

Universidade da Beira Interior

Departamento de Informática



**Departamento de
Informática**

Simulador de Algoritmos de Escalonamento

Elaborado por:

Leonardo Ferreira 52981

Rafael Gomes 51923

Tomás Simões 52585

Gonçalo Simões 52091

Orientador:

Professor Doutor Paul Crocker

27 de abril de 2025

Conteúdo

1	Introdução	1
1.1	Contexto e Objetivos	1
2	Estrutura da Aplicação	2
2.1	Módulo de Geração de Processos	2
2.2	Módulo do <i>Clock</i>	2
2.3	Módulo de Escalonamento	3
2.4	Interface Gráfica	4
2.5	Conclusão	5
3	Como Utilizar a Aplicação	6
3.1	Configuração Inicial	6
3.2	Iniciar Simulação	8
3.3	Janela de Simulação	8
3.4	Painel de Processos	8
3.4.1	Processo em Execução	8
3.5	Painel de Tempo	9
3.6	Painel de Processos Completos	10
3.7	Gráficos de Métricas	11
4	Conclusão	12

Capítulo 1

Introdução

1.1 Contexto e Objetivos

Neste relatório apresenta-se o desenvolvimento e o funcionamento de uma aplicação de simulação de algoritmos de escalonamento de processos, concebida no âmbito da Unidade Curricular de Sistemas Operativos. Os principais objetivos do projeto são:

1. Estudar e comparar diferentes algoritmos de escalonamento;
2. Avaliar métricas de desempenho, designadamente *turnaround time*, *waiting time* e *response time*;
3. Visualizar o comportamento de cada algoritmo através de uma interface gráfica;
4. Investigar o impacto de diferentes conjuntos de processos, gerados aleatoriamente, no desempenho dos algoritmos.

Espera-se, com este trabalho, proporcionar uma compreensão prática das vantagens e limitações de cada algoritmo de escalonamento, bem como uma ferramenta interactiva que facilite a análise dos mesmos.

Capítulo 2

Estrutura da Aplicação

2.1 Módulo de Geração de Processos

O Módulo de Geração de Processos é responsável pela criação dos processos que serão submetidos ao escalonador. Existem duas formas de obtenção:

Modo Aleatório: Utiliza um modelo de Poisson para gerar os *arrival times* dos processos, em que os *inter-arrival times* são amostrados através uma distribuição exponencial de parâmetro λ . Num processo de Poisson, o número de chegadas num intervalo de duração t segue uma $\text{Poisson}(\lambda t)$. De forma análoga, para o *burst time* de cada processo é usado uma distribuição exponencial, produzindo predominantemente *burst times* curtos, com algumas exceções mais longas. A cada novo processo são atribuídos:

- Um PID unico incremental;
- Prioridade selecionada por meio de uma *weighted distribution*;
- Período igualmente por meio de uma *weighted distribution*;

Leitura Estática: carrega processos pré-definidos de um ficheiro JSON (“`staticProcesses.json`”), contendo os mesmos campos de cima.

2.2 Módulo do *Clock*

O módulo do *Clock* centraliza a lógica de tempo da simulação e garante a coerência entre o tempo virtual e o tempo real. Os pontos mais relevantes são:

- **Velocidade de Simulação:** O `tick` define o fator de aceleração em relação ao tempo real. Por exemplo, `tick = 5` faz com que cada segundo real seja simulado como cinco segundos de sistema.

- **Ticks Discretos:** A execução do *scheduler* avança em passos fixos (ticks), correspondentes a um segundo de simulação de cada vez. Após cada tick, o *Clock* avança e novos processos entram para a *ready queue* tendo em conta os seus *arrival times*.
- **Separação de Clocks:** Para manter a interface grafica fluida, utilizam-se duas threads distintas:
 - *Clock* de Simulação: executa o avanço do tempo virtual em ticks, fazendo *sleep* entre cada passo sem bloquear o restante programa.
 - *Clock* Real: corre na *main thread*, combinando o tempo de simulação com o tempo real decorrido para apresentar um relógio contínuo e fluido na interface.

Esta estrutura garante controlo sobre a dinâmica temporal da simulação, ao mesmo tempo que mantém a interface responsiva e corretamente sincronizada com o decorrer dos eventos internos.

2.3 Módulo de Escalonamento

O Módulo de Escalonamento concentra toda a lógica de gestão dos processos. Esta recebe os processos do Módulo de Geração de Processos e é responsável pelo seu escalonamento correto. Os pontos mais relevantes deste módulo são:

- **Receção de Processos:** Processos novos são inseridos na *ready queue* e notificados ao algoritmo ativo, de modo ao algoritmo avaliar o que fazer.
- **Ciclo de Escalonamento:** A cada *tick* (consistente com o Módulo de *Clock*), incrementa-se o tempo corrente, atualizam-se métricas de execução e decide-se, via `algorithm.schedule()`, qual o próximo processo a correr.
- **Estados de Processo:** Cada processo transita entre *Ready*, *Running* e *Completed*. — quando o processo é concluído, calculam-se as métricas de *turnaround* e *waiting times*. — em preempções (*time quantum* expirado ou por prioridade superior), interrompe-se o processo atual e reintroduz-no na *ready queue*, fazendo em seguida o escalonamento correto.
- **Emissão de Sinais:** A cada atualização relevante, são emitidos sinais para atualizar a interface gráfica.
- **Algoritmos Implementados:** Os algoritmos suportados pelo nosso escalonador são:

- *First-Come, First-Served (FCFS)*
- *Shortest Job First (SJF)*
- *Round Robin (RR)*
- *Priority Scheduling* (preemptivo e não-preemptivo)
- *Multilevel Queue*
- *Earliest Deadline First (EDF)*
- *Rate Monotonic (RM)*

Este *design* permite comparar, de forma directa e consistente, o comportamento e as métricas de cada estratégia de escalonamento.

2.4 Interface Gráfica

A interface gráfica permite ao utilizador configurar a simulação, visualizar os processos ativos e concluídos, e acompanhar estatísticas de desempenho através de gráficos em tempo real. Para o seu desenvolvimento foi utilizada a biblioteca *PyQt6*.

A aplicação é composta por duas janelas principais:

- **Configuration Window:** Nesta janela, o utilizador pode definir todos os parâmetros da simulação, nomeadamente:
 - Configurações de geração de processos;
 - Configurações do *clock*;
 - Configurações do algoritmo de escalonamento.

Adicionalmente, é possível carregar configurações a partir de um ficheiro JSON.

- **Simulation Window:** Após a configuração, a *Simulation Window* apresenta a execução em tempo real do algoritmo de escalonamento. A interface encontra-se organizada em quatro painéis:
 - Processos em execução pelo escalonador;
 - Processos concluídos;
 - Relógio e evolução do sistema ao longo do tempo;
 - Gráficos de métricas médias e distribuição das mesmas.

Estes elementos são atualizados dinamicamente conforme a simulação progride, proporcionando uma análise visual do comportamento do sistema.

Este *design* interativo em tempo real visa facilitar a experiência do utilizador, permitindo acompanhar de forma clara e intuitiva o desempenho dos diferentes algoritmos de escalonamento.

2.5 Conclusão

Neste capítulo foi detalhada a arquitetura da aplicação, especialmente a sua divisão em módulos independentes e especializados: geração de processos, controlo de tempo, escalonamento e interface gráfica. Esta organização modular permitiu não só uma implementação mais limpa e eficiente, como também tornou o sistema facilmente extensível para futuras melhorias, como a adição de novos algoritmos ou parâmetros de simulação.

Capítulo 3

Como Utilizar a Aplicação

Disclaimer: Tivemos uns problemas em relação a responsabilidade da interface gráfica em resoluções menores. Esta está feita para ser utilizada em ecrãs 1920x1080 ou superior

3.1 Configuração Inicial

Antes de iniciar a simulação, é necessário definir corretamente as configurações da aplicação. Para tal, o utilizador deve preencher os seguintes parâmetros:

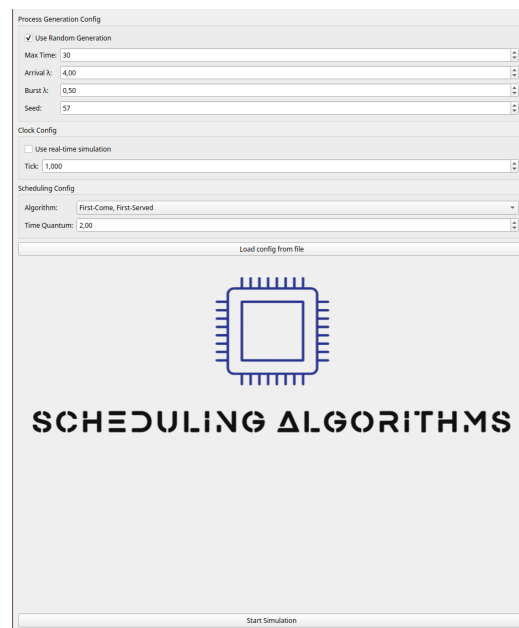


Figura 3.1: Janela de configuração da simulação

- **Process Generation Config:** Definir se pretende utilizar geração aleatória de processos, especificando o tempo máximo, o valor de λ para os tempos de chegada e para os tempos de execução, e ainda a *seed* para garantir a reprodutibilidade na simulação. Caso não utilizar a geração aleatória, vai ser carregado os processos em *staticProcesses.json*
- **Clock Config:** Definir a velocidade da simulação, ajustando o valor do *tick*.
- **Scheduling Config:** Selecionar o algoritmo de escalonamento desejado e configurar o *time quantum* se aplicável (por exemplo, no algoritmo *Round Robin*).

Alternativamente, é possível carregar uma configuração pré-definida, clicando no botão "*Load config from file*". O ficheiro deve estar em formato JSON e conter todos os parâmetros necessários para a execução da simulação.

3.2 Iniciar Simulação

Depois de configurar os parâmetros manualmente ou através de um ficheiro, o utilizador pode dar início à simulação clicando no botão *"Start Simulation"*.

3.3 Janela de Simulação

A interface da simulação foi projetada para fornecer uma visualização completa e intuitiva do comportamento dos diferentes algoritmos de escalonamento, dividida em várias secções que permitem o monitoramento em tempo real da execução dos processos, métricas de desempenho e estatísticas.

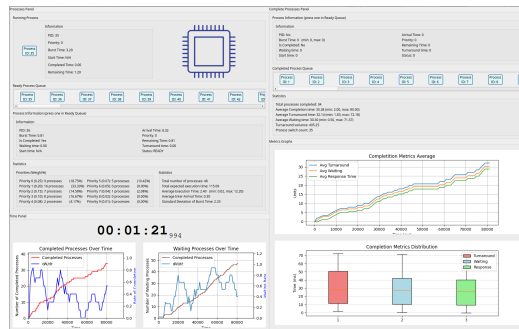


Figura 3.2: Janela de Simulação

3.4 Painel de Processos

3.4.1 Processo em Execução

Neste painel podemos visualizar informações detalhadas sobre os processos que estão a ser executados em tempo real pelo escalonador. O painel fornece uma visão abrangente com as seguintes informações:

- O processo atualmente em execução, com todos os seus parâmetros e características;
- A *ready queue* contendo todos os processos aguardando execução;
- Uma secção interativa que permite seleccionar qualquer processo da *ready queue* para visualizar os seus parâmetros;
- Uma secção de estatísticas que apresenta métricas sobre o comportamento do escalonador e dos processos;

Esta visualização em tempo real possibilita o acompanhamento completo do ciclo de vida dos processos no sistema, desde a sua chegada até à conclusão da execução.

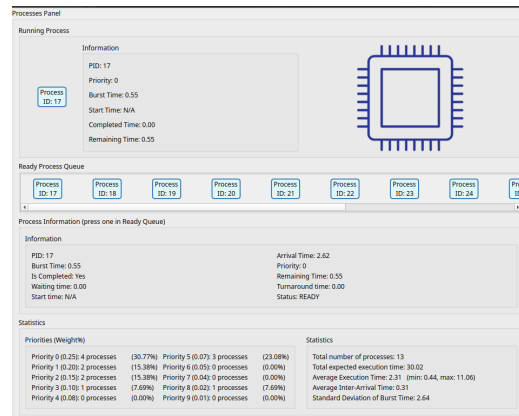


Figura 3.3: Painel de Processos

3.5 Painel de Tempo

No Painel de Tempo podemos observar informações relacionadas com a progressão temporal da simulação e o comportamento dos processos ao longo do tempo. Este painel fornece uma visualização dinâmica que permite analisar:

- O tempo atual de simulação exibido em formato horas:minutos:segundos e milissegundos
- Gráficos detalhados sobre a evolução dos processos ao longo do tempo
- Métricas de desempenho representadas visualmente

O painel inclui dois gráficos principais que ilustram o comportamento do escalonador ao longo do tempo:

- **Completed Processes Over Time:** Mostra o número acumulado de processos concluídos e a taxa de conclusão dos processos (dN/dt) ao longo do tempo. Este gráfico permite avaliar a eficiência do escalonador em termos de *throughput*.
- **Waiting Processes Over Time:** Apresenta o número de processos em espera e a taxa de variação de processos em espera (dN/dt) ao longo do tempo. Este gráfico ajuda a identificar potenciais congestionamentos na fila de processos.

Estes gráficos são fundamentais para a análise do desempenho do algoritmo de escalonamento, permitindo visualizar padrões, tendências e anomalias que podem não ser evidentes apenas com estatísticas numéricas.

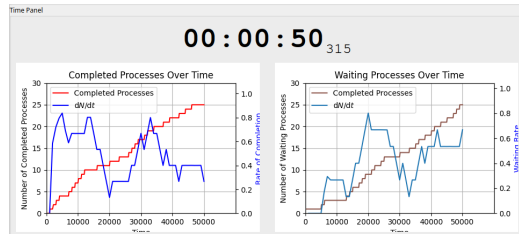


Figura 3.4: Painel do Tempo

A análise destes gráficos permite identificar momentos críticos na execução, como picos de processos em espera ou variações na taxa de conclusão, fornecendo uma compreensão melhor do algoritmo de escalonamento utilizado.

3.6 Painel de Processos Completos

Neste painel podemos visualizar informações detalhadas sobre os processos que já foram concluídos pelo escalonador. O painel fornece uma visão abrangente com as seguintes informações:

- Similar ao primeiro painel, uma secção interativa que permite seleccionar qualquer processo da *completed queue* para visualizar os seus parâmetros;
- A *completed queue* contendo todos os processos já completos;
- Uma secção de estatísticas que apresenta métricas sobre a performance do algoritmo;



Figura 3.5: Painel de Processos Completos

3.7 Gráficos de Métricas

Neste painel podemos visualizar representações gráficas detalhadas das principais métricas de desempenho dos algoritmos de escalonamento. O painel fornece visualizações analíticas com as seguintes informações:

- Gráficos de evolução temporal das métricas médias de *turnaround*, espera e resposta;
- Diagramas de caixas mostrando a distribuição estatística das métricas;

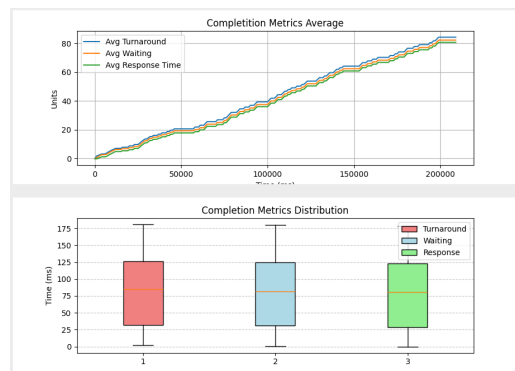


Figura 3.6: Painel dos Gráficos

Estas visualizações são fundamentais para uma análise aprofundada do comportamento do algoritmo, permitindo identificar não apenas os valores médios, mas também a variabilidade e consistência no tratamento dos processos.

Capítulo 4

Conclusão

Neste trabalho, abordamos a implementação e análise de algoritmos de escalonamento de processos, com o objetivo de entender como diferentes estratégias afetam a execução e o desempenho dos processos em um sistema operacional.

Em conclusão, este trabalho forneceu uma visão prática sobre como os algoritmos de escalonamento operam e como suas decisões impactam o desempenho de sistemas reais. Além disso, destacou a importância da análise cuidadosa dos processos e das condições de carga para obter um escalonador eficiente.