RESUMO GERENCIAMENTO DE MEMÓRIA

VINCULAÇÃO DE ENDEREÇOS

1. Memória e Acessos:

- A CPU acessa diretamente a memória principal, registradores e cache, com foco em otimizar a velocidade.
- Dados devem estar na memória principal para serem processados pela CPU.
- Para concluir o um acesso a memória, que muitas vezes pode ser lento, usa-se cache, memoria intermediaria que pode acelerar o processo.

2. Controle de Acesso à Memória:

- A multiprogramação permite múltiplos processos na memória simultaneamente.
- Uso de registradores base e limite para definir os intervalos de memória permitidos para cada processo.
- Um processo de usuário pode residir em qualquer parte da memória.
- Um processo de um usuário não pode modificar ou alterar estruturas de outros processos.

Registrador base: o menor endereço legal da memória física

Registrador limite: o tamanho do intervalo

3. Vinculação de Endereços:

- Mapeamento de endereços **simbólicos** no código-fonte para endereços **reais** na memória.
- A vinculação pode ocorrer em tempo de compilação, carga ou execução.
- Separação entre endereços lógicos (gerados pela CPU) e físicos (visíveis pela unidade de memória).

Espaço de endereçamento lógico é o conjunto de endereços lógicos gerados por um programa.

Espaço de endereçamento físico é o conjunto de endereços físicos correspondentes aos lógicos.

4. Mapeamento Lógico-Físico:

- Feito pela Unidade de Gerenciamento de Memória (MMU).
- Esquema simples de MMU adiciona o valor do registrador base ao endereço lógico.

5. Carga Dinâmica:

- Partes do processo são carregadas na memória conforme necessário, economizando espaço.
- Permite executar programas grandes utilizando apenas os trechos frequentemente usados.

6. Vinculação de Bibliotecas:

- Pode ser estática (integrada ao programa) ou dinâmica (carregada em tempo de execução via stubs).

Stubs -> Stub é um fragmento de código que indica como localizar na memória as rotinas da biblioteca ou como carregá-las

7. Permuta (Swapping):

- Troca de processos entre memória principal e memória de retaguarda para otimizar recursos.

Swap out -> Quando um processo sai da memória principal e vai para a memória de retaguarda.

Swap in -> Quando um processo sai da memória de retaguarda e vai para a memória principal.

ALOCAÇÃO DE MEMÓRIA CONTÍGUA E O PROBLEMA DA FRAGMENTAÇÃO DA MEMÓRIA

1. Alocação de Memória Contígua:

- Definição: Cada processo ocupa uma única seção contígua de memória.

Divisão:

- Partição do sistema operacional (SO).
- Partição dos processos do usuário.

2. Tipos:

1. Simples: Apenas um processo na memória, gerando subutilização.

2. Múltipla:

- Múltiplos processos simultâneos.
- Pode usar partições fixas (MFT) ou variáveis (MVT).

3. Particionamento Fixo (MFT):

- Memória dividida em partições de tamanhos fixos.

Desvantagens:

- Espaço não utilizado em partições maiores que o necessário (fragmentação interna).

4. Funcionamento:

- Processos são carregados em partições disponíveis.

Filas de acesso podem ser:

- Unica: Todos os processos competem pelas partições.
- Múltiplas: Cada partição possui sua própria fila.

5. Particionamento Variável (MVT):

- Memória particionada dinamicamente.
- Mantém tabelas de lacunas disponíveis.

Funcionamento:

- Lacunas podem ser divididas ou mescladas para acomodar processos.

Métodos de seleção:

- *First-fit*: Primeira lacuna suficiente.
- *Best-fit*: Menor lacuna suficiente.
- *Worst-fit*: Maior lacuna.

O Primeiro Apto (First-fit) e o Mais apto (Best-fit) são melhores que o Menos Apto (Worst-fit) em termos de tempo e uso de memória.

6. Problemas de Fragmentação:

1. Interna:

- Espaço ocioso dentro de partições fixas.

Exemplo: Processos menores que o tamanho das partições.

2. Externa:

- Lacunas não contíguas (espaço suficiente, mas fragmentado), mesmo com espaço suficiente disponível.

Soluções:

- Compactação: Reorganizar os processos para criar uma única grande lacuna.
- Relocação: Mover processos para consolidar memória livre.

ESQUEMAS DE PAGINAÇÃO E SEGMENTAÇÃO

Paginação:

Definição:

- Permite que o espaço de endereçamento físico de um processo não seja contíguo.
- Divide a memória física em quadros (frames) e a memória lógica em páginas (pages) de tamanho fixo.
- Evita fragmentação externa, mas pode gerar fragmentação interna.

Funcionamento:

- Cada processo possui uma tabela de páginas para mapear suas páginas em quadros da memória física.

Endereços lógicos gerados pela CPU são divididos em:

- Número de página (p): Índice na tabela de páginas.

- Deslocamento de página (d): Define a posição dentro da página.

Exemplo de Cálculo:

- *Fórmula: (número de página × tamanho do quadro) + deslocamento interno*.

Proteção:

- Cada entrada da tabela de páginas possui um bit válido/inválido para indicar se a página está dentro do espaço de endereçamento lógico do processo.

Segmentação:

Definição:

- Divide o espaço de endereçamento lógico em segmentos que correspondem a partes lógicas de um programa (ex.: código, pilha, heap).
- Evita fragmentação interna, mas pode gerar fragmentação externa.

Funcionamento:

- Cada segmento possui uma base (endereço inicial na memória física) e um limite (tamanho do segmento).
- O compilador cria segmentos que podem crescer ou diminuir independentemente.

Exemplo:

- Um segmento pode ocupar diferentes posições na memória física e ser acessado de forma independente.

Comparação entre Paginação e Segmentação:

Característica	**Paginação**	**Segmentação**
Tamanho	Fixos (quadros e páginas)	Variáveis (Segmentos)
Fragmentação	Interna	Externa
Organização	Foco em blocos de memória	Foco em elementos do programa
Tabela	Tabela de Páginas	Tabela de Segmentos

MEMÓRIA VIRTUAL

Definição:

- Separa memória lógica (vista pelos processos) da memória física (hardware disponível).
- Permite que programas usem mais memória do que o disponível fisicamente.
- Memória principal é tratada como uma cache para armazenamento secundário.

Paginação por Demanda:

Funcionamento:

- Divisão da memória lógica em páginas.
- As páginas são carregadas na memória física apenas quando acessadas.
- Páginas não acessadas permanecem no disco.

Bit "Válido - Inválido":

- Válido: Página está na memória.
- Inválido: Página está no disco ou é inválida.

- Erro de página (Page Fault):

- Ocorre quando um processo tenta acessar uma página inválida.

- Tratamento:

1. Verificar validade do acesso.
2. Carregar a página no quadro de memória.
3. Atualizar a tabela de páginas.
4. Reiniciar a instrução interrompida.

Substituição de Páginas:

Quando ocorre:

- Se não há quadros livres na memória.
- O sistema escolhe um quadro para substituir usando algoritmos específicos.

Critérios para avaliação de algoritmos:

- Taxa de erros de página: Número de erros por total de requisições.
- Algoritmos eficientes minimizam os erros de página.

Algoritmos de Substituição de Páginas:

1. **FIFO** (First In, First Out):

- Substitui a página mais antiga.
- Simples, mas pode gerar alta taxa de erros.

2. **Ótimo** (OPT):

- Substitui a página que não será usada pelo maior tempo.
- Ideal em teoria, mas difícil de implementar.

3. **LRU** (Least Recently Used):

- Remove a página menos usada recentemente.
- Usa o passado recente como predição do futuro.

4. **LFU** (Least Frequently Used):

- Substitui a página menos frequentemente usada.
- Pode ser combinado com FIFO ou LRU para desempate.

5. **MFU** (Most Frequently Used):

- Substitui a página mais frequentemente usada.
- Parte do princípio de que a página recém-carregada ainda será utilizada.

Atividade Improdutiva (Thrashing):

- Ocorre quando o sistema gasta mais tempo transferindo páginas entre memória principal e secundária do que executando processos.
- Problema comum em sistemas que implementam memória virtual.

ALOCAÇÂO DE QUADROS DA MEMÓRIA

Tamanho de Páginas:

Páginas pequenas:

- Transferências mais rápidas entre disco e memória.
- Suporta mais páginas na memória, mas exige grandes tabelas de páginas.
- Ideal para instruções.

Páginas grandes:

- Tabelas de páginas menores.
- Transferências mais lentas.
- Melhor para dados.

Prática:

- Tamanhos variam entre 64 bytes e 4 MB.
- 4 KB é o mais comum em sistemas gerais; 1 KB*em sistemas embarcados.

Alocação de Quadros:

Problema:

- Como dividir um número fixo de quadros de memória entre processos?

Regras básicas:

Sistemas monoprogramados:

- Apenas um processo usa todos os quadros.

Mínimo de quadros:

- Depende da arquitetura e do número de páginas referenciadas por instruções.

Métodos de Alocação:

1. Igual:

- Divide os quadros igualmente entre os processos.

Ex.: 93 quadros e 5 processos → Cada processo recebe 18 quadros (restante serve como buffer).

2. Proporcional:

- Aloca quadros de acordo com o tamanho do processo.

Ex.: 62 quadros e 2 processos (um com 10 páginas e outro com 127):

Processo 1: 4 quadros.

Processo 2: 57 quadros (restante como buffer).

Alocação Global vs. Local:

1. Local:

- Substituição de quadros ocorre apenas dentro do conjunto do processo.

2. Global:

- Considera todos os quadros disponíveis no sistema, mesmo os alocados para outros processos.