade: indica se a página é válida.
 - Caso este bit for 0, tentativas de acesso à página irão gerar uma interrupção de falta de página ou page fault.
 - Se o bit for igual a 1, a página está já carregada na memória principal.
- Writable: Página é leitura e escrita (1) ou somente leitura (0).
- User: Se for 1, acessa a página em modo usuário. Senão, só acessível ao núcleo do sistema.
 - Present: indica se a página está presente na memória RAM ou se foi transferida para um armazenamento secundário, como ocorre nos sistemas com paginação em disco.

Política de alocação de páginas

- Define quantos frames cada processo poderá manter na memória principal.

Há 2 tipos de alocação:

- Alocação fixa: há um número máximo de frames. Caso seja insuficiente, uma página do processo é descartada para que uma nova página seja carregada.
- Alocação variável: número máximo de páginas alocadas varia dependendo da taxa de paginação e da ocupação da memória principal.

Busca por páginas

- Define quando uma página deve ser carregada na memória principal.

Paginação por demanda:

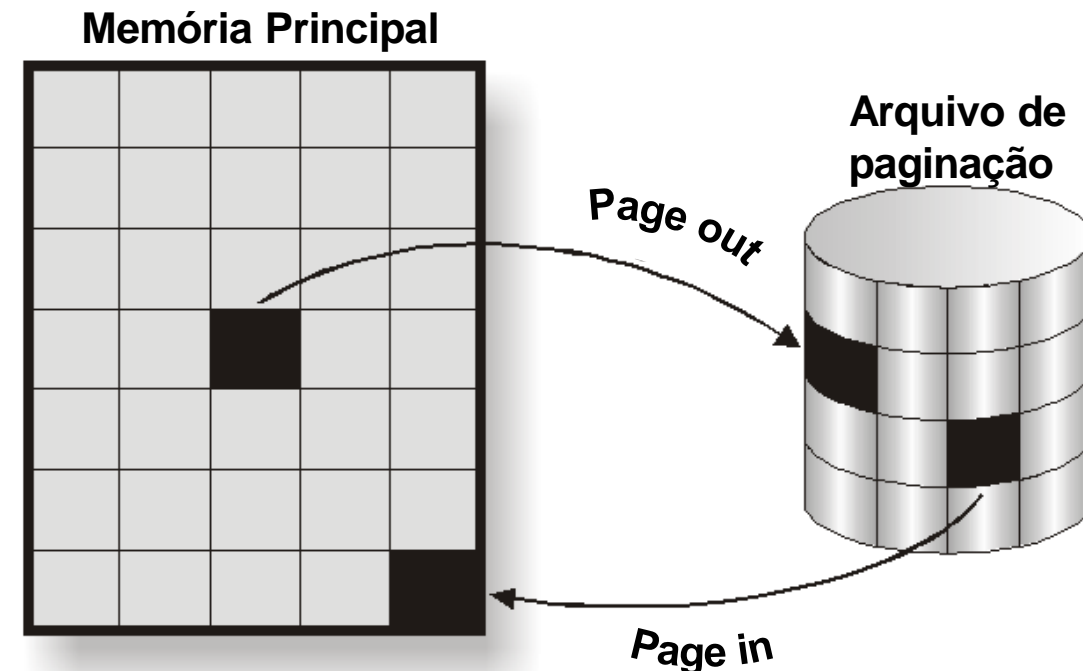
- As páginas dos processos são transferidas da memória secundária para a memória principal somente quando são referenciadas.
- Somente as páginas necessárias à execução do programa são carregadas.

Paginação antecipada:

- Além da página referenciada, outras páginas que podem ou não ser necessárias ao processo ao longo do processamento são carregadas na memória principal.
- Páginas adicionais podem não ser utilizadas durante a execução do processo.

Política de substituição de páginas

- Determina como SO seleciona qual página será liberada da memória principal para que outra a substitua.
- Política de substituição de páginas local:
 - Quando um processo necessitar de uma nova página, o sistema selecionará uma página do processo em questão para ser substituída.
- Política de substituição de páginas global:
 - Todas as páginas alocadas na memória principal são candidatas à substituição, independentemente do processo que a gerou.



Algoritmos de substituição de página

Diversos algoritmos foram implementados:

- Ótimo;
- Fifo: Primeira página que entra, é a primeira que sai;
- Menos Usado Recentemente (NRU);
- Segunda chance;
- Não Utilizada Recentemente (LRU);
- Relógio.

Interatividade

Na tarefa de gerenciamento de memória, existe um hardware de extrema importância que é conhecido como Unidade de Gerenciamento de Memória (MMU). Qual a função do MMU?

- a) Manter rodando na memória principal somente as partes necessárias de um programa.
- b) Definir o tamanho máximo de uma área de swapping.
- c) Liberar a memória de processos inativos.
- d) Informar ao programador endereços de alocação da memória.
- e) Traduzir endereços virtuais em endereços físicos.

Resposta

Na tarefa de gerenciamento de memória, existe um hardware de extrema importância que é conhecido como Unidade de Gerenciamento de Memória (MMU). Qual a função do MMU?

- a) Manter rodando na memória principal somente as partes necessárias de um programa.
- b) Definir o tamanho máximo de uma área de swapping.
- c) Liberar a memória de processos inativos.
- d) Informar ao programador endereços de alocação da memória.
- e) Traduzir endereços virtuais em endereços físicos.

Segmentação

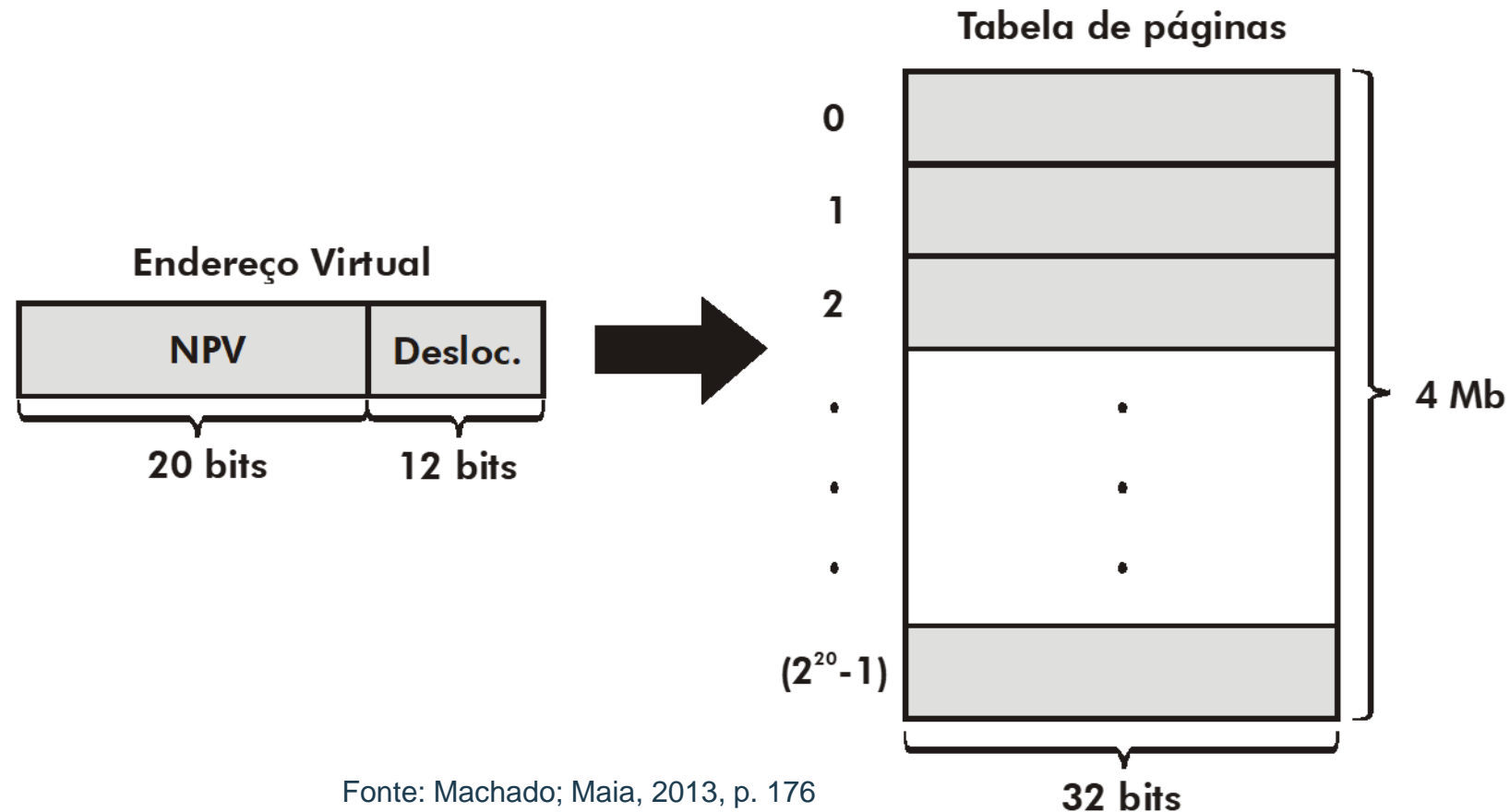
- Além da paginação, a segmentação de memória é uma das formas mais simples para se obter a proteção da memória.

Com o uso da segmentação são atendidos os seguintes requisitos:

1. Pode haver vários segmentos distintos.
2. Cada segmento pode ter um tamanho próprio.
3. Cada segmento é constituído de uma sequência linear de endereços.
4. O tamanho dos segmentos pode variar durante a execução.
5. O tamanho de cada segmento de pilha pode ser expandido sempre que algo é colocado sobre ela e diminuído sempre que algo é retirado dela.
6. Segmentos diferentes podem crescer ou diminuir independentemente e quando for necessário.

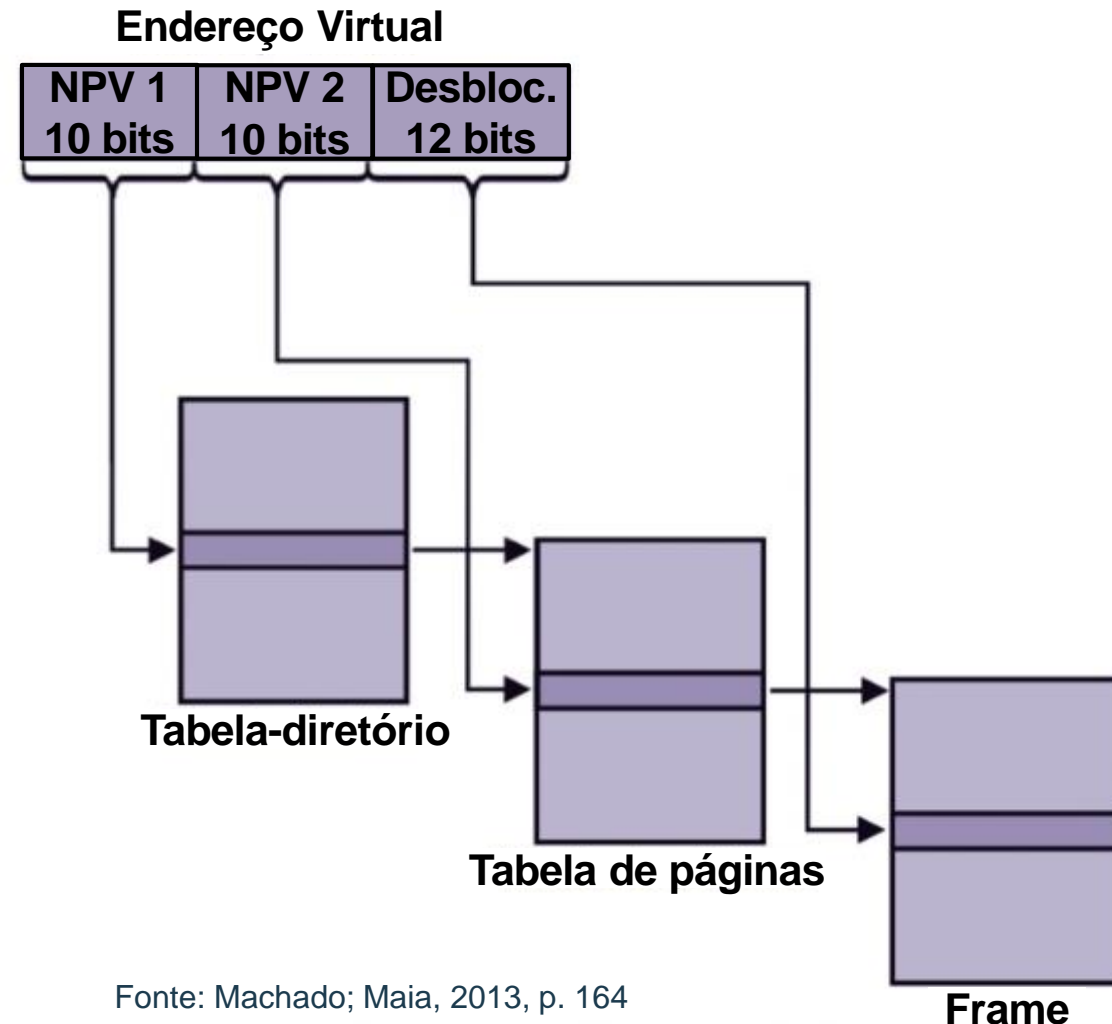
Paginação em um nível

- Problema crítico: qual o tamanho das páginas.
- NPV: número da página virtual.
- Para diversos processos residentes na memória principal, como seria possível manter e gerenciar tabelas desse tamanho para cada processo?



Paginação em dois níveis

- Existe uma tabela-diretório, em que cada entrada aponta para uma tabela de página.
- Divisão do campo NPV em duas partes:
 - número da página virtual de nível 1 (NPV1):
localiza a tabela de páginas na tabela-diretório.
 - número da página virtual de nível 2 (NPV2):
localiza o frame desejado na tabela de páginas.



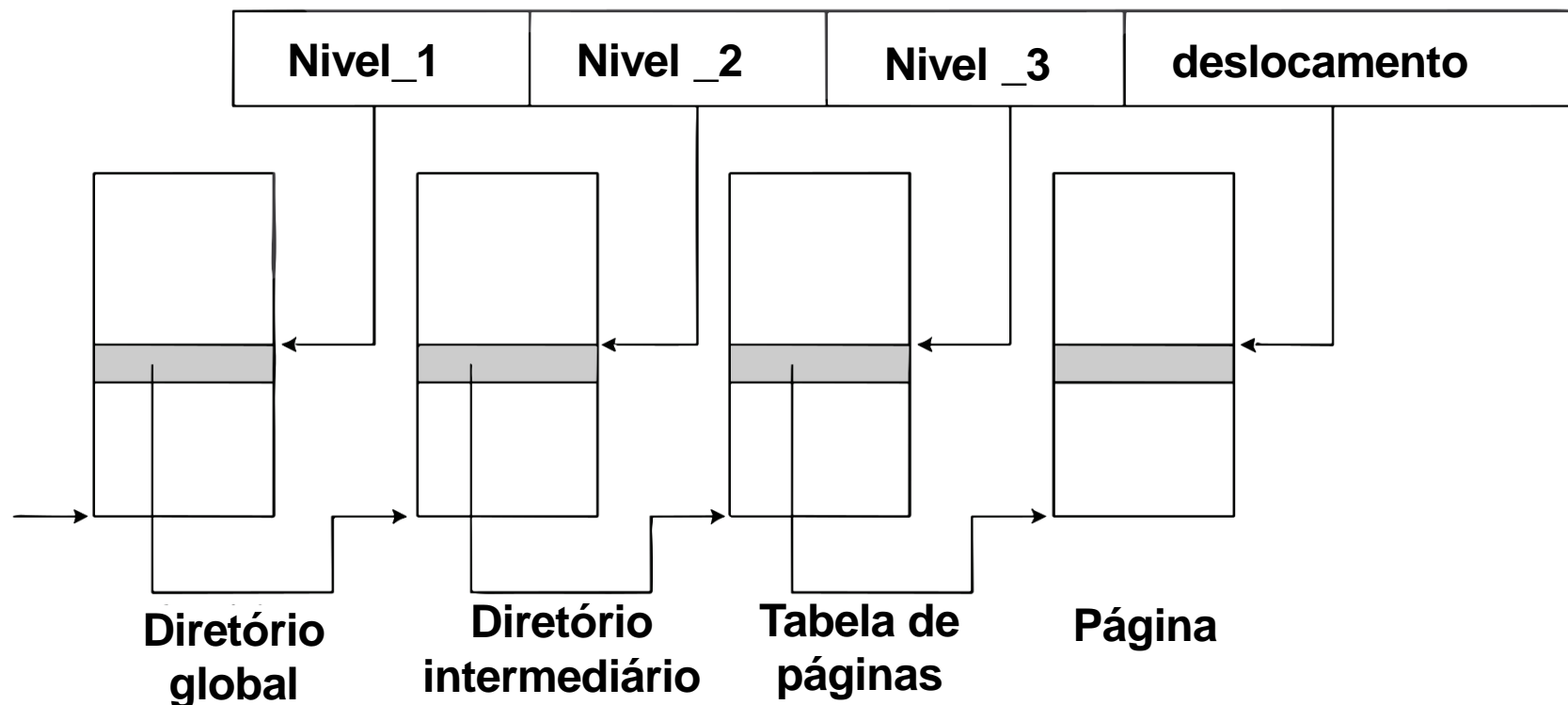
Paginação em múltiplos níveis

- Benefício: redução do espaço ocupado na memória.
 - Somente estarão residentes na memória principal as tabelas realmente necessárias aos processos.
- A técnica de paginação em múltiplos níveis pode ser estendida para três, quatro, cinco ou mais níveis.

Paginação no Linux : em 3 níveis

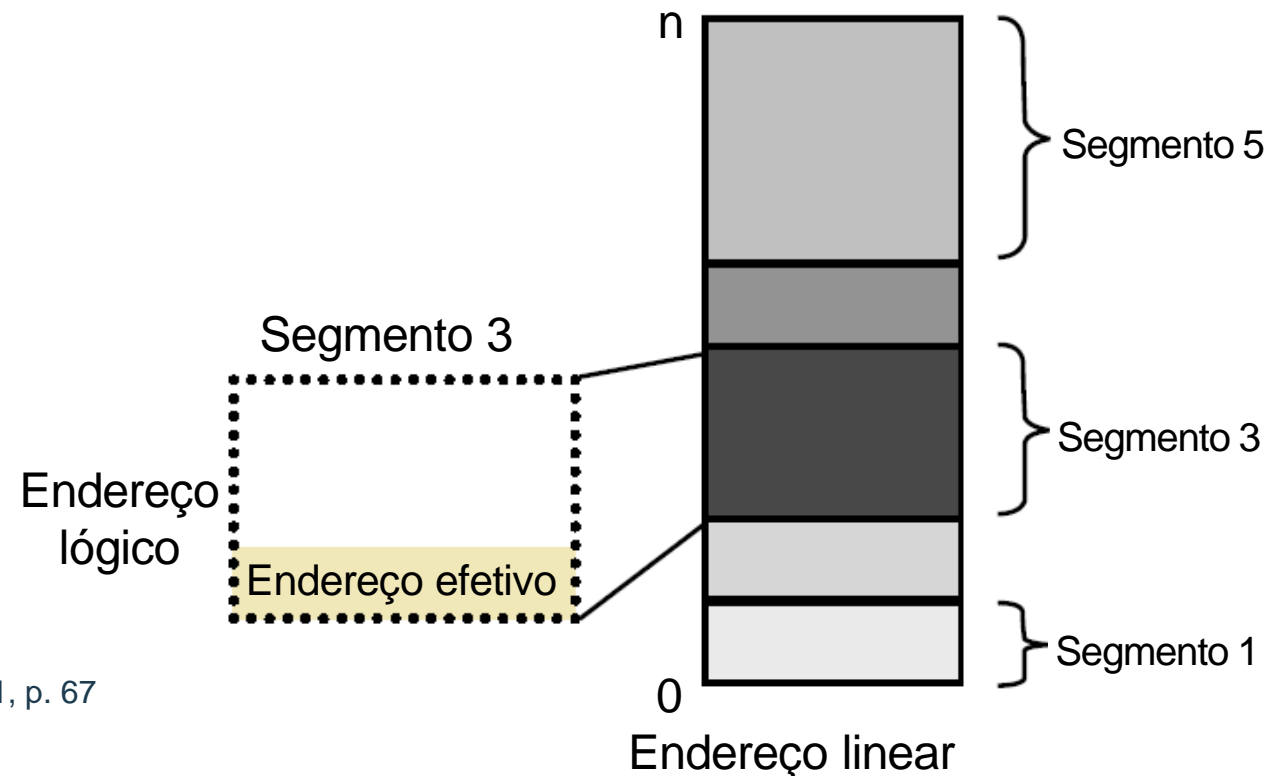
- Visando fornecer suporte a diferentes arquiteturas de processadores, o Linux considera uma paginação em três níveis.

Endereço Virtual



Memória virtual por segmentação

- Espaço de endereçamento virtual é dividido em blocos de tamanhos diferentes e arbitrários chamados de segmentos.
- Um programa será dividido logicamente em sub-rotinas e estruturas de dados, que são alocadas em segmentos na memória principal.
- Vantagem: flexibilidade que permite que o tamanho do segmento seja alterável.
- Desvantagem: causa fragmentação externa.
 - Há várias áreas livres na memória principal, mas nenhuma é suficiente para alocar um novo segmento.



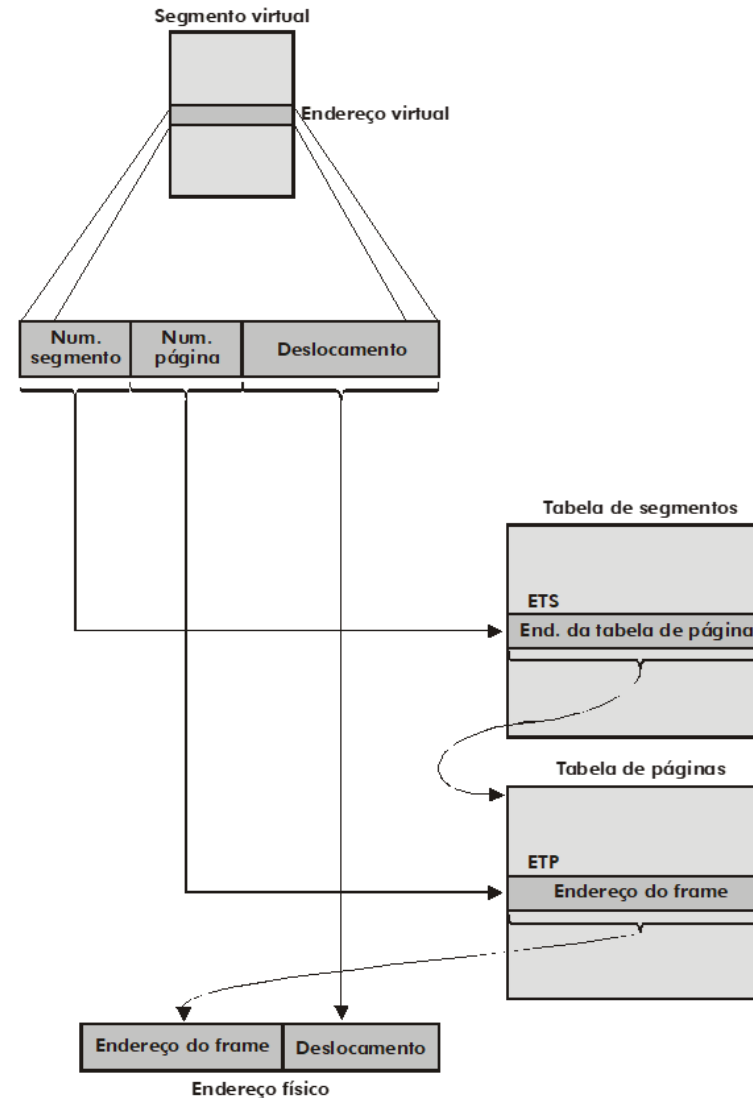
Paginação X Segmentação

- A tabela compara as técnicas de memória virtual por paginação e segmentação.

| Característica | Paginação | Segmentação |
|-------------------------------|-----------------|---------------|
| Tamanho dos blocos | Iguais | Diferentes |
| Proteção | Complexa | Mais simples |
| Compartilhamento | Complexo | Mais simples |
| Estruturas de dados dinâmicas | Complexo | Mais simples |
| Fragmentação interna | Pode existir | Inexistente |
| Programação modular | Dispensável | Indispensável |
| Alteração do programa | Mais trabalhosa | Mais simples |

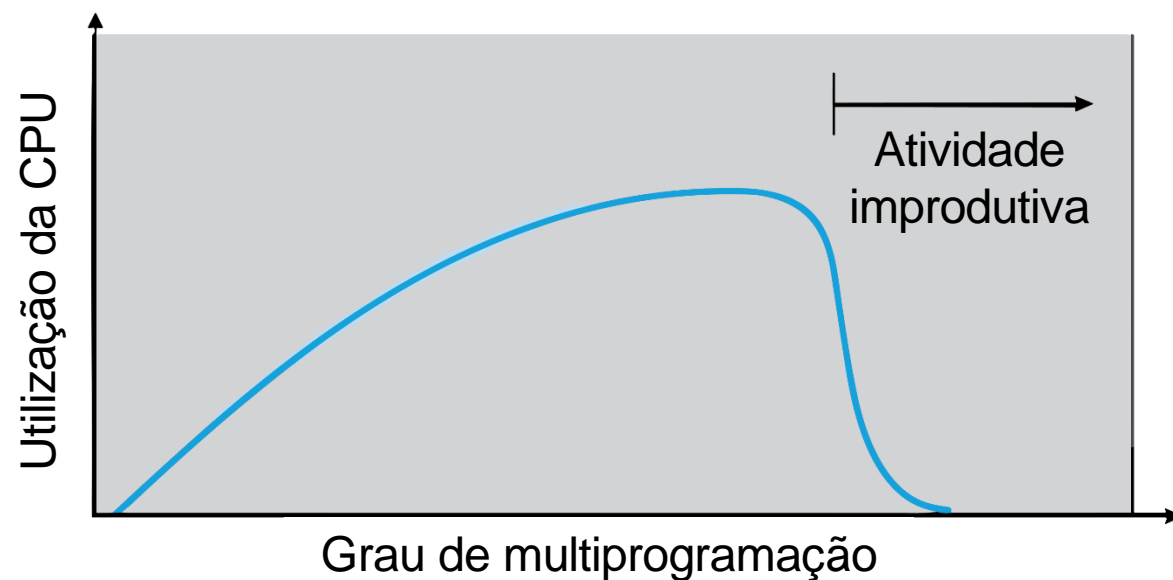
Paginação com segmentação

- O espaço de endereçamento é dividido em segmentos, e por sua vez cada segmento é dividido em páginas.
- Elimina-se a fragmentação externa.



Thrashing ou atividade improdutiva

- Representa uma situação do sistema operacional resultante da excessiva transferência de páginas ou segmentos entre a memória principal e a memória secundária.
- É possível limitar os efeitos da atividade improdutiva por meio de algoritmo de substituição por prioridades.
- Conforme o grau de multiprogramação aumenta, a utilização da CPU também aumenta, embora mais lentamente, até que um nível máximo seja alcançado.
- Se o grau de multiprogramação aumentar ainda mais, a atividade improdutiva se instala e a utilização da CPU cai significativamente.



Interatividade

Sobre gerenciamento de memória virtual, quando se divide a memória em blocos de tamanho diferentes e permite que as páginas sejam carregadas em quaisquer quadros disponíveis, está sendo implementando qual tipo de gerenciamento?

- a) Fifo.
- b) Segmentação de memória.
- c) Enquadramento de memória.
- d) Paginação de memória.
- e) Memória cache.

Resposta

Sobre gerenciamento de memória virtual, quando se divide a memória em blocos de tamanho diferentes e permite que as páginas sejam carregadas em quaisquer quadros disponíveis, está sendo implementando qual tipo de gerenciamento?

- a) Fifo.
- b) Segmentação de memória.
- c) Enquadramento de memória.
- d) Paginação de memória.
- e) Memória cache.

Referências

- FURUKAWA, F.; NUNES, R. *Fundamentos de Sistemas Operacionais*. São Paulo: Editora Sol, 2011.
- HENNESSY, J. L.; PATTERSON, D. A. *Organização e projeto de computadores: a Interface Hardware/software*. 5. ed. São Paulo: Gen/Ltc, 2017.
- MACHADO, F. B.; MAIA, L. P. *Arquitetura de sistemas operacionais*. 5. ed. Rio de Janeiro: LTC, 2013.
- MAZIERO, C. A. *Sistemas operacionais: conceitos e mecanismos [recurso eletrônico]*. Curitiba: DINF - UFPR, 2019.
- MONTEIRO, Mario A. *Introdução à organização de computadores*, 5. ed. Rio de Janeiro: LTC, 2007. p. 96.

Referências

- OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. *Sistemas operacionais* - V.11. 4 ed. Porto Alegre: Bookman, 2010.
- SILBERSCHATZ, A.; GALVIN, P. B. GAGNE, G. *Fundamentos de sistemas operacionais*. 9. ed. Rio de Janeiro: LTC, 2015.
- STALLINGS, William. *Arquitetura e organização de computadores*. 8. ed. São Paulo: Pearson Prantice Hall, 2010, p. 129.
- TANENBAUM, A. S.; BOS, H. *Sistemas operacionais modernos*. 4. ed. São Paulo: Pearson, 2016.
- TANENBAUM, S. A.; WOODHULL, S. *Sistemas operacionais: projetos e implementação - o livro do Minix*. 3. ed. Porto Alegre: Bookman, 2008.

ATÉ A PRÓXIMA!