**1.Gerenciamento básico de memória**

**->Hierarquia de memórias**

Memória ideal: grande, rapida e não volátil. O que sairia muito caro para computadores comuns, daí a idéia de hierarquia como meio de compensar o desempenho.

Cache -> pequena, rapida e muito cara

Memória Principal -> média em todos os aspectos

Disco Magnético -> grande, velocidade baixa e barata

**->Relocação e Proteção**

Não se sabe onde o programa será carregado na memória, daí a necessidade de registradores base e limite. Estes são utilizados para dar a cada processo um espaço de endereçamento.

**->Mapeamento da memória**

Trata-se de gerenciar os espaços livres ou alocados na memória. Os meios mais comuns são Mapa de Bits e Lista encadeada.

**2.Troca de processos (Swapping)**

**->Swapping**

Consiste na “troca de processos”, isto ocorre quando é preciso carregar um processo e não há espaço na memória .

**3.Memória virtual**

**->Página/Memória virtual**

Quando um programa não cabe na memória é preciso quebra-lo em “páginas”, e o espaço onde ficam essas “páginas” é conhecido por “memória virtual”.

**4.Paginação (Paging)**

**->Paginação**

Paginação: troca de páginas entre memória virtual e memória real (fisica) , a relação é feita através de uma “tabela de páginas”.

**->Acelerando a paginação**

O mapeamento deve ser rapido, se o espaço de endereçamento for grande a tabela também será. Para tentar minimizar o problema:

Memória associativa ou TLB: Tabela das traduções de endereços mais recentes, funciona como uma “cache”.

Tablea de páginas Multi-Níveis: Onde uma tabela pode apontar para outra tabela reduzindo assim o tamanho das tabelas.

**->Substituição de páginas**

Substituiçao de páginas: Alocação de espaço para a página a ser trazida para a memória.

-Quando ocorre uma falta de página na memória qual deverá ser removida?

-A página modificada deve primeiro ser salva, senão é apenas sobreposta

-Melhor escolher uma página que não está sendo muito usada

**->Algoritmos de substituição de páginas**

Ótimo – impraticável, usado apenas para comparação

FIFO – Mantém uma lista encadeada de todas as páginas, na ocorrência de falta de página a mais antiga é removida. Desvantagem: página há mais tempo na memória pode ser usada com muita frequência.

NRU – Cada página tem os bits referenciada e modificada e são classificadas em:

Classe 0: não referenciada(0), não modificada(0)

Classe 1: não referenciada(0), modificada(1)

Classe 2: referenciada(1), não modificada(0)

Classe 3: referenciada(1), modificada(1)

remove página aleatoriamente da classe de ordem mais baixa que não esteja vazia.

SC – Lista de páginas em ordem FIFO, porém na falta de página a mais antiga é tratada como mais recentemente carregada.

LRU – Lista encadeada de páginas, onde a mais recentemente usada no inicio da lista e mantém um contador(tempo) em cada entrada da tabela. Retira página com contador de menor valor.

WS – Varre todas as páginas examinando o bit R(referenciada)

Se (R = 1), faça instante da última referência igual ao tempo virtual atual

Se (R = 0 e idade > t), remova esta página

Clock – Quando ocorre uma falta de página, a página apontada é examinada. Se R = 0 retira a página. Se R = 1: faz R = 0 e avança o ponteiro.

WSClock – Igual ao WS, porém quando vai examinar as paginas não começa do inicio, começa da pagina apontada.

NFU – Lista encadeada de páginas cada uma com seu contador. Onde a cada referencia a página incrementa o seu contador. Remove paginas com o menor contador.

Envelhecimento – A cada interrupção de relogio o algoritmo guarda o bit R de todas as paginas e desloca a direita. Remove a pagina com o menor valor.

**->Moldura de página**

É de se esperar que quanto mais molduras maior o sucesso, ou seja, menor a chance de falta de página.

Anomalia de Belady é quando ocorre mais faltas de páginas para mais molduras.

**->Controle de carga**

Para evitar paginação excessiva (trashing), isto ocorre quando alguns processos precisam de mais memória mas nenhum outro processo pode ceder.

Solução: reduzir o número de processos que competem pela memória, levar alguns deles para disco (swap) e liberar a memória a eles alocada.

**->Tamanho de página**

Tamanho de página pequeno:

Vantagem: menos fragmentação interna

Desvantagem: programas precisam de mais páginas, tabelas de páginas maiores.

**->Envolvimento do SO com paginação**

1.Criação de processo

-determina tamanho do programa e cria tabela de página

2.Execução de processo

-Inicia MMU para novos processos

3.Ocorrência de falta de página

-determina endereço virtual que causou a falta

-descarta, se necessário, página antiga

-carrega página requisitada para a memória (swap)

4.Terminação de processo

-Libera tabela de páginas, páginas e espaço em disco que elas ocupam

**->Lidando com uma falta de página**

**->Fixação de páginas na memória >>>> Procurar melhor no livro <<<<<**

**5.Segmentação**

**->Segmentação**

Problema: espaço de endereçamento unidimensional com tabelas crescentes, uma tabela pode atingir a outra.

Segmentação: permite que cada tabela cresça ou encolha, independentemente.

“Para permitir que programas e dados sejam quebrados em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção.”

**->Segmentação com paginação**

Descritores de segmentos apontam para tabelas de páginas