

Um Estudo Comparativo de Simuladores de Rotinas de Sistemas Operacionais para Auxílio às Aulas Teóricas

A Comparative Study of Operating System Routine Simulators to Assist Lecture Classes

Recebimento dos originais: 01/03/2023

Aceitação para publicação: 05/04/2023

Adilson Oliveira Cruz

Mestre em Informática

Instituição: Libera Università di Bolzano

Endereço: Rua Professor Belmiro Siqueira, 85, Ed. Victoria Bay, Torre 2, Enseada do Suá, Vitória – ES, CEP: 29050-580

E-mail: adilson.cruz@ifes.edu.br

Átila Alves de Jesus

Bacharel em Sistemas de Informação

Instituição: Centro Universitário Salesiano

Endereço: Av. Vitória, 950, Forte São João, Vitória – ES, CEP: 29017-950

E-mail: atilaalvesj@gmail.com

Fabio Pio Marques Filho

Bacharel em Sistemas de Informação

Instituição: Centro Universitário Salesiano

Endereço: Av. Vitória, 950, Forte São João, Vitória – ES, CEP: 29017-950

E-mail: contato@fabio-filho.com

RESUMO

A disciplina de Sistemas Operacionais trata de assuntos que, muitas vezes, parecem abstratos à primeira vista. É possível minimizar a dificuldade no aprendizado da disciplina utilizando simuladores para os conceitos fundamentais. Este artigo apresenta uma introdução aos principais tópicos dessa disciplina (gerenciamento de processos, gerenciamento de memória e sistemas de arquivos) e, em seguida, apresenta e compara os principais simuladores utilizados pela academia no aprendizado da disciplina de Sistemas Operacionais. Futuramente, espera-se poder realizar estudos sobre o desenvolvimento desses simuladores, norteando, se possível, futuras soluções de simulação de rotinas de sistemas operacionais.

Palavras-chave: simuladores de rotinas sistemas operacionais, gerenciamento de processos, gerenciamento de memória, sistemas de arquivos.

ABSTRACT

The Operating Systems course touches on subjects that sometimes seem abstract at first sight. It's possible minimizing the difficulties learning these concepts through simulators of operating systems routines. This paper will show an introduction to the main subjects of this course – processing management, memory management and file systems – and introduce and compare the main simulators of operating system routines used by the academy during the Operating Systems course. We intend, in the future, to study about the operating systems simulators development, to start, if possible, future solutions of operating systems simulations.

Keywords: operating system routine simulators, process management, memory management, file systems.

1 INTRODUÇÃO

Diversos cursos integram à sua grade curricular a disciplina de Sistemas Operacionais, pois entender o funcionamento dos mesmos é pré-requisito fundamental para quem estuda ou trabalha com tecnologia.

Um desenvolvedor de *software*, por exemplo, ao desenvolver sua aplicação, para que a mesma seja estável e produtiva, precisa entender as rotinas dos sistemas operacionais, para que tal programador possa orientar seu desenvolvimento à tecnologia que utilizará. Porém, a disciplina de Sistemas Operacionais, muitas vezes, trata de assuntos abstratos e que podem confundir um iniciante.

Uma das grandes vantagens do uso de simuladores é a maneira facilitada de apresentar conceitos complexos e abstratos utilizados nos Sistemas Operacionais. Os simuladores apresentam esses conceitos de forma gráfica e simplificada, melhorando a didática e facilitando o processo de ensino. Além disso, após ter contato com esses conceitos abstratos, o aluno consegue de forma autônoma identificá-los ao utilizar no dia-a-dia os próprios Sistemas Operacionais, unindo teoria à prática.

Outra vantagem do uso de simuladores é percebida na medida em que os mesmos permitem um método ativo e interativo de ensino-aprendizagem. Um grande desafio do uso de métodos ativos de aprendizagem é a maneira de apresentar os conceitos aos alunos, mas, como citado no parágrafo anterior, a simulação desses conceitos adapta-se bem como uma estratégia de desenvolvimento de conteúdo.



2 OBJETIVOS

De uma forma geral, este artigo pretende apresentar uma introdução básica ao funcionamento dos sistemas operacionais no gerenciamento de processos, memória e sistemas de arquivos, bem como a apresentação dos principais simuladores de rotinas de sistemas operacionais disponíveis para a comunidade acadêmica e um comparativo entre as principais funções das mesmas, com o objetivo de trazer uma abordagem intuitiva e facilitada a pesquisadores e alunos da disciplina de Sistemas Operacionais.

Para isso, será necessária a apresentação de:

- A introdução à disciplina de Sistemas Operacionais;
- O gerenciamento de processos, comentando o funcionamento de seus algoritmos;
- O gerenciamento de memória, comentando o funcionamento de seus algoritmos;
- Os sistemas de arquivos, listando os algoritmos de escalonamento de disco;
- Simuladores de rotinas de sistemas operacionais e comparativos de funcionalidades dos mesmos.

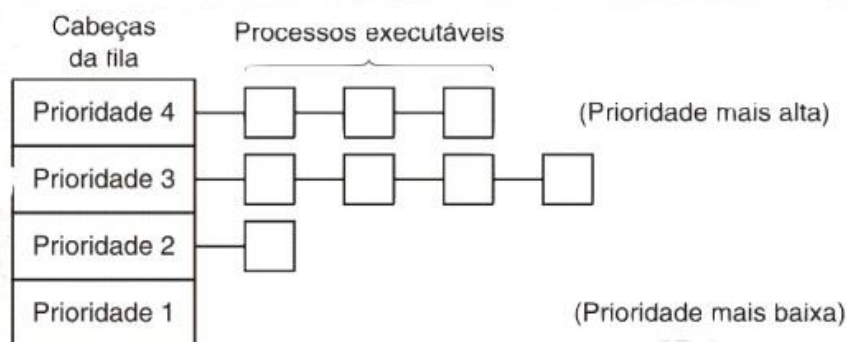
3 REFERENCIAL TEÓRICO

Para a compreensão da disciplina de Sistemas Operacionais, é necessário definir aspectos básicos. Para [TANENBAUM et al., 2000], o estudo dessa disciplina envolve a definição de processos, comunicações interprocessos, semáforos, *drivers*, gerenciamento de memória, etc. Para isso, serão definidos abaixo os principais entidades de gerenciamento dos sistemas operacionais.

3.1 GERENCIAMENTO DE PROCESSOS

Nos sistemas operacionais, um processo é a forma de representar um programa em execução. É o processo que utiliza os recursos do computador – processador, memória, etc. – para a realização das tarefas para as quais a máquina é destinada.

Figura 1 – esboço da execução de processos no Linux segundo sua prioridade. (Fonte: Oficina da Net)



Os sistemas operacionais modernos conseguem, através de um sistema de compartilhamento de tempo, fazer com que pareça que existem vários processos em execução simultânea, mas os mesmos, internamente, dividem juntos o tempo do processador. Para garantir a eficácia da execução simultânea, é necessário um bom conjunto de algoritmos de gerenciamento de processos. O escalonador de processos, presente no *kernel*, é responsável por gerenciar essa simultaneidade. O escalonador de processos, para realizar o gerenciamento, precisa conhecer o estado de cada processo, como se segue:

- TASK RUNNING – sendo executado ou aguardando o momento de execução;
- TASK INTERRUPTIBLE – processo pausado até que determinada condição seja atendida;
- TASK UNINTERRUPTIBLE – o que difere este do anterior é o fato de que o mesmo não muda seu estado ao receber resposta positiva da condição;
- TASK STOPPED – processo cuja execução foi parada;
- TASK ZOMBIE – processo finalizado cujo processo pai não realizou um *system-call* para retornar informações sobre o processo morto.

3.2 GERENCIAMENTO DE MEMÓRIA

A Memória Principal do computador é um recurso indispensável à multitarefa e às operações que o processador executa. Para tratar de todos os pormenores, os sistemas

operacionais contam com algoritmos de gerenciamento de memória. Estes monitoram recursos utilizados e disponíveis, alocam e disponibilizam a memória que os processos executam e gerenciam a troca de processos entre as memórias principal e secundária. O Linux gerencia a memória usando um algoritmo que caminha entre os blocos de memória em potências de dois. Entretanto, esse algoritmo gera fragmentações, pois, se algum processo solicitar um bloco de 65 páginas, seria retornado um bloco de 128 páginas. [DA SILVA, 2007]

Para reduzir o impacto dessas fragmentações, o Linux trabalha com uma segunda locação de memória que, após obter os blocos, retalha os mesmos em unidades menores para, então, realizar o gerenciamento. Há, também, um terceiro alocador usado quando a memória solicitada tiver de ser contígua somente virtualmente, não na memória física. [TANEMBAUM, 2000].

3. 3 SISTEMAS DE ARQUIVOS

Através dos sistemas de arquivos ocorrem a gravação e a recuperação dos dados nas mídias de armazenamento do computador. O sistema de arquivos é especificado pelo *software* do sistema operacional no momento da instalação do mesmo ou na instalação e configuração posterior de mídias de armazenamento. Atualmente, são suportados mais de vinte sistemas de arquivos pelo Linux, dos quais, os mais usados são o *ReiserFS* e o *ext3* [DA SILVA, 2007]. Novas versões do *Windows* utilizam versões cada vez mais recentes do sistema de arquivos NTFS.

Existem também diversos algoritmos de escalonamento de discos, como os que seguem [Illinois Institute of Technology]:

- First Come – First Serve (FCFS): todos os pedidos de uso da mídia são colocados no final da fila, ou seja, a utilização do disco ocorre na ordem das requisições;
- Shortest Seek Time First (SSTF): nesse caso, se houverem muitas requisições de acesso ao disco, as mesmas são atendidas de acordo com a proximidade;
- Elevator (SCAN): essa abordagem lembra o funcionamento de um elevador. O acesso às regiões do disco se dá em varreduras regulares. Embora este



método seja melhor que o anterior, ainda assim não é o mais eficiente;

- Circular Scan (C-SCAN): funciona exatamente como o anterior, entretanto sem a definição de extremos, o que torna o mesmo circular;
- C-LOOK: uma versão cujo algoritmo é ainda melhor do que o observado na abordagem anterior.

4 METODOLOGIA

Foram consultadas, para a construção deste artigo, diversas fontes – principalmente em portais de tecnologia e pesquisa científica, livros e comunidades acadêmicas – com o objetivo de apresentar informações atualizadas e coerentes a respeito do assunto.

Durante o processo de pesquisa, alguns simuladores foram testados, com o objetivo de perceber, na prática, os pontos fortes e as fragilidades dos mesmos.

Para facilitar a consulta de informações e detalhes dos simuladores abordados neste artigo, os dados colhidos foram diagramados numa tabela contendo, em suas dimensões:

- Nome dos simuladores;
- Simulações de gerenciamento de memória suportadas: no que diz respeito a alocação de paginação, *swapping* e substituição de páginas;
- Suporte a gerenciamento de processador;
- Suporte a gerenciamento de processos: políticas de escalonamento e sistemas de proteção;
- Linguagem de programação utilizada no desenvolvimento: com o objetivo de verificar a possibilidade de utilização ou *port* para outras plataformas;
- Principais características: informações que não puderam ser catalogadas nas demais colunas.

Vale ressaltar que nem todas as funções de alguns simuladores puderam ser observadas. Por isso, em alguns casos, tais campos foram dadas como indefinidas. Essas indefinições ocorreram devido a alguns motivos, dentre eles:

1. Houve indicações claras da ausência da simulação;



2. A documentação oficial pareceu confusa quanto à presença ou ausência da funcionalidade, ou;
3. A função não pôde ser observada durante os testes, ou;
4. O funcionamento da função teve resultado duvidoso.

O objetivo da tabela é servir como um suporte ao estudante que desejar conhecer detalhes a respeito dos simuladores considerados nesta abordagem.

5 RESULTADOS E DISCUSSÃO

A tabela a seguir compara os simuladores de rotinas de sistemas operacionais mais utilizados pela academia. Alguns desenvolvedores escreveram artigos sobre seus simuladores.

Alguns dos simuladores foram testados, portanto a seção de características dessa tabela terá enfoque nas detalhes observados durante o uso dos simuladores, bem como dados que foram reunidos a partir da documentação oficial dos simuladores.

Tabela 1 – comparação entre simuladores de rotinas de sistemas operacionais.

Simulador	Ger. de Memória			Ger. de Processador	Ger. de Processos (princ.. políticas de escalonamento)	Escalonamento de disco	Linguagem programação usada	Principais Características
	Alocação de Paginação	Swapping	Substituição de Páginas					
SOSIM	Sim	Sim	Não	Sim	Não-Preemp., Preemp., FIFO, SJF, Por Prioridade	Não	Pascal, utilizando o software Borland Delphi.	Facilidade de uso, eficiência, permite que desenvolvedores alterem o código-fonte para moldá-lo segundo seu interesse
SSOG	Sim	Não	Não	Sim	Não-Preemp., Preemp., FIFO, SJF	Não	Java Standard Edition (SE) e utiliza paradigma orientado a objetos	A opção pela orientação a objetos foi, principalmente, permitir o aumento da produtividade. Graças ao paradigma, foi possível



								classificar as rotinas em entidades
WXPR OC	Não	Não	Não	Não	RJF, FIFO, SJF	Não	C++ e as bibliotecas Standard Template Library (STL) e wxwindows	O <i>software</i> é multiplataforma, o que garante seu uso por uma grande comunidade de usuários
S²O	Não	Não	Não	Sim	Por Prioridade, Não-Preemp., FCFS, SJF, Round Robin	Não	Java	Por ser em Java, oferece suporte multiplataforma, tal qual permite que usuários de diferentes plataformas possam executar o mesmo sem que, por isso, percam desempenho
TBC-SO/ WEB	Sim	Não	Não	Não	Por Prioridade, FIFO, SJF, HRRN, Não-Preemp., Preemp., SRTF, Round-Robin	Não	Java <i>applets</i>	O <i>software</i> simula as políticas de alocação de memória First-Fit, Next-Fit, Best-Fit e Worst-Fit
Simula RSO	Não	Não	Sim	Não	FCFS, SJF, Round Robin, SRT	FC-FS, SSTF, CSCAN, SCAN, LOOK	Web (Java, HTML5, VRaptor, etc.)	O <i>software</i> funciona inteiramente via <i>web</i> , possuindo o código-fonte aberto com a descrição das tecnologias utilizadas.

(Fonte: autores)

6 CONSIDERAÇÕES FINAIS

Dentre os simuladores citados no presente artigo, o TBC-SO/WEB chamou a atenção por permitir a execução de diversas simulações via *web*, através de *Java Applets*, oferecendo, portanto, suporte multiplataforma sem a necessidade de instalação de qualquer programa no computador, apenas exigindo que o usuário possua o Java instalado. Outro simulador que chamou atenção foi o SOSIM, cujo código-fonte pode ser editado por desenvolvedores que desejarem adicionar ou melhorar funcionalidades. Dentre os simuladores utilizados, este foi o que apresentou suporte à simulação da maior quantidade de rotinas, apresentando uma interface simples e completa, que permite que o estudante utilize a criatividade na criação de simulações altamente interativas. O único simulador que apresentou, durante os testes, possibilidade de simulação de algoritmos de escalonamento de discos foi o SimulaRSO, que funciona via *web*.

Os conceitos e comparativos contidos nesse artigo pretendem informar o leitor a respeito dos fundamentos de sistemas operacionais e traçar paralelos entre os simuladores de rotinas de sistemas operacionais. Existem muitos simuladores disponíveis para a utilização. Inclusive, alguns dos mesmos, tem fins mais específicos. Esse comparativo auxiliará alunos de graduação no conhecimento das ferramentas de simulação de rotinas de sistemas operacionais através de métodos ativos no processo de ensino-aprendizagem.

Evidentemente, à medida em que o tempo passa, novos desenvolvedores criam outras soluções para esses fins, bem como novas tecnologias vão sendo desenvolvidas. Portanto, de tempos em tempos, é importante que o profissional de tecnologia da informação tenha a curiosidade de pesquisar as ferramentas mais recentes que possam colaborar com o seu aprendizado.

APOIO FINANCEIRO

Esse artigo foi publicado graças ao aporte financeiro disponibilizado pelo IFES através do Programa Institucional de Difusão Científica (PRODIF) - Auxílio para publicações científicas.

REFERÊNCIAS

Almeida, R. de A. P., Lima, T. F., & Rabelo, P. S. SAE: Simulador para Algoritmos de Escalonamento. Recuperado de <https://uspdigital.usp.br/siicusp/cdOnlineTrabalhoVisualizarResumo?numeroInscricaoTrabalho=3488&numeroEdicao=18>

Clube do Hardware. Recuperado de <http://e.cdn-hardware.com.br/static/20101216/index-html-6dc00340.png.resized.jpg>

Silva, A. C. da. (2007). Estudo sobre os tipos de gerenciamento do SO Linux. Recuperado de <http://www.vivaolinux.com.br/artigo/Estudo-sobre-os-tipos-de-gerenciamento-do-SO-Linux?pagina=1>

Illinois Institute of Technology. Disk Scheduling Algorithms. Recuperado de <http://www.cs.iit.edu/~cs561/cs450/disksched/disksched.html>

Kioki, E. Y. (2008). Um simulador didático como ferramenta de apoio ao ensino da disciplina de sistemas operacionais. Recuperado de http://www.fai-mg.br/portal/download/revista_inicia_2008/pub_dw_artigo_simulador.pdf

Machado, F. B., & Maia, L. P. (2013). Um framework construtivista no aprendizado de Sistemas Operacionais: uma proposta pedagógica com o uso do simulador SOsim. Recuperado de <http://www.training.com.br/sosim/sbcwei04.pdf>

Silva, J. C. M. da. (2013). PSSAV: Simulação de Escalonamento de Processos. Publicado em 2013. Recuperado de <http://www.vivaolinux.com.br/artigo/PSSAV-Simulacao-de-Escalonamento-de-Processos>

Novato, D. (2014). Sistemas Operacionais: O que é Escalonamento de Processos? Recuperado de <http://www.oficinadanet.com.br/post/12781-sistemas-operacionais-o-que-e-escalonamento-de-processos>

Reis, F. P. (2009). Um software educacional para o ensino de políticas de escalonamento de processos e de alocação de memória em sistemas operacionais. Recuperado de <http://www.br-ie.org/pub/index.php/sbie/article/view/1129/1032>

Ribeiro, T. P. Simulador de rotinas do sistema operacional para auxílio às aulas teóricas. Recuperado de <http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2014/0063.pdf>

Rocha, A. de R. (2014). wxProc: Um simulador de políticas de escalonamento multiplataforma. Recuperado de <http://www.ic.unicamp.br/~rocha/pub/papers/wxProcUmSimuladorPoliticasEscalonamento.pdf>



Rodrigues, A. de A. (2011). SimulaRSO: Simulador de Recursos de Sistemas Operacionais. Recuperado de <http://www.simula-rso.appspot.com> Tanenbaum, A. S. (2000). Sistemas operacionais: projeto e implementação. Porto Alegre: Bookman.