



SisOS - SIMULADOR DE ALGORITMOS DE SUBSTITUIÇÃO DE PÁGINAS EM GERENCIAMENTO DE MEMÓRIA

Renan Rabelo Soeiro – renan.ifce@gmail.com

Jones Monteiro Jacinto – jonesmonteirojacinto@gmail.com

Maurício Barros de Almeida Neto – mauricio.ifce.br@gmail.com

Fillipe Diego Ferreira Carneiro – fillipediego@gmail.com

Hugaleno da Costa Bezerra – hugaleno@gmail.com

José Alexandre de Castro Bezerra Filho – j.alexandre.castro@gmail.com

Grupo de Desenvolvimento de Sist. de Telecomunicações e Embarcados - GDEST

Avenida 13 de Maio, 2081

60040-531 – Fortaleza – CE

Carlos Maurício Jaborandy de Mattos Dourado Júnior – cmauriciojd@gmail.com

Jose Wally Mendonça Menezes – wally@ifce.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Avenida 13 de Maio, 2081

60040-531 – Fortaleza – CE

***Resumo:** Este trabalho apresenta o desenvolvimento do simulador SisOS, um simulador de algoritmos de substituição de página com a finalidade de auxiliar alunos e professores nas cadeiras de sistemas operacionais e disciplinas de gerenciamento de memória. Para facilitar as aulas ministradas por professores nesta área, estamos oferecendo um sistema que pode simular estes algoritmos ajudando na compreensão do conteúdo pelos alunos.*

***Palavras-chave:** Sistemas Operacionais, Simulador, Algoritmos de Substituição, Paginação.*



1. INTRODUÇÃO

A ideia inicial deste trabalho veio a partir dos alunos que cursam a disciplina de Sistemas Operacionais, tendo em vista que a lógica desses algoritmos apresentam certas peculiaridades e que seu funcionamento exige certo grau de abstração e interpretação de acordo com a solução proposta pelos algoritmos.

A partir disso, foi iniciado a implementação do SisOS (Simulador de algoritmos de substituição de páginas), com o intuito de auxiliar alunos no estudo de Sistemas Operacionais, ilustrando melhor o funcionamento de cada algoritmo.

Para o desenvolvimento, foi utilizado a plataforma JSE 7.21 (Java Standard Edition), plataforma capaz de desenvolver aplicativos para desktops. Ela foi escolhida por uma de suas principais vantagens, a portabilidade, posto que sua linguagem é interpretada por uma JVM (Java Virtual Machine), então basta que o sistema operacional a possua para que a aplicação seja executada, podendo assim rodar no sistema operacional Windows, GNU/Linux ou Mac. [3]

2. ALGORITMOS SIMULADOS

Na primeira versão do simulador, foram implementados os algoritmos FIFO, SC, NRU e o Clock. Nesta seção, será apresentado uma noção básica dos métodos de substituição de cada algoritmo.

2.1. FIFO (First-in, First-out)

No algoritmo FIFO, a página que foi carregada a mais tempo será substituída primeiro, porque esta página está no final da fila. Como o próprio nome sugere, a primeira página a entrar na fila é a primeira a sair. Além disso, na ocorrência de falta de páginas, então o algoritmo remove a página mais antiga e a nova página é adicionada no final da fila.

Esse algoritmo possui a desvantagem no tempo de acesso das páginas, caso elas estejam a muito tempo na fila podem ser usadas mais vezes, em detrimento da mais recentes.

2.2. SC (Second Chance)

No algoritmo Second Chance, as páginas são ordenadas numa Fila e cada página possui um bit de referência, o qual se inicia com o valor 1. Após um certo período de tempo todas as páginas terão seus bits de referência mudados para zero, retornando ao valor 1 somente quando a página for referenciada novamente.

Na necessidade de substituição, se a página no final da fila possuir seu bit de referência em 1, ela não será substituída e será colocada no início da fila com seu bit de referência em 0, caso contrário ela será substituída.

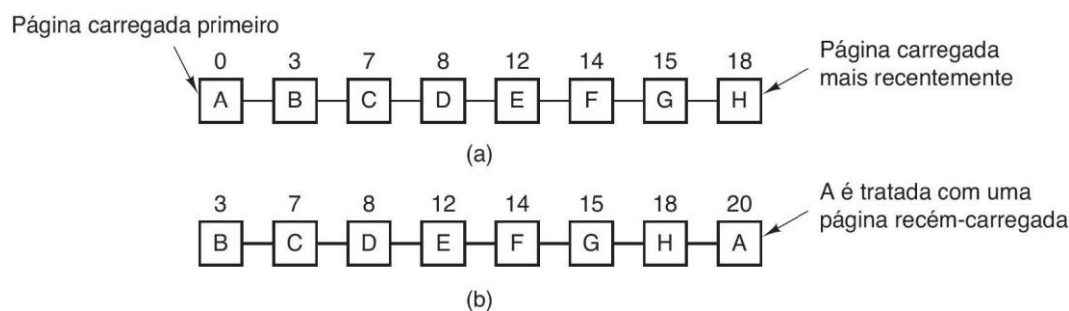


Figura 1 - Fig. 1. [1] Operação de segunda chance. (a) Páginas na ordem FIFO. (b) Lista de páginas se uma falta de página ocorre no tempo 20 e o bit R de A possui o valor 1. Os números acima das páginas são o tempo de carregamento.

2.3. NRU (Not Recently Used)

Esse Algoritmo possui dois bits, um para referência (R) e um para modificação (M), ambos assumem o valor 1 quando ocorre essas ações. A partir disso, as páginas são divididas em quatro classes de acordo com o valor dos bits R e M.

Tabela 1 – Relação entre classes e Bits.

| Classe | bit R | bit M |
|----------|-------|-------|
| Classe 0 | 0 | 0 |
| Classe 1 | 0 | 1 |
| Classe 2 | 1 | 0 |
| Classe 3 | 1 | 1 |

Cada página inicia com seu bit de referência com o valor 1, que será alterado para 0 após um tempo pré-determinado, e seu bit de modificação com o valor 0.

Na falta de páginas o algoritmo faz uma varredura nos bits de todas as páginas seguindo uma ordem de prioridade, serão substituídas primeiro: páginas com bits de referência e modificação com o valor 0 (R=0 e M=0), páginas não referenciadas (R=0 e M=1), páginas não modificadas (R=1 e M=0) e, em último caso, páginas referenciadas e modificadas (R=1 e M=1).

2.4. CLOCK (Relógio)

Cria uma lista circular e, como no SC, utiliza bits de referência. Na necessidade de substituição, o algoritmo verifica a página na qual está sendo apontada, caso seu bit de referência tenha valor igual a 1, a página não é substituída e tem seu bit de referência alterado para 0, caso contrário ela será substituída, o algoritmo aponta para a próxima página. Lembrando o funcionamento de um relógio, como mostrado na “Fig. 2”.

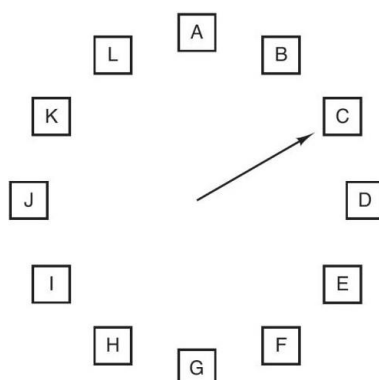


Figura 2 - Fig. 2. [1] Algoritmo de substituição de página Relógio.

3. SIMULADOR

Nesta secção, será dado enfoque ao simulador, tanto por sua parte gráfica quanto por seu funcionamento. Sua janela inicial é dividida em dois painéis, o painel “Configurações” e o painel “Simulação”, elaborados de maneira a deixar seu uso mais intuitivo.

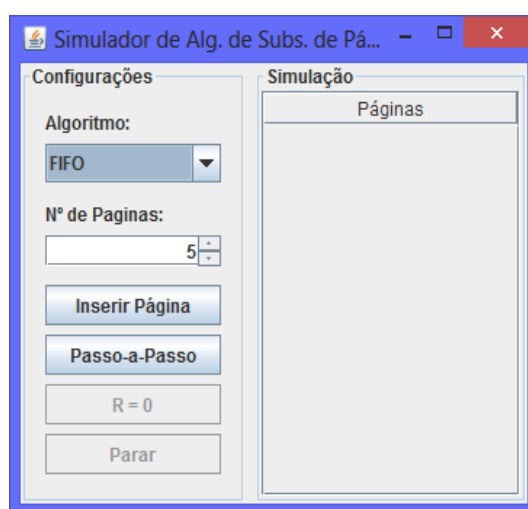


Figura 3 – Tela inicial do simulador.

No painel “Configurações” o usuário seleciona qual algoritmo que será usado e quantas páginas sua simulação deve possuir. Para dar início a simulação é necessário clicar no botão “Inserir Página” ou no botão “Passo-a-Passo”, posteriormente será explicado melhor a diferença entre essas duas funções.

Para alterar os bits de referência deve se pressionar o botão “R = 0”, alterando todos os bits de referência para o valor 0, simulando o tempo limite utilizados nos algoritmos SC, NRU e Clock. O usuário poderá alterar o bit de modificação e de referência para o valor 1 clicando na linha da página.

Para a página ser modificada ela tem que ser referenciada, por isso são modificados os dois bits para o valor 1. Assim que o número de páginas atingir o limite de páginas definidos pelo usuário, será possível visualizar o mecanismo do algoritmo utilizado.

No painel “Simulação”, o usuário pode observar o decorrer do teste observando a lista de páginas da simulação. Cada página possui um identificador (Id), um bit R (referência) e um bit M (modificação).

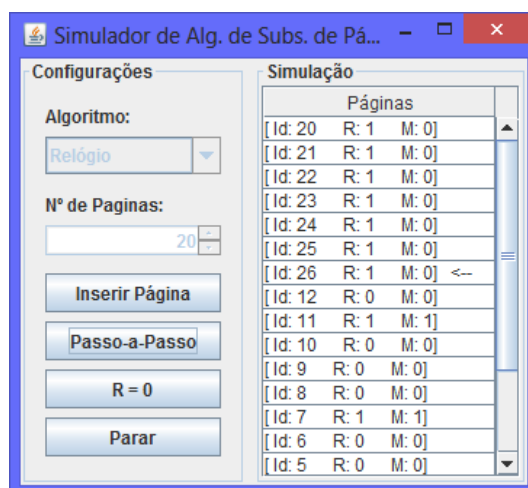


Figura 4 – Simulação do algoritmo relógio.

Para finalizar a simulação o usuário deve clicar no botão “Parar”, para assim poder dar início a uma nova simulação.

4. FUNÇÕES “INSERIR PÁGINA” E “PASSO-A-PASSO”

Ambas as funções são utilizadas para se obter o resultado final da execução desses algoritmos, no entanto há uma diferenciação na maneira de exibir o resultado para o usuário do sistema.

Ao pressionar o botão “Inserir Página” o sistema mostra a lista de páginas já com a substituição, portanto há um resultado final sem a interação com os ponteiros de referência e de modificação. Já ao pressionar o botão “Passo-a-Passo” o sistema mostra a página a qual está sendo apontada pelo ponteiro, mostra os bits que foram alterados e, ao final, mostra a lista de páginas realizando a substituição.

Dessa forma, a função “Passo-a-Passo” é capaz de mostrar mais a fundo o funcionamento de cada algoritmo, auxiliando na aprendizagem de alunos, enquanto a função “Inserir Página” retorna o resultado da substituição de forma rápida, auxiliando na resolução de exercícios.

5. SIMULAÇÕES

Para um melhor entendimento de sua aplicação será mostrado um exemplo onde existe a necessidade de substituição de páginas e será mostrado o resultado obtido para cada algoritmo simulado.

| Simulação | | | |
|-----------|------|-------|---|
| Páginas | | | |
| [Id: 4 | R: 1 | M: 0] | ← |
| [Id: 3 | R: 1 | M: 1] | |
| [Id: 2 | R: 0 | M: 1] | |
| [Id: 1 | R: 0 | M: 0] | |
| [Id: 0 | R: 0 | M: 1] | |
| | | | |

Figura 5 - Lista de páginas utilizada para o exemplo de simulação.

FIFO (First-in, First-out)

A página a ser substituída será a de Id igual a 0, pois ela se encontra no final da fila.

| Simulação | | | |
|-----------|------|-------|---|
| Páginas | | | |
| [Id: 4 | R: 1 | M: 0] | ← |
| [Id: 3 | R: 1 | M: 1] | |
| [Id: 2 | R: 0 | M: 1] | |
| [Id: 1 | R: 0 | M: 0] | |
| [Id: 5 | R: 1 | M: 0] | |
| | | | |

Figura 6 - Resultado da simulação do algoritmo FIFO.

SC (Second Chance)

A página a ser substituída será a de Id igual a 0, pois além de se encontrar no final da fila seu bit de referência possui valor igual a zero, caso contrário seu bit seria setado para 1 e a página retornaria para o início da fila.

| Simulação | | | |
|-----------|------|-------|-----|
| Páginas | | | |
| [Id: 4 | R: 1 | M: 0] | <-- |
| [Id: 3 | R: 1 | M: 1] | |
| [Id: 2 | R: 0 | M: 1] | |
| [Id: 1 | R: 0 | M: 0] | |
| [Id: 5 | R: 1 | M: 0] | |
| | | | |

Figura 7 - Resultado da simulação do algoritmo SC.

NRU (Not Recently Used)

A página a ser substituída será a de Id igual a 1, pois dentre as páginas presentes na fila esta é a de menor prioridade para o algoritmo, tendo em vista que seu bit de referência e de modificação possuem valor igual a 0.

| Simulação | | | |
|-----------|------|-------|-----|
| Páginas | | | |
| [Id: 4 | R: 1 | M: 0] | <-- |
| [Id: 3 | R: 1 | M: 1] | |
| [Id: 2 | R: 0 | M: 1] | |
| [Id: 5 | R: 1 | M: 0] | |
| [Id: 0 | R: 0 | M: 1] | |
| | | | |

Figura 8 - Resultado da simulação do algoritmo NRU.

Clock (Relógio)

Como o ponteiro está apontando para a página de Id igual a 4, então a página a ser substituída será a de Id igual a 2, pois é a primeira página não referenciada após o ponteiro.

As páginas de Id igual a 4 e 3 terão seus bits de referência setado para o valor 0, já que elas foram apontadas e não foram substituídas.

| Simulação | | | |
|-----------|------|-------|-----|
| Páginas | | | |
| [Id: 4 | R: 0 | M: 0] | |
| [Id: 3 | R: 0 | M: 1] | |
| [Id: 5 | R: 1 | M: 0] | |
| [Id: 1 | R: 0 | M: 0] | <-- |
| [Id: 0 | R: 0 | M: 1] | |

Figura 9 - Resultado da simulação do algoritmo Clock.

6. OUTROS SIMULADORES

Existem outros trabalhos e simuladores na mesma área, porém cada um com suas peculiaridades. Será mostrado um pouco sobre alguns desses simuladores desenvolvidos.

P.R.A - Page Replacement Algorithms[6]

Desenvolvido por Ting Yu do Departamento de Ciências da Computação de Illinois, o software foi desenvolvido para a plataforma web tendo como base para as suas simulações os algoritmos FIFO, LRU e Optimal (Ótimo - onde a melhor página a remover da memória em um dado instante é aquela que ficará mais tempo sem ser usada pelos processos).

System Configuration:

Page Replacement Algorithm:

Page Frame Number:

Simulation:

Page Refs:

| |
|---|
| A |
| B |
| C |
| D |
| A |
| B |
| E |
| A |
| B |
| C |
| D |
| E |

Page Contents:

| | | |
|---|---|---|
| E | B | A |
|---|---|---|

Hit! Page A is in memory

Page reference: 8
Page faults: 7

Issues:

FIFO Algorithm:
replace the page stays longest in memory
Note: Belady's Anomaly:
number of page faults increases with the increase of page frame number

| #Page Refs | #Page Frames | #Page Faults |
|------------|--------------|--------------|
| 12 | 3 | 9 |
| 12 | 4 | 10 |

Start Stop Step Cont

Figura 10 - Sistema “Page Replacement Algorithms”.

SamSol - Page Replacement Algorithm Simulation [7]

Desenvolvido pelo Engenheiro de Software Samir Solanki, o sistema foi feito utilizando a plataforma java, simulando os algoritmos FIFO e LRU.



Figura 11 - Sistema “SamSol - Page Replacement Algorithm Simulation”.

SimulaRSO – Sim. de Recursos de Sist. Operacionais [8]

Desenvolvido pelos Engenheiros André Luiz Vizine Pereira, André de Araújo Rodrigues e Caio Ribeiro Pereira, da Universidade Católica de Santos, o sistema foi desenvolvido para a plataforma web, simulando os algoritmos FIFO, Optimal, LRU e MRU.

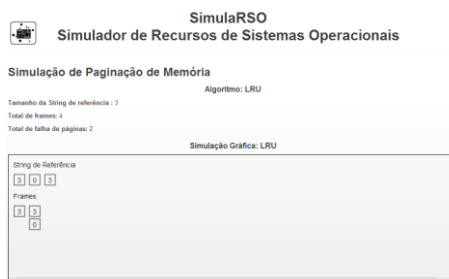


Figura 12 - Sistema “SimulaRSO – Simulador de Recursos de Sistemas Operacionais”.

7. COMPARAÇÃO ENTRE OS SIMULADORES

Tabela 2 – Quanto aos Algoritmos

| ALGORITMOS/SIM. | SISOS | P.R.A | SAMSOL | SIMULARSO |
|-----------------|-------|-------|--------|-----------|
| FIFO | X | X | X | X |
| SC | X | | | |
| OPTIMAL | | X | | X |
| LRU | X | X | X | X |
| MRU | X | | | X |

Tabela 3 – Quanto a Plataforma

| PLATAFORMA/SIM. | SISOS | P.R.A | SAMSOL | SIMULARSO |
|-----------------|-------|-------|--------|-----------|
| JAVA | X | | X | |
| WEB | | X | | X |

Tabela 4 – Quanto as Funções

| FUNÇÕES/SIM. | SISOS | P.R.A | SAMSOL | SIMULARSO |
|------------------|-------|-------|--------|-----------|
| EXIBIR RESULTADO | X | X | X | X |
| PASSO-A-PASSO | X | X | | |
| ANIMAÇÃO | | X | | X |

Tabela 5 – Pontos Positivos

| SISOS | P.R.A | SAMSOL | SIMULARSO |
|--|-----------------------------------|---|---------------------------|
| QUANTIDADE DE ALGORITMOS E APRESENTA A FUNÇÃO PASSO-A-PASSO. | APRESENTA A FUNÇÃO PASSO-A-PASSO. | MOSTRA OS VALORES DE PAGE HITS E PAGE FAULTS. | QUANTIDADE DE ALGORITMOS. |

Tabela 6 – Pontos Negativos

| SISOS | P.R.A | SAMSOL | SIMULARSO |
|--|---|---|---------------------------------------|
| NÃO APRESENTA O ALGORITMO OPTMAL E POSSUI UMA INTERFACE POUCO ROBUSTA. | POUCOS ALGORITMOS E SIMULA ATÉ 4 PÁGINAS. | POUCOS ALGORITMOS E NÃO APRESENTA A FUNÇÃO PASSO-A-PASSO. | NÃO APRESENTA A FUNÇÃO PASSO-A-PASSO. |

8. SUGESTÕES FUTURAS

Para futuras versões do simulador o usuário pode esperar a implementação dos algoritmos LRU (Least Recently Used), MRU (Most Recently Used), LFU (Least Frequently Used) e MFU (Most Recently Used).

Também está nos planos a melhoria de sua interface gráfica, pois para a primeira versão a prioridade do simulador era mostrar o mecanismo dos algoritmos de forma correta e não uma interface robusta.

9. CONSIDERAÇÕES FINAIS

Ao final deste artigo, pode-se perceber como um simulador deste tipo pode auxiliar no ensino e na aprendizagem de assuntos relacionados a sistemas operacionais e gerenciamento de memória, mostrando de forma simples e clara o funcionamento dos principais algoritmos de substituição de páginas.

Agradecimentos

Agradeço ao IFCE (Instituto Federal de Educação, Ciência e Tecnologia do Ceará) e ao laboratório GDESTE (Grupo de Desenvolvimento de Sistemas de Telecomunicações e Sistemas Embarcados), os quais me deram suporte no desenvolvimento desse artigo.

REFERÊNCIAS BIBLIOGRÁFICAS

Livros:

- [1] Andrew S. Tanenbaum, Sistemas Operacionais Modernos, Ed. Prentice Hall - Br, 3ª Ed., 2010, pp. 122-133.
- [2] Silberschatz, Abraham; Galvin, Peter Baer, Sistemas Operacionais com Java, Ed. Campus, 7ª Ed., 2008, pp. 238-274.
- [3] Sierra, Kathy; Bates, Bert, Use a Cabeça!: Java, Ed. Alta Books, 2ª Ed., 2007, 470p.
- [4] Fraizer, Colin; Bond, Jill, API Java: manual de referência, Ed. Makron, 1997, 371p.

Artigos de jornais:

- [5] Kumari, Mukesh; **SIMULATION OF LRU PAGE REPLACEMENT ALGORITHM FOR IMPROVING PERFORMANCE OF SYSTEM**, International Journal of Computer Applications in Technology vol. 4, 2013, ISSN:2229-6093.

Internet:

- [6] Solanki, Samir; **P.R.A PAGE REPLACEMENT ALGORITHM**, 1996. Disponível: <http://www.cs.uiuc.edu/class/sp06/cs241/Animations/PageReplac/>.
- [7] Yu, Ting; SamSol - **PAGE REPLACEMENT ALGORITHM SIMULATION**, 2008. Disponível: <http://www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=5924&lngWId=2>.
- [8] Pereira, André; **SimulaRSO – SIMULADOR DE RECURSOS DE SISTEMAS OPERACIONAIS**, 2011. Disponível: <http://www.simula-rso.appspot.com/>.



SisOS - SIMULATOR REPLACEMENT ALGORITHMS PAGES IN MEMORY MANAGEMENT

Abstract: *This paper presents the development of SisOS simulator, a page replacement algorithms simulator with the purpose of assisting students and teachers in the operating systems and memory management subjects. To facilitate the classes taught by teachers in this area, we are offering a system that can simulate these algorithms and helping in the understanding of the content by the students.*

Key-words: *Operating Systems, Simulation Systems, Replacement Algorithm, Paging.*