

[< VOLTAR](#)

Gerenciamento de Memória

Abordar o conceito da Memória, a organização das informações na Memória e seu gerenciamento.

NESTE TÓPICO

- [> Introdução](#)
- [> Referências](#)

Marcar
tópico



Introdução

Desde os primeiros computadores, os Mainframes, a memória sempre teve um papel importante nos processamentos. Essa importante área armazena todos os processos (programas) que estão em processamento (em execução) e, assim que estes finalizam, a área se libera automaticamente para novas execuções, pois ela é volátil (a RAM perde os dados após o processamento do programa).

O verbo utilizado para a “reserva” de espaço na memória é “Alocar”, exemplo: um espaço é alocado ou desalocado na memória.

A memória pode ser organizada de diferentes formas:

- Um processo pode usar todo o espaço de memória: situação difícil hoje em dia em função dos grandes espaços de memória nos computadores;
- Cada processo obtém uma partição exclusiva na memória: situação comum;
- Alocada dinâmica ou estaticamente.

Tendência: a necessidade de mais espaço alocado na memória, por parte das aplicações, sempre foi aumentando e tende a continuar assim, pois a cada dia, as aplicações tornam-se mais complexas.

Hierarquia de memória

1 - Memória principal: É conhecida como RAM (Random Access Memory) nos microcomputadores e armazena as instruções de programa que estão sendo executados (estão em processamento) e também os dados processados por estes.

2 - Armazenamento secundário: São os discos (disco rígido, Hard Disk – HD, fitas e cartuchos), estes armazenam dados (em arquivos) e programas (em bibliotecas) que são acessados sempre que necessários: quando um programa é chamado para ser executado, o SO faz uma busca na biblioteca referente e o carrega na RAM.

3 - Memória cache:

- Sua velocidade é extremamente alta;
- Normalmente se localiza no próprio processador;
- Os dados mais comumente usados são copiados para o cache para que possam ser acessados mais rapidamente;
- Uma pequena quantidade de cache é suficiente para melhorar o desempenho.

Dessa forma, os programas são armazenados em memórias secundárias (armazenamento secundário), como discos ou fitas, por serem um meio não volátil (que não se perde), pois seu custo é baixo se comparado ao da memória principal. Essa diferença de custo se dá em função do material utilizado na RAM ser mais caro que o material utilizado na construção dos HDs e demais meios de armazenamento.

O processador somente executa tarefas que estiverem alocadas na memória principal. Para isso, o sistema operacional deve sempre transferir programas da memória secundária para a memória principal antes de serem executados. O tempo de acesso à memória secundária é maior do que o do acesso à memória principal. Para resolver isso, o sistema operacional deve buscar reduzir o número de operações de I/O (Input/Output: E/S, Entrada e Saída) à memória secundária, senão poderá ocorrer problemas no desempenho do sistema. Surge então a primeira função da gerência de memória: manter na memória principal o maior número de processos, permitindo aperfeiçoar o compartilhamento do processador e demais recursos.

Os processos só serão recusados no caso da capacidade da memória principal e da memória virtual estarem 100% utilizadas (caso raro hoje em dia). Em tempo, a memória virtual é uma extensão da memória principal,

localizada no HD.

O mecanismo que faz a transferência de processos da memória principal para a memória secundária (liberando espaço na memória principal) é conhecido como swapping.

Na multiprogramação, o sistema operacional deve proteger as áreas de memória ocupadas por cada processo e a área onde está alocado o próprio sistema. Se uma tarefa tentar acessar indevidamente algum endereço de memória, o sistema deverá impedir. Mesmo a gerência garantindo a proteção de áreas da memória, os SO (Sistemas Operacionais) não estão livres de possuir mecanismos de compartilhamento, para que os processos possam trocar dados de forma segura.

Alocação de Espaço na Memória

Esse algoritmo foi desenvolvido e implementado nos primeiros sistemas operacionais, porém ainda está presente em alguns sistemas monoprogramáveis. Visa dividir a memória principal em duas áreas: uma para o SO e outra para o programa do usuário final. O programador, ao desenvolver suas aplicações, deve se preocupar apenas em não ultrapassar o espaço de memória disponível à sua aplicação, que será a diferença entre o tamanho total da memória principal e a área ocupada pelo SO. Nessa arquitetura o usuário possui total controle sobre a memória principal, podendo acessar inclusive a área destinada ao sistema operacional. Para impedir violação do endereço de memória, os sistemas operacionais implementam um registrador que limita a área do usuário e do sistema operacional. Sendo assim, ao tentar acessar a memória, o sistema faz a verificação, se não estiver dentro do permitido, a tarefa será cancelada e uma mensagem de erro será demonstrada para o usuário.

Dessa forma temos a alocação de memória contígua (espaços sequentes) e não contígua (espaços não sequentes):

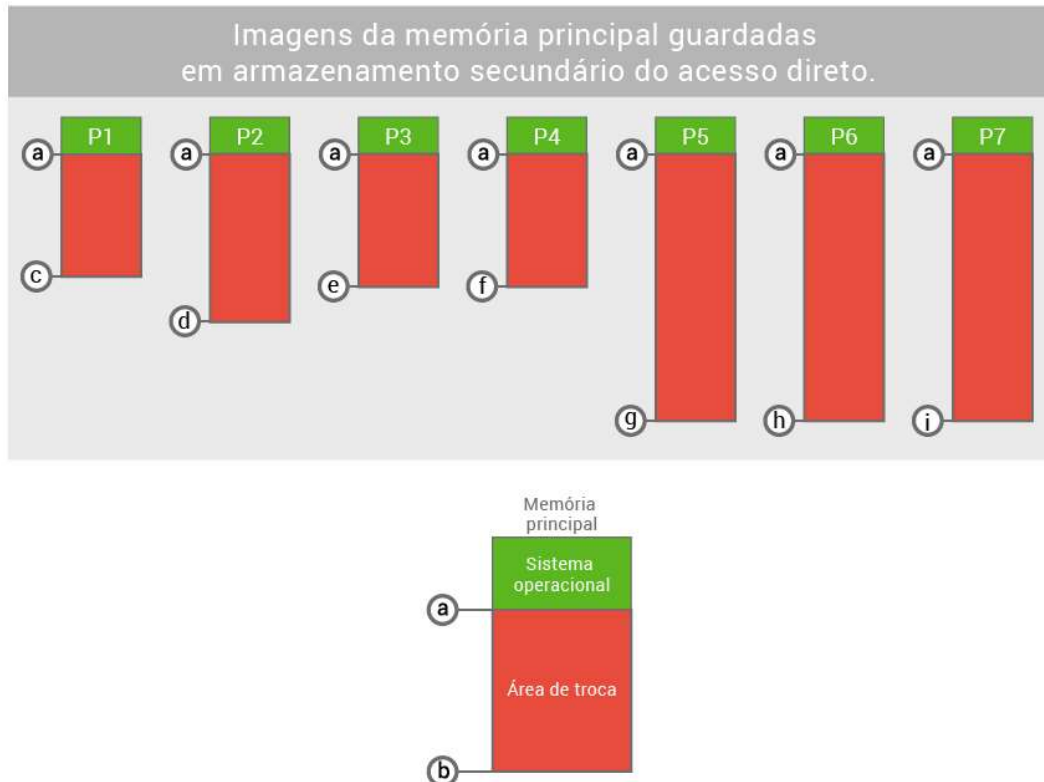
- Alocação contígua:
 - O programa deve estar em um bloco único de endereços contíguos alocados na memória;
 - Às vezes é impossível encontrar um bloco grande o suficiente;
 - Sua sobrecarga é baixa.
- Alocação não contígua:
 - O programa é dividido em porções denominadas segmentos;
 - Os segmentos podem ser posicionados em partes diferentes da memória;
 - É fácil encontrar “lacunas” nas quais o segmento possa se encaixar;
 - Pelo fato de poder haver mais processos simultâneos na memória, isso compensa a sobrecarga própria dessa técnica.

Swapping

Segundo Machado (2007), swapping é uma técnica que foi introduzida para contornar o problema de insuficiência de memória principal, aplicada à gerência de memória para programas que esperam por memória livre para serem executados. Nessa situação, o sistema escolhe um processo residente, que é transferido da memória principal para a memória secundária (swap out), geralmente disco. Posteriormente, o processo é carregado de volta da memória secundária para a memória principal (swap in) e pode continuar sua execução como se nada tivesse ocorrido. Para selecionar a tarefa a ser retirada da memória principal, o sistema operacional deve priorizar aquele com menores chances de ser escalonado para evitar o swapping desnecessário. Normalmente esses processos encontram-se no estado de espera, mas ainda existe a possibilidade de um processo no estado de pronto também ser selecionado para swapping.

Multiprogramação em um sistema de troca de processos (swapping) no qual há somente um único processo por vez na memória principal:

SEGMENTAÇÃO



1. Somente um processo por vez reside na memória principal.
2. Esse processo executa até:
 - a) ser emitida um E/S
 - b) o temporizador expirar
 - c) ocorrer término voluntário
3. Então o sistema retira o processo copiando a área de troca (memória principal) para armazenamento secundário.
4. O sistema admite o próximo processo lendo sua imagem na memória principal para a área de troca. o novo processo executa até ser eventualmente trocado pelo próximo processo e assim por diante.

Swapping.

Fonte: DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. Sistemas operacionais. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

Paginação

O processo de paginação faz com que o programa seja alocado em áreas não contíguas na memória. Dessa forma:

- O espaço de endereçamento lógico de um processo (programa em execução) é dividido em páginas lógicas de tamanho fixo;
- A memória física é dividida em páginas com tamanho fixo, com tamanho igual ao da página lógica;
- O programa é carregado página a página, cada página lógica ocupa uma página física;
- As páginas físicas não são necessariamente contíguas.

O endereço lógico é inicialmente dividido em duas partes:

- Um número de página lógica (usado como índice no acesso a tabela de páginas, de forma a obter o número da página física correspondente);
- Um deslocamento dentro da página.

PAGINAÇÃO

PAGINA 0	Inicialização
PAGINA 1	Laço de Repetição While { }
PAGINA 2
PAGINA 3	Fim do laço ... }
PAGINA 4	Finalização

Fonte: DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. Sistemas operacionais. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

Segmentação

É uma técnica onde programas são divididos em segmentos de tamanhos variados cada um com seu próprio espaço de endereçamento. A principal diferença entre a paginação e a segmentação é a alocação da memória de maneira não fixa, a alocação depende da lógica do programa.

O mapeamento é feito através das tabelas de mapeamento de segmentos e os endereços são compostos pelos:

- Número do segmento;
- Deslocamento dentro do segmento.

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 ocorrer sem problemas, o SO mantém uma tabela com as áreas livres e ocupadas da memória e somente segmentos referenciados são transferidos para a memória principal.

A seguir um exemplo de Segmentação:



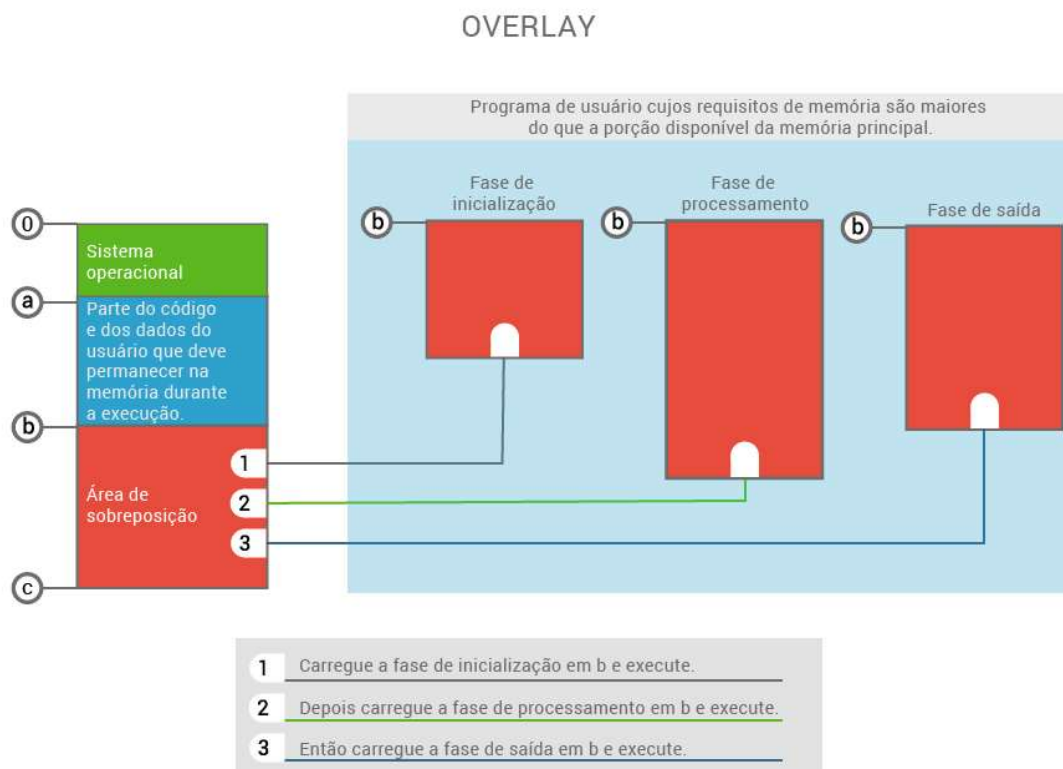
Fonte: DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. Sistemas operacionais. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

Overlay

Para Deitel (2005, p. 243), uma maneira de o desenvolvedor superar a limitação da memória principal seria criar overlays (sobreposições) que permitam ao sistema executar programas maiores do que a memória

principal. Quando um programa não necessita de memória para uma seção, o sistema poderá substituí-la, total ou parcialmente, pela memória para uma seção que necessite.

A figura a seguir ilustra o Overlay:



Fonte: DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. Sistemas operacionais. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

Este tópico finaliza aqui, esperamos que você tenha entendido como funciona a memória principal de um computador.

Bom estudo!

Quiz

Exercício

Gerenciamento de Memória

INICIAR ➤

Quiz

Exercício Final

Gerenciamento de Memória

INICIAR ➤

Referências

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. Sistemas operacionais. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

MACHADO, F. B., MAIA, L. P.; Arquitetura de Sistemas Operacionais; 4a Ed., Rio de Janeiro: LTC; 2007.

TANENBAUM, Andrew. Sistemas operacionais modernos, New Jersey; Prentice Hall; 2a; 2008.

Site Terminal de Informação. Disponível em: <terminaldeinformacao.com
(<http://terminaldeinformacao.com/2013/01/28/tudo-sobre-paginacao-e-segmentacao/>)>. Acesso em 27/04/2016.



Avalie este tópico



ANTERIOR

<

Sincronização de Processos

>

PRÓXIMO?

Biblioteca

(https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

Portal Uninove

(http://www.uninove.br)

Mapa do Site

Índice

Gerenciamento de Memória virtual

(https://ava.uninove.br/ava/curso=)

© Todos os direitos reservados

>

>