

Alocação de Quadros

Referência:

Silberschatz, Abraham. *Sistemas Operacionais com Java*. 7 ed. Rio de Janeiro: Elsevier, 2008.

- Cada processo precisa do número *mínimo* de páginas
- Exemplo: IBM 370 – 6 páginas para tratar da instrução SS MOVE:
 - instrução tem 6 bytes, pode espalhar por 2 páginas
 - 2 páginas para tratar *de*
 - 2 páginas para tratar *para*
- Dois esquemas de alocação principais
 - alocação fixa
 - alocação por prioridade

Alocação fixa

- Alocação igual – Por exemplo, se houver 100 quadros e 5 processos, dá a cada processo 20 quadros.
- Alocação proporcional – Aloca de acordo com o tamanho do processo

Alocação por prioridade

- Usa um esquema de alocação proporcional ao invés de tamanho
- Se processo P_i gera uma falta de página,
 - seleciona para substituição um de seus quadros
 - seleciona para substituição um quadro de um processo com número de prioridade menor

Alocação global versus local

- Substituição global – processo seleciona um quadro de substituição do conjunto de todos os quadros; um processo pode apanhar um quadro de outro

- **Substituição local** – cada processo seleciona apenas do seu próprio conjunto de quadros alocados

Thrashing

- Se um processo não tem páginas “suficientes”, a taxa de falta de página é muito alta. Isso leva a:
 - baixa utilização de CPU
 - sistema operacional pensa que precisa aumentar o grau de multiprogramação
 - outro processo acrescentado ao sistema
- Thrashing \equiv um processo está ocupado trocando páginas pra dentro e pra fora

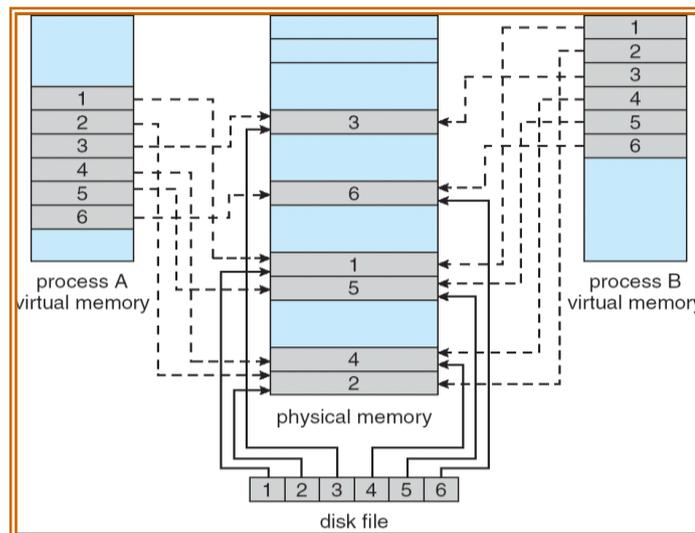
Paginação por demanda e thrashing

- Por que a paginação por demanda funciona?
Modelo de localidade
 - Processo migra de uma localidade para outra
 - Localidades podem se sobrepor
- Por que ocorre o thrashing?
 Σ tamanho da localidade > tamanho total da memória

Arquivos mapeados na memória

- E/S de arquivo mapeado na memória permite que a E/S de arquivo seja tratada como acesso de rotina à memória, mapeando um bloco de disco para uma página na memória
- Um arquivo é lido inicialmente usando paginação por demanda. Uma parte do arquivo com tamanho da página é lida do sistema para uma página física. Leituras/escritas subseqüentes de/para o arquivo são tratadas como acessos comuns à memória.
- Simplifica o acesso ao arquivo, tratando a E/S do arquivo por meio da memória, ao invés das chamadas do sistema read() e write()

- Também permite que vários processos sejam mapeados para o mesmo arquivo, permitindo que as páginas na memória sejam compartilhadas



Alocando memória do kernel

- Tratado diferente da memória do usuário
- Normalmente alocado por um pool de memória livre
 - Kernel solicita memória para estruturas de tamanhos variáveis
 - Alguma memória do kernel precisa ser contígua

Sistema buddy

- Aloca memória a partir de segmento de tamanho fixo, consistindo em páginas de memória fisicamente contíguas
- Memória alocada usando alocador de potência de 2
 - Satisfaz requisitos em unidades com tamanho de potência de 2
 - Requisição arredondada para próxima potência de 2 mais alta
 - Quando precisa de alocação menor que o disponível, pedaço atual dividido em dois buddies da próxima potência de 2 menor
 - Continua até haver chunk de tamanho apropriado

Outras questões – pré-paginação

- **Pré-paginação**
 - Reduzir o grande número de faltas de página que ocorre no início do processo
 - Pré-paginar todas ou algumas das páginas que um processo precisará.
 - Mas se as páginas pré-paginadas não forem usadas, E/S e memória foram desperdiçadas
 - Suponha que s páginas sejam preparadas e α das páginas seja usado
 - O custo de $s * \alpha$ faltas de página de salvamento > ou < que o custo de preparar $s * (1 - \alpha)$ páginas desnecessárias?
 - α perto de zero \Rightarrow preparando perdas

Outras questões – tamanho de página

- **Coleção de tamanho de página deve levar em consideração:**
 - fragmentação
 - tamanho da tabela
 - overhead de E/S
 - localidade

Outras questões – Alcance do TLB

- **Alcance do TLB – A quantidade de memória acessível pelo TLB**
- **Alcance do TLB = (Tamanho TLB) X (Tamanho da página)**
- **Ideal que o conjunto de trabalho de cada processo seja armazenado no TLB**
 - **Caso contrário, há um alto grau de faltas de página**
- **Aumenta o tamanho da página**
 - **Isso pode levar a um aumento na fragmentação, pois nem todas as aplicações exigem um tamanho de página grande**
- **Oferece múltiplos tamanhos de página**

- **Isso permite às aplicações que exigem tamanhos de página maiores oportunidade de usá-las sem aumento na fragmentação**

Outras questões – interlock de E/S

- **Interlock de E/S – Páginas às vezes precisam ser bloqueadas na memória**
- **Considere E/S – Páginas que são usadas para copiar um arquivo de um dispositivo precisam evitar serem selecionadas para evicção por um algoritmo de substituição de página**