

# wxProc — Um Simulador de Políticas de Escalonamento Multiplataforma

ANDERSON DE REZENDE ROCHA<sup>1</sup>  
ANDRÉIA SCHNEIDER<sup>2</sup>  
JÚLIO CÉSAR ALVES<sup>2</sup>  
RICARDO MARTINS DE ABREU SILVA<sup>2</sup>

<sup>1</sup> Universidade Estadual de Campinas (Unicamp)  
Instituto de Computação  
Caixa Postal 6176 – CEP 13084-971, Campinas, SP  
anderson.rocha@ic.unicamp.br

<sup>2</sup> Universidade Federal de Lavras (UFLA)  
Depto. de Ciência da Computação  
Caixa Postal 37 – CEP 37200-000, Lavras, MG  
<sup>2</sup>(andreaia, jcalves, rmas)@comp.ufla.br

**Resumo.** Cada vez mais ferramentas pedagógico-computacionais são desenvolvidas, facilitando o processo de ensino-aprendizagem. O objetivo deste artigo consiste em apresentar uma ferramenta de auxílio ao ensino de políticas de escalonamento contextualizando-a no ensino a distância. Espera-se contribuir, de forma efetiva, com a melhoria da qualidade de ensino nas disciplinas (Sistemas Operacionais, Redes de Computadores, entre outras) que fazem uso dessas políticas.

**Palavras-Chave:** Escalonamento de processos, Ensino a distância, Ferramentas didáticas, wxProc

## 1 Introdução

Com o avanço tecnológico atual, os computadores passaram a fazer parte do dia a dia das pessoas. Percebe-se claramente o aumento do uso da Internet, principalmente como uma ferramenta auxiliar na educação (pesquisas escolares, ensino a distância, entre outros).

De acordo com Araújo [J. Manzoli, 2003], o uso de recursos tecnológicos, como o computador e a Internet, não só desperta nos alunos o interesse em estudar quanto os prepara para a integração com uma sociedade altamente tecnológica.

A utilização do computador como ferramenta educacional começa a concretizar um novo conceito educacional: o aluno é estimulado a construir o conhecimento e não apenas esperar a transferência deste para si através do professor. Outro fato importante é que o uso de *software* educacional poder melhorar o aprendizado do aluno, pois a rápida resposta dada pelas ferramentas computacionais encoraja os alunos a se auto corrigirem, os leva a experimentação e descoberta do conhecimento [J. Manzoli, 2003], além de promover a imediata discussão dos erros e acertos conduzindo-os a uma melhor compreensão do assunto abordado.

Segundo Valente [Valente, 2003], a aquisição do conceito por parte do aluno ocorre de duas formas: primeiro o conhecimento origina-se no aluno, passa pelo *software* e vai para o computador. No segundo, o conhecimento parte do computador, passa pelo *software* alcançando o aluno. O *wxProc* contempla estas duas formas de interação.

De acordo com Fischer [Fischer, 2003], a comunicação existente entre um *software* e o aluno é fator relevante ao sucesso do mesmo. Assim, para a construção de um *software* educativo de qualidade, deve-se buscar alcançar uma interface amigável. A ênfase deve preocupar-se não apenas com o conteúdo, mas também com a apresentação do mesmo, tendo como alvo o usuário e sua satisfação em usar o *software*.

Outro fator importante que deve ser considerado em um *software* educativo consiste na velocidade da resposta a ação do usuário. Um *software* educativo deve também indicar de forma clara seus erros e apresentar propostas para corrigi-los. Desta forma, o usuário sentir-se-á motivado a corrigir suas falhas e assimilará melhor o conteúdo.

No caso de uma ferramenta como a *wxProc*, que serve como auxiliadora no aprendizado, os fatores mais

importantes são:

- interface clara e fácil de usar;
- resultado claro e fácil de ser entendido pelo usuário, além de mostrar os passos seguidos para se chegar ao resultado;
- transmissão, de forma clara, de todos os conceitos a serem aprendidos.

O simulador *wxProc* foi desenvolvido para permitir que os alunos da disciplina de Sistemas Operacionais, do curso de Ciência da Computação da Universidade Federal de Lavras, acompanhassem visualmente a simulação do escalonamento de processos no computador. Para cada conjunto de processos, o aluno pode visualizar como seria o resultado deste conjunto quando submetido a uma das políticas de escalonamento implementadas. Foram implementadas as quatro políticas apresentadas na seção 2.

Atualmente, encontra-se em desenvolvimento uma versão em Java do simulador, habilitando a sua execução diretamente através de um navegador *Web*. Esta atualização está sendo desenvolvida porque a ferramenta foi aprovada como parte integrante do Curso de Pós-graduação *latu-sensu* de Administração de Redes Linux (ARL) desta universidade.

A inserção em um curso a distância visa sanar as enormes dificuldades dos alunos em aprender este conteúdo. Estas dificuldades surgem, principalmente, porque o processo de aprendizagem fica extremamente limitado na ausência de uma ferramenta visual.

Para facilitar a leitura das seções subsequentes, este artigo segue a seguinte estrutura: primeiramente apresentam-se os fundamentos teóricos das políticas de escalonamento de processos. Em seguida, têm-se uma visão e funcionamento do simulador desenvolvido. Finalmente, algumas considerações finais detalham o estágio atual de desenvolvimento e algumas perspectivas futuras.

## 2 Fundamentos teóricos

O escalonamento de tarefas pode ser visto como uma técnica de propósito geral para diferentes áreas tais como teoria de controle, otimização de sistemas e gerenciamento de produção, entre outros.

Um caso específico de escalonamento de tarefas é o escalonamento de processos em uma CPU. Através da troca de uso da CPU entre os processos é possível fazer com que o computador torne-se mais eficiente. O maior problema é decidir qual processo deve ser executado primeiro.

Para implementar o compartilhamento da CPU entre diversos processos, um sistema operacional multiprogramável deve possuir um critério para determinar, entre os diversos processos no estado pronto (*ready*), qual o próximo processo a executar. Esse procedimento de seleção é realizado por um importante componente do sistema operacional denominado escalonador, daí o nome de escalonamento de processos. O escalonamento procura dividir o tempo do processador de forma justa entre os processos aptos a executar, de modo a atingir os seguintes objetivos<sup>1</sup>:

- maximizar a utilização do processador;
- maximizar a produção do sistema, ou seja, o número de processos executados por unidade de tempo;
- garantir que todos os processos recebam o processador;
- minimizar o tempo total de execução dos processos;
- minimizar o tempo que um processo fica na lista de espera;
- minimizar o tempo de reposta, isto é, o tempo decorrido entre uma requisição e a sua realização.

Um algoritmo de escalonamento seleciona qual o processo deve executar em um determinado instante de tempo. Existem vários algoritmos para atingir os objetivos do escalonamento. Eles buscam obter bons tempos médios ao invés de maximizar ou minimizar um determinado critério e busca privilegiar a variância em relação a tempos médios.

Existem dois tipos básicos de escalonamento:

- não-preemptivo: o processo que está sendo executado não pode ser interrompido. Tal tipo de escalonamento foi utilizado nos primeiros sistemas multiprogramáveis, onde predominava o processamento em *batch*. As políticas que implementam escalonamento não-preemptivo não são aplicáveis a sistemas de tempo compartilhado, pois em processos interativos é necessário um tempo de reposta ao usuário razoável;
- preemptivo: o processador pode ser retirado do processo que está sendo executado. Permite atenção imediata aos processos mais prioritários (tempo real), melhores tempos de resposta (tempo compartilhado), compartilhamento uniforme do processador;

<sup>1</sup>Muitas vezes não é possível alcançar, de uma vez, todos estes objetivos

Há diferentes algoritmos que tratam disto [Stalings, 2002], dentre eles estão: *First-Come-First-Served*, *Shortest-Job-First*, *Round Robin* e *Remaining-Job-First* preemptivo, descritos a seguir.

### 2.1 First-Come-First-Served

É o mais simples de todos os algoritmos de escalonamento. Ele parte da idéia de fila (primeiro a entrar é o primeiro a sair). Suas características são descritas a seguir:

- o primeiro processo que chegar é o primeiro a ser selecionado para execução;
- os processos que passam para o estado pronto são colocados no final de uma fila. Quando chegarem ao seu início, serão selecionados para execução;
- também utilizado para processos bloqueados;
- algoritmo de implementação bastante simples, utilizando uma única fila;
- ao receber o processador, o processo utiliza-o sem sofrer interrupções;
- processos pequenos esperam tanto quanto os grandes.

### 2.2 Round Robin

Round-Robin é um dos mais antigos algoritmos de escalonamento. É largamente usado, e foi projetado especialmente para sistemas de tempo compartilhado. A idéia do algoritmo é a seguinte:

- o primeiro processo na fila de prontos recebe o processador;
- cada processo executa durante uma fatia de tempo;
- se a fatia de tempo for excedida, o processo que estava executando é removido (preempção) e inserido, do mesmo modo que os processos que são desbloqueados e os recém-criados, no final da fila pronto;
- o tratamento dispensado aos processos é o mesmo, indistintamente;
- o valor da fatia de tempo é crucial. Caso seja pequeno, o tempo de troca de contexto torna-se significativo, gerando sobrecarga (*overhead*). Caso seja grande, aumenta o tempo de resposta dos processos no final da fila de prontos;

- bastante adequado para sistemas de tempo compartilhado. Favorece processos pequenos, que terminam primeiro, sem penalizar os grandes;
- simplicidade de implementação.

### 2.3 Shortest-Job-First

A idéia deste algoritmo é executar sempre o processo que tenha menor tempo de execução. Ele possui as seguintes características:

- cada processo possui um tempo de uso do processador;
- é ativado o processo com menor tempo de processador;
- o processo não perde o processador para outro de menor duração (não-preemptivo);
- minimiza o tempo médio de espera de um conjunto de processos, pois os processos menores são colocados a frente e terminam mais rápido;
- processos pequenos têm espera média menor que os grandes;
- dificuldade de se determinar antecipadamente o tempo de processador de cada processo. Na prática o tempo é estimado aproximadamente;
- não considera outros tipos de seleção (prioridades);
- pode causar postergação indefinida quando um processo com elevado tempo de processamento não receber nunca o processador;
- processos grandes possuem espera média maior que no FCFS.

### 2.4 Remaining-Job-First

Este algoritmo diferencia-se do algoritmo da seção 2.3 quanto a preempção. Nesta abordagem um processo ativo pode vir a perder o processador caso o seu tempo restante de processamento seja maior que o tempo de processamento de algum outro processo.

## 3 Visão geral do Simulador

Para atingir este objetivo foram utilizadas a linguagem de programação C++ e as bibliotecas *Standard Template Library* (STL) e *wxWindows*. Estas escolhas tornam o *software* multi-plataforma abrangendo um maior número de usuários possível.

As recomendações de Interface Homem Máquina (IHM) foram criteriosamente seguidas, tornando a interface clara, amigável e consistente [Minasi, 2003].

#### 4 O funcionamento do software

Ao se executar o programa, é apresentada na tela do computador uma janela gráfica inicial conforme pode ser visto na Figura 1.

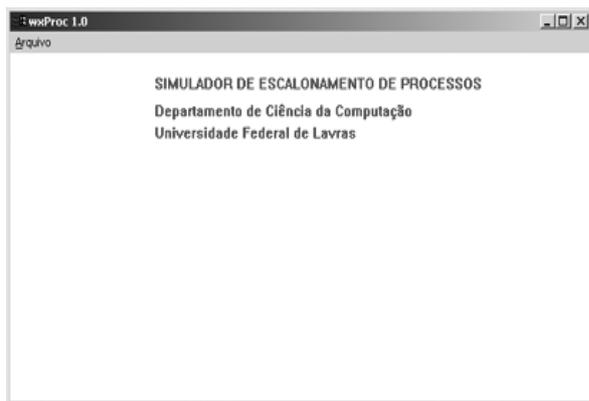


Figura 1: Tela inicial do aplicativo

Para utilizar as funcionalidades do programa acesse o *menu* principal, formado pelas seguintes opções: *Configurar Processos*, *Escolher Política*, *Tic*, *Sobre* e *Sair*.

##### Configurar Processos

Permite a inserção dos processos a serem escalonados. O usuário deve digitar o tempo de criação e de processamento de cada processo, além das opções abaixo:

- Adicionar processo → clica-se no botão **Adicionar** ou tecla-se ENTER.
- Finalizar configuração de processos → clica-se no botão **Terminado**.

##### Escolher Política

Permite a escolha da política de escalonamento. O usuário escolhe a política desejada e o modo de simulação (passo-a-passo ou automático). No modo *passo-a-passo* é executada uma fatia de tempo por vez, enquanto no modo *automático* toda a simulação é executada de uma vez (Figura 3).

Caso a política escolhida seja *Round Robin*, apresenta-se uma nova janela na qual o usuário deve informar o valor da fatia de tempo (*quantum*).

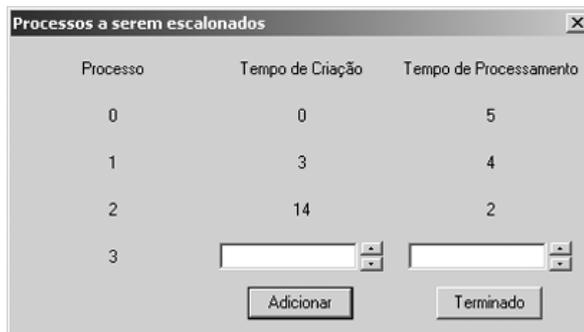


Figura 2: Tela de escolha dos processos

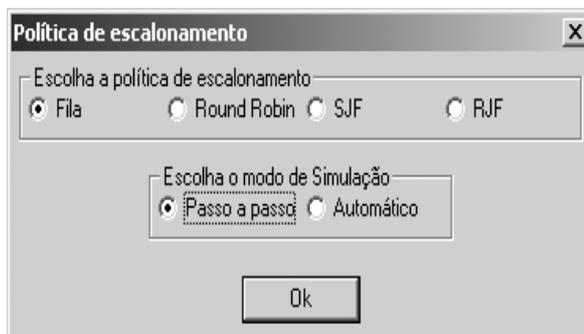


Figura 3: Tela de escolha das políticas de escalonamento e do modo de simulação

##### Tic

Esta é a funcionalidade que permite a execução *passo-a-passo* da simulação, além de apresentar graficamente ao usuário qual processo utilizou o processador no momento em questão (Figura 4).

##### 4.1 Término da simulação

Quando a simulação termina, apresenta-se ao usuário uma tabela contendo o identificador, *turnaround* e os tempos de inicialização, processamento e finalização de cada processo (Figura 5).

#### 5 Considerações finais

Os autores deste trabalho acreditam que esta ferramenta poderá contribuir de maneira significativa para o aprendizado de processos de escalonamento. Para comprovar a veracidade desta suposição, iniciou-se um estudo de caso juntamente aos alunos do ARL. Os *applets* desenvolvidos ficarão em uma página integrada ao ambiente [Teleduc, 2003] para acesso dos alunos do curso. Este ambiente permite que os alunos postem mensagens nos

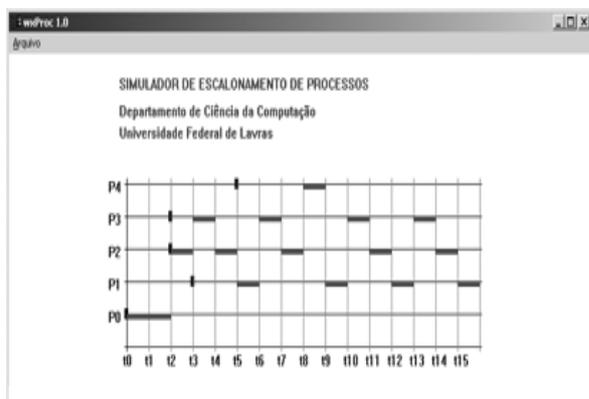


Figura 4: *wxProc* em execução

Processo	Criação	Processamento	Finalização	Turnaround
0	0	2	2	2
1	3	4	16	13
2	2	5	15	13
3	2	4	14	12
4	5	1	9	4

Figura 5: Tabela de finalização

fóruns e/ou tirem dúvidas nos *chats* enquanto utilizam a ferramenta. Além disso, os autores pretendem passar um questionário com a finalidade de estudar a aceitabilidade da ferramenta desenvolvida.

Além do módulo *Web*, pretende-se desenvolver um módulo de impressão. Nada impede que novas funcionalidades sejam adicionadas a partir do retorno proporcionado pelo estudo de caso proposto.

## Referências

[J. Manzoli, 2003] J. Manzoli, C. R. A. (2003). *Javamusic*: uma proposta para criação de um cenário para educação musical baseado na utilização de applets java e xml.

[Fischer, 2003] Fischer, M. C. B. O. (2003). Estudo de requisitos para um software educativo de apoio ao ensino de introdução à computação.

[Minasi, 2003] Minasi, M. (2003). Segredos de projeto de interface gráfica com o usuário.

[Stalings, 2002] Stalings, W. (2002). *Operating systems: Internals and design principles*.

[Teleduc, 2003] Teleduc (2003). *Teleduc, um ambiente de ensino a distância*.

[Valente, 2003] Valente, J. A. (2003). *Diferentes usos do computador na educação*.