Sistemas Operacionais

Gerência de Memória

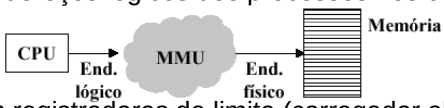
Gerência de Memória

- Multiprogramação:
  - diversos processos através da divisão do tempo de processador
- Rápido chaveamento:
  - necessidade de processos em memória.
- Gerência de memória:
  - prover mecanismos necessários para que diversos processos compartilhem a memória de forma segura e eficiente.
- Espaço de Endereçamento Lógico:
  - espaços formados por todos os endereços lógicos que o processo pode acessar;
- Espaço de Endereçamento Físico:
  - Implementado pela eletrônica do computador, corresponde a todos os end. Aceitos pelos CI de memória.

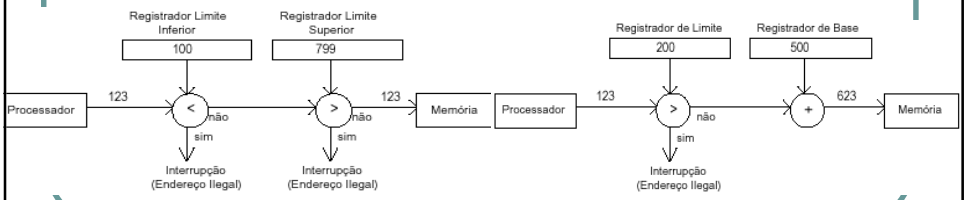
2

Unidade de Gerenciamento de Memória

- Provê mecanismos básicos que serão usados pelo SO para gerenciar a memória.
- Mapear os endereços lógicos dos processos nos correspondentes end. físicos.



- Proteção com registradores de limite (carregador absoluto); e
- Proteção com mecanismos de base e limite (carregador relocador)



3

Grau de Multiprogramação

- Consideração simplista
  - Se cada processo gasta 20% de seu tempo de CPU → 5 processos ocuparão o processador o tempo todo.
  - Modelo otimista e irreal → nunca dois ou mais processos aguardarão E/S ao mesmo tempo.
- Consideração Probabilística
  - Um processo gasta uma fração *p* de seu tempo fazendo E/S.
  - Com *n* processos na memória, a probabilidade de todos aguardarem E/S ao mesmo tempo (processador ocioso) é *pn*: UP = 1 - *pn*
  - Sabendo que os processos gastam 80% de seu tempo fazendo E/S...
    - 1 processo, UP = 1 - 0,8 → 20% de utilização (GO = 80%)
    - 2 processos, UP = 1 - 0,64 → 36% de utilização (GO=64%)
    - 10 processos, UP = 1 - 0,11 → ≅90% de utilização (GO≅10%)
    - Traçar o gráfico com o Grau de Multiprogramação (0-100%) até 10 processos com 20%, 50% e 80% de E/S.

4

Modelo para previsão aproximada

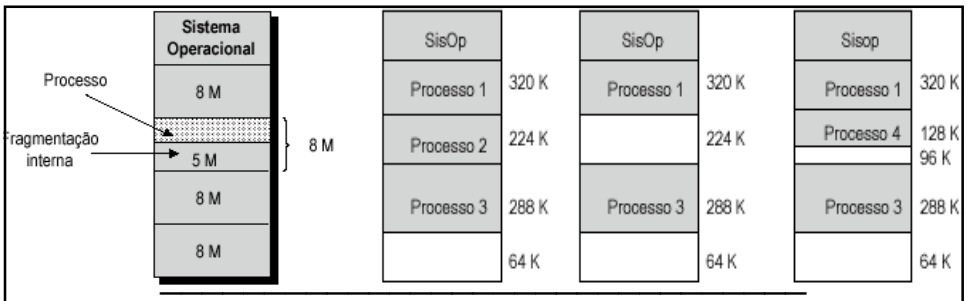
- Ex.: 1 MB de memória, SO (200 KB) e programa de usuários (200 KB)
  - 1 MB: 4 programas → UP ≅ 59%
  - adição de +1 MB: 9 programas → UP ≅ 87% (incr.de 45%)
  - adição de +1 MB: 14 programas → UP ≅ 96% (incr. de 10%)
  - Conclusão: considerando o sistema inicial, um 2.° MB é um ótimo investimento, um 3.° MB não.
- Mecanismos Básicos de Gerência de Memória
- Partições Fixas (alocação particionada fixa)
  - Forma mais simples de gerência de memória.
  - Divisão da memória em duas partes: uso do SO e uso dos proc. usuários.
  - Parte dedicada aos processos usuários é subdividida em partições fixas (iguais ou diferentes). Processo é colocado na menor partição que tenha tamanho suficiente para armazená-lo.

5

Mecanismos Básicos de Gerência de Memória

- Todo o espaço não utilizado pelo processo na partição é perdido.
- Múltiplas Filas de Entrada X Fila de Entrada Única.
- Desvantagens: uso ineficiente da memória; e fragmentação interna e externa.
- Partições Variáveis (alocação particionada dinâmica)
  - Tamanho das partições é ajustado dinamicamente às necessidades exatas do processo.
  - Fragmentação?
  - Formação de lacunas (buracos) na memória
  - Uma lacuna de tamanho igual ou maior será usada.
  - O espaço que eventualmente sobrar será transformado em uma nova lacuna, porém menor que a original.
  - Lacunas adjacentes são unificadas.
  - Compactação de memória: recurso geralmente não utilizado, pois gasta muito tempo de processador.

6



- **Estratégias de Gerenciamento**
- Gerência com Mapeamento de Bits
  - Divisão da memória em unidades de alocação
  - Tamanho variado determina relação tam. mapa X tam. unidade.
  - Necessidade de buscar quantas unidades consecutivas forem necessárias.
  - Percorrer todo o mapa → Busca lenta → pouco uso na prática.

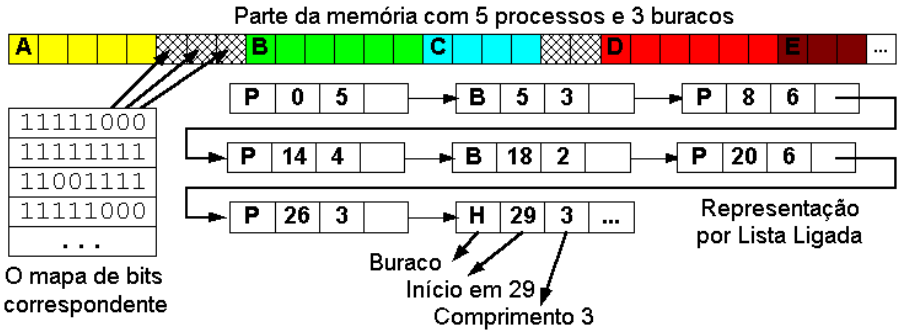
Gerência com Listas Ligadas

- Manutenção de uma Lista Ligada dos segmentos Livres e Ocupados.
  - Segmento: um processo ou uma lacuna (buraco)
- Rápida atualização da Lista quando um processo terminar ou for removido.
- Tal lista é percorrida quando um novo processo for executar.
  - Quatro formas básicas de percorrer a lista de lacunas:
  - **First Fit**: utiliza a primeira lacuna que encontrar com tamanho suficiente.
  - **Best Fit**: utiliza a lacuna que resultar a menor sobra.
  - **Worst Fit**: utiliza a lacuna que resultar a maior sobra.
  - **Circular/Next Fit**: inicia a busca a partir da última sobra.
  - **Considerações sobre desempenho e desperdício:**
  - Lista de processos e buracos separadas e Ordenação das Lacunas.

8

Sistema Buddy

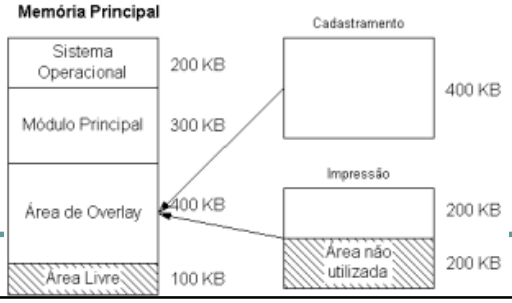
- Divisão da lacuna em espaços na base 2:
  - 1, 2, 4, 8, ..., até o limite da memória.
- Até encontrar uma lacuna suficiente.
  - Eficiente quanto à velocidade e Ineficiente quanto à fragmentação.



9

Conceitos adicionais

- Regra dos 50%: em média, para *n* processos, existirão *n/2* buracos.
- Regra da mem. não utilizada: fração *f* da memória com buracos.
- Swap:
  - movimento de processos entre memória e disco e vice-versa.
  - Espaço garantido na ordem de blocos de disco (*cluster*).
- Problema histórico de prog. serem maiores que a memória.
  - 1.ª iniciativa: *overlays*→divisão do programa extremamente onerosa.

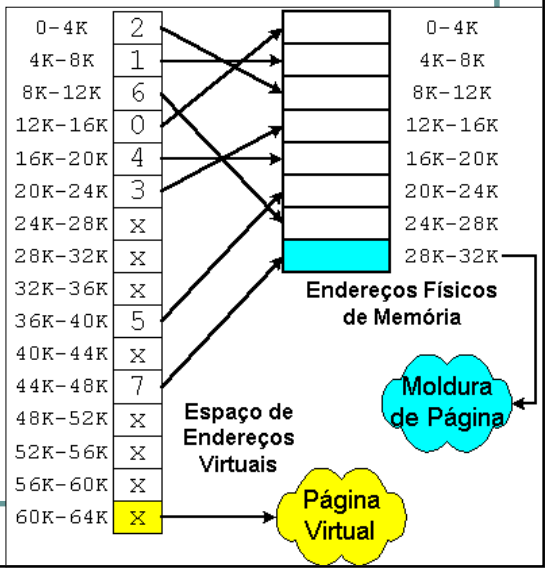


10

Memória Virtual

- O programa pode exceder os limites da memória.
- SO responsável por manter na memória partes dos programa efetivamente em uso, mantendo o resto em disco.

- **Técnica da Paginação**
  - Mapeamento de endereços virtuais (*pages*) em endereços físicos (*page frames*) ➔ MMU
  - Cópia em disco dos endereços mapeáveis.



Paginação (cont.)

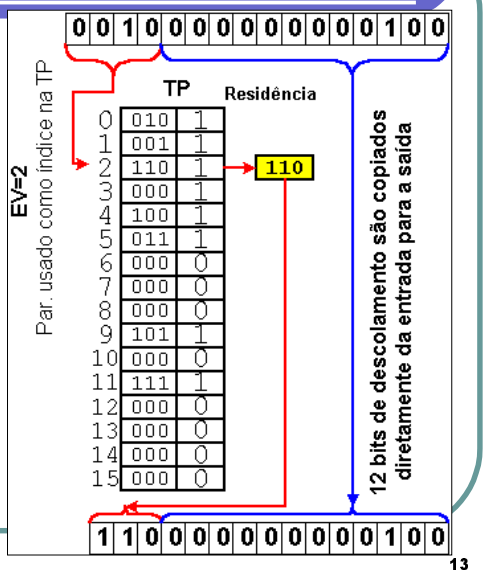
- Ex. 1: programa acessa o endereço 0; a MMU conclui que este end. virtual cai na página 0 (0 ~ 4095) que está mapeada na moldura 2 (8192 ~12287).
- Ex. 2: *move REG,* ***20550*** → (20KB = 20480) - pág. 5, frame 3: 12KB = 12288 → **12358**.

- Bit de presença/ausência: informa se página pode ser mapeada:
  - Processo acessa end. 32780 → pág. não mapeável → ***page fault*** (falta de página)
  - SO transfere a moldura menos acessada para o disco;
  - Busca a página referenciada e copia na moldura liberada e reexecuta a instrução → Substituir moldura 1; carregar página 8 no end. 4 KB e:
    - a) marcar página virtual como não mapeável; e
    - b) substituir a entrada correspondente à página 8 por 1.

12

Funcionamento da MMU

- **Ex.: 16 bits**
  - 4 bits de moldura = 2^4
    - 16 *page frames*
  - 12 bits de deslocam. = 2^{12}
    - 4096 endereçamentos
  - End. 8196: 16 bits (página) → MMU → 15 bits (moldura)
  - Função (arg:EV) ret. (ER)
- **Ponderações:**
  - 1 - a tabela pode ficar muito grande; e
  - 2 - o mapeamento deve ser rápido.



13

Tabelas de Página Multinível

- Evitar manter as tabelas de páginas na memória durante todo o tempo.
  - End. Virtual de 32 bits particionado em dois campos de 10 bits (TP1 e TP2) e um terceiro campo de 12 bits (deslocamento)
  - TP1: 1024 entradas (nível superior); TP2: 1024 entradas (nível inferior) para 4 KB.
- Exemplo:
  - 4206596 (origem = 4194304, sendo 12292 depois)
  - TP1 = 1 (0: 0MB ~ 4MB; 1: 4MB ~ 8MB) → 4194304
  - TP2 = 3 (0: 0~4; 1: 4~8; 2: 8~12; 3: 12KB ~16KB) → 12288
  - deslocamento: 4 bytes.

14

Memória Associativa

- Maioria dos programas tende a referenciar um pequeno nr. de páginas.
- Memória Associativa:
  - Dispositivo de hardware que traduz EV em ER sem usar a memória principal.

Entrada válida	Página virtual	Modificado	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

15

Memória Associativa

- EV → MMU → acesso à memória associativa (rápido)
  - Se encontra → pag. Real (hit ratio)
  - Se falhar → acesso à tabela de páginas e traz para a MA. (miss ratio)
- MA → funciona como um cache para a tabela de páginas.
- **ALGORITMOS PARA TROCA DE PÁGINAS**
  - **Not Recently-Used Page Replacement (NRU)**
  - Troca de página não usada recentemente
  - Normalmente há 2 bits → R e M
  - fácil de se entender, eficiente para implementar e fornece bom desempenho
  - **First-in First-out Page Replacement (FIFO)**
  - Simples mas pode levar a não eficiência pois uma página que está em uso constante pode ser retirada.

16

Algoritmos de Substituição de Páginas

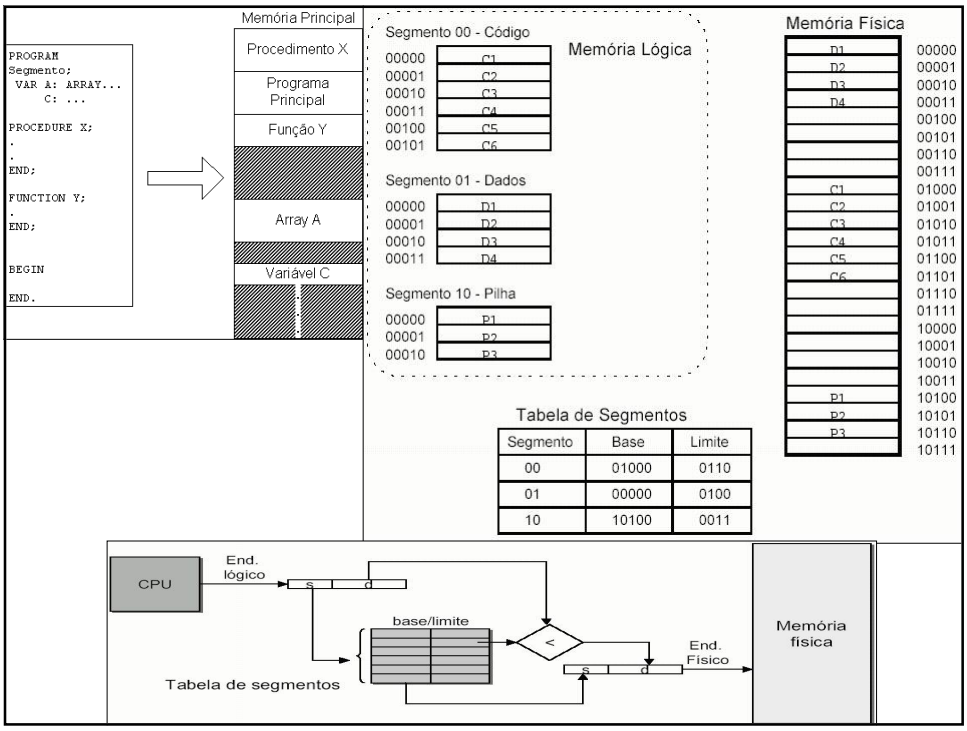
- **Least Recently Used Page Replacement (LRU)**
- seleciona a página menos utilizada recentemente, ou seja, a página que está a mais tempo sem ser referenciada..
- **Segunda Oportunidade**
- FIFO + R bit
- Página mais velha é a candidata em potencial; se o bit R é 0 → fim, caso contrário, R := 0 e se dá uma nova chance à página, colocando-a no final da lista
- **Least-Frequently-Used (LFU)**
- nesse esquema, a página menos freqüentemente utilizada será a escolhida. Existe um contador do número de referência feitas às páginas.

17

Segmentação

- Programadores e compiladores não consideram a memória lógica como páginas, mas **segmentos**. Segmentos são entidades lógicas definidas pelo programador.
- Um programa pode ser dividido em quatro segmentos: código, dados estáticos, dados dinâmicos e pilha de execução.
- Quando a gerência de memória suporta segmentação, a memória passa a ser endereçada por um número de segmento e um deslocamento.
- Cada segmento é copiado para a memória física, e uma **tabela de segmentos** é construída → para cada segmento, há um endereço na memória física, onde ele foi colado e qual seu tamanho. A cada acesso, necessidade de verificar a se ele é válido. **Vantagens:** compartilhamento e proteção.

18



Segmentação X Paginação

- Paginação **(P)**
  - Obter espaço de endereçamento maior que a memória física;
- Segmentação **(S)**
  - Auxiliar na divisão lógica de programas para facilitar o compartilhamento e proteção.
- Comparações
  - Facilidade de compartilhamento

  (S) Única entrada. **(P)** Várias entradas
  - Facilidade de ajuste de tamanho

  (S) Definição de novo limite. **(P)** Atualização da Tabela.

20

Trashing

- Excessiva transferência entre memória principal e memória secundária
  - Nível do processo:
    - ↑ *miss rate* (tam. do *working set* pequeno) → aumentar o WS;
    - Não respeitar a localidade → aplicação deve ser reescrita.
  - Nível do sistema:
    - Mais processos que memória disponível;
    - O sistema tenta garantir que todos os processos sejam executados:
      - **1ª iniciativa: diminuir o WS; 2ª iniciativa: swapping.**
  - **Swapping:** selecionar processo que deixará a memória principal.
    - **Prioridade (menor) ou estado do processo (espera).**

21