

Gerência de Memória

Introdução e Particionamento

Marcelo Johann

Aulas anteriores...

- Ciclo de Compilação, ligação e carga

Plano da aula

- Introdução
- Hardware para Proteção de Memória
- Alocação contígua simples
 - Vantagens e limitações
- Alocação particionada estática e dinâmica
 - Fragmentação (!)
 - Gerenciamento de lacunas.
- Uma palavra sobre *swap* de processos

Novo capítulo

Gerência de Memória

Memória lógica & física

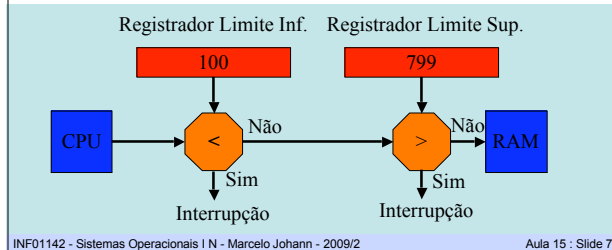
- Do lado do processo, enxerga-se um espaço de endereçamento **lógico**.
 - Ele acessa endereços lógicos.
 - Seu segmento de código também é armazenado em endereços lógicos.
 - Cada processo tem seu espaço lógico independente, contíguo, indo do endereço 0 até o endereço N-1.
- O HW disponibiliza **memória física**
 - Implementada em circuitos.
 - Fisicamente limitada (e.g. 512 Mbytes).
 - Usa vários níveis de Cache (hierarquia).

Memory Management Unit - MMU

- O HW disponibiliza um componente específico para fazer a tradução lógico/físico: a **MMU**.
 - Unidade de Gerenciamento da Memória.
 - Em geral, integrada com o processador.
- O Sis. Op. também participa dessa tradução.
 - Gerencia tabelas de tradução.
 - Organiza os vários níveis de memória acessíveis.
- Juntos, definem uma função de **mapeamento** dos endereços lógicos para os endereços físicos.
 - A interação exata é complexa.

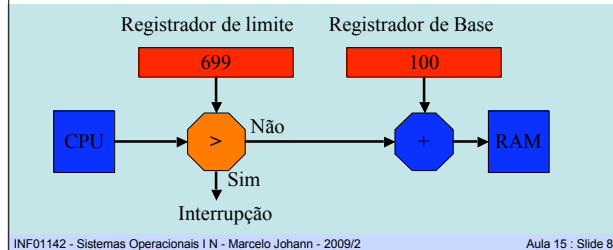
MMU e controle de acesso

- A MMU é frequentemente usada para controlar os acessos na memória.
- Exemplo: controle com **registradores de limite**.
 - Apenas possibilita um mapeamento limitado na memória.
 - Mapeamento direto
 - Aqui, controla acessos entre 100 e 799.



MMU e controle de acesso (2)

- Controle com registrador de base
 - possibilita um deslocamento do endereço
 - Aqui controla acessos físicos entre 100 e 799.
 - Mapea endereços de 0 até 699 para 100 até 799.



Interação MMU / Sis. Op. em ambos casos

- Cada processo tem seus acessos à memória limitados pelos dois valores nos registradores.
- Os mesmos devem ser armazenados no contexto do processo.
 - O salvamento de contexto deve limpar os registradores e restaurá-los!
 - Esses registradores devem ser protegidos dos usuários!
 - Acesso apenas em modo supervisor.

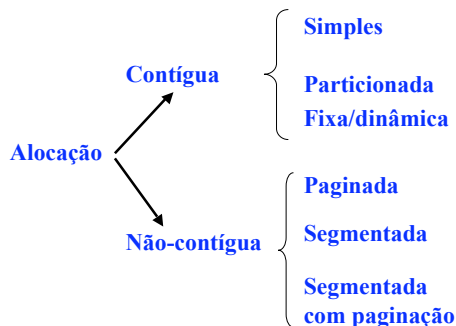
INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 15 : Slide 9

O que se espera da memória?

- Cada processo deve enxergar um espaço de endereçamento lógico
 - Grande (2-4 GBytes)
 - Contíguo (indo de 0 até N-1).
- A memória física é
 - Escassa
 - "compartilhada" entre os processos
 - Quer dizer que ela deve ser repartida entre os processos!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 15 : Slide 10

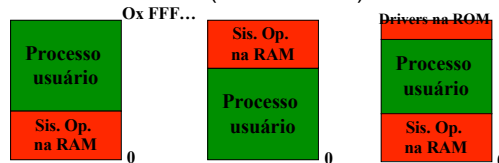
Como alocar memória?



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 15 : Slide 11

Alocação simples

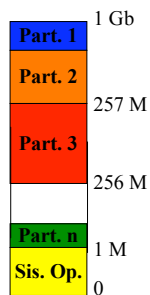
- O esquema mais simples possível compartilha a memória entre o Sis. Op. e um processo usuário.
 - Executa apenas um processo usuário por vez (MONO-programação).
- Basicamente, existem 3 possibilidades:
 - MS-DOS usa a 3a (BIOS = ROM)



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2 Aula 15 : Slide 12

Alocação particionada fixa

- Para possibilitar a multi-programação, a memória é dividida em partições:
 - de tamanho igual;
 - Ou de tamanhos diferentes.
- O tamanho das partições é fixo.
- OBS.: Fixo \neq idêntico

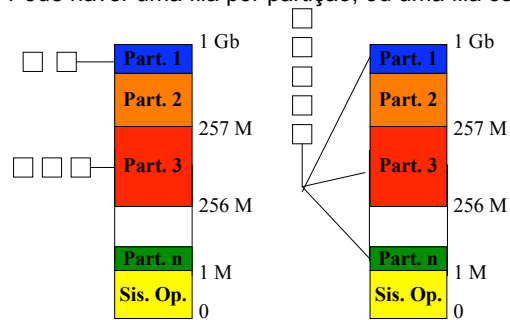


INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 13

Filas de espera de processos

- Pode haver uma fila por partição, ou uma fila central



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 14

Qual partição alocar a qual processo?

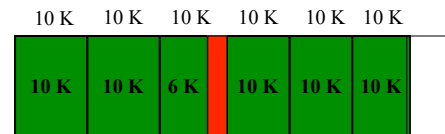
- Sistema com múltiplas filas**
 - Ordenação dos processos;
 - Problema quando uma fila para partições grandes é vazia, apesar de ter fila para partições menores!
- Sistema com fila única**
 - Deve varrer a fila para escolher o "melhor" processo para uma dada partição.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 15

Problema de Fragmentação (1)

- Problema: qual é o tamanho para as partições?
- O que fazer com processos cujo uso de memória varia durante sua execução?
- Quando é alocada uma partição maior do que o necessário a um processo, há desperdício.
 - Fragmentação interna.**



- Problema minimizado com o esquema com partições de tamanhos diferenciados.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 16

Alocação com particionamento dinâmico

- O tamanho da partição pode variar sob-demanda.
 - Inicialmente, toda a memória é considerada como sendo um bloco alocável;
 - A medida que blocos vão sendo alocados;
 - Quando um processo se encerra, ele libera o bloco que esteve usando.
 - Complica o gerenciamento!
- O Sis. Op. deve manter uma lista dos "buracos" (blocos disponíveis).
 - Os buracos são chamados "lacunas".
 - Conforme for a evolução dos processos, as lacunas vão se espalhando na memória.
 - O Sis. Op. deve, para alocar um bloco, percorrer a lista de lacunas até achar a lacuna mais bem adaptada.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

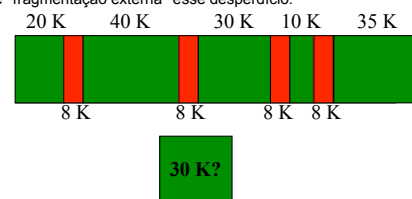
Aula 15 : Slide 17

Gerenciar as lacunas (1)

– A fragmentação contra-ataca

- Dois problemas de gerenciamento:**

- As lacunas criam fragmentação externa.
 - Com a multiplicação de lacunas pequenas, aumenta a memória que não pode ser aproveitada pelos processos.
 - Chama-se "fragmentação externa" esse desperdício.



- Qual lacuna alocar a um processo que pede por um tamanho X?

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 18

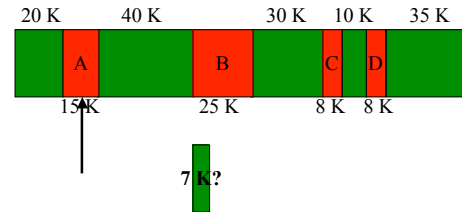
Gerenciamento de lacunas (2) – alocação de lacuna

- O Sis. Op. gerencia uma lista de lacunas.
- Quando se aloca uma lacuna, o espaço não usado da mesma cria uma nova lacuna, menor.
- Qual lacuna alocar a um processo que pede por um tamanho X?
 - **FIRST-FIT**: a primeira lacuna onde cabem X bytes;
 - Procura pode começar sempre no início da lista, ou a partir do último bloco alocado ("next-fit").
 - **BEST-FIT**: a lacuna de tamanho mais parecido a X (sendo maior do que X);
 - Pode se aproveitar da ordenação da lista de lacunas.
 - **WORST-FIT**: a maior lacuna disponível.
 - Bom, pois a nova lacuna criada será grande.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 19

Exemplos de escolha de lacuna



First-Fit: retorna a lacuna A

Best-Fit: retorna a lacuna C

Worst-Fit: retorna a lacuna B

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 20

Fragmentação (3o episódio)

- Uma boa estratégia de escolha de lacunas ajuda minimizar a fragmentação externa...
 - ... mas ainda existe!
 - Estudos estatísticos mostram "**lei dos 50%**": usando N blocos, se perdem N/2 blocos...
- Solução radical: **compactação**
 - Unir as lacunas de forma a agregá-las e formar uma lacuna gigante.
 - Deslocar todos os processos para os endereços baixos, e as lacunas para cima.
 - Extremamente lento!
 - 512 Mbytes, 1 Ghz/32 bits => 1/8 de segundo.
 - Necessita um mecanismo de amarramento dinâmico.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 21

Implementação de listas de lacunas

- **Bitmap**
 - Define-se um a unidade básica de memória (bloco): 1 palavra, 1 KB, ...
 - Armazena-se 1 bit por unidade
 - 0: unidade disponível; 1: unidade alocada.
 - Maior a unidade, menor o bitmap e maior a fragmentação interna!
 - Tamanho do bitmap proporcional ao tamanho da memória.
 - Procura sequencial.
- **Lista encadeada**
 - Cada elemento armazena o endereço de início e o tamanho da lacuna;
 - Também deve ser armazenado um ponteiro para o próximo elemento da lista.
 - Usa às próprias lacunas para armazenar a lista!
 - Necessidades de deletar/compactar impõem listas duplamente encadeadas.
 - Possibilidade de ordenar os elementos para agilizar o gerenciamento.
 - Tamanho proporcional ao número de blocos disponíveis.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 22

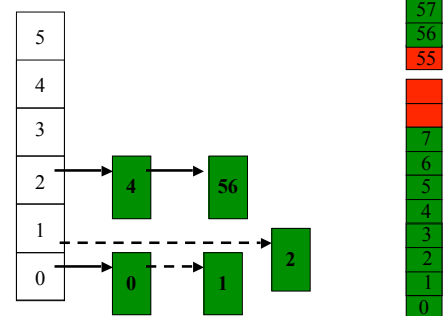
Compactação "on-line"

- O Linux usa o algoritmo chamado "buddy" para agregar as lacunas a cada alocação/liberação de um bloco.
- Vetor de listas de blocos disponíveis (lacunas)
 - A entrada i aponta para uma lista de blocos de tamanho 2^i
- À alocação:
 - O que sobra da lacuna alocada é decomposto em potências de 2 e entra no vetor, nas devidas entradas.
 - Se podem ser fusionados com vizinhos, os blocos se agregam.
- À liberação: mesma coisa.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 23

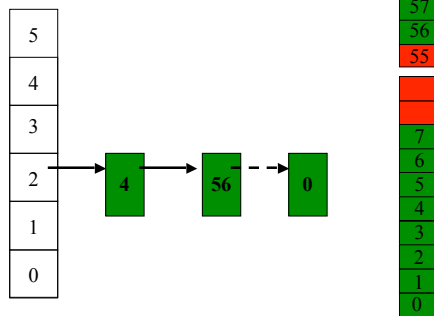
Ilustração do algoritmo buddy



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 24

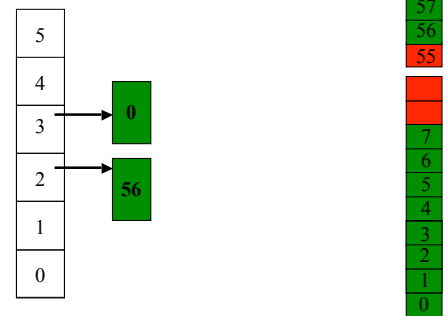
Ilustração do algoritmo buddy



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 25

Ilustração do algoritmo buddy



INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 26

Swap de Processos

- Sobra ainda um problema: nem todos os processos cabem na memória!
- Há necessidade de descarregar da memória parte dos processos, num dado momento
 - Eles vão para o disco
- A operação de descarga se chama “**swap-out**”; a carga se chama “**swap-in**”.
- É uma operação lenta... mas crítica.
- Nada obriga o Sis. Op. a alocar, no swap-in, a mesma partição a um processo que tinha já executado.
 - Necessidade de ligação dinâmica.

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 27

Conclusão sobre particionamento

- Na verdade, boa parte de nossos problemas (fragmentação...) vêm da exigência de se alocar um espaço contíguo de memória.
- Idéia genial: alocar memória de forma não contígua!
- = Paginação & Tabela de Páginas...
- Será o assunto da próxima semana!

INF01142 - Sistemas Operacionais I N - Marcelo Johann - 2009/2

Aula 15 : Slide 28