



**INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DO ACRE
IFAC**

Elis Ananda, Marisol Assunção, Pâmela Bruna e Rheygys Patrick

**ESCALONAMENTO DE PROCESSOS
FIFO**

Rio Branco
2025

Elis Ananda, Marisol Assunção, Pâmela Bruna e Rheygs Patrick

ESCALONAMENTO DE PROCESSOS

FIFO

Documentação focada na explicação do sistema de escalonamento de processos FIFO, desenvolvidos pelos alunos de Estrutura de dados, do Curso de Sistemas para Internet, do campus Ifac em Rio Branco-Acre.

Orientador: Prof. Jonas Pontes

Alunos: Elis Ananda; Marisol Assunção; Pâmela Bruna e Rheygs Patrick.

Rio Branco

2025

SUMÁRIO

1. INTRODUÇÃO.....	4
2. ALGORITMO DE ESCALONAMENTO FIFO.....	4
3. COMO FOI DESENVOLVIDO O ALGORITMO FIFO.....	5
4. DECISÕES DO PROJETO.....	6
5. COMO COMPIRAR E EXECUTAR O PROGRAMA.....	7
6. FORMA DE CALCULAR AS MÉTRICAS.....	7
7. CONCLUSÃO.....	8
REFERÊNCIAS.....	9

1. INTRODUÇÃO

Esta documentação tem como propósito apresentar o desenvolvimento do sistema simulador de escalonamento de processos, requerida pelo professor Jonas Pontes, utilizando o algoritmo de escalonamento FIFO (First In, First Out), conforme proposto na disciplina de Estrutura de Dados do curso de Sistemas para Internet do Instituto Federal do Acre, Campus Rio Branco.

O sistema foi desenvolvido para permitir a inserção manual de processos, executar o escalonamento de acordo com a ordem de chegada, bem como é proposto o modelo FIFO, e calcular as principais métricas de desempenho utilizadas nos sistemas operacionais, como o tempo de término, turnaround time, waiting time, response time e throughput, apresentando ao final da entrada de dados uma tabela de clara compreensão, apresentando o desempenho calculado ao final dos processos.

2. ALGORITMO DE ESCALONAMENTO FIFO

O algoritmo de escalonamento de processos FIFO (First In, First Out), também conhecido como FCFS (First Come, First Served) nos livros mais clássicos, se trata de um dos mais simples métodos de escalonamento de processos, se fazendo presente com mais intensidade nos campos de estudo de Estrutura de dados e Sistemas Operacionais. Nesse tipo de algoritmo, os processos são atendidos estritamente na ordem em que estes entram no sistema, formando uma fila. O algoritmo FIFO se trata de um sistema de algoritmo não preemptivo, ou seja, uma vez que um processo inicia sua execução, ele não é interrompido até a sua conclusão.

Segundo Silberschatz, um especialista no campo da ciência da computação, o FIFO é um tipo de algoritmo de fácil compreensão e implementação, sendo amplamente utilizado para fins didáticos e como base para o estudo de algoritmos mais complexos. No entanto, esse tipo de modelo de escalonamento pode apresentar resultados insatisfatórios em determinados cenários, especialmente quando processos longos bloqueiam a execução de processos menores, fenômeno

conhecido como efeito comboio. De acordo com Tanenbaum e Bos, apesar de suas limitações, o FIFO é importante para os meios acadêmicos por permitir uma melhor análise inicial do comportamento de escalonadores, servindo como referência para comparação com outros algoritmos que buscam otimizar métricas como tempo médio de espera e tempo de resposta.

Neste trabalho, o algoritmo FIFO é utilizado como base para a construção de um simulador de escalonamento de processos, possibilitando a observação prática de seu funcionamento e a análise das métricas de desempenho associadas.

3. COMO FOI DESENVOLVIDO O ALGORITMO FIFO

Neste estudo, o algoritmo de escalonamento FIFO foi desenvolvido modelando cada processo com uma estrutura de dados do tipo struct, que inclui informações fundamentais como identificador, tempo de chegada e tempo de execução (burst time). Além das informações de entrada, a estrutura guarda os valores processados ao longo da execução do escalonamento, como tempo de início, tempo de término e métricas de desempenho.

Primeiramente, os processos são inseridos pelo usuário de forma manual e guardados em uma estrutura sequencial, no caso, um vetor, mantendo a ordem de chegada. Na sequência, o algoritmo atravessa essa estrutura de maneira linear, simulando a execução dos processos um por um, de acordo com o princípio do FIFO, em que o primeiro processo a chegar é o primeiro a ser executado.

Uma variável auxiliar controla o tempo atual da CPU durante a simulação. Se o tempo atual for menor que o tempo de chegada do próximo processo, assume-se que a CPU fica ociosa até a chegada desse processo. Como o FIFO é um algoritmo não preemptivo, um processo usa a CPU do início ao fim de sua execução.

Ao final da execução de cada processo, são determinadas as métricas de desempenho individuais, como tempo de conclusão, turnaround time, waiting time e response time. Esses valores são guardados na estrutura do processo em si, o que

possibilita sua apresentação em formato de tabela posteriormente. Finalmente, após a execução de todos os processos, calcula-se a média das métricas do algoritmo, além da vazão (throughput) do sistema.

4. DECISÕES DO PROJETO

O algoritmo FIFO foi selecionado para a realização do trabalho sobre os métodos de escalonamento de processos justamente por ser um dos clássicos, estando presente em diversas literaturas, principalmente no campo de sistemas operacionais, a exemplo do clássico livro “Fundamentos de Sistemas Operacionais”, de Silberschatz. Mesmo sendo de certa forma, ineficiente em alguns casos, o estudo do algoritmo FIFO ainda se faz extremamente necessário para a compreensão dos sistemas de escalonamento mais complexos.

Para o armazenamento dos processos, optou-se pelo uso de uma estrutura sequencial do tipo vetor. Essa escolha se justifica pelo fato de que o algoritmo FIFO executa os processos estritamente na ordem de chegada, não sendo necessária a utilização de estruturas de dados mais complexas, como filas encadeadas ou listas.

A implementação do algoritmo foi realizada de forma não preemptiva, respeitando as características originais do FIFO. Dessa forma, uma vez que um processo inicia sua execução, ele permanece utilizando a CPU até sua finalização, sem interrupções. Essa decisão garante aderência ao comportamento descrito nos livros clássicos de sistemas operacionais.

Além disso, a estrutura do processo foi projetada para armazenar tanto os dados de entrada quanto as métricas de desempenho calculadas durante a execução. Essa abordagem facilita a organização do código e a apresentação dos resultados em formato de tabela ao final da simulação.

5. COMO COMPILAR E EXECUTAR O PROGRAMA

O sistema foi desenvolvido na linguagem de programação C, utilizando o Visual Studio Code como ambiente de desenvolvimento e o compilador GCC para a compilação do código-fonte. Para compilar o programa, o usuário deve acessar, por meio do terminal, o diretório onde se encontra o arquivo fonte do sistema. Em seguida, deve ser realizado o processo de compilação utilizando o compilador GCC, o que gera um arquivo executável do programa.

Após a compilação, o sistema pode ser executado diretamente pelo terminal. Durante a execução, o programa solicita ao usuário a quantidade de processos a serem simulados, bem como os dados de cada processo, como o tempo de chegada e o tempo de execução. Ao final da entrada dos dados, o algoritmo FIFO é executado automaticamente e os resultados são apresentados na tela, incluindo a tabela final dos processos e as métricas de desempenho calculadas.

6. FORMA DE CALCULAR AS MÉTRICAS

Os tempos registrados durante a simulação do algoritmo FIFO foram usados para calcular as métricas de desempenho do escalonamento para cada processo individualmente. Para isso, usaram-se os valores de tempo de chegada, tempo de início e tempo de término de cada processo.

O tempo de término (completion time) é o momento em que o processo conclui totalmente sua execução na CPU. Esse valor é calculado adicionando o tempo de início da execução ao tempo total de processamento do processo.

O turnaround time é o tempo total que um processo leva para ser concluído no sistema, desde sua chegada até sua finalização. Esse valor é determinado pela diferença entre o tempo de chegada e o tempo de término do processo.

O waiting time representa o tempo total que um processo ficou na fila de prontos antes de ser executado. Esse período é calculado subtraindo o tempo de execução do processo de seu turnaround time.

O response time é o período que vai desde a chegada do processo ao sistema até o momento em que ele começa a ser executado pela primeira vez. No algoritmo FIFO, que é não preemptivo, o tempo de resposta é igual ao tempo de espera do processo.

Depois de calcular as métricas individuais, as métricas globais do algoritmo foram estabelecidas, utilizando as médias do tempo de espera, tempo de resposta e tempo de execução, levando em conta todos os processos realizados. A vazão (throughput) do sistema foi determinada pela relação entre o número total de processos realizados e o tempo total de execução da simulação.

7. CONCLUSÃO

A realização deste trabalho permitiu a aplicação prática dos conceitos de escalonamento de processos aprendidos na disciplina de Estrutura de Dados, através da criação de um simulador fundamentado no algoritmo FIFO. A modelagem dos processos e a simulação de sua execução permitiram uma compreensão concreta do funcionamento desse algoritmo e uma análise de seu impacto nas métricas de desempenho do sistema.

O simulador criado atendeu às demandas estabelecidas, possibilitando a inserção manual dos processos, a execução do escalonamento de acordo com a ordem de chegada e o cálculo das métricas mais relevantes empregadas em sistemas operacionais, como tempo de término, turnaround time, waiting time, response time e vazão. Ademais, a organização do código em módulos e a documentação detalhada auxiliaram na clareza e na manutenção do sistema.

Assim, é possível concluir que os objetivos do estudo foram atingidos, oferecendo uma compreensão mais aprofundada do algoritmo FIFO e funcionando como um ponto de partida para comparações futuras com outros algoritmos de escalonamento mais complexos.

REFERÊNCIAS

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. **Sistemas Operacionais: Conceitos**. 9. ed. Rio de Janeiro: LTC, 2015.

TANENBAUM, Andrew S.; BOS, Herbert. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson, 2016.