Paginação por Demanda

Sistemas Operacionais

Prof. Pedro Ramos
pramos.costar@gmail.com

Pontifícia Universidade Católica de Minas Gerais
ICEI - Departamento de Ciência da Computação

PAGINAÇÃO POR DEMANDA

Até agora, assumimos que o tamanho da memória virtual é igual ao tamanho da memória física.

Ilusões do sistema operacional:

-> tratar o disco (ou outro armazenamento auxiliar) como uma memória principal muito maior, porém mais lenta.

De maneira análoga a cache, porém mais lento.

A ilusão de uma memória virtual infinita:

um processo pode ser maior que a memória física,

um processo pode executar mesmo se não estiver todo em memória,

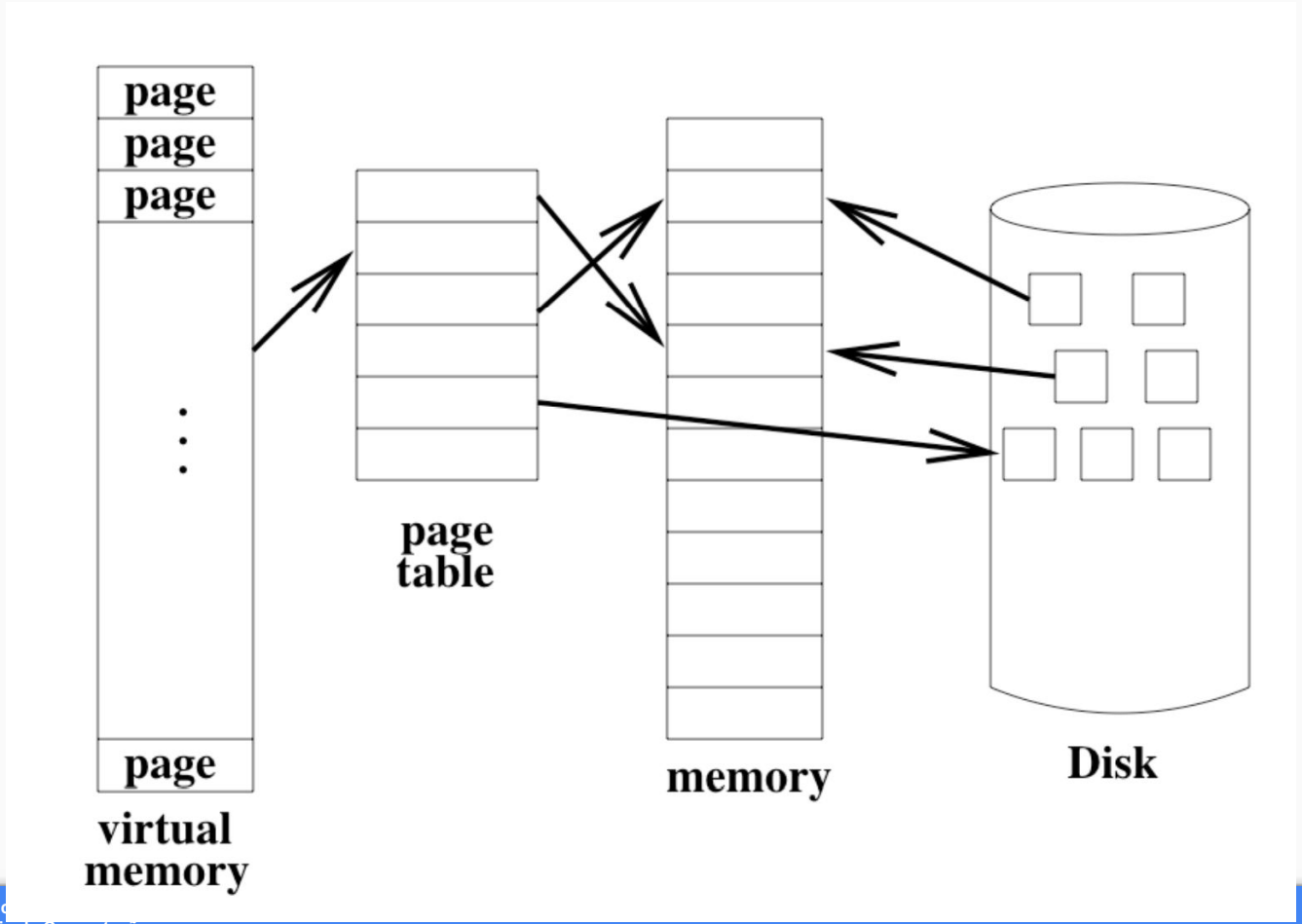
e mais processos do que cabem na memória podem rodar simultaneamente.

PAGINAÇÃO POR DEMANDA

- A paginação por demanda usa a memória como uma cache para o disco.
- A tabela de páginas (mapeamento de memória) **indica se a página está no disco ou na memória**, usando um bit de validade.
- Quando uma página é carregada do disco para a memória, o sistema operacional atualiza a tabela de páginas e o *bit de validade*.
- Por **razões de eficiência**, os acessos à memória precisam ***referenciar páginas que estão na memória na maior parte do tempo***.
  - *Caso contrário, o tempo efetivo de acesso à memória se aproximará do tempo de acesso ao disco.*

Ideia principal: Localidade – o conjunto de trabalho de um processo precisa caber na memória e permanecer lá (regra dos 90/10).

PAGINAÇÃO POR DEMANDA



QUANDO CARREGAR UMA PÁGINA?

No início do processo: o espaço de endereço virtual não deve ser maior do que a memória física.

Overlays: o programador da aplicação indica quando carregar e remover páginas.

- *Permite que o espaço de endereço virtual seja maior que o espaço de endereço físico*
- *É difícil de implementar e propenso a erros*

Paginação sob demanda: o processo avisa o sistema operacional antes de precisar de uma página e, em seguida, quando termina de usá-la.

QUANDO CARREGAR UMA PÁGINA?

Paginação sob demanda: o sistema operacional carrega uma página na primeira vez que ela é referenciada

- *Pode remover uma página da memória para abrir espaço para a nova página.*
- *O processo precisa liberar a CPU enquanto a página está sendo carregada*
- ***Page fault:*** *interrupção que ocorre quando uma instrução referencia uma página que não está na memória.*

Pré-paginação: o sistema operacional tenta prever quais páginas o processo precisará e as carrega previamente na memória.

- *Permite* ***maior sobreposição entre CPU e E/S*** *se o sistema operacional* ***acertar na previsão.***
- *Se o sistema operacional errar => page fault*
- ***Erros*** *podem resultar na* ***remoção de páginas úteis****.*
- *Difícil de acertar devido a bifurcações no código.*

PAGINAÇÃO POR DEMANDA

Uma cópia completa do programa deve estar armazenada no disco.
→ *PORQUÊ?*

O bit de validade na tabela de páginas indica se a página está na memória.

1: na memória 0: não está na memória (ou está no disco ou em um endereço inválido)

PAGINAÇÃO POR DEMANDA

Se a página não está na memória: interrupção para o SO na primeira referência.

O SO verifica se o endereço é válido. Se for, o SO:

- seleciona uma página para substituir (algoritmo de substituição de páginas),
- invalida a página antiga na tabela de páginas,
- começa a carregar a nova página do disco para a memória,
- alterna o contexto para outro processo enquanto a E/S está em andamento,
- recebe uma outra interrupção informando que a página foi carregada na memória,
- atualiza a entrada da tabela de páginas,
- retoma o processo que causou a falha (por que não continuar o processo atual?).

ESPAÇO DE SWAP (TROCA)

O que acontece quando uma página é removida da memória?

- Se a página contiver código, podemos simplesmente removê-la, pois ela pode ser recarregada do disco.
- Se a página contiver dados, precisamos salvar esses dados para que possam ser recarregados caso o processo a que pertencem precise usá-los novamente.
- Espaço de swap: uma parte do disco é reservada para armazenar páginas que são removidas da memória.

Uma página de memória virtual pode existir em um ou mais dos seguintes locais:

- O sistema de arquivos
- Memória física
- Espaço de swap

A tabela de páginas precisa ser **mais sofisticada para saber onde encontrar uma página.**

DESEMPENHO PAGINAÇÃO POR DEMANDA

Teoricamente, um processo poderia acessar uma nova página a cada instrução.

Localidade temporal: se um processo acessa um item na memória, tende a referenciar o mesmo item novamente em breve.

Localidade espacial: se um processo acessa um item na memória, tende a referenciar um item adjacente em breve.

DESEMPENHO PAGINAÇÃO POR DEMANDA

Seja *p* a probabilidade de um *page fault* (0 ≤ p ≤ 1).

Tempo de acesso efetivo = (1 - p) x *ma* + p x tempo de *page fault*

- Se o tempo de acesso à memória é 200 ns e um *page fault* leva 25 ms
- Tempo de acesso efetivo = (1 - p) x 200 + p x 25.000.000

Se queremos que o tempo de acesso efetivo seja apenas 10% mais lento que o tempo de acesso à memória, qual valor *p* deve ter?

ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINA

Em um *page fault*, precisamos escolher uma página para remover.

- **Aleatório**: surpreendentemente, esse algoritmo funciona bem.
- **FIFO**: *First-In, First-Out*. Remove a página mais antiga. É simples de implementar, mas o sistema operacional *pode facilmente remover uma página que está sendo acessada com frequência*.
- **MIN**: (também conhecido como ***OPT***) Olha para o futuro e remove a página que será acessada mais longe no futuro (provavelmente ótimo [Belady'66]). Problema?
- **LRU**: *Least Recently Used* (Menos Recentemente Usada). Aproximação do MIN que funciona bem se o passado recente for um bom indicador do futuro. Remove a página que não foi usada há mais tempo.

PAGINAÇÃO POR DEMANDA - FIFO

3 Quadros na memória
4 páginas virtuais A, B, C, D
Fluxo: A B C A B D A D B C B

FIFO - Primeiro a entrar, primeiro a sair

	A	B	C	A	B	D	A	D	B	C	B
QUADRO 1											
QUADRO 2											
QUADRO 3											

Qual é o número de PAGE FAULTS?

PAGINAÇÃO POR DEMANDA - MIN

3 Quadros na memória
4 páginas virtuais A, B, C, D
Fluxo: A B C A B D A D B C B

MIN - Remove a página que será acessada mais longe no futuro

	A	B	C	A	B	D	A	D	B	C	B
QUADRO 1											
QUADRO 2											
QUADRO 3											

Qual é o número de PAGE FAULTS?

PAGINAÇÃO POR DEMANDA - LRU

3 Quadros na memória
4 páginas virtuais A, B, C, D
Fluxo: A B C A B D A D B C B

LRU (Menos Usada Recentemente) - Remove a página que não foi usada há mais tempo.

	A	B	C	A	B	D	A	D	B	C	B
QUADRO 1											
QUADRO 2											
QUADRO 3											

Qual é o número de PAGE FAULTS?

ADICIONAR MEMÓRIA - FIFO

Adicionar memória, vai reduzir o número de page faults?

	A	B	C	D	A	B	E	A	B	C	D	E
QUADRO 1												
QUADRO 2												
QUADRO 3												
QUADRO 1												
QUADRO 2												
QUADRO 3												
QUADRO 4												

ADICIONAR MEMÓRIA - FIFO

ANOMALIA DE BELADY - Adicionar quadros de páginas podem causar mais page faults em alguns algoritmos como FIFO.

	A	B	C	D	A	B	E	A	B	C	D	E
QUADRO 1												
QUADRO 2												
QUADRO 3												
QUADRO 1												
QUADRO 2												
QUADRO 3												
QUADRO 4												

ADICIONAR MEMÓRIA - LRU

NO LRU, PORÉM, AUMENTAR O NÚMERO DE FRAMES SEMPRE ABAIXA O NÚMERO DE *PAGE FAULTS*. Porquê?

	A	B	C	D	A	B	E	A	B	C	D	E
QUADRO 1	**A***	A	A	**D***	D	D	**E***	E	E	**C***	C	C
QUADRO 2		**B***	B	B	**A***	A	A	A	A	A	**D***	D
QUADRO 3			**C***	C	C	**B***	B	B	B	B	B	B
QUADRO 1	**A***	A	A	A	A	A	A	A	A	A	A	**E***
QUADRO 2		**B***	B	B	B	B	B	B	B	B	B	B
QUADRO 3			**C***	C	C	C	**E***	E	E	E	**D***	D
QUADRO 4				**D***	D	D	D	D	D	**C***	C	C

RESUMO

Vantagens da paginação sob demanda:

- O espaço de endereço virtual pode ser maior que o espaço de endereço físico.
- Os processos podem rodar sem estarem totalmente carregados na memória.
  - Os processos **iniciam mais rápido** porque **só precisam carregar algumas páginas** (de código e dados) para começar a rodar.
  - Os processos podem **compartilhar memória de forma mais eficaz, reduzindo os custos quando ocorre uma troca de contexto.**
- Um bom algoritmo de substituição de páginas pode **reduzir o número de *page faults* e melhorar o desempenho.**

PERGUNTAS?

REFERÊNCIAS

- **TANENBAUM, Andrew.** Sistemas operacionais modernos.
- **SILBERSCHATZ, Abraham et al.** Fundamentos de sistemas operacionais: princípios básicos.
- **MACHADO, Francis; MAIA, Luiz Paulo.** Arquitetura de Sistemas Operacionais.
- **CARISSIMI, Alexandre et al.** Sistemas operacionais.