
	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

ALGORITMOS DE ESCALONAMENTO DE TAREFAS: IMPLEMENTAÇÃO E ANÁLISE

Objetivo do trabalho prático: aplicar o conhecimento teórico obtido na disciplina de Sistemas Operacionais (12035) para implementar e analisar diferentes algoritmos de escalonamento de tarefas, analisando na prática. Além dos conceitos de algoritmos de escalonamento de tarefas, serão também implementados conceitos relacionados à comunicação entre processos, mais especificamente, a utilização de *sockets*.

Linguagens de programação que podem ser utilizadas no desenvolvimento do trabalho prático: Python 3.x; C; ou C++11. (Para utilização de linguagens de programação além das 3 estabelecidas, o aluno deve entrar em contato com o professor por e-mail, para obter autorização de utilização).

SO a ser utilizado: Windows 10 ou superior; Distribuições Linux com *release* de kernel posteriores ao ano de 2022; macOS versão 14 ou superior.

Pontuação máxima: 7,0 pontos (correspondentes à nota parcial de composição da 3ª avaliação periódica) + 1,0 ponto extra (a ser alocado como pontuação complementar na nota de uma das duas provas escritas).

Modo de realização: em duplas (preferível) ou individual.

Data final de entrega: 01 de agosto de 2025, às 23h59min, exclusivamente via Google Classroom da disciplina.

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

Descrição do Trabalho Prático

Este trabalho prático consiste na simulação de um serviço de escalonamento prestado por um **escalonador de tarefas** (*task scheduler*) num SO com características *batch*, ou seja, que não possuem interação com o usuário. A simulação realizada deve prover a análise de execução de tarefas entre diferentes algoritmos de escalonamento, baseado no cálculo de métricas de tempo de execução (*turnaround time*) e tempo de espera (*waiting time*), além da observação da linha do tempo de execução de tarefas no processador.

Devem ser implementados 7 (sete) diferentes algoritmos de escalonamento, de acordo com o material bibliográfico de Maziero (2019):

- First-Come, First-Served (FCFS)
- Round-Robin (RR) com *quantum* fixo de 3 unidades de *clock*
- Shortest Job First (SJF)

- Shortest Remaining Time First (SRTF)
- Escalonamento por prioridades fixas cooperativo (PRIOc)
- Escalonamento por prioridades fixas preemptivo (PRIOp)
- Escalonamento por prioridades dinâmicas (PRIOd)

Além dos algoritmos, a simulação deve considerar três componentes distintos, que devem ser implementados como **processos separados** e devem se comunicar por meio de sockets. Os componentes são descritos a seguir:

- Clock:

- Responsável por simular o clock da CPU.
- O clock deve ser inicializado em 0 e ser incrementado sempre em 1 unidade de tempo.
- Deve haver um delay de 100ms por incremento, simulando o avanço da linha do tempo.
- A cada incremento, o clock deve enviar uma mensagem:

■ Primeiro ao Emissor de Tarefas;

2

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

- Após 5ms, ao Escalonador de Tarefas (isso garante que o Emissor insira as tarefas antes do Escalonador tentar escaloná-las).
- Emissor de tarefas: esse componente deve informar o escalonador sobre as tarefas que estão prontas para serem executadas para a fila de tarefas prontas.
 - A emissão é baseada na leitura de um arquivo de entrada informado na execução do processo. Cada linha do arquivo representa uma tarefa, no formato:
 - ID;tempo de ingresso;duração prevista;prioridade
 - Exemplo:

- `t0;0;5;1`

- **Atenção:** quanto **menor** o valor numérico de prioridade informado, **maior** a prioridade para execução. Ou seja: uma tarefa com prioridade 1 tem prioridade **maior** de execução do que uma tarefa com prioridade 2.

- Com cada novo valor de clock recebido via socket, o Emissor verifica se uma ou mais tarefas devem ser inseridas na fila de prontas.

■ Exemplo:

- Se a tarefa t1 tiver tempo de ingresso 5, ela só será enviada ao Escalonador quando o clock chegar em 5.

- Quando o Emissor inserir a última tarefa, deve enviar uma mensagem ao Escalonador informando que todas as tarefas já foram emitidas.

● Escalonador de tarefas:

- Implementa todos os algoritmos de escalonamento listados acima.
- O algoritmo ativo é determinado por um argumento de entrada, com os seguintes identificadores:

- `fcfs, rr, sjf, srtf, prioc, priop, priod`

3

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

- A cada novo valor de *clock* recebido via *socket*, o Escalonador executa o algoritmo ativo e seleciona qual tarefa deve ocupar o processador.
- Após o término da última tarefa, o Escalonador deve:
 - Enviar uma mensagem ao Clock e ao Emissor sinalizando o fim da simulação;
 - Escrever o arquivo de saída com os dados da execução.
- O arquivo gerado pelo Escalonador ao final da simulação deve conter, exatamente nesta ordem:
 - Linha com a sequência de tarefas escalonadas a cada unidade de clock, separadas por “;”

○ Exemplo:

■ $t_0; t_0; t_0; t_1$

● Uma linha por tarefa, contendo (separados por “;”) :

ID;clock de ingresso na fila de prontas;clock de finalização;turnaround time;waiting time

● Por fim, a média dos tempos de execução e espera, separados por “;” e arredondados para cima com 1 casa decimal.

○ Exemplo:

■ 6.0;2.7

Vamos considerar um cenário hipotético para ficar mais claro como as informações devem ser escritas. Suponha uma simulação do algoritmo FCFS com 3 tarefas: t_0 , t_1 e t_2 . Nessa simulação, a tarefa t_0 ingressa na fila de prontas na unidade de tempo 0, a tarefa t_1 em 1, e a tarefa t_2 em 3. O tempo previsto de execução de t_0 é 5, de t_1 é 2, e de t_2 é 3. Nesse caso, o arquivo de saída deveria ser o seguinte:

$t_0; t_0; t_0; t_0; t_0; t_1; t_1; t_2; t_2; t_2$
 $t_0; 0; 5; 5; 0$
 $t_1; 1; 7; 6; 4$
 $t_2; 3; 10; 7; 4$
6.0;2.7

4

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

Estruturação dos componentes do trabalho prático

A estruturação dos componentes a serem implementados neste trabalho prático podem (e devem) ser planejados pela dupla (ou pelo aluno solo). No entanto, algumas estruturas básicas devem ser observadas. A Figura 1 apresenta um esquema básico dessas estruturas, que serão detalhados a seguir. Tenha em mente que o esquema e descrição apresentados servem apenas como um guia para o desenvolvimento do trabalho, na qual as decisões de projeto são de responsabilidade da dupla (ou do aluno solo).

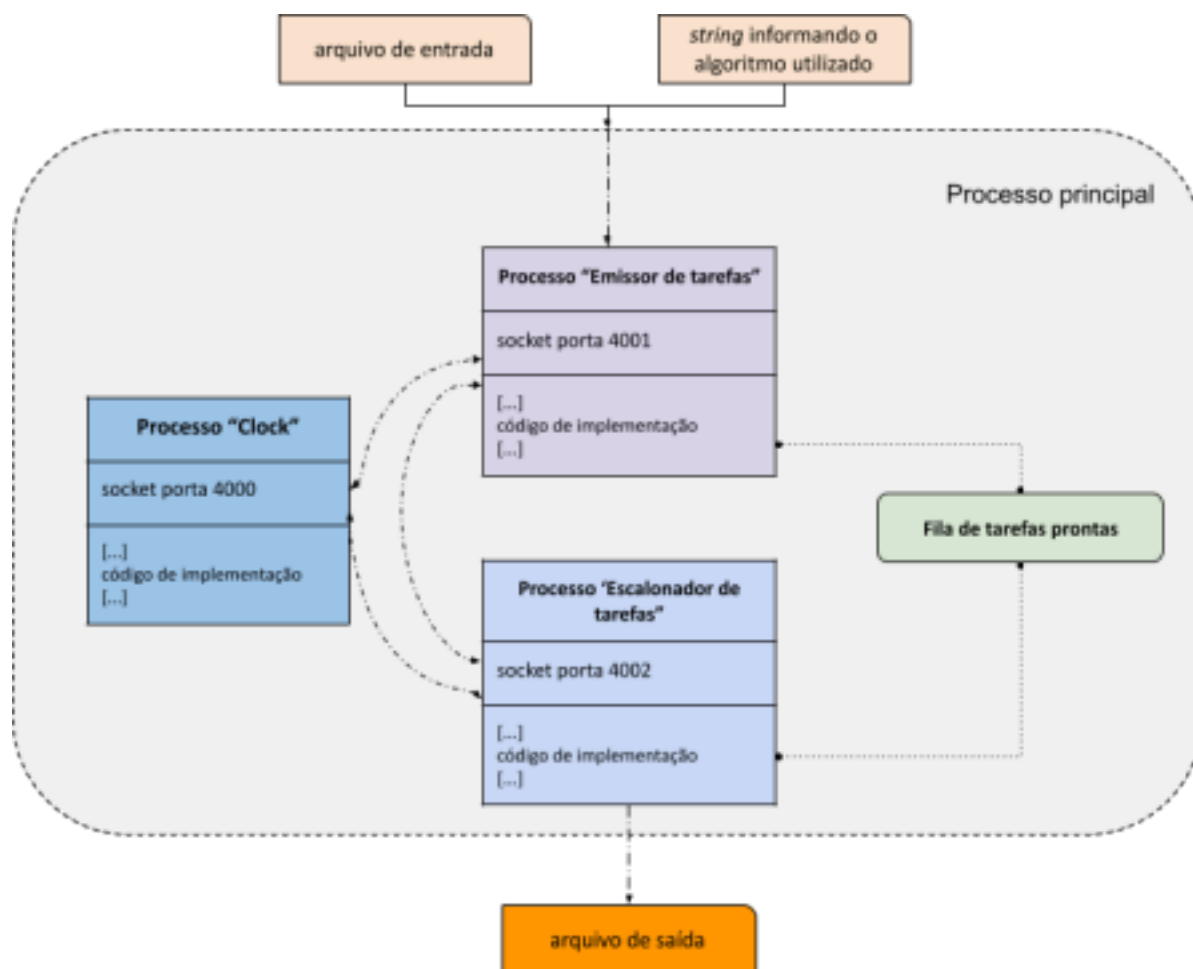


Figura 1: Esquemático dos componentes básicos do trabalho prático
A seguir, tem-se a discussão dos componentes:

5

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

- Ao se executar o código implementado, deve-se considerar 2 argumentos de entrada:
 - 1º argumento: caminho relativo do arquivo que contém a descrição das tarefas de simulação
 - 2º argumento: indicação do algoritmo de escalonamento a ser utilizado na simulação, seguindo a seguinte nomenclatura:

■ `fcfs`: First-Come, First-Served (FCFS)

- `rr`: Round-Robin (RR) [considere sempre o quantum = 3]
 - `sjf`: Shortest Job First (SJF)
 - `srtf`: Shortest Remaining Time First (SRTF)
 - `prioc`: Escalonamento por prioridades fixas cooperativo (PRIOc)
 - `priop`: Escalonamento por prioridades fixas preemptivo (PRIOp)
 - `priod`: Escalonamento por prioridades dinâmicas (PRIOd)
- Após a inicialização e leitura dos argumentos de entrada, o programa implementado deve criar e inicializar os 3 processos anteriormente descritos, representando os componentes desse trabalho: “Clock”; “Emissor de Tarefas”; “Escalonador de Tarefas”;
 - Para as portas dos *sockets*, considere a seguinte numeração:
 - Clock: porta 4000
 - Emissor de Tarefas: porta 4001
 - Escalonador de Tarefas: porta 4002
 - Os componentes “Emissor de Tarefas” e “Escalonador de Tarefas” devem aguardar a comunicação do componente “Clock” via *socket* para realizar qualquer tipo de ação.

6

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

- Quando o componente “Emissor de Tarefas” inserir na fila de tarefas prontas a última tarefa (considerando o arquivo descritivo de entrada), ele deve enviar uma mensagem *via socket* para o “Escalonador de Tarefas” informando que todas as tarefas já foram emitidas;
- Após a execução da última tarefa, o “Escalonador de tarefas” deve enviar uma

mensagem *via socket* tanto para o “Emissor de Tarefas” quanto para o componente “Clock” informando que eles podem ser encerrados, pois a simulação chegou ao fim. Nesse momento, é o “Escalonador de tarefas” quem deve escrever os dados no arquivo de saída.

Itens a serem entregues

A data final de entrega do trabalho prático é dia 01 de agosto de 2025, às 23h59min, exclusivamente via Google Classroom da disciplina. **Não haverá prorrogação do prazo.** A entrega do trabalho prático deve ser feita por todos os alunos, inclusive se o trabalho foi desenvolvido em duplas. A entrega deve consistir num arquivo zipado contendo os itens descritos a seguir:

- **Código fonte:** todo o código fonte implementado na linguagem escolhida deve constar na entrega. Obrigatoriamente, o código fonte deve conter:
 - A implementação dos 3 componentes básicos (Clock, Emissor de Tarefas e Escalonador de Tarefas);
 - A implementação dos 7 algoritmos de escalonamento;
 - A utilização de uma estrutura para representar a fila de tarefas prontas;
 - A utilização da estrutura de *sockets*;

A implementação das demais operações relacionadas ao trabalho (como a leitura dos dados de entrada e a escrita dos dados de saída) são decisões de projeto da dupla (ou do aluno solo).

- **Manual de compilação/execução:** Deve ser entregue um arquivo no formato PDF com as instruções de compilação (caso o código necessite ser compilado) e execução do código fonte. A execução do código desenvolvido por parte do professor avaliador seguirá o que constar neste manual. Caso o manual não forneça informações


7

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

suficientes para compilação/execução do código fonte desenvolvido, impossibilitando a execução do mesmo, não será atribuída pontuação para o desenvolvimento do código fonte.

- **Relatório técnico de desenvolvimento do trabalho:** Deve ser escrito um relatório técnico sobre o trabalho, constando os seguintes itens:
 - Discussão das decisões de projeto: Listagem e descrição das estruturas básicas e das funções utilizadas. Dentre outros aspectos, recomenda-se fortemente que as seguintes decisões de projeto sejam discutidas:
 - Como foi implementada e gerenciada a fila de tarefas prontas?
 - Como foi utilizada a estrutura de *socket*? Quais as dificuldades enfrentadas?
 - Foi necessário utilizar alguma estrutura adicional? Se sim, discuta a necessidade delas. (exemplo: mais filas, variáveis globais, mecanismos de sincronização, etc.)
 - Como foi realizada a leitura dos dados de entrada e a escrita dos dados de saída?
 - Discussão técnica sobre o funcionamento dos algoritmos implementados: baseado no arquivo de entrada “*entrada00.txt*” (disponibilizado no Google Classroom), deve ser apresentado o conteúdo do arquivo de saída proveniente da execução de cada algoritmo implementado. **Não insira os dados do arquivo de maneira “crua”, faça uma tabulação dos dados.** A partir dos dados tabulados, deve ser realizada uma análise técnica sobre o resultado obtido com as métricas, relacionando-os com as características de cada algoritmo.
 - Diagramas de Gantt [pontuação extra]: Conforme visto ao longo da disciplina, a utilização do diagrama de Gantt fornece uma melhor visualização das tarefas que estão escalonadas para execução a cada ciclo de *clock*. Desenvolver o código fonte e inserir os diagramas na discussão técnica proverá à dupla (ou ao aluno solo) pontuação extra no trabalho.

8

	UNIVERSIDADE ESTADUAL DE MARINGÁ CENTRO DE TECNOLOGIA DEPARTAMENTO DE INFORMÁTICA	
Disciplina 12035 - Sistemas Operacionais (Ano letivo de 2025, 1º Semestre)		
Bacharelado em Ciência da Computação		
Professor Dr. Alisson Renan Svaigen		
<u>Trabalho Prático da Disciplina: Algoritmos de Escalonamento</u>		

Critérios de avaliação

Os 7,0 pontos regulares do trabalho + 1,0 ponto extra estão distribuídos conforme a tabela a seguir.

Item avaliado	Pontuação
<i>Implementação</i>	
Implementação correta do algoritmo FCFS	0,5
Implementação correta do algoritmo RR	0,5
Implementação correta do algoritmo SJF	0,5
Implementação correta do algoritmo SRTF	0,5
Implementação correta do algoritmo PRIOc	0,5
Implementação correta do algoritmo PRIOp	0,5
Implementação correta do algoritmo PRIOd	0,5
Implementação e utilização correta da fila de tarefas prontas	0,5
Implementação correta dos demais itens do código fonte	0,5
Pontuação total da parte de implementação	4,5
<i>Manual + Relatório Técnico</i>	
Apresentação de manual de compilação/execução com descrição sugestiva e em acordo com o código desenvolvido	0,25
Discussão das decisões de projeto no relatório técnico	0,75
Discussão técnica sobre funcionamento dos algoritmos implementados no relatório técnico	1,5
Apresentação dos diagramas de Gantt [extra]*	1,0
Pontuação total da parte de relatórios e manual	2,5 + 1,0 [extra]
Total Geral	7,0 + 1,0 [extra]

* A pontuação extra será alocada como nota adicional na pontuação da avaliação escrita de menor valor 9