

Simulador de LRU

Autores

- I. Pedro Pillon Vanzella <pedro@pedrovanzella.com> Cartão 213990

Plataforma de Desenvolvimento

O projeto foi desenvolvido e testado em um Mac, rodando OSX Lion, um sistema Unix baseado no BSD. O processador é um Intel Core i7 de 4 núcleos, com 8 threads por HT. Foi compilado tanto com o clang 3.1 quanto com o gcc 4.2.1.

Estruturas de dados

- I. A tabela de páginas é uma estrutura que contém um inteiro referente ao tamanho da mesma, um vetor de estruturas de página e um contador, que serve de índice para o vetor de páginas. Quando ele chega ao fim (o tamanho da estrutura), o algoritmo o zera. Foi escolhido por ser a forma mais simples de resolver o problema, eliminando possíveis bugs de referências a ponteiros. A desvantagem desta solução é o uso de memória ligeiramente maior, além de um tamanho máximo para a lista (que pode ser modificado no header include/memory.h)
- II. Cada página tem os seguintes campos:
 - process_id: a ID do processo dono da página
 - page_id: a ID da página dentro do processo
 - referenciada: flag que indica se a página foi referenciada
 - suja: flag que indica se a página sofreu escrita recentemente
 - not_in_use_anymore: flag que indica que o processo dono da página já terminou e, portanto, a página pode ser removida da lista.
 - acessos: quantos acessos a página sofreu
 - page_faults: quantos page_faults foram gerados ao acessar a página
 - substituicoes: quantas vezes a página foi retirada da memória

Algoritmo

O algoritmo percorre a lista, primeiramente verificando se a posição atual no relógio de páginas está disponível (nunca foi utilizado ou contém uma página de processo morto). Em seguida, ele verifica se a flag de referência está setada. Caso esteja, limpa a flag e chama o algoritmo recursivamente. Caso não esteja, substitui a página.

O que funciona

Dados os arquivos de teste, tudo está funcionando corretamente.

O que não funciona

A flag de 'sujo' nunca é checada. Parte-se do princípio que, caso a página devesse ser substituída, ao escrever de volta no swap, esta flag seria checada e a página corretamente reescrita. Não pareceu estar no escopo do trabalho tratar este caso.

Metodologia de Desenvolvimento

O projeto foi desenvolvido top-down: a função `main()` foi escrita, chamando funções que não existiam ainda, e estas foram sendo escritas conforme se mostravam necessárias. Isso permitiu uma API mais limpa e clara, com uma boa divisão de responsabilidades entre os módulos e funções. Para debugar o projeto foram utilizadas chamadas de `printf()` estrategicamente posicionadas, algumas delas mantidas na versão final. O algoritmo de substituição, por exemplo, reporta o que está fazendo, e as decisões sendo tomadas.

Nos arquivos de teste utilizados foi tentado prever edge cases, como uma memória muito pequena, acessar páginas de processos que não existem e acessar páginas inexistentes de processos. Com estes testes, o projeto parece estar funcionando corretamente.

Dificuldades

O projeto não apresentou grandes dificuldades. O parser dos arquivos de testes foi escrito com extensões do GNU para o tratamento de strings, principalmente a chamada `getline()`, que lê um arquivo de texto linha por linha e corretamente realoca o buffer caso ela seja maior que o buffer original.

Inicialmente foi tentado utilizar uma lista encadeada de páginas e processos, mas isto se demonstrou muito complicado, dada a alternativa e o custo da mesma. Para ambas estruturas utilizou-se vetores, com tamanhos máximos definidos em seus respectivos headers.

Exemplo de saída do programa

```
niflheim SisOp/lrusimul <master*> » bin/lrusimul testes/tester.txt
```

```
[+] MEMSIZE 4
[+] PROCSIZE 1 IO
[+] PROCSIZE 2 IO
[+] PROCSIZE 3 IO
[+] PROCSIZE 4 IO
[+] READ 1 1
    [+] substitute: [page (1)] [id (1)]
    [+] substitute: Empty position at [0]
[+] READ 2 1
    [+] substitute: [page (2)] [id (1)]
    [+] substitute: Empty position at [1]
[+] READ 1 3
    [+] substitute: [page (1)] [id (3)]
    [+] substitute: Empty position at [2]
```

[+] WRITE 1 1

[+] READ 3 1

[+] substitute: [page (3)] [id (1)]

[+] substitute: Empty position at [3]

[+] READ 3 3

[+] substitute: [page (3)] [id (3)]

[+] substitute: Page's referenced bit set. Trying next.

[+] substitute: [page (3)] [id (3)]

[+] substitute: Page's referenced bit set. Trying next.

[+] substitute: [page (3)] [id (3)]

[+] substitute: Page's referenced bit set. Trying next.

[+] substitute: [page (3)] [id (3)]

[+] substitute: Page's referenced bit set. Trying next.

[+] substitute: [page (3)] [id (3)]

[+] substitute: removing page from memory: [page (1)] [id (1)]

[+] READ 2 3

[+] substitute: [page (2)] [id (3)]

[+] substitute: removing page from memory: [page (3)] [id (3)]

[+] WRITE 1 2

[+] substitute: [page (1)] [id (2)]

[+] substitute: removing page from memory: [page (2)] [id (3)]

[+] WRITE 1 1

[+] substitute: [page (1)] [id (1)]

[+] substitute: removing page from memory: [page (1)] [id (2)]

[+] READ 2 1

niflheim SisOp/lrusimul <master*> » cat perf/log.txt

PROCESSO 1

Página Acessos (R/W)			NroPageFault	NroSubst
0	0	0	0	
1	3	2	1	
2	2	1	0	
3	1	1	0	
4	0	0	0	
5	0	0	0	
6	0	0	0	
7	0	0	0	

8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

PROCESSO₂

Página	Acessos (R/W)	NroPageFault	NroSubst
0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1	I	I	I
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

PROCESSO₃

Página	Acessos (R/W)	NroPageFault	NroSubst
0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1	I	I	<input type="radio"/>
2	I	I	I
3	I	I	I
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

PROCESSO₄

Página	Acessos (R/W)	NroPageFault	NroSubst
0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

niflheim SisOp/lrusimul <master*> »