

MAC0422 – 2016/2

Sistemas Operacionais

Exercício-programa 3
Profº Daniel Macêdo Batista

Vítor Kei Taira Tamada – 8516250
Luiz Felipe Moumdjian Giroto - 8941189

Estrutura do programa

process

table

Função para cada algoritmo

Funções auxiliares

Structs

process

Cada processo é um struct que armazena diversas informações relevantes para o uso dos algoritmos de gerenciamento de espaço livre e de substituição de página

table

A tabela de páginas é um struct com três vetores: um para o bit de presente/ausente, um para o endereço da página e um para o bit R

Funções

Gerenciamento de espaço livre

firstFit()

nextFit()

bestFit()

worstFit()

Substituição de página

optimal()

secondChance()

clk()

LRU4()

Auxiliares

executa()

writeMem()

freeProc()

fillPageTable()

pageFit()

binToDec()

clearR()

printPhyVec()

printVirVec()

printPageTable()

printMem()

Algoritmos de gerenciamento de espaço

Variáveis globais:

vector<bool> virVec – bitmap da memória virtual

fstream virMem – arquivo /tmp/ep3.vir

vector<process> procVec – vetor de processos

firstFit():

Nenhuma variável específica para este algoritmo

Busca no virVec o primeiro espaço, a partir do começo, em que o processo cabe e coloca-o nele, realizando a operação análoga em virMem

Algoritmos de gerenciamento de espaço

`nextFit()`:

`int nextPos` – armazena a próxima posição da memória virtual que será verificada (variável global)

Busca o primeiro espaço no `virVec` que o processo cabe a partir da posição determinada por `nextPos`. `nextPos` percorre `virVec` circularmente

`bestFit()` & `worstFit()`:

`vector<int> gapSize` – armazena o tamanho de cada espaço livre da memória (variável local)

`vector<int> gapIni` – armazena a posição inicial de cada espaço livre da memória (variável local)

Armazena o tamanho e o início de todos os espaços da memória virtual que o processo cabe e, após encontrar todos, aloca o processo conforme o algoritmo (menor espaço para o `bestFit()` e maior para o `worstFit()`)

Algoritmos de substituição de página

Variáveis globais:

`fstream phyMem` – arquivo `/tmp/ep3.mem`

`vector<bool> phyVec` – bitmap da memória física

`vector<process> procVec` – vetor de processos

`table pageTable` – tabela de páginas

`pageFit()`:

Antes de executar um algoritmo de substituição de página, o programa verifica se há espaço livre na memória física para alocar a página. Quando uma página é alocada desta forma, as estruturas próprias de cada algoritmo já são preenchidas adequadamente.

Algoritmos de substituição de página

`optimal()`:

`vector<int> acc_freq` – vetor de frequência de acessos das páginas de um processo. Cada espaço do vetor representa o número de acessos restantes da respectiva página (variável do struct process)

O algoritmo se baseia neste vetor para escolher qual página remover: escolhe a página cujo número de acessos futuros é o maior possível.

Algoritmos de substituição de página

`secondChance()`:

`queue<int> scq` – fila que recebe as páginas presentes na memória física (variável global)

Quando uma página precisa ser substituída pelo critério do algoritmo Segunda Chance, primeiro remove a página da memória física, em seguida, aloca a página que foi acessada. As alterações na tabela de páginas e na fila do algoritmo são realizadas de acordo em cada um desses momentos.

`clk()`:

`vector<int> clkv` – Lista circular que recebe as páginas presentes na memória física (variável global)

Funciona da mesma forma que o algoritmo Segunda Chance, mas utiliza uma lista circular ao invés de uma fila.

Algoritmos de substituição de página

LRU4 ():

`vector<list<bool>> age` – Vetor de listas que armazena a “idade” de uma página. A idade é armazenada em um número em representação binária de 8 bits (variável do struct process)

O algoritmo LRU4 (*least recently used* 4ª versão) remove a página com menor idade de acordo com o vetor age.

Funções auxiliares

`executa()` - Executa o comando `executa` quando inserido no prompt

`writeMem()` - Escreve na memória física (`ep3.mem`) ou virtual (`ep3.vir`)

`freeProc()` - Quando um processo acaba, remove-o da tabela de páginas, da memória virtual, dos bitmaps e da memória física se algumas de suas páginas estiver nela

`fillPageTable()` - Insere as páginas de um processo na tabela de páginas. Chamada apenas quando o processo chega

`pageFit()` - Insere uma página de processo na memória física caso haja espaço para isso

`binToDec()` - Converte um número binário em sua representação em decimal. Utilizado no algoritmo *least recently used*

Funções auxiliares

`clearR()` - Zera os bits R da tabela de páginas. Chamado periodicamente

`printPhyVec()` - Imprime o bitmap da memória física

`printVirVec()` - Imprime o bitmap da memória virtual

`printPageTable()` - Imprime a tabela de páginas

`printMem()` - Imprime o conteúdo da memória física (`ep3.mem`) ou da memória virtual (`ep3.vir`)

Resultados dos testes

Especificações do computador utilizado para realizar os testes:

Intel Core i3-3227U CPU 1.90GHz x 4

ubuntu 14.04 LTS 64-bit

3.7 GB de RAM

Arquivo de trace:

1000 processos

32 testes para cada algoritmo de gerenciamento de espaço livre

8 testes para cada algoritmo de substituição de página

Resultados dos testes

Page faults:

O número de page faults de um algoritmo de substituição de páginas não muda para um mesmo arquivo de trace, pois cada acesso de cada página é sempre feito no mesmo instante

Optimal: 2871

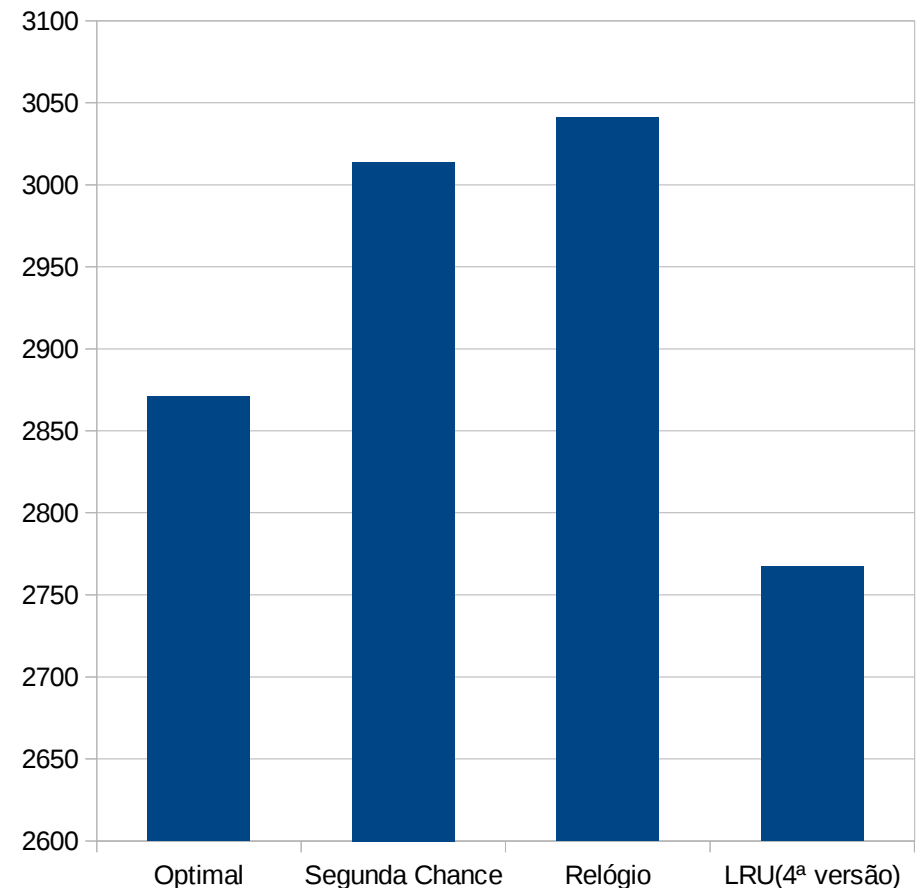
Segunda Chance: 3014

Relógio: 3041

LRU (quarta versão): 2767

Eixo x: algoritmo de substituição de páginas

Eixo y: número de page faults



Resultados dos testes

Gerência de espaço livre:

First fit: [0.0406; 0.0526]

Next fit: [0.0359; 0.0422]

Best fit: [0.1294; 0.1451]

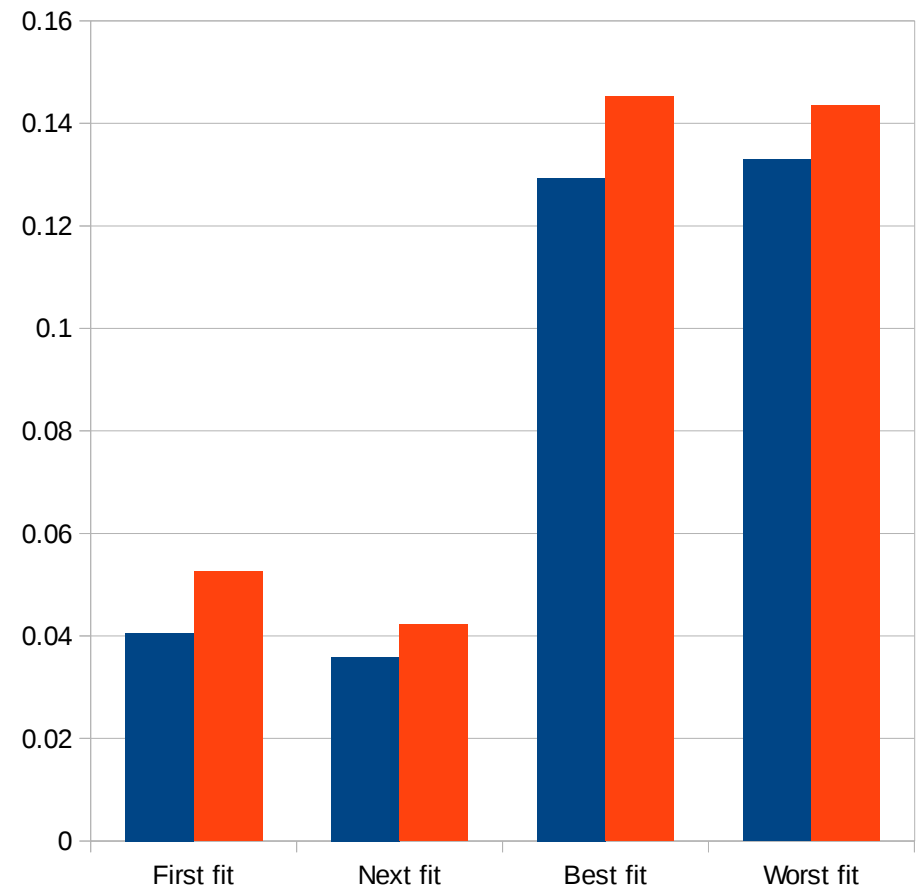
Worst fit: [0.1329; 0.1435]

First fit e Next fit tem intervalos de confiança semelhantes, pois seus funcionamentos são parecidos. O mesmo vale para Best fit e Worst fit entre si.

First fit e Next fit mostraram-se mais rápidos, pois não buscam e comparam todos os espaços que o processo cabe para fazer a alocação.

Eixo x: algoritmo de gerenciamento de espaço

Eixo y: tempo em segundos



FIM