

Trabalho prático 3º unidade - Memória virtual com paginação

Raí Emanuel T. de Freitas
raiemmanuel50.50@gmail.com

Departamento de Engenharias e Tecnologia - DETEC - Campus Pau dos Ferros.
Universidade Federal Rural do Semi-Árido

13 de fevereiro de 2025

Material em construção

Esse slide está em construção! Novas atualizações ou correções serão aplicadas.

Sumário

- 1 Discord
- 2 Github
- 3 Sorteio dos temas
- 4 Políticas de busca de páginas
- 5 Algoritmos de substituição de páginas
 - Algoritmo aleatório
 - Algoritmo FIFO (First In, First Out)
 - Algoritmo LFU (Least-Frequently-Used)
 - Algoritmo LRU (Least-Recently-Used)
- 6 Entrega do trabalho
- 7 Referências

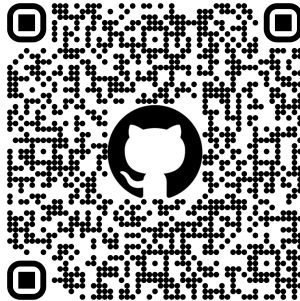
Discord

Chat no Discord para dúvidas.



Repositório Github

Repositório no Github com materiais da disciplina de sistemas operacionais.
Slide: Slide trabalho prático 3 unidade.pdf



Sorteio dos temas

- Podem-se repetir os grupos dos outros relatórios
 - 5 ou 6 pessoas
- Cada grupo será responsável por sortear um algoritmo de substituição de páginas para implementar e redigir um relatório técnico
 - 0 - Aleatório
 - 1 - FIFO (First In, First Out)
 - 2 - LRU (Least Recently Used)
 - 3 - LFU (Least Frequently Used)

Link da planilha

Link da planilha de sorteio - Clique aqui para acessar

Políticas de busca de páginas

- A política de busca de páginas determina quando uma página deve ser buscada
 - Paginação por demanda
 - Paginação antecipada
- Paginação por demanda
 - As páginas são transferidas do disco para a RAM somente quando necessário
 - Traz apenas as páginas necessárias naquele momento
 - Talvez partes pouco usadas dos programas nunca sejam carregadas na RAM
- Paginação antecipada
 - Traz páginas além da solicitada naquele momento
 - Há apenas uma expectativa de uso futuro dessas páginas
 - Considere um programa armazenado sequencialmente no disco
 - Faz sentido levar várias instruções próximas de uma vez
 - Uma busca traz todas as instruções de uma vez, em vez de buscá-las uma a uma
 - Se não for necessário, o programa terá perdido tempo e espaço

Algoritmos de substituição de páginas

- A maior dificuldade não é escolher qual página carregar, mas sim qual página liberar
 - A escolha errada pode ter consequências negativas no desempenho
- Considere que um processo necessita de um frame, mas não há nenhum disponível
 - O sistema deve escolher, entre todas as páginas na memória RAM, qual será liberada.
- Os algoritmos de substituição objetivam remover páginas que não serão referenciadas tão cedo
 - O page out de uma página que em seguida será referenciada fará um acesso ao disco desnecessário
- O algoritmo perfeito seria aquele que adivinharia qual página não será mais referenciada ou demorará a ser chamada

Algoritmos de substituição de páginas - Aleatório

- “joga um dado” e remove uma página, sem se preocupar se ela será referenciada em breve
- Não há critério de seleção
 - Elimina uma página de forma aleatória, sem critérios bem definidos
- Todas as páginas são equiprováveis de sofrerem *page out*
 - Páginas de alta frequência de uso também estão vulneráveis
- Fácil de implementar
- Consome pouco recurso do sistema
 - “pensa pouco”
- Pouco implementado devido à sua baixa eficiência
 - Pode remover uma página que ainda é muito usada
- O algoritmo não é sensível às referências das páginas na memória

Algoritmos de substituição de páginas - Aleatório

- Dica de implementação
 - Link sobre a classe de geração de números pseudoaleatórios em Java
 - Link de tutorial de geração de números pseudoaleatórios
 - Link de arquivo exemplo - 3 unidade/trabalho prático/NumerosAleatorios.java

Algoritmos de substituição de páginas - FIFO (First In, First Out)

- A primeira página a entrar na memória será a primeira a sair
- Seleciona-se a página que está há mais tempo na RAM
- Nem sempre escolher o mais antigo para sair é uma boa escolha
 - Uma página com dados frequentemente acessados pode tornar-se antiga
- É interessante levar em conta não apenas a ordem em que as páginas foram adicionadas
- Fácil de implementar
- O algoritmo não é sensível às referências das páginas na memória

Algoritmos de substituição de páginas - FIFO (First In, First Out)

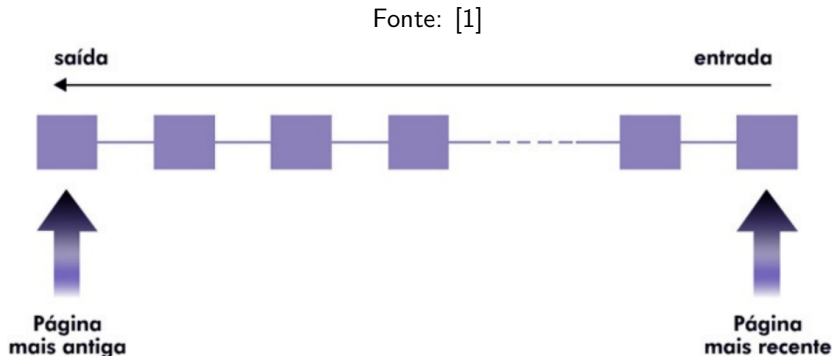


Figura: Funcionamento do algoritmo de substituição de páginas FIFO

Algoritmos de substituição de páginas - FIFO (First In, First Out)

- Dica de implementação
 - Estrutura de dados fila
 - Link de arquivo exemplo - 3 unidade/trabalho prático/FilaExemplo.java
 - Collections, interface Queue, LinkedList em Java
 - Array básico com deslocamento dos elementos
 - `Page [] x = new Page[10];`

Algoritmos de substituição de páginas - LFU (Least-Frequently-Used)

- Seleciona-se a página menos referenciada
- É mantido um contador para cada página que está na memória principal
 - O contador conta quantas vezes a página foi referenciada
- A página com o menor contador vai ser eliminada quando for necessário liberar espaço na memória
- O algoritmo evita selecionar páginas que são muito referenciadas
- Um problema surge por existir páginas que acabaram de sofrer page in
 - Páginas que acabaram de entrar na RAM estão em desvantagem porque o seu contador é recente
- Uma página que foi muito referenciada no passado pode ter deixado de ser tão referenciada. Páginas antigas podem levar vantagem em relação às páginas novas
- Reduz a chance de substituir uma página muito utilizada
- O algoritmo é sensível às referências das páginas na memória

Algoritmos de substituição de páginas - LFU (Least-Frequently-Used)

- Dica de implementação
 - Adicionar uma variável int cont na classe que representa a página. A cada referência, incrementa-se 1 no contador: `pagina.cont++`;
 - Registra a contagem de referências das páginas
 - Zera-se o contador ao sair da RAM
 - Busca linear pelo menor contador
 - Substitui a página com menor contagem

Algoritmos de substituição de páginas - LRU (Least Recently Used)

- Seleciona-se a página que está há mais tempo sem ser referenciada
- Baseado na localidade temporal
 - Uma página que foi referenciada recentemente tende a ser referenciada em breve
 - Prioriza referências recentes
- É necessário registrar o momento da última referência à página acessada
 - Timestamp/epoch
 - A cada referência ao frame, atualiza-se o timestamp
- O sistema seleciona o frame que está há mais tempo sem ser referenciado quando for necessário substituir uma página
 - Busca linear pelo timestamp mais antigo
- Alto custo de implementação
- Processamento extra para gerenciar timestamps
- O algoritmo é sensível às referências das páginas na memória

Algoritmos de substituição de páginas - LRU (Least Recently Used)

- Dica de implementação
 - Adicionar uma variável `long` tempo na classe que representa a página. A cada referência, atualiza-se o valor de tempo
 - Registra o tempo da última referência
 - Busca linear pelo menor timestamp
 - Substitui a página com menor timestamp
 - `System.currentTimeMillis()` em Java
 - Tutorial de uso do timestamp em Java
 - Link de arquivo exemplo - 3 unidade/trabalho prático/EpochExemplo.java

Entrega do trabalho

- E-mail: raiemmanuel50.50@gmail.com
 - Apenas um relatório por equipe
 - 5 ou 6 pessoas
 - Não há limite mínimo ou máximo de laudas
- Implementar o algoritmo de substituição de páginas sorteado e elaborar um relatório técnico
 - Qualquer linguagem pode ser usada
 - Link de código de exemplo - 3 unidade/trabalho prático/GerenciadorMemoria.java e 3 unidade/trabalho prático/Page.java
- Entrega: Até as 23h59min59s do dia 15/02/2025
- Link do slide base do trabalho - 3 unidade/trabalho prático/Slide trabalho prático 3 unidade.pdf
- Link do trabalho - 3 unidade/trabalho prático/Trabalho prático 3 unidade.pdf

Referências



F. B. Machado and L. P. Maia, *Arquitetura de Sistemas Operacionais*.

Rio de Janeiro, Brasil: Editora LTC, 3 ed., 2005.

Inclui exercícios.



A. S. Tanenbaum and H. Bos, *Sistemas Operacionais Modernos*.

Porto Alegre, RS: Bookman, 5 ed., 2024.

E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 09 fev. 2025.

Obrigado!