

MAC0422 - EP3

Gabriel de Russo e Carmo
Victor Wichmann Raposo
Novembro de 2016

Sumário

- Visão geral
- Organização do código
- Implementação e algoritmos
- Resultados dos experimentos

Visão Geral

- Cinco módulos
 - ep3.py - shell
 - pages.py - gerência da paginação
 - memory.py - gerência das memórias
 - utility.py - estruturas de dados, globais, etc
 - execution.py - coordenador da simulação

Visão Geral

- Modularizar o máximo possível
- Implementação de partes independentes
- Programação orientada a objetos
- Instantes de tempo **não** são segundos, mas sim iterações de um laço

Implementação

Processos

- Cada processo é um objeto da classe Process
- Base, tamanho, acessos e PID
- Estruturas de dados para processos (heaps)
 - RunningQueue (processos rodando, ordenados pelo tempo final)
 - ProcessQueue (processos que irão acessar a memória, ordenados pelo tempo do próximo acesso)
- Necessário pois o simulador não pode parar no último acesso

Memórias

- Cada memória é um objeto da classe Memory
- Tamanho, tamanho da página, arquivo de pids e bitmap (python bytearray)
- Escrita e leitura no modo binário com seek e funções auxiliares do Python
- Os algoritmos de alocação de memória são métodos
- Memória é limpa quando um processo acaba (ambas)

Paginação

- Classe Pages
- Quantidade de páginas, tamanho do quadro, bit R, e tabela de páginas
- Os algoritmos de gerência são métodos da classe
- Dependendo do algoritmo, novos atributos são definidos (por exemplo a fila do second chance)

Gerência de memória

First fit

- Simplesmente itera-se na memória virtual, de página em página, procurando um espaço
- Laço simples e utilização do bitmap

Next fit

- Ponteiro da última inserção
- Itera-se a partir do ponteiro na memória virtual, de página em página, procurando um espaço
- Laço modular e utilização do bitmap

Best fit

- Itera-se na memória mantendo a menor posição que consegue alocar nosso processo
- Dois ponteiros e variável best
- Caso de borda: melhor tamanho vai até o final

Worst fit

- Itera-se na memória mantendo a maior posição que consegue alocar nosso processo
- Dois ponteiros e variável worst
- Caso de borda: pior tamanho vai até o final

Substituição de páginas

Geral

- Páginas fantasmas em todos os algoritmos
- Dependendo da estrutura associada, a página fantasma tem informações diferentes
- Páginas fantasmas pré-carregadas nos quadros
- Facilita muito a implementação
- Muitos page faults no início
- Quadros de página de processos 'mortos' são eventualmente substituídos pelos algoritmos

Optimal

- Cada quadro de página tem como rótulo seu próximo acesso (infinito quando não existe)
- Vetor de rótulos (fantasmas com rótulo infinito)
- Invariante: rótulos são sempre futuros
- Basta atualizar **um** rótulo por acesso
- Atualização feita iterando-se nos próximos acessos do processo

Second chance

- Fila de páginas (inicialmente fantasmas com bit R valendo 0)
- Diversos push e pop enquanto não encontramos uma página com bit R valendo 0
- Sempre encontramos pois reinserimos páginas na fila com bit R valendo 0
- Fila do Python

Clock

- Vetor circular de tamanho fixo (páginas fantasmas com bit R valendo 0)
- Difere apenas na implementação: ponteiro modular avança enquanto não encontramos uma página com bit R valendo 0
- Sempre encontramos pois mudamos o bit R de páginas visitadas para 0

Least Recently Used

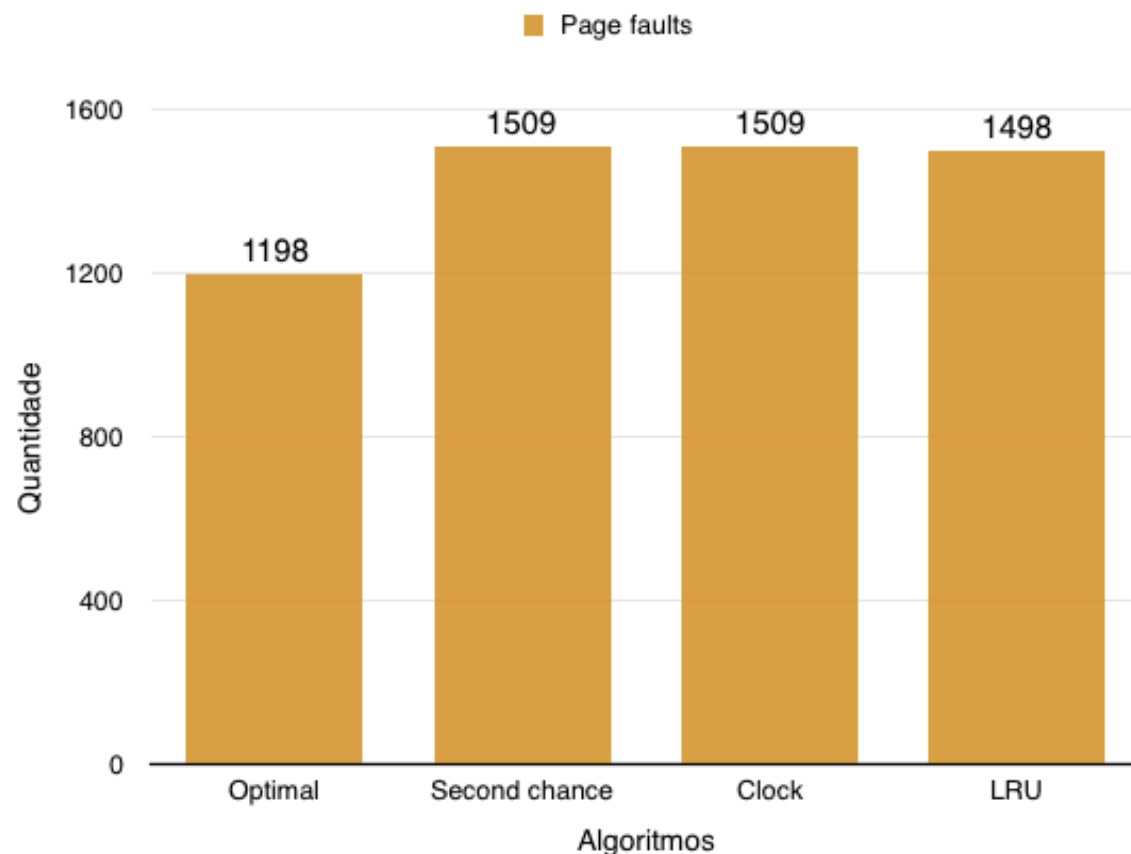
- Vetor de tamanho fixo de contadores de 32 bits (páginas fantasma com contador valendo 0)
- Os contadores são atualizados sempre que ocorre um acesso
- O bit R tem um papel fundamental. O intervalo de zerar o bit R tem impacto direto no desempenho
- Busca no vetor pelo menor contador

Experimentos

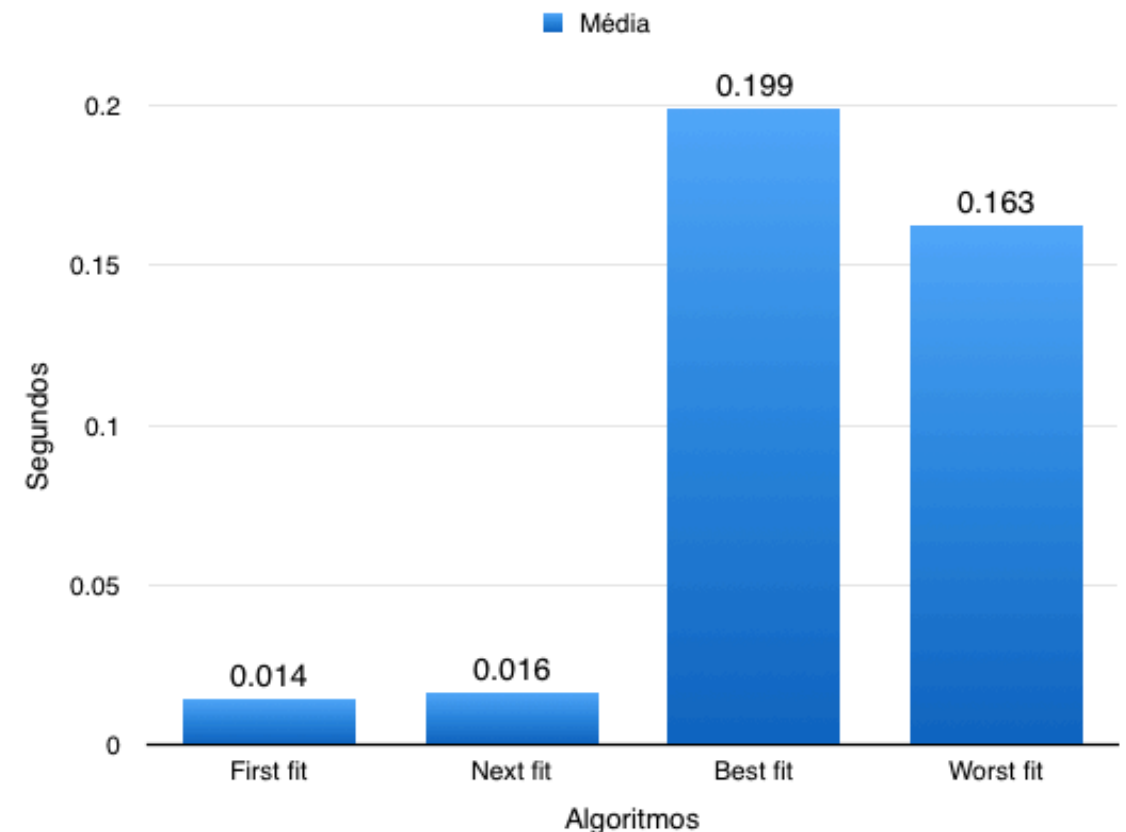
Nosso trace

- Tentamos gerar um trace que não tivesse viés
- Para o nosso tamanho de trace, zeramos o bit R a cada 32 instantes
- Outros traces podem apresentar resultados drasticamente diferentes
- 30 experimentos para o tempo médio
- macOS com i5 @ 1.5 ghz

Resultados



Falhas de página



Tempo médio de alocação

IC 0.95

First fit: 0.0007

Next fit: 0.0024

Best fit: 0.0155

Worst fit: 0.0083