

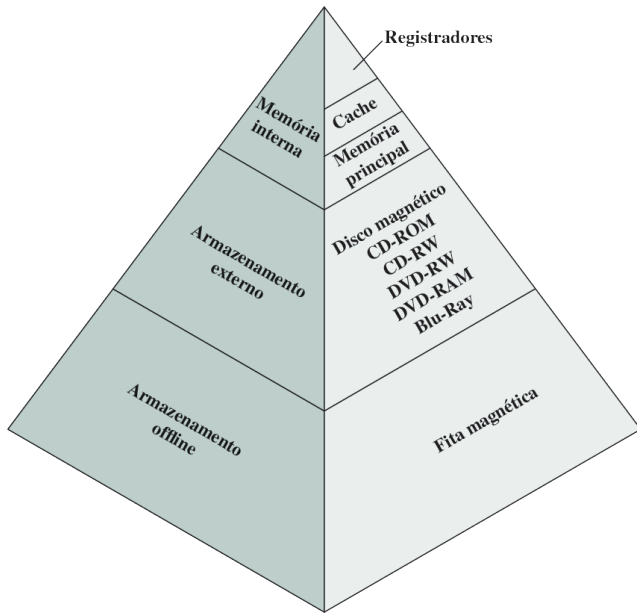
Gerenciamento de memória

Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

14 de Abril de 2023

Revisão memórias



- Conceito descreve a criação de uma representação simplificada;
- Uma visão abstraída de uma memória;
- Escondendo detalhes complexos;
- Gerenciar a memória;
- Várias técnicas;
- Linguagens de programação de alto nível;
- C++, Java, Python;
- Complexidade do gerenciamento de memória;
- Implicação direta no desempenho e segurança;
- Acesso ineficiente ou incorreto.

Sem abstração de memória

- A abstração de memória mais simples é não ter abstração alguma;
- Até 1980 os computadores não tinham abstração de memória;
- Cada programa apenas via a memória física;
- Várias opções eram possíveis:
- (a) Grande porte (b) Sistemas Embarcados (c) MS-DOS

Três maneiras simples de organizar a memória com um sistema operacional e um processo de usuário. Também existem outras possibilidades.



Sem abstração de memória

- Paralelismo em um sistema sem abstração de memória;
- Múltiplas threads;
- Mesma imagem da memória;
- Uso limitado;
- Programas não relacionados estejam executando ao mesmo tempo;
- Abstração de threads não realiza;
- Abstração de memória é improvável que proporcione uma abstração de threads;
- Mesmo sem uma abstração de memória, é possível executar múltiplos programas ao mesmo tempo.

- Falta de uma abstração de memória ainda é comum em sistemas embarcados e de cartões inteligentes.
- Casos em que o software se endereça à memória absoluta: rádios, máquinas de lavar roupas e fornos de micro-ondas.
- Smartphones, por exemplo, possuem sistemas operacionais elaborados.

- Um **espaço de endereçamento** é o conjunto de endereços que um processo pode usar para endereçar a memória.
- Cada processo tem seu próprio espaço de endereçamento, independente daqueles pertencentes a outros processos.
- Exemplos:
 - números de telefones;
 - portas de E/S;
 - endereços de IPv4;
 - conjunto de domínios da internet.

Registradores base e registradores limite

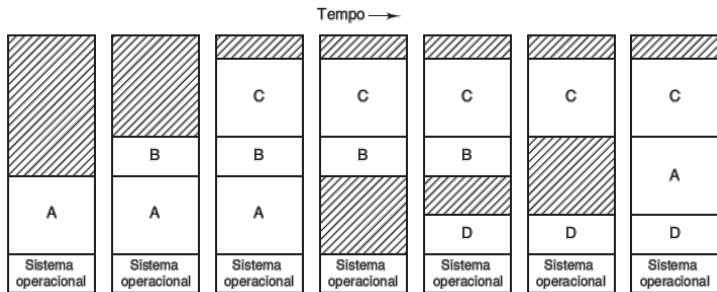
- São carregados em posições de memória consecutivas sempre que haja espaço e sem realocação durante o carregamento.
- Usar registradores base e limite é uma maneira fácil de dar a cada processo seu próprio espaço de endereçamento privado;
- Cada endereço de memória gerado automaticamente tem o conteúdo do registrador base adicionado a ele antes de ser enviado para a memória;
- Desvantagem: necessidade de realizar uma adição e uma comparação em cada referência de memória.

Troca de processos (Swapping)

- Estratégia simples;
- Lida com a sobrecarga de memória;
- Cada processo em sua totalidade, executá-lo por um tempo e então colocá-lo de volta no disco;
- Processos ociosos estão armazenados em disco;
- Processos podem criar múltiplos espaços na memória;
- É possível combiná-los em um grande espaço movendo todos os processos para baixo;
- **Compactação de memória.**

Troca de processos (Swapping)

- Mudanças na alocação de memória à medida que processos entram nela e saem dela. As regiões sombreadas são regiões não utilizadas da memória:



Quanta memória deve ser alocada para um processo quando ele é criado ou trocado?

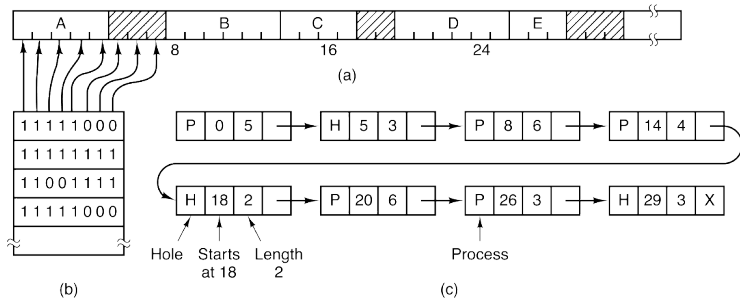
- Processos criados com um tamanho fixo que nunca muda é simples;
- O sistema operacional aloca exatamente o que é necessário;
- Um problema ocorre sempre que um processo tenta crescer;
- Se houver um espaço adjacente ao processo ele poderá ser alocado e o processo será autorizado a crescer naquele espaço;
- Processo for adjacente a outro, aquele que cresce terá de ser movido para um espaço na memória grande o suficiente para ele, ou um ou mais processos terão de ser trocados para criar um espaço grande o suficiente.

Quanta memória deve ser alocada para um processo quando ele é criado ou trocado?

- Se um processo não puder crescer em memória e a área de troca no disco estiver cheia, ele terá de ser suspenso até que algum espaço seja liberado (ou ele pode ser morto);
- Se o esperado for que a maioria dos processos cresça à medida que são executados, provavelmente seja uma boa ideia alocar um pouco de memória extra sempre que um processo for trocado ou movido, para reduzir a sobrecarga associada com a troca e movimentação dos processos que não cabem mais em sua memória alocada.

Gerenciamento de memória com mapas de bits

- Correspondendo a cada unidade de alocação há um bit no mapa de bits, que é 0 se a unidade estiver livre e 1 se ela estiver ocupada;



- Outra maneira de controlar o uso da memória é manter uma lista encadeada de espaços livres e de segmentos de memória alocados.
- Algoritmos:
 - **first fit** (procura por um espaço livre que tenha o tamanho suficiente);
 - **next fit** (uma pequena variação do first fit, exceto por memorizar a posição que se encontra um espaço livre adequado sempre que o encontra);
 - **best fit** (faz uma busca em toda a lista, do início ao fim, e escolhe o menor espaço livre que seja adequado);
 - **worst fit** (sempre escolhe o maior espaço livre);
 - **quick fit** (mantém listas em separado para alguns dos tamanhos mais comuns solicitados).

- Apesar de os tamanhos das memórias aumentarem depressa, os tamanhos dos softwares estão crescendo muito mais rapidamente;
- Necessidade de executar programas que são grandes demais para se encaixar na memória;
- Necessidade de ter sistemas que possam dar suporte a múltiplos programas executando em simultâneo;
- Cada um deles encaixando-se na memória, mas com todos coletivamente excedendo-a.

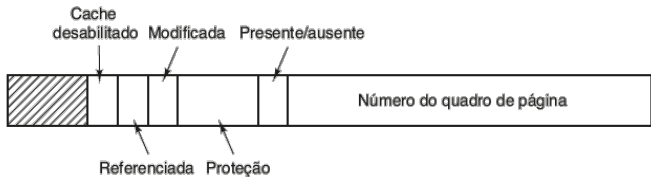
- Método encontrado para passar todo o programa para o computador.
- Cada programa tem seu próprio espaço de endereçamento, o qual é dividido em blocos chamados de **páginas**;
- Generalização da ideia do registrador base e registrador limite;
- Funciona bem em um sistema de multiprogramação, com pedaços e partes de muitos programas na memória simultaneamente;
- Enquanto um programa está esperando que partes de si mesmo sejam lidas, a CPU pode ser dada para outro processo.

- Técnica usada pela maioria dos sistemas de memória virtual;
- O espaço de endereçamento virtual consiste em unidades de tamanho fixo chamadas de **páginas**;
- As unidades correspondentes na memória física são chamadas de **quadros de página**;
- Transferências entre a memória RAM e o disco são sempre em páginas inteiras;
- No hardware real, um bit Presente/ausente controla quais páginas estão fisicamente presentes na memória;
- Muitos processadores dão suporte a múltiplos tamanhos de páginas que podem ser combinados e casados como o sistema operacional preferir;
- Há quatro momentos em que o sistema operacional tem de se envolver com a paginação: **na criação do processo, na execução do processo, em faltas de páginas e no término do processo.**

- O objetivo da tabela de páginas é mapear as páginas virtuais em quadros de páginas;
- Função com o número da página virtual como argumento e o número do quadro físico como resultado;
- Usando o resultado dessa função, pode ser substituído por um campo de quadro de página, desse modo formando um endereço de **memória física**.

Tabelas de páginas

- Estrutura de uma entrada da tabela de páginas:



- O tamanho varia de computador para computador (32 bits);
- O campo mais importante é o Número do quadro de página;
- Objetivo do mapeamento de páginas é localizar esse valor.

- Sistema de paginação, duas questões fundamentais precisam ser abordadas:
 - O mapeamento do endereço virtual para o endereço físico precisa ser rápido;
 - Se o espaço do endereço virtual for grande, a tabela de páginas será grande.
- O projeto mais simples é ter uma única tabela de página consistindo de uma série de registradores, com uma entrada para cada página virtual, indexada pelo número da página virtual.

Memória Associativa

- Pequeno dispositivo de hardware para mapear endereços virtuais para endereços físicos sem ter de passar pela tabela de páginas.

Válida	Página virtual	Modificada	Proteção	Quadro de página
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

- **Ausência leve** (soft miss): ocorre quando a página referenciada não se encontra na TLB, mas está na memória;
- **Ausência completa** (hard miss): ocorre quando a página em si não está na memória e também não está na TLB;
- **Passeio na tabela de páginas** (page table walk): procurar o mapeamento na hierarquia da tabela de páginas.

Algoritmos de substituição de páginas

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usado recentemente)	Aproximação muito rudimentar do LRU
FIFO (primeiro a entrar, primeiro a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
LRU (usada menos recentemente)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usado)	Aproximação bastante rudimentar do LRU
Envelhecimento (<i>aging</i>)	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

Políticas de alocação local versus global

- Algoritmos locais correspondem a alocar a todo processo uma fração fixa da memória;
- Algoritmos globais alocam dinamicamente quadros de páginas entre os processos executáveis. O número de quadros de páginas designadas a cada processo varia com o tempo;
- Algoritmos globais funcionam melhor;
- Especialmente quando o tamanho do conjunto de trabalho puder variar muito.

Políticas de alocação local versus global

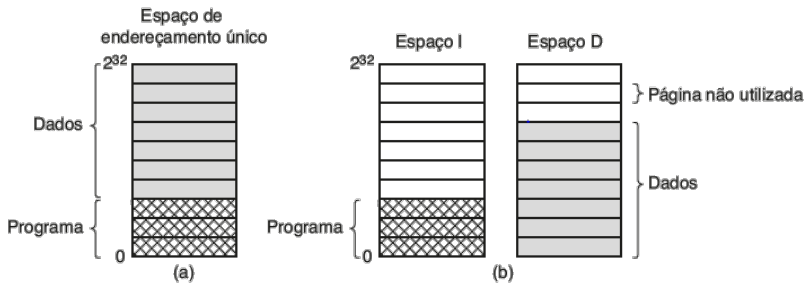
- Outra abordagem é ter um algoritmo para alocar quadros de páginas para processos.
- Uma maneira de gerenciar a alocação é usar o algoritmo **PFF (Page Fault Frequency** — frequência de faltas de página). Ele diz quando aumentar ou diminuir a alocação de páginas de um processo, mas **não** diz nada sobre qual página substituir em uma falta. Ele apenas **controla o tamanho do conjunto de alocação**.
- A escolha de local versus global, em alguns casos, é independente do algoritmo.

- Caso ocorra uma ultrapaginação, será necessário dar mais memória a alguns processos;
- Pelo fato de não haver um modo de dar mais memória àqueles processos que precisam dela sem prejudicar alguns outros, a única solução real é livrar-se temporariamente de alguns processos;
- Uma boa maneira de reduzir o número de processos competindo pela memória é levar alguns deles para o disco e liberar todas as páginas que eles estão segurando;
- É preciso considerar também o grau de multiprogramação;
- Tamanho dos processos;
- Frequência da paginação;
- Decidir qual processo deve ser trocado.

- É um parâmetro que pode ser escolhido pelo sistema operacional.
- Determinar o melhor tamanho de página exige equilibrar vários fatores, pois não há um tamanho ótimo geral.
- O espaço extra de uma página que é desperdiçado é denominado fragmentação interna.
- Um tamanho de página grande pode causar mais desperdício de espaço na memória.
- Páginas pequenas ocupam muito espaço no TLB.

Espaços separados de instruções e dados

- Se o espaço de endereçamento for grande o suficiente, tudo funcionará bem;
- Se for pequeno demais, ele força os programadores a encontrar uma saída para fazer caber tudo nesse espaço.
- Uma solução é ter dois espaços de endereçamento diferentes para instruções e dados, chamados de espaço I e espaço D.



- É eficiente compartilhar as páginas para evitar ter duas cópias da mesma página na memória ao mesmo tempo.
- Pesquisar todas as tabelas de páginas para ver se uma página está sendo compartilhada normalmente é muito caro;
- Portanto estruturas de dados especiais são necessárias para controlar as páginas compartilhadas;

Bibliotecas compartilhadas

- São chamadas de DLLs ou Dynamic Link Libraries — Bibliotecas de Ligação Dinâmica — no Windows;
- Tornam arquivos executáveis menores e também salvar espaço na memória;
- Vantagem: se uma função em uma biblioteca compartilhada for atualizada para remover um erro, não será necessário recompilar os programas que a chamam e os antigos arquivos binários continuam a funcionar;
- Desvantagem: a realocação durante a execução não funciona. É preciso criar novas páginas para cada processo compartilhando a biblioteca, realocando-as dinamicamente.

- Trata-se de um processo que pode emitir uma chamada de sistema para mapear um arquivo em uma porção do seu espaço virtual;
- Os processos acessem arquivos em disco como se fossem parte da memória principal do sistema;
- As bibliotecas podem usar esse mecanismo se arquivos mapeados em memória estiverem disponíveis.

- Os sistemas de paginação geralmente têm um processo de segundo plano;
- Chamado de daemon de paginação;
- Periodicamente inspeciona o estado da memória e garante que todos os quadros estejam limpos;
- Assim eles não precisam ser escritos às pressas para o disco quando requisitados;
- O objetivo é otimizar o uso da memória;
- Permitindo que os processos acessem mais memória do que está fisicamente disponível na RAM;
- Minimiza a sobrecarga de transferência de dados entre a RAM e o armazenamento secundário.

- Ajuda a lidar com estruturas de dados que podem mudar de tamanho durante a execução e simplifica a ligação e o compartilhamento;
- Facilita proporcionar proteção para diferentes segmentos;
- Às vezes a segmentação e a paginação são combinadas para proporcionar uma memória virtual bidimensional.

Segmentação X Paginação

Consideração	Paginação	Segmentação
O programador precisa saber que essa técnica está sendo usada?	Não	Sim
Há quantos espaços de endereçamento linear?	1	Muitos
O espaço de endereçamento total pode superar o tamanho da memória física?	Sim	Sim
Rotinas e dados podem ser distinguidos e protegidos separadamente?	Não	Sim
As tabelas cujo tamanho flutua podem ser facilmente acomodadas?	Não	Sim
O compartilhamento de rotinas entre os usuários é facilitado?	Não	Sim
Por que essa técnica foi inventada?	Para obter um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física	Para permitir que programas e dados sejam divididos em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção

Segmentação com paginação

- Se os segmentos forem grandes, talvez não seja possível mantê-los na memória principal em sua totalidade;
- Paginação dos segmentos, de maneira que apenas aquelas páginas de um segmento que são realmente necessárias tenham de estar na memória;
- O sistema MULTICS e o x86 de 32 bits da Intel dão suporte à segmentação e à paginação;
- Não dão mais suporte a uma segmentação de verdade.

- Hoje, a pesquisa sobre a paginação se concentra em tipos novos de sistemas.
- **Paginação em sistemas com múltiplos núcleos:** esses tipos de sistemas tendem a possuir muita memória cache compartilhada de maneiras complexas.
- **Paginação em sistemas NUMA:** diversas partes da memória podem ter diferentes tempos de acesso.
- Smartphones e tablets.
- Sistemas em tempo real.

- TANENBAUM, A S. **Sistemas Operacionais Modernos**. 2 ed. Prentice-Hall, 2003;

Obrigado! Dúvidas?

Guilherme Henrique de Souza Nakahata

guilhermenakahata@gmail.com

<https://github.com/GuilhermeNakahata/UNESPAR-2023>