SIMULADOR DOS ALGORITMOS DE ESCALONAMENTO DE PROCESSOS

Wilson Prates de Oliveira[1], Danilo Silveira Martins[1], Carlos Miguel dos Santos Alves[1]

[1]Instituto Federal do Norte de Minas Gerais – IFNMG, Arinos-MG, Brasil
(wilson.oliveira@ifnmg.edu.br; danilo.silveira@ifnmg.edu.br;
cmsa@aluno.ifnmg.edu.br)

Resumo: Este projeto desenvolve um simulador de algoritmos de escalonamento de processos, integrando conceitos das disciplinas de Estrutura de Dados II e Sistemas Operacionais. Visa facilitar a compreensão teórica e prática dos alunos sobre gerenciamento de processos e utilização da CPU. O simulador demonstra na prática como diferentes algoritmos afetam o desempenho do sistema, promovendo uma aprendizagem significativa e preparando os alunos para desafios reais no campo da computação.

Palavras-chave: Escalonamento de Processos; Simulador Educacional; Sistemas Operacionais; Algoritmos de CPU

INTRODUÇÃO

Nos últimos anos, a crescente complexidade dos sistemas computacionais e a necessidade de otimização dos recursos têm exigido um entendimento mais profundo das técnicas de gerenciamento de processos em Sistemas Operacionais. Um dos principais desafios no campo da computação é garantir que a Unidade Central de Processamento (CPU) seja utilizada de maneira eficiente, maximizando o desempenho do sistema. A gestão adequada dos processos é vital para alcançar essa eficiência, sendo realizada através de algoritmos de escalonamento. Historicamente, o desenvolvimento dos Sistemas Operacionais remonta aos primeiros computadores multiprogramados, onde a necessidade de executar múltiplos programas simultaneamente levou à criação de técnicas para gerenciar e priorizar processos. Desde então, diversas políticas e algoritmos de escalonamento foram propostos e implementados para lidar com diferentes cargas de trabalho e requisitos de desempenho (TANENBAUM; 2024).

Apesar dos avanços, um desafio persistente na educação em Ciência da Computação é a compreensão teórica e prática dos algoritmos de escalonamento de processos. Os alunos frequentemente encontram dificuldades em associar os conceitos teóricos de Estrutura de Dados e Sistemas Operacionais com aplicações práticas reais. A falta de uma abordagem integrada entre essas disciplinas pode resultar em um entendimento fragmentado e superficial dos mecanismos que regem o funcionamento de sistemas computacionais modernos. Essa desconexão entre teoria e prática evidencia a necessidade de métodos de ensino que combinem ambos os aspectos de forma coesa e eficaz.

A compreensão dos diferentes algoritmos de escalonamento de processos e das estruturas de dados utilizadas em Sistemas Operacionais é fundamental para a formação de profissionais competentes em Ciência da Computação (CORMEN, 2012). Questões como o funcionamento prático desses algoritmos, as estruturas de dados mais adequadas para representar processos, e as diferenças de desempenho entre as diversas políticas de escalonamento são cruciais para o desenvolvimento de sistemas eficientes. Além disso, entender a aplicação prática desses conceitos pode facilitar a internalização teórica pelos alunos, proporcionando uma aprendizagem mais significativa e duradoura.

Este estudo tem como objetivo principal projetar e implementar um simulador de escalonamento de processos, visando auxiliar os alunos do curso de Bacharelado em Sistemas de Informação do Instituto Federal do Norte de Minas Gerais - IFNMG Campus Arinos a compreenderem e aplicarem os conceitos fundamentais das disciplinas Estrutura de Dados II e Sistemas Operacionais de maneira interdisciplinar. Entre os objetivos específicos, destaca-se o ensino de conceitos teóricos de Algoritmos e Estruturas de Dados a partir de um ponto de vista prático. Também se pretende demonstrar como os processos são representados internamente em um Sistema Operacional, criar uma estrutura de dados simplificada para armazenar os dados de um processo, e explicar as principais estruturas de dados usadas pelo SO para organizar processos na memória e no disco.

Ensinar os conceitos de escalonamento e os principais algoritmos utilizados para selecionar processos para execução é outro objetivo crucial deste projeto. A implementação prática de um simulador permitirá que os alunos vejam na prática como esses algoritmos funcionam, quais são suas vantagens e desvantagens, e como eles afetam o desempenho geral do sistema. Esse conhecimento prático é essencial para a formação de profissionais capazes de projetar e manter sistemas operacionais eficientes e confiáveis.

A construção de um simulador de algoritmos de escalonamento é relevante tanto do ponto de vista pedagógico quanto técnico. Pedagogicamente, oferece uma ferramenta prática que conecta teoria e aplicação, facilitando a aprendizagem e retenção dos conceitos. Tecnicamente, proporciona um ambiente de experimentação e análise, permitindo aos alunos explorar o impacto de diferentes políticas de escalonamento em cenários diversos. Dessa forma, o projeto não só melhora a qualidade do ensino, mas também prepara os alunos para enfrentar desafios reais no mercado de trabalho (REIS et al., 2009).

O objeto de estudo deste projeto é a implementação e avaliação de um simulador de algoritmos de escalonamento de processos. Este simulador será utilizado como ferramenta educacional para demonstrar na prática os conceitos teóricos de Sistemas Operacionais e Estruturas de Dados II, promovendo uma compreensão mais profunda e integrada dos conteúdos dessas disciplinas. A relevância do projeto se manifesta na potencial melhoria do processo de ensino-aprendizagem, capacitando os alunos a entenderem e aplicarem conceitos complexos de forma prática e interativa, resultando em uma formação mais completa e robusta.

MATERIAL E MÉTODOS

A metodologia para o desenvolvimento do projeto de construção do simulador de escalonamento de processos é dividida em três etapas interligadas: Planejamento e Preparação, Implementação do Simulador e Avaliação do Aprendizado. Esta abordagem visa garantir uma compreensão aprofundada dos conceitos, combinando teoria com aplicação prática. Cada etapa é cuidadosamente planejada para maximizar a eficácia educacional e técnica do projeto.

Na primeira etapa, focamos no Planejamento e Preparação. Definimos claramente os objetivos educacionais, delineando as competências que esperamos que os alunos adquiram. Realizamos uma pesquisa minuciosa para selecionar os algoritmos de escalonamento mais relevantes a serem implementados no simulador. Paralelamente, planejamos a estrutura geral do simulador, desde a representação dos processos até a interface do usuário. Preparamos também materiais de apoio, incluindo conceitos básicos de escalonamento de CPU, documentação dos algoritmos, tutoriais de programação e cenários de simulação.

Na segunda etapa, procedemos com a Implementação do Simulador. Os alunos têm a oportunidade de aplicar o conhecimento teórico adquirido. Os algoritmos de escalonamento são codificados na linguagem de programação escolhida, e a interface do usuário é desenvolvida, permitindo interação com os algoritmos em diferentes cenários. Estruturas de dados relevantes são integradas para representar processos e informações relacionadas. Testes rigorosos são conduzidos para identificar e resolver quaisquer erros ou problemas que possam surgir durante o desenvolvimento.

A terceira etapa concentra-se na Avaliação do Aprendizado. Avaliamos a eficácia do projeto em termos educacionais, envolvendo os alunos em simulações dirigidas onde aplicam algoritmos e observam os resultados. Eles realizam análises comparativas entre diferentes algoritmos e fornecem feedback qualitativo sobre a usabilidade do simulador. Avaliações teóricas são aplicadas para verificar a compreensão dos alunos sobre os conceitos abordados, garantindo que os objetivos educacionais foram alcançados.

Essas três etapas se interconectam para criar uma experiência educacional abrangente e prática. Desde o planejamento até a avaliação, o projeto busca não apenas ensinar teoria, mas também permitir que os alunos apliquem os conhecimentos em um contexto realista. A metodologia escolhida promove uma compreensão profunda dos conceitos de escalonamento de processos e possibilita uma avaliação significativa do aprendizado alcançado pelos alunos.

RESULTADOS E DISCUSSÃO

O projeto resultou na criação de um Simulador de Escalonamento de Processos, desenvolvido pelos alunos com a orientação dos professores. O código fonte do projeto pode ser encontrado em: https://bit.ly/4bzP8km. Este simulador permite explorar e compreender o funcionamento de diversos algoritmos de escalonamento.

Um escalonador de processos é uma parte crucial do sistema operacional, responsável por gerenciar a ordem de execução dos processos no computador. Ele garante que todos os processos tenham a oportunidade de usar a CPU (unidade central de processamento) de forma eficiente e justa (NÉRY et al., 2018)

Por exemplo, em um sistema computacional quando são executados vários programas ao mesmo tempo, como um navegador, um editor de texto e um reprodutor de música, o escalonador decide qual programa (processo) deve ser executado a cada

momento. Ele leva em consideração as prioridades dos processos e o tempo que cada um já foi executado para decidir a ordem de execução. O escalonador alterna entre esses processos rapidamente, dando a impressão de que todos estão sendo executados simultaneamente, mesmo que a CPU só esteja processando um deles por vez, este é o conceito denominado pseudoparalelismo (MARQUES et al.,2019).

A interface gráfica inicial do simulador desenvolvido, exibe um menu dos tipos de algoritmos de escalonamento que serão simulados neste projeto: FCFS (First-Come, First-Served), Round Robin, Fila de Prioridade (Priority Scheduling), Job Menor (ver Figura 1).

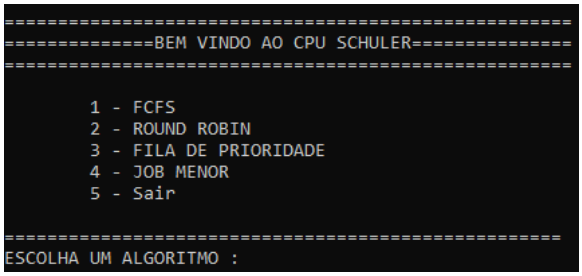

Figura 1. Tela inicial do simulador.

Após selecionar um número que representa o algoritmo que será utilizado para escalonamento, o usuário precisa definir outros parâmetros como a quantidade de processos que deseja simular, o tempo de CPU de cada processo, e ainda a prioridade de cada processo (1 – 5), sendo 1, baixa prioridade, e 5, alta prioridade (ver Figura 2).

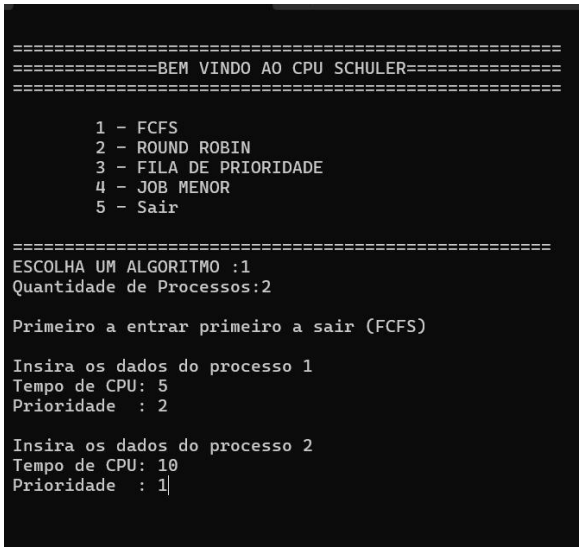

Figura 2. Parâmetros adicionais a serem configurados após escolha do algoritmo de escalonamento.

Após estas definições, o simulador realiza a execução dos algoritmos conforme selecionado.

A gestão eficiente dos processos em um sistema operacional é crucial para garantir o bom desempenho e a responsividade do sistema. Diversos algoritmos de escalonamento de processos foram desenvolvidos para otimizar a utilização da CPU e minimizar o tempo de espera dos processos (SANTOS et al., 2017). Neste trabalho, mencionamos o funcionamento, as vantagens e desvantagens de quatro algoritmos comumente utilizados.

1 - FCFS (First-Come, First-Served)

Primeiro a chegar, primeiro a ser servido, esse algoritmo prioriza a execução com base na ordem de chegada dos processos à fila de prontos (estado dos processos em um sistema operacional quando estão aguardando para serem executados pela CPU). A Figura 3 demonstra a simulação dos processos sendo executados por ordem de chegada.

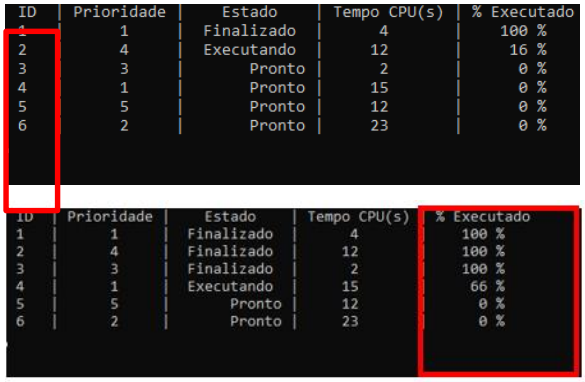

Figura 3. Exemplo de execução do algoritmo FCFS (First-Come, First-Served)

Por sua simplicidade de implementação, se destaca por atender os processos na ordem de chegada. Essa característica garante justiça entre os processos que chegam no mesmo instante. No entanto, pode resultar em tempos de espera longos, especialmente em sistemas com processos de duração variada. Um processo longo, por exemplo, bloqueia a execução de outros, gerando longos tempos de espera. A ausência de consideração de prioridade ou tempo de execução dos processos também limita a aplicabilidade deste algoritmo em cenários mais complexos (SILBERSCHATZ et al., 2015; TANENBAUM, 2024).

2 - ROUND ROBIN

Este algoritmo atribui um quantum de tempo (ou fatia de tempo) fixo a cada processo, alternando sua execução em uma fila circular. Um quantum é um intervalo de tempo fixo em que um processo pode ser executado antes de ser interrompido para dar espaço a outro processo. Essencial no algoritmo Round Robin, o quantum assegura que todos os processos tenham uma chance equitativa de uso da CPU. Por exemplo, com um quantum de 20 milissegundos, cada processo é executado por esse período antes de ceder lugar ao próximo na fila de execução.

Quando selecionado no simulador, além dos parâmetros já solicitados nos demais algoritmos, esse algoritmo Round Robin pede para inserir o Quantum (veja Figura 4).

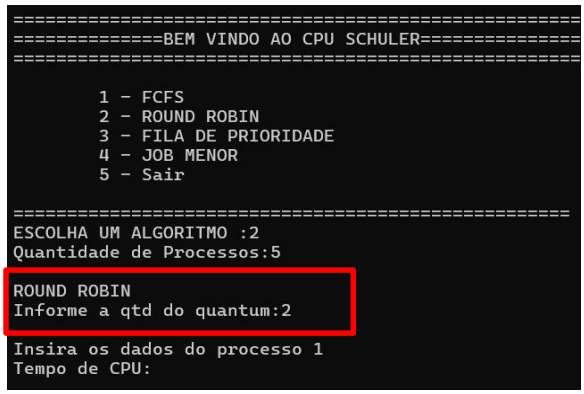

Figura 4. Demonstração da parametrização adicional do algoritmo Round Robin (quantum).

Após isso, o algoritmo inicia a partir do primeiro processo na fila e o executa por um quantum de tempo. Ao término do quantum, o processo em execução é movido para o final da fila, e o próximo processo na fila é selecionado e executado até que todos os processos sejam finalizados ou a fila esteja vazia (veja a Figura 5).

ID	Prioridade	Estado	Tempo CPU(s)	% Executado
1	1	Finalizado	4	100 %
2	4	Finalizado	12	100 %
3	3	Finalizado	2	100 %
4	1	Executando	15	80 %
5	3	Finalizado	12	100 %
6	2	Executando	23	52 %
7	2	Finalizado	10	100 %
8	4	Finalizado	3	100 %
9	2	Finalizado	12	100 %
10	1	Executando	18	55 %
11	3	Finalizado	6	100 %
12	3	Executando	21	47 %
13	3	Executando	21	47 %

Figura 5. Execução do algoritmo Round Robin.

O algoritmo Round Robin melhora a equidade e o tempo de resposta dos processos ao compartilhar o tempo da CPU em intervalos iguais, porém pode introduzir sobrecarga devido à frequente comutação de contexto (SILBERSCHATZ et al., 2015; TANENBAUM, 2024).

3 - FILA DE PRIORIDADE (PRIORITY SCHEDULING)

O algoritmo Priority Scheduling determina a ordem de execução com base nas prioridades atribuídas no início de cada processo. Processos com maior prioridade recebem tempo de CPU preferencialmente, garantindo que tarefas críticas sejam executadas com mais rapidez (ver Figura 6). Essa característica garante que processos críticos recebam atenção imediata, minimizando o tempo de espera para tarefas relevantes. O algoritmo apresenta comportamento previsível, pois processos com alta prioridade sempre têm precedência. No entanto, a Fila de Prioridade é suscetível à inanição de processos com baixa prioridade, que podem ficar "esquecidos" se a fila de alta prioridade estiver sempre ocupada (SILBERSCHATZ et al., 2015; TANENBAUM, 2024).

ID	Prioridade	Estado	Tempo CPU(s)	% Executado
3	3	Finalizado	2	100 %
8	4	Finalizado	3	100 %
1	1	Executando	4	50 %
7	2	Pronto	10	0 %
2	4	Pronto	12	0 %
5	3	Pronto	12	0 %
9	2	Pronto	12	0 %
4	1	Pronto	15	0 %
10	1	Pronto	18	0 %
6	2	Pronto	23	0 %

ID	Prioridade	Estado	Tempo CPU(s	% Executad
3	3	Finalizado	2	100 %
8	4	Finalizado	3	100 %
1	1	Finalizado	4	100 %
7	2	Finalizado	10	100 %
2	4	Finalizado	12	100 %
5	3	Finalizado	12	100 %
9	2	Finalizado	12	100 %
4	1	Finalizado	15	100 %
10	1	Executando	18	33 %
6	2	Pronto	23	0 %

Figura 6. Execução do algoritmo Priority Scheduling.

4 - JOB MENOR (SHORTEST JOB FIRST - SJF)

Este algoritmo visa minimizar o tempo médio de espera dos processos, executando primeiro os processos com a menor duração estimada ou tempo de execução restante (ver Figura 7). Essa estratégia garante que processos curtos sejam finalizados rapidamente, liberando o processador para outros processos e reduzindo significativamente o tempo médio de espera. No entanto, o SJF pode levar à inanição de processos longos se estes forem continuamente preteridos por processos curtos (SILBERSCHATZ et al., 2015; TANENBAUM, 2024).

ID	Prioridade	Estado	Tempo CPU(s)	% Executado
1	1	Finalizado	4	100 %
4	1	Executando	15	46 %
10	1	Pronto	18	0 %

ID	Prioridade	Estado	Tempo CPU(s)	% Executado
6	2	Executando	23	82 %
7	2	Pronto	10	0 %
9	2	Pronto	12	0 %

ID	Prioridade	Estado	Tempo CPU(s)	% Executado
3	3	Finalizado	2	100 %
5	3	Executando	12	8 %

Figura 7. Execução do algoritmo Job Menor.

CONCLUSÃO

O desenvolvimento e implementação do simulador de escalonamento de processos demonstraram ser uma ferramenta educacional eficaz para o ensino dos conceitos fundamentais abordados nas disciplinas de Estrutura de Dados II e Sistemas Operacionais de maneira interdisciplinar. Através da simulação interativa, os alunos puderam explorar diferentes algoritmos de escalonamento, compreender suas características e analisar seu impacto no desempenho do sistema computacional.

Os resultados obtidos nas simulações destacaram as vantagens e desvantagens de cada algoritmo, proporcionando uma compreensão prática dos princípios teóricos discutidos em sala de aula. O

algoritmo SJF mostrou-se mais eficiente em termos de tempo de espera e tempo de retorno médio, corroborando com a literatura existente sobre escalonamento de processos.

Além dos benefícios quantitativos, observou-se uma melhoria substancial na capacidade dos alunos em aplicar os conceitos aprendidos. O feedback positivo recebido dos alunos e do professor da disciplina de Sistemas Operacionais reflete uma maior confiança e competência dos estudantes na explicação e análise dos algoritmos de escalonamento.

A utilização do simulador não apenas fortaleceu o entendimento teórico dos alunos, mas também preparou-os melhor para enfrentar desafios práticos no campo da computação. A experiência proporcionada pelo simulador serviu como um complemento valioso ao ensino tradicional, incentivando uma abordagem mais hands-on e interativa para o aprendizado.

Este projeto destacou a importância de integrar teoria e prática no processo de ensino-aprendizagem. O simulador não apenas facilitou a assimilação dos conteúdos acadêmicos, mas também promoveu uma compreensão mais profunda e holística dos algoritmos de escalonamento de processos. Espera-se que este estudo inspire futuros desenvolvimentos em metodologias educacionais, oferecendo novas perspectivas sobre como abordar temas complexos na educação em Ciência da Computação.

REFERÊNCIAS

CORMEN, Thomas et al. Algoritmos-Teoria e Prática (3a. ediçao). Editora Campus, 2012.

MARQUES, Bruna Lopes; DA ROSA, Guilherme Silveira; MOREIRA, João Padilha. Processos Em Sistemas Operacionais. SEMINÁRIO DE TECNOLOGIA GESTÃO E EDUCAÇÃO, v. 1, n. 1, p. 1-6, 2019. Recuperado de https://raam.alcidesmaya.com.br/index.php/SGTE/article/view/2

NÉRY, Jhonatan T. Cabral; BARRETO, Franciny M.; DE FREITAS, Joslaine C. Jeske. Construção de um Modelo de Simulação para Escalonamento de Processos não-preemptivos. In: WORKSHOP EM DESEMPENHO DE SISTEMAS COMPUTACIONAIS E DE COMUNICAÇÃO (WPERFORMANCE), 17., 2018, Natal. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2018. ISSN 2595-6167. DOI: https://doi.org/10.5753/wperformance.2018.3320.

REIS, Fabrício Pereira; JÚNIOR, Paulo Afonso Parreira; COSTA, Heitor Augustus Xavier. TBC-SO/WEB: Um software educacional para o ensino de políticas de escalonamento de processos e de alocação de memória em sistemas operacionais. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2009.

SANTOS, A. M. et al. OSLive: Protótipo de simulação de algoritmos de escalonamento de processos. In: ENCOINFO-Congresso de Computação e Tecnologias da Informação. ENCOINFO, 2017. p. 80-87.

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. Fundamentos de Sistemas: Operacionais. 9ª ed. Editora LTC, São Paulo, 2015.

TANENBAUM, Andrew S. et al. Sistemas operacionais modernos. 5ª ed. Bookman Editora, São Paulo, 2024.