

Nota: Este material complementar, disponível em <https://prettore.github.io/lectures.html> representa uma cópia resumida de conteúdos bibliográficos disponíveis gratuitamente na Internet.

Gerenciamento de Memória

Introdução	1
Conceitos Fundamentais	1
Gerenciamento de Memória RAM com e sem Abstração	2
Gerenciamento sem Abstração (Bare Machine)	2
Gerenciamento com Abstração (Alocação Contígua Simples)	2
Memória Virtual Paginada	2
Otimizações de Memória Virtual Paginada	3
Translation Lookaside Buffer (TLB)	3
Tabela de Páginas Invertidas (Inverted Page Table)	4
Algoritmos de Substituição de Páginas	4
Questões de Projeto e de Implementação de Memórias Paginadas	5
Memória Virtual Segmentada	5
Segmentação com Paginação	6
Conclusão	6
Referências	6

Introdução

O gerenciamento de memória é uma das responsabilidades mais críticas de um sistema operacional. A memória principal (RAM) é um recurso essencial, mas geralmente limitado, que deve ser compartilhado entre o sistema operacional e múltiplos processos de usuário. O gerenciador de memória do SO tem a tarefa de alocar e desalocar espaço de memória para os processos, manter o controle de quais partes da memória estão em uso e por quem, e garantir que os processos não interfiram uns com os outros. Além disso, sistemas modernos empregam técnicas de memória virtual para permitir que os processos utilizem um espaço de endereçamento maior do que a memória física disponível, melhorando a utilização da CPU e a facilidade de programação. Este capítulo explora os conceitos fundamentais do gerenciamento de memória, incluindo o gerenciamento de RAM com e sem abstração, a memória virtual paginada, otimizações como TLB e tabelas de páginas invertidas, algoritmos de substituição de páginas, questões de projeto e implementação, e a memória virtual segmentada.

Conceitos Fundamentais

- **Espaço de Endereçamento (Address Space):** O conjunto de todos os endereços lógicos que um processo pode gerar. Cada processo tem seu próprio espaço de endereçamento virtual, que é independente do espaço de endereçamento físico.

- **Endereço Lógico vs. Endereço Físico:**
 - **Endereço Lógico (ou Virtual):** Gerado pela CPU; também conhecido como endereço virtual. Deve ser traduzido para um endereço físico antes de ser usado.
 - **Endereço Físico:** O endereço visto pela unidade de memória; o endereço real na RAM.
- **Unidade de Gerenciamento de Memória (MMU - Memory Management Unit):** Um dispositivo de hardware que mapeia endereços virtuais para endereços físicos em tempo de execução.
- **Alocação Contígua vs. Não Contígua:**
 - **Contígua:** Cada processo ocupa uma única seção contígua de memória física.
 - **Não Contígua:** Um processo pode ser alocado em várias seções fisicamente dispersas da memória (e.g., paginação, segmentação).

Gerenciamento de Memória RAM com e sem Abstração

Gerenciamento sem Abstração (Bare Machine)

Em sistemas muito simples ou antigos, os programas tinham acesso direto à memória física. Não havia proteção entre processos ou entre processos e o SO. Essa abordagem é raramente usada hoje devido à sua falta de segurança e flexibilidade.

Gerenciamento com Abstração (Alocação Contígua Simples)

- **Partições Fixas (Fixed Partitions):** A memória é dividida em partições de tamanho fixo. Cada partição pode conter um processo. Pode levar à fragmentação interna (espaço desperdiçado dentro de uma partição alocada se o processo for menor que a partição).
- **Partições Variáveis (Variable Partitions):** O SO mantém uma tabela de partes da memória disponíveis (buracos) e partes ocupadas. Quando um processo chega, ele é alocado a um buraco grande o suficiente. Pode levar à fragmentação externa (espaço total de memória existe para satisfazer uma requisição, mas não é contíguo).
 - **Algoritmos de Alocação:** First-fit (primeiro buraco grande o suficiente), Best-fit (menor buraco grande o suficiente), Worst-fit (maior buraco).

Memória Virtual Paginada

A memória virtual é uma técnica que permite a execução de processos que podem não estar completamente na memória física. Isso permite que os programadores tenham um espaço de endereçamento virtual muito grande, independentemente da memória física disponível, e que mais processos possam ser mantidos na memória, aumentando a utilização da CPU e o throughput.

A **paginação** é o esquema de gerenciamento de memória mais comum para implementar memória virtual.

- **Páginas (Pages):** O espaço de endereçamento virtual de um processo é dividido em blocos de tamanho fixo chamados páginas.
- **Quadros (Frames):** A memória física é dividida em blocos de tamanho fixo chamados quadros (ou molduras). O tamanho da página e do quadro são iguais.
- **Tabela de Páginas (Page Table):** Para cada processo, o SO mantém uma tabela de páginas que mapeia cada página virtual para o quadro físico correspondente onde ela está armazenada. Cada entrada na tabela de páginas contém o número do quadro e bits de controle (e.g., bit de validade/invalidade, bit de proteção, bit de presença, bit de modificado, bit de referenciado).
- **Tradução de Endereços:** Um endereço lógico gerado pela CPU é dividido em duas partes: um número de página (p) e um deslocamento dentro da página (d). O número da página é usado como índice na tabela de páginas para encontrar o número do quadro base. O endereço físico é formado pela concatenação do número do quadro base com o deslocamento.
- **Page Fault (Falta de Página):** Se um processo tenta acessar uma página que não está atualmente na memória física (indicado pelo bit de presença/validade na tabela de páginas), ocorre uma falta de página. O SO deve então:
 1. Verificar se o acesso é válido.
 2. Encontrar um quadro livre na memória física.
 3. Agendar uma operação de disco para carregar a página necessária do dispositivo de armazenamento secundário (e.g., disco) para o quadro livre.
 4. Modificar a tabela de páginas para indicar que a página está agora na memória.
 5. Reiniciar a instrução que causou a falta de página.

Otimizações de Memória Virtual Paginada

Translation Lookaside Buffer (TLB)

Acessar a tabela de páginas (que geralmente está na memória principal) para cada referência à memória pode ser lento. O TLB é um cache de hardware pequeno e rápido, associativo e gerenciado pelo hardware da MMU, que armazena um subconjunto das entradas da tabela de páginas usadas recentemente.

- **Funcionamento:** Quando um endereço virtual é gerado, o número da página é apresentado ao TLB. Se houver um acerto no TLB (TLB hit), o número do quadro é obtido rapidamente do TLB e o endereço físico é formado. Se houver uma falha no TLB (TLB miss), a tabela de páginas na memória principal deve ser consultada. A entrada da tabela de páginas é então carregada no TLB (possivelmente substituindo uma entrada existente) para uso futuro.
- **Impacto no Desempenho:** Uma alta taxa de acertos no TLB é crucial para o desempenho de sistemas paginados.

Tabela de Páginas Invertidas (Inverted Page Table)

Em sistemas com espaços de endereçamento virtual muito grandes (e.g., 64 bits), as tabelas de páginas tradicionais (uma por processo) podem se tornar excessivamente grandes. Uma tabela de páginas invertida tem uma entrada para cada quadro de memória física real, em vez de uma para cada página virtual.

- **Estrutura:** Cada entrada na tabela de páginas invertida indica qual par (processo, página virtual) está localizado naquele quadro físico.
- **Busca:** Para traduzir um endereço lógico (processo, página), o sistema deve pesquisar na tabela de páginas invertida para encontrar uma entrada que corresponda ao par (processo, página). Essa busca pode ser lenta e geralmente requer o uso de tabelas hash para acelerá-la.
- **Vantagem:** O tamanho da tabela de páginas é proporcional ao tamanho da memória física, não ao tamanho total do espaço de endereçamento virtual de todos os processos.

Algoritmos de Substituição de Páginas

Quando ocorre uma falta de página e não há quadros livres na memória física, o sistema operacional deve escolher uma página residente na memória para ser substituída (swapped out) para o disco, a fim de liberar um quadro para a página recém-solicitada. O objetivo dos algoritmos de substituição de páginas é selecionar a página que tem menor probabilidade de ser referenciada no futuro próximo, minimizando assim o número de faltas de página.

- **Algoritmo Ótimo (OPT ou MIN):** Substitui a página que não será usada pelo maior período de tempo no futuro. É o algoritmo ideal e garante a menor taxa de faltas de página, mas é irrealizável na prática, pois requer conhecimento do futuro.
- **First-In, First-Out (FIFO):** Substitui a página que está na memória há mais tempo. Simples de implementar, mas pode ter um desempenho ruim (e.g., Anomalia de Belady, onde aumentar o número de quadros pode aumentar o número de faltas de página).
- **Least Recently Used (LRU) / Menos Recentemente Usada:** Substitui a página que não foi usada pelo maior período de tempo. Baseia-se na premissa de localidade temporal (páginas usadas recentemente provavelmente serão usadas novamente em breve). Geralmente tem bom desempenho, mas é difícil de implementar perfeitamente, pois requer hardware para rastrear o tempo da última referência a cada página.
 - **Implementações Aproximadas de LRU:**
 - **Algoritmo do Bit de Referência Adicional (Additional-Reference-Bits Algorithm):** Associa um contador a cada página. Periodicamente, o bit de referência é adicionado ao contador. A página com o menor contador é substituída.
 - **Algoritmo da Segunda Chance (Second-Chance Algorithm / Clock Algorithm):** Uma variação do FIFO. Usa um bit de referência. Quando uma página é selecionada para substituição, seu bit de

referência é verificado. Se for 0, a página é substituída. Se for 1, o bit é zerado e a página recebe uma “segunda chance”, e o ponteiro avança para a próxima página na fila circular.

- **Counting-Based Algorithms:**
 - **Least Frequently Used (LFU):** Substitui a página com o menor contador de referências.
 - **Most Frequently Used (MFU):** Substitui a página com o maior contador de referências (baseado na ideia de que páginas muito usadas podem não ser mais necessárias).

Questões de Projeto e de Implementação de Memórias Paginadas

- **Alocação de Quadros:** Quantos quadros alocar para cada processo? Pode ser alocação fixa (igual para todos ou proporcional ao tamanho) ou alocação prioritária.
- **Escopo da Substituição:**
 - **Substituição Global:** Uma página vítima pode ser selecionada de qualquer quadro na memória, independentemente de qual processo ela pertence.
 - **Substituição Local:** A página vítima é selecionada apenas dentre os quadros alocados ao processo que causou a falta de página.
- **Thrashing:** Ocorre quando um sistema passa a maior parte do tempo paginando (trocando páginas entre disco e memória) em vez de executar. Geralmente acontece quando os processos não têm quadros suficientes para manter seu conjunto de trabalho (working set - o conjunto de páginas ativamente usadas por um processo). O sistema pode detectar thrashing monitorando a utilização da CPU e a taxa de paginação.
- **Tamanho da Página:** Um tamanho de página menor leva a menos fragmentação interna, mas a tabelas de páginas maiores e mais sobrecarga de E/S. Um tamanho de página maior reduz o tamanho da tabela de páginas e a sobrecarga de E/S, mas aumenta a fragmentação interna.

Memória Virtual Segmentada

A **segmentação** é outro esquema de gerenciamento de memória que suporta a visão do usuário sobre a memória. Um programa é uma coleção de segmentos. Um segmento é uma unidade lógica, como um procedimento principal, uma função, um objeto, variáveis locais, variáveis globais, a pilha, uma tabela de símbolos, etc.

- **Espaço de Endereçamento Lógico:** É uma coleção de segmentos. Cada segmento tem um nome (ou número) e um tamanho.
- **Endereço Lógico:** Consiste em um par: (número_do_segmento, deslocamento).
- **Tabela de Segmentos (Segment Table):** Mapeia o endereço lógico bidimensional para um endereço físico unidimensional. Cada entrada na tabela de segmentos tem uma base do segmento (endereço físico inicial do segmento) e um limite do segmento (tamanho do segmento).

- **Proteção e Compartilhamento:** A segmentação facilita a proteção (associando bits de proteção a cada segmento) e o compartilhamento de segmentos entre processos.

Segmentação com Paginação

Alguns sistemas combinam segmentação com paginação para obter as vantagens de ambos. O espaço de endereçamento lógico é dividido em segmentos, e cada segmento é então dividido em páginas. Isso é usado, por exemplo, na arquitetura Intel x86.

Conclusão

O gerenciamento de memória é uma tarefa complexa e crucial para o desempenho e a funcionalidade dos sistemas operacionais. Desde as técnicas básicas de alocação contígua até os sofisticados esquemas de memória virtual paginada e segmentada, o objetivo é fornecer aos processos um espaço de endereçamento conveniente e eficiente, enquanto se gerencia o recurso limitado da memória física. O uso de tabelas de páginas, TLBs e algoritmos de substituição de páginas são componentes chave na implementação da memória virtual, permitindo que os sistemas executem programas maiores que a memória física e melhorem a utilização dos recursos. A compreensão desses mecanismos é fundamental para entender como os sistemas operacionais modernos operam e como otimizar o desempenho das aplicações.

Referências

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson Education.
- Stallings, W. (2018). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.

Isenção de Responsabilidade:

Os autores deste documento não reivindicam a autoria do conteúdo original compilado das fontes mencionadas. Este documento foi elaborado para fins educativos e de referência, e todos os créditos foram devidamente atribuídos aos respectivos autores e fontes originais.

Qualquer utilização comercial ou distribuição do conteúdo aqui compilado deve ser feita com a devida autorização dos detentores dos direitos autorais originais. Os compiladores deste documento não assumem qualquer responsabilidade por eventuais violações de direitos autorais ou por quaisquer danos decorrentes do uso indevido das informações contidas neste documento.

Ao utilizar este documento, o usuário concorda em respeitar os direitos autorais dos autores originais e isenta os compiladores de qualquer responsabilidade relacionada ao conteúdo aqui apresentado.