

Aula 12

Sistemas Operacionais I

Gerenciamento de Memória - Parte 1

Prof. Julio Cezar Estrella

jcezar@icmc.usp.br

Material adaptado de

Sarita Mazzini Bruschi

baseados no livro Sistemas Operacionais Modernos de A. Tanenbaum

Gerenciamento de Memória

- Hierarquia de Memória
- Alocação particionada estática e dinâmica
- Gerenciamento dos espaços
- Swapping
- Memória virtual
 - Paginação e Segmentação

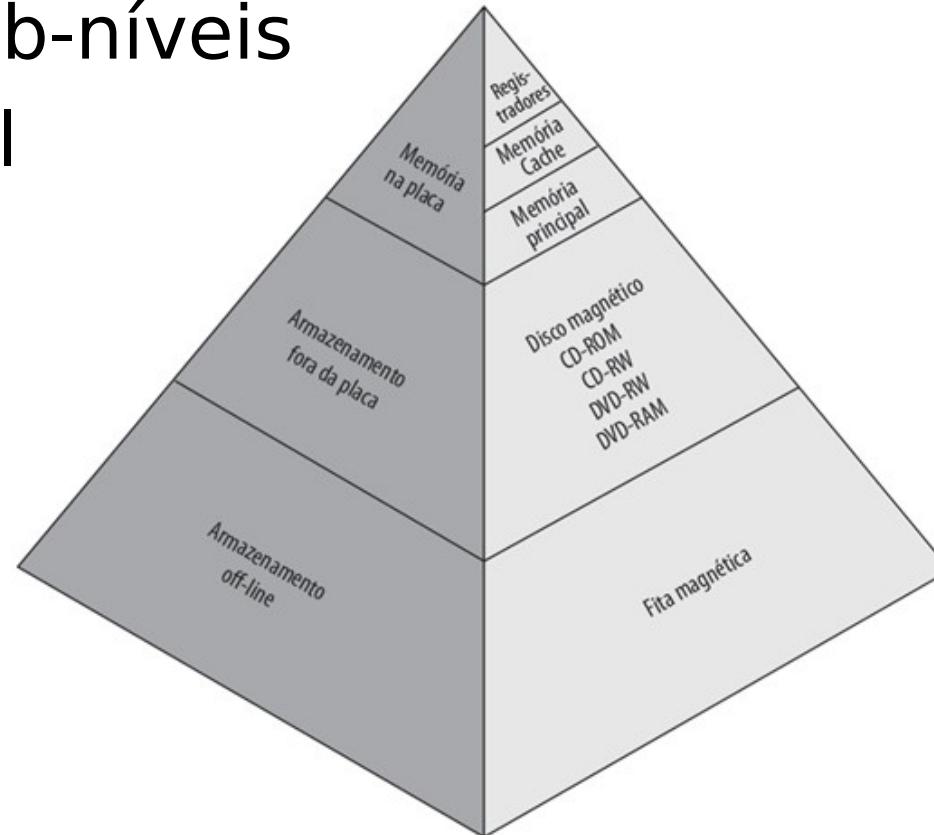
Gerenciamento de Memória

- Memória - recurso muito importante;
- Tendência atual do software
 - Lei de *Parkinson*: “Os programas se expandem para preencher a memória disponível para eles” (adaptação);
- Requisitos:
 - Muito grande;
 - Rápida;
 - Não volátil;
 - Baixo custo.

Gerenciamento de Memória

Hierarquia de Memória

- Cache – vários sub-níveis
- Memória Principal
- Disco



Gerenciamento de Memória

Hierarquia de Memória

- Cache
 - Pequena quantidade – k/M bytes
 - Alto custo por byte
 - Muito rápida
 - Volátil
- Memória Principal
 - Quantidade intermediária – M/G bytes
 - Custo médio por byte
 - Velocidade média
 - Volátil
- Disco
 - Grande quantidade – G/T bytes
 - Baixo custo por byte
 - Lenta
 - Não volátil

Gerenciamento de Memória

Hierarquia de Memória

- Para cada tipo de memória, deve-se:
 - Gerenciar os espaços livres/ocupados
 - Alocar processos/dados na memória
 - Localizar dado
- Entre os níveis de memória, deve-se:
 - Gerenciar as trocas

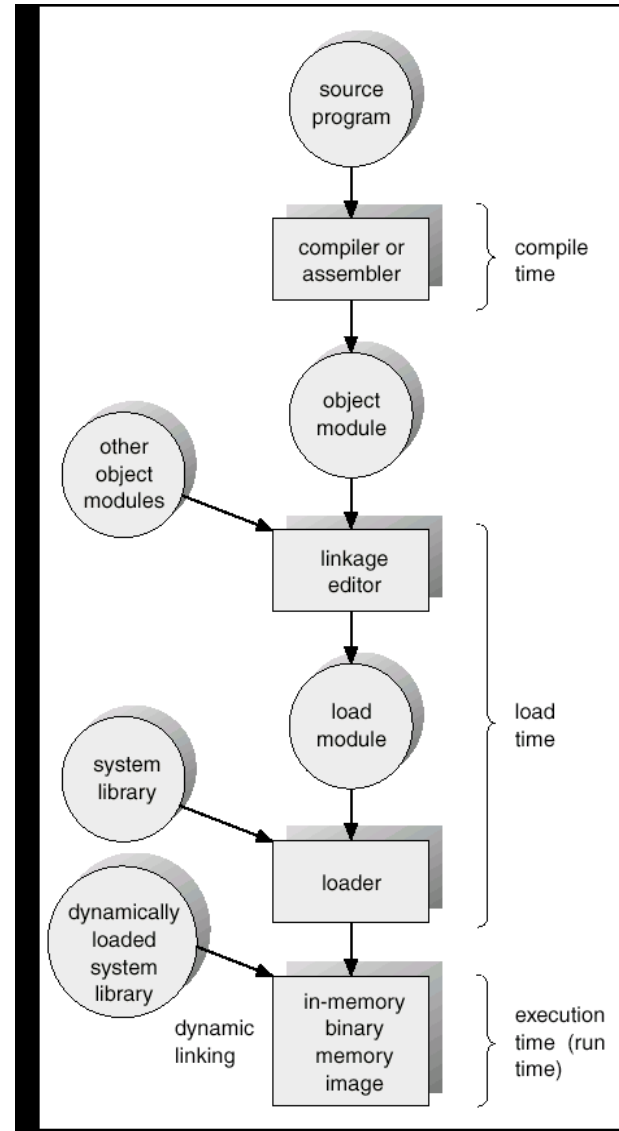
Gerenciamento de Memória

- O Gerenciador de Memória é responsável por:
 - Alocar e liberar espaços na memória para os processos em execução;
 - Gerenciar chaveamento entre os níveis de memória: principal e disco;

Vinculação de endereços

- Os programas em linguagem de alto nível normalmente não referenciam endereços específicos de memórias
- A associação com um endereço fixo pode ser feita em diversos momentos, dependendo da linguagem, do SO e do HW
 - Tempo de compilação
 - Tempo de carga
 - Tempo de execução

Vinculação de endereços

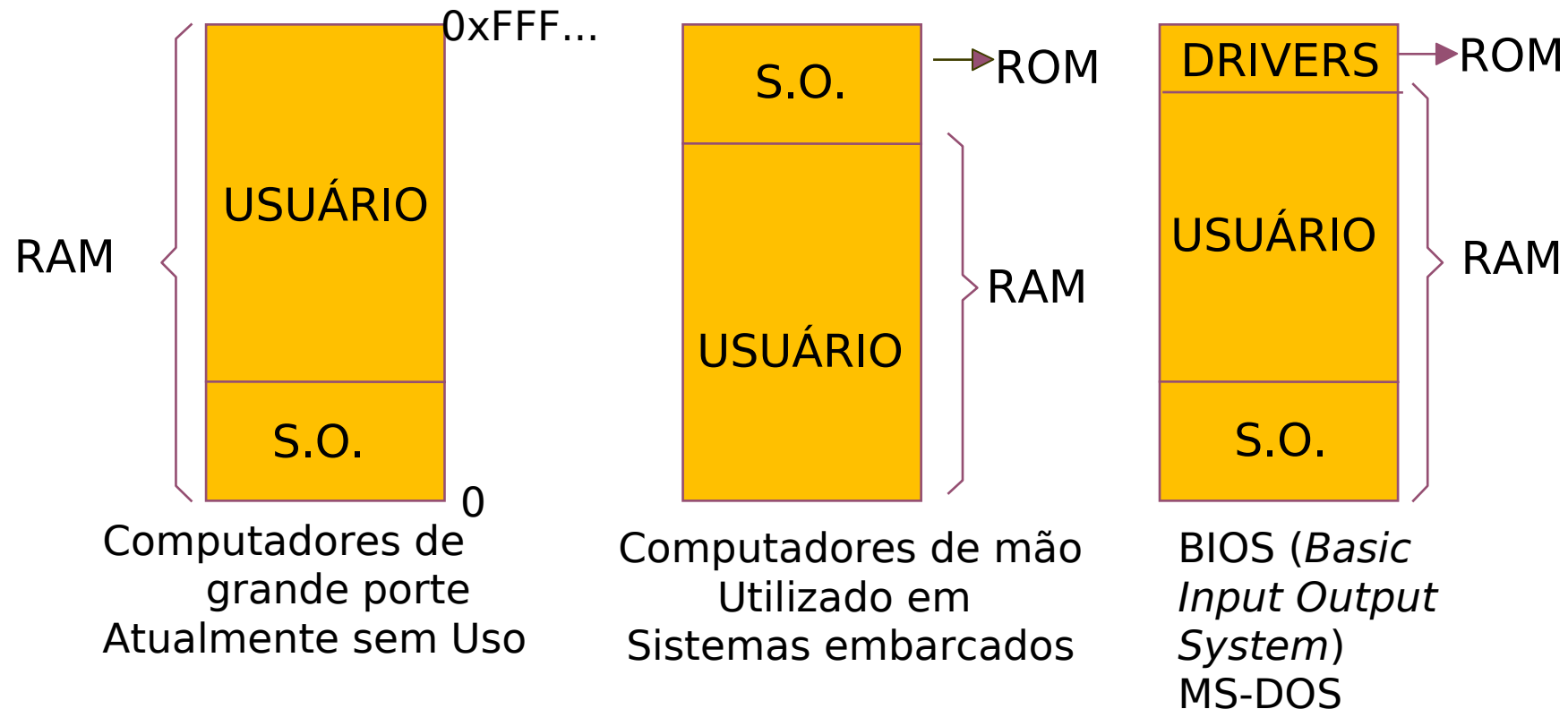


Gerenciamento de Memória

- Tipos básicos de gerenciamento:
 - Com troca (swap) ou com memória virtual:
 - Processos são movidos entre a memória principal e o disco;
 - Artifício usado para resolver o problema da falta de memória;
 - Se existe MP suficiente não há necessidade de se ter troca;
 - Sem troca:
 - Não há chaveamento;
 - Apenas um processo na memória;
 - Monoprogramação

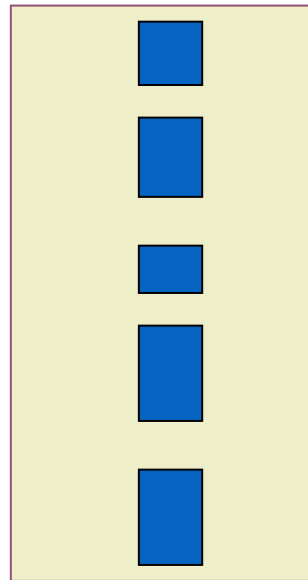
Gerenciamento de Memória

- Monoprogramação:
 - Sem troca: gerenciamento mais simples;
- Apenas um processo na memória



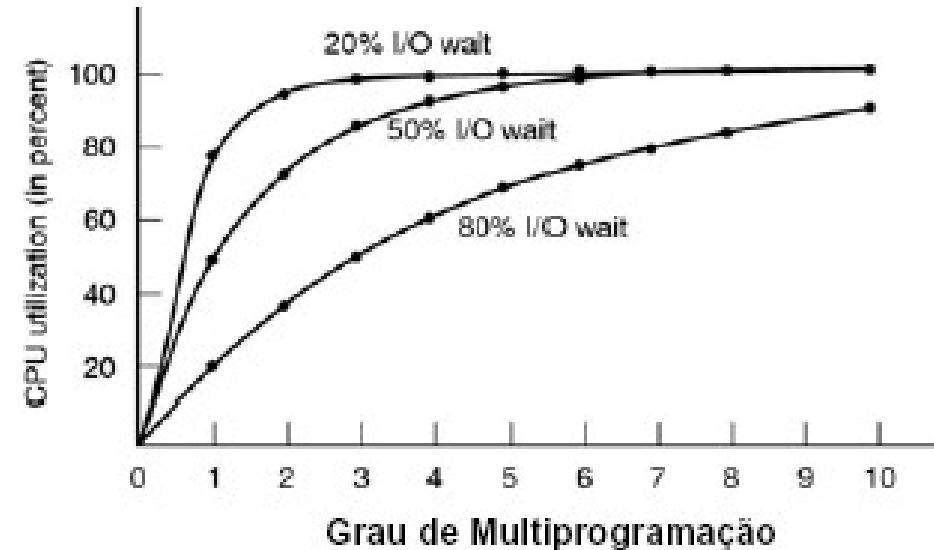
Gerenciamento de Memória

- Modelo de Multiprogramação:
 - Múltiplos processos sendo executados;
 - Eficiência da CPU;



■ Processo

Memória Principal - RAM



**Necessidade de
Particionamento da Memória
Principal**

Gerenciamento de Memória

- **Multiprogramação:**

- Vários processos na memória:
- Proteção: como proteger os processos uns dos outros?
- Necessidade de realocação: processo pode estar em diferentes posições da memória

- Todas as soluções envolvem equipar a CPU com um hardware especial:

- **MMU (*Memory Management Unit*);**

Gerenciamento de Memória

- **Realocação:**

- Quando um programa é montado (*link*), i.e. programa principal + rotinas do usuário + rotinas da biblioteca → executável, o montador (*linker*) deve saber **em que endereço** o programa irá iniciar na memória;
- Nesse caso, para que o montador não escreva em um local indevido (por exemplo na área do SO), é preciso realocação:

#100 + Δ → que depende da partição!!!

- **Proteção:**

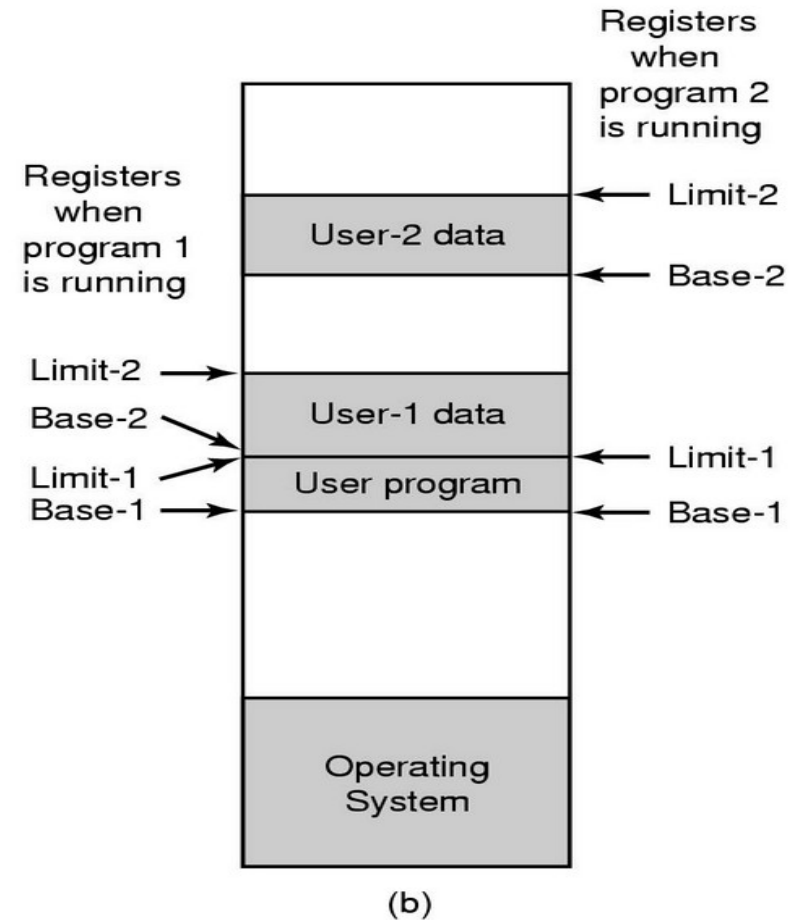
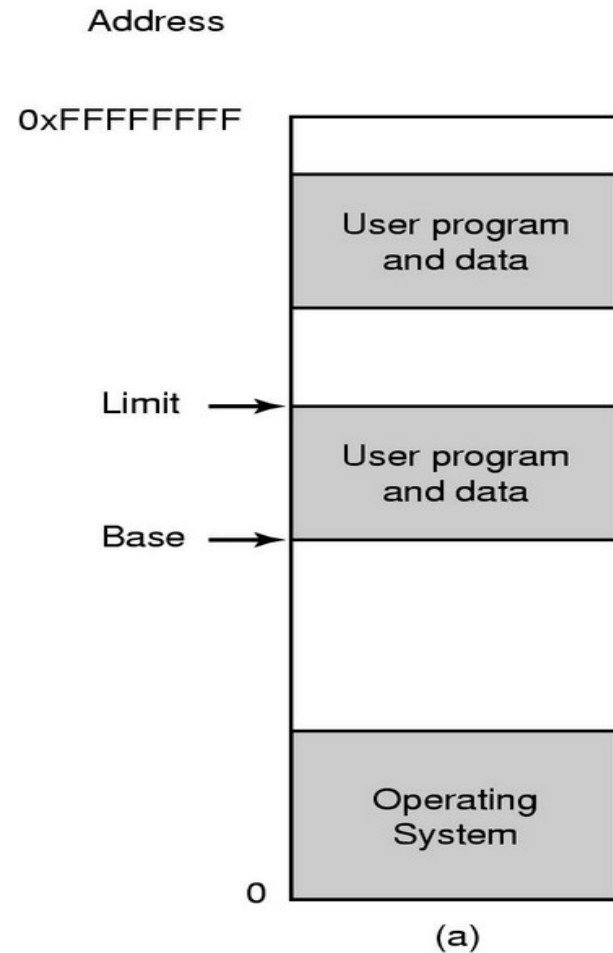
- Com várias partições e programas ocupando diferentes espaços da memória é possível acontecer um acesso indevido;

Gerenciamento de Memória

- **Solução para ambos os problemas do slide anterior:**
 - 2 registradores → base e limite
 - Quando um processo é escalonado o **registrador-base** é carregado com o endereço de início da partição e o **registrador-limite** com o tamanho da partição;
 - O registrador-base torna impossível a um processo um acesso a qualquer parte de memória abaixo de si mesmo;
 - Automaticamente, a **MMU** adiciona o conteúdo do **registrador-base** a cada endereço de memória gerado;
 - Endereços são comparados com o **registrador-limite** para prevenir acessos indevidos;

Gerenciamento de Memória

Registradores base e limite



Gerenciamento de Memória

Partições

- Particionamento da memória pode ser realizado de duas maneiras:
 - Partições fixas (alocação estática);
 - Partições variáveis (alocação dinâmica);
- **Partições Fixas:**
 - Tamanho e número de partições são fixos (estáticos);
 - Não é atrativo, porque partições fixas tendem a desperdiçar memória (Qualquer espaço não utilizado é literalmente perdido)
 - Mais simples;

Gerenciamento de Memória

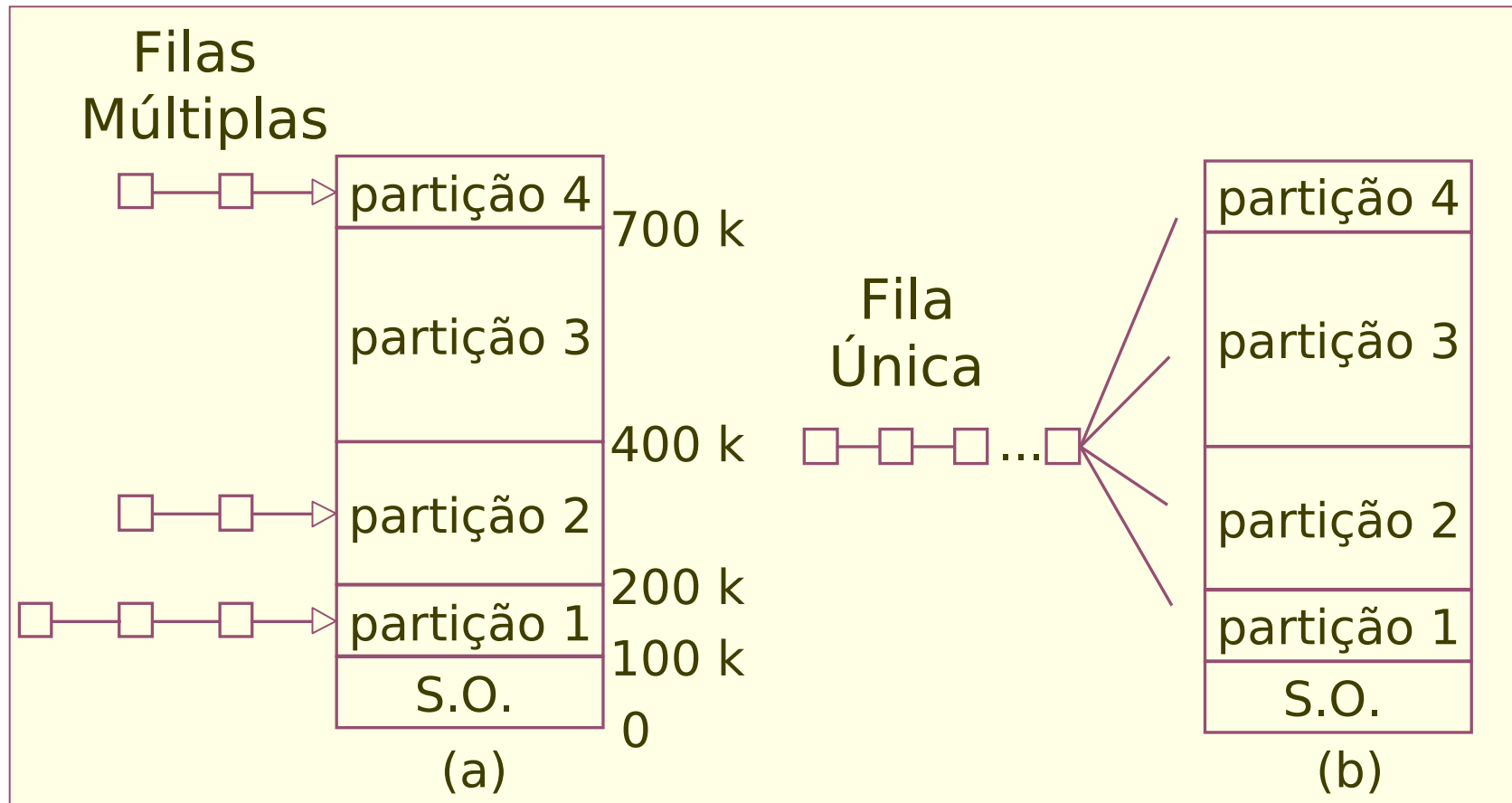
Partições Fixas

- Filas múltiplas:
 - Problema: filas não balanceadas;
- Fila única:
 - Melhor utilização da memória, pois procura o melhor processo para a partição considerada;
 - Diferentes algoritmos podem ser considerados para alocar os processos;

Gerenciamento de Memória

Partições Fixas

- Divisão da Memória em Partições Fixas:



Gerenciamento de Memória

Partições Fixas

- Partições Fixas: problemas com fragmentação:
 - **Interna**: desperdício dentro da área alocada para um processo;
 - Ex.: processo de tamanho 40K ocupando uma partição de 50k;
 - **Externa**: desperdício fora da área alocada para um processo;
 - Duas partições livres: PL1 com 25k e PL2 com 100k, e um processo de tamanho 110K para ser executado;
 - Livre: 125K, mas o processo não pode ser executado;

Gerenciamento de Memória

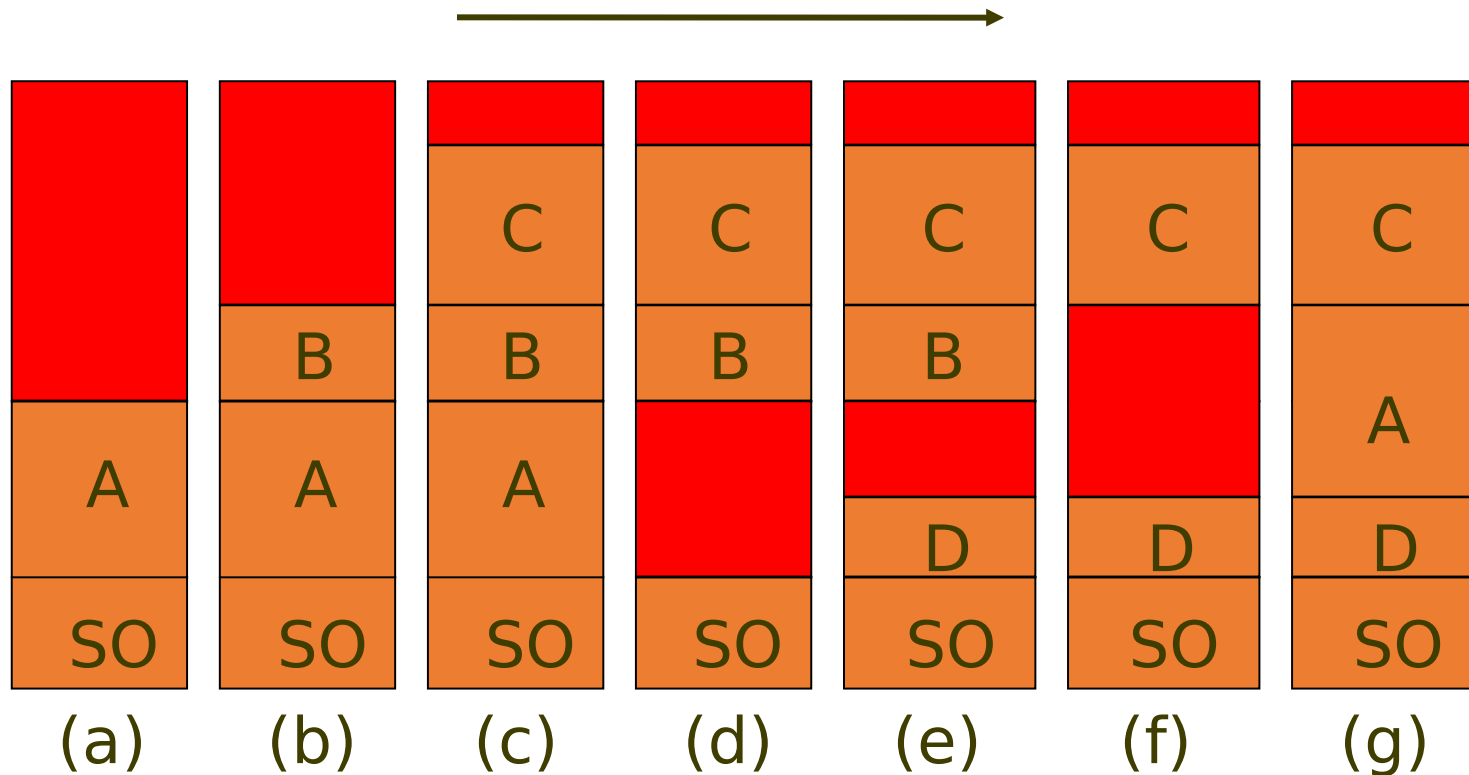
Partições Variáveis

- **Partições Variáveis:**

- Tamanho e número de partições variam;
- Otimiza a utilização da memória, mas complica a alocação e liberação da memória;
- Partições são alocadas dinamicamente;
- SO mantém na memória uma lista com os espaços livres;
- Menor fragmentação interna e grande fragmentação externa;
 - Solução: Compactação;

Gerenciamento de Memória

Partições Variáveis



■ Memória livre

Gerenciamento de Memória

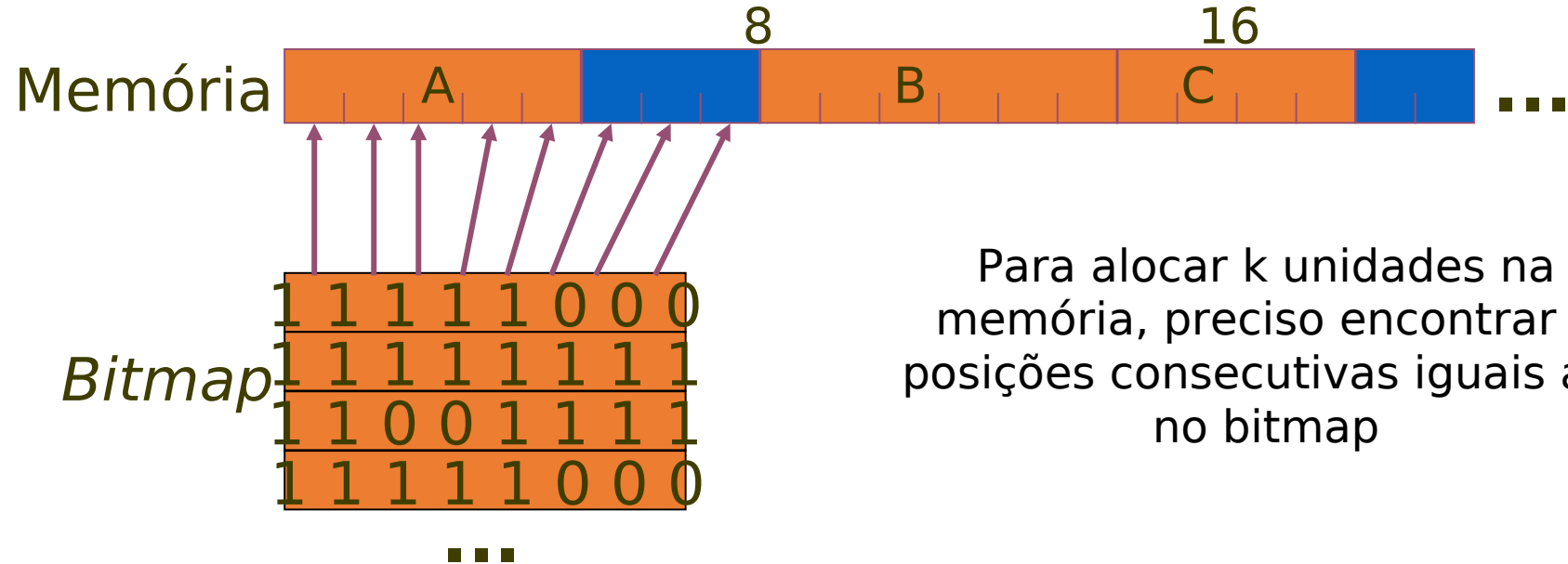
- Minimizar espaço de memória inutilizados:
 - Compactação: necessária para recuperar os espaços perdidos por fragmentação; no entanto, muito custosa para a CPU;
- Técnicas para alocação dinâmica de memória:
 - *Bitmaps*;
 - Listas Encadeadas;

Gerenciamento de Memória

- Técnica com *Bitmaps*:
 - Memória é dividida em unidades de alocação em kbytes;
 - Cada unidade corresponde a um *bit* no *bitmap*:
 - 0 → livre
 - 1 → ocupado
 - Tamanho do *bitmap* depende do tamanho da unidade e do tamanho da memória;
 - Ex.:
 - unidades de alocação pequenas → *bitmap* grande;
 - unidades de alocação grandes → perda de espaço;

Gerenciamento de Memória

- Técnica com *Bitmaps*:

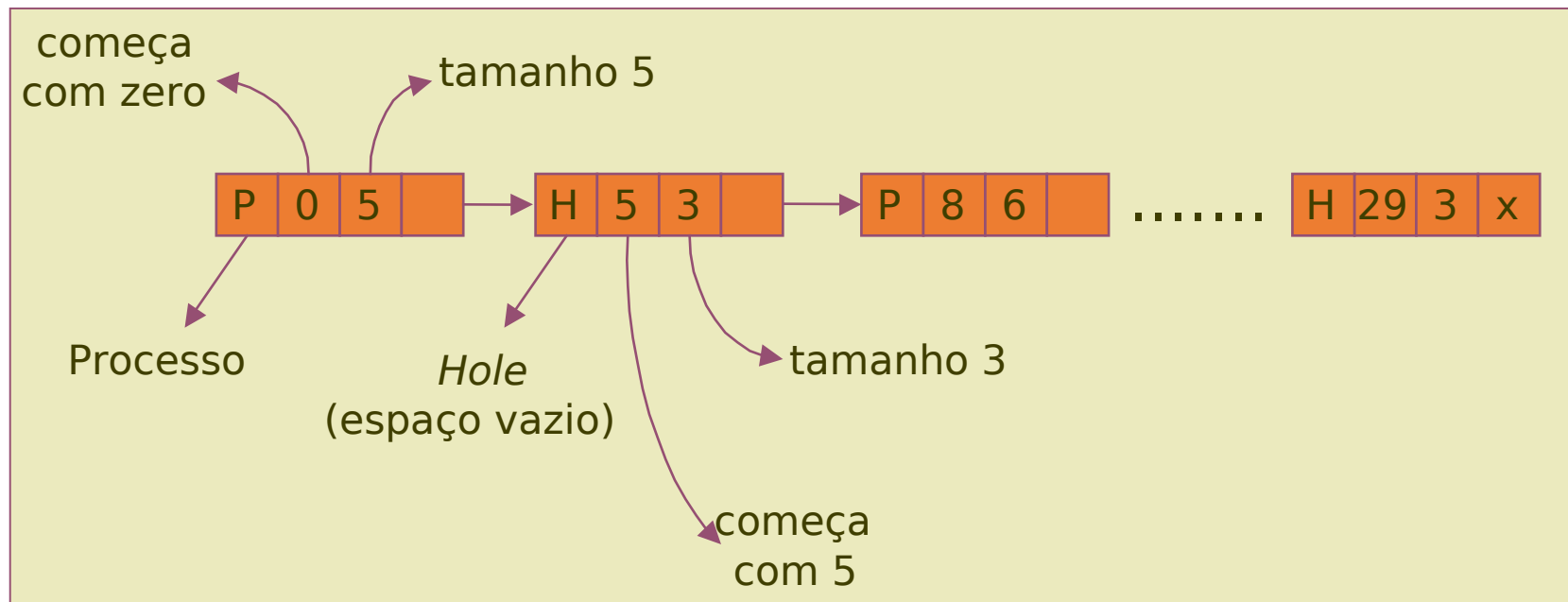


Para alocar k unidades na memória, preciso encontrar k posições consecutivas iguais a 0 no bitmap

■ Memória ocupada
■ Memória livre

Gerenciamento de Memória

- Técnica com Listas Encadeadas:
 - Uma lista para os espaços vazios e outra para os espaços cheios, ou uma lista para ambos!
 - “espaço \equiv segmento”



Gerenciamento de Memória

- Algoritmos de Alocação → quando um novo processo é criado:
 - *FIRST FIT*
 - 1º segmento é usado;
 - Rápido, mas pode desperdiçar memória por fragmentação;
 - *NEXT FIT*
 - 1º segmento é usado;
 - Mas na próxima alocação inicia busca do ponto que parou anteriormente;
 - Possui desempenho inferior;

Gerenciamento de Memória

- *BEST FIT*
 - Procura na lista toda e aloca o espaço que mais convém;
 - Menor fragmentação;
 - Mais lento;
- *WORST FIT*
 - Aloca o maior espaço disponível;
- *QUICK FIT*
 - Mantém listas separadas para os espaços mais requisitados;

Gerenciamento de Memória

- Cada algoritmo pode manter listas separadas para processos e para espaços livres:
 - Vantagem:
 - Aumenta desempenho;
 - Desvantagens:
 - Aumenta complexidade quando espaço de memória é liberado – gerenciamento das listas;
 - Fragmentação;

Gerenciamento de Memória

O que fazer quando não existe espaço suficiente para todos os processos ativos?

- **Swapping**
 - Chaveamento de processos inteiros entre a memória principal e o disco
- **Overlays → Memória Virtual**
 - Programas são divididos em pedaços menores
 - Pedaços são chaveados entre a memória principal e o disco

Gerenciamento de Memória

- ***Swapping:***

- Chaveamento de processos inteiros entre a memória principal e o disco;
- Transferência do processo da memória principal para a memória secundária (normalmente disco): *Swap-out*;
- Transferência do processo da memória secundária para a memória principal: *Swap-in*;
- Pode ser utilizado tanto com partições fixas quanto com partições variáveis;

Gerenciamento de Memória

Memória Virtual (MV)

- Programas maiores que a memória eram divididos em pedaços menores chamados *overlays* → programador;
 - Desvantagem: custo muito alto;
- Memória Virtual:
 - Sistema operacional é responsável por dividir o programa em *overlays*;
 - Sistema operacional realiza o chaveamento desses pedaços entre a memória e o disco;