

Sistemas Operativos: Gestão de Memória

Pedro F. Souto (pfs@fe.up.pt)

April 28, 2012

Sumário

Conceitos e Técnicas Básicas

Swapping

Gestão da Memória Física

Sumário

Conceitos e Técnicas Básicas

Swapping

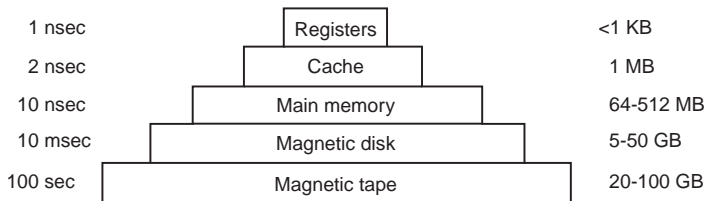
Gestão da Memória Física

Hierarquia de Memória

- ▶ Praticamente qualquer programador gostaria de dispôr de memória:
 - ▶ em grande quantidade;
 - ▶ rápida;
 - ▶ não volátil.
- ▶ SO e os compiladores exploram a hierarquia de memória para satisfazer estes “desejos”:

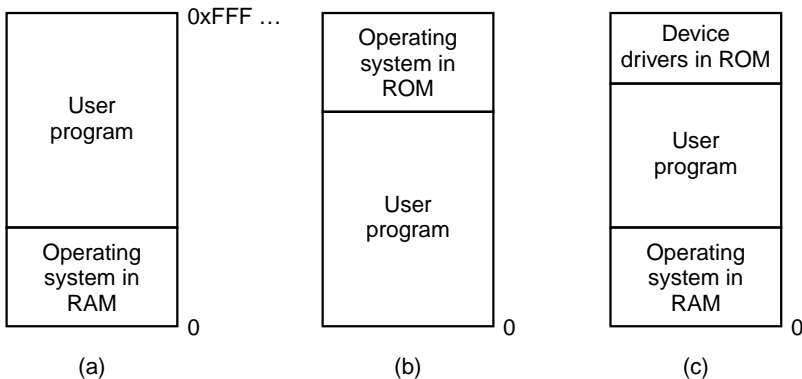
Typical access time

Typical capacity



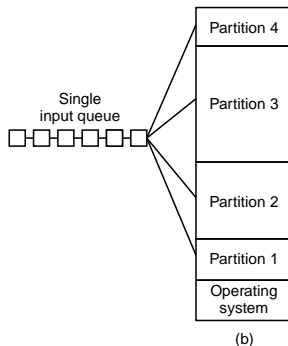
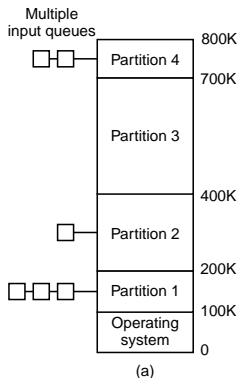
Gestão de Memória em Monoprogramação

- ▶ Executa apenas um processo de cada vez.



- b) Disposição típica em *palmtops* (~ sistemas embebidos).
- c) Disposição em MS-DOS (A parte em ROM é conhecida por BIOS.)

Multiprogramação e Partições Fixas



- A memória é dividida em partições (possivelmente \neq s).
- a) o SO atribui a partição de menor tamanho capaz de conter o processo;
- b) quando uma partição fica disponível, o SO atribui-a ao processo capaz de a usar mais à frente na fila.

Problemas Introduzidos por Multiprogramação

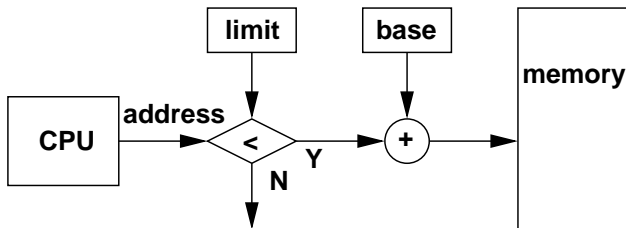
Recolocação: a posição do código pode variar entre execuções. Duas soluções:

SW O *loader* pode alterar os endereços absolutos de acordo com a posição onde o código é carregado.

HW Usar um *base register* a inicializar com o endereço inicial da partição atribuída ao processo.

Protecção: impedir que um processo acesse a código ou a dados de outros processos ou do SO:

- ▶ Usar um *limit register* além do *base register*.



Sumário

Conceitos e Técnicas Básicas

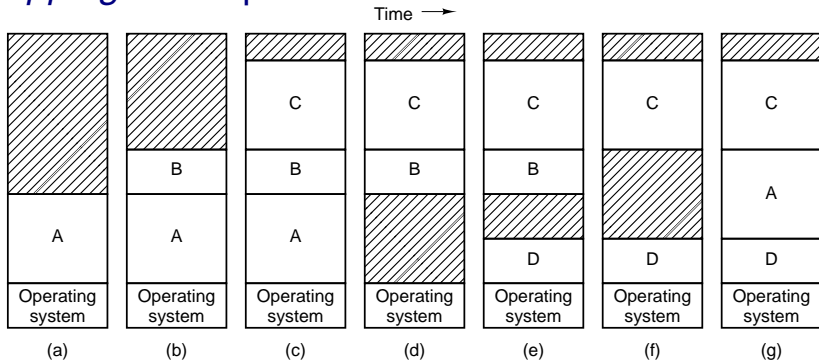
Swapping

Gestão da Memória Física

Swapping: Ideia

- ▶ Com partições fixas um processo é carregado numa partição quando chega à cabeça da fila correspondente, e depois fica em memória até terminar
- ▶ A memória física pode tornar-se insuficiente para executar todos os processos.
- ▶ Uma solução possível é o recurso a *swapping*:
 - ▶ Passar um processo para o disco (*swap out*) e, posteriormente
 - ▶ Transferi-lo de novo do disco para a memória (*swap in*)

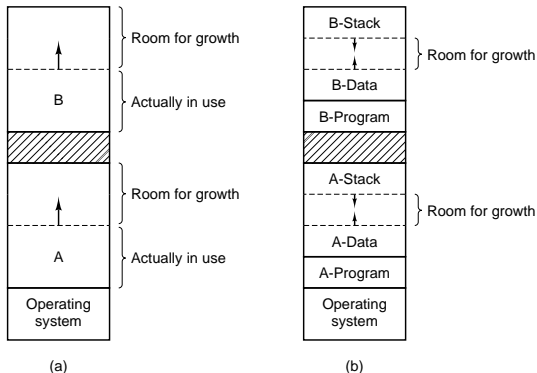
Swapping: Exemplo



- ▶ Com *swapping* o número, localização e tamanho das partições varia dinamicamente
 - + menor fragmentação interna – melhor utilização;
 - torna a gestão de memória mais complexa.
- ▶ Contudo *swapping* pode conduzir à fragmentação da memória:
 - ▶ pode ser atenuada usando compactação;
 - ▶ mas compactação consome bastante CPU

Swapping: Tamanho das Regiões a Alocar

- Qual o tamanho da região de memória a alocar quando um processo é criado ou trazido para memória (*swapped in*)?



- Se o processo crescer em demasia e não houver mais memória para alocar, pode-se passá-lo para o disco (*swap-out*)
 - Se disco estiver cheio, pode-se bloquear processo ... mas nesse caso o processo ocupará memória

Sumário

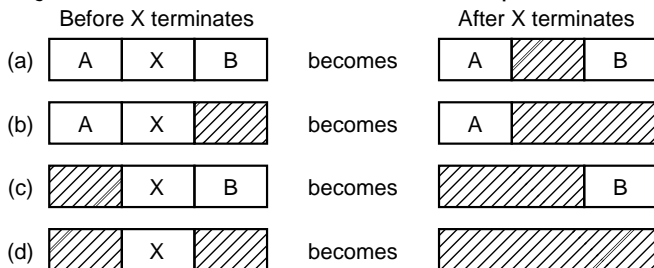
Conceitos e Técnicas Básicas

Swapping

Gestão da Memória Física

Algoritmos de Alocação Dinâmica de Memória

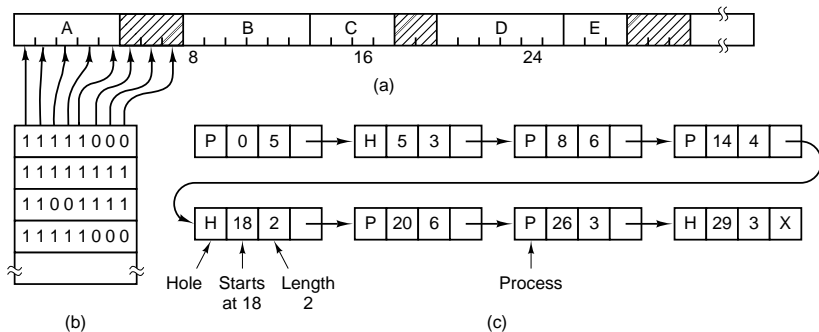
- ▶ O SO tem que gerir o espaço de memória físico
- ▶ A libertação de memória é relativamente simples:



- ▶ A alocação é mais interessante:
 - first fit*: simples e eficiente;
 - next fit*: começa varrimento onde parou da última vez - de facto pior do que *first fit*;
 - best fit*: tenta evitar fragmentação, mas ...
 - worst fit*: tenta evitar fragmentos muito pequenos, mas ...

Alocação Dinâmica de Memória

- ▶ Tipicamente decompõe-se a memória física em unidades de alocação, cujo tamanho é da ordem dos KB
- ▶ Normalmente usa-se uma de 2 estruturas de dados
 1. bitmaps;
 2. listas ligadas.



Bitmaps vs. Listas

- + Bitmaps ocupam um espaço de memória fixo que depende do tamanho da unidade de alocação
 - ▶ blocos pequenos exigem mais memória para os bitmaps;
 - ▶ blocos grandes podem conduzir a fragmentação interna;
- ? Depende (ver a seguir)
 - Pesquisa de espaço disponível para alocação pode ser lenta

Alocação Dinâmica de Memória com Listas

1. Manter listas separadas para a memória ocupada e a memória livre
 - + facilita alocação;
 - + pode-se usar a própria memória livre para implementar os elementos da lista de memória livre;
 - penaliza a liberação da memória.
2. Manter listas de memória livre, uma para cada um dos tamanhos de blocos mais comuns (*quick fit*)
 - problemas análogos aos do uso de listas separadas