Simulador de Processos

MAC0422 - EP1

Leonardo Heidi Almeida Murakami - 11260186

Algoritmo de Escalonamento com Prioridade

LST - Least Slack Time

Este escalonador gerencia processos com o objetivo de maximizar o cumprimento de deadlines. A prioridade é dinâmica e baseada na urgência: calcula-se a "folga/slack" de cada processo (`deadline - tempo_atual - tempo_restante`). Quanto menor a folga, maior a prioridade, indicando que o processo está mais próximo de perder sua deadline.

Para garantir que as tarefas mais críticas tenham acesso à CPU, o algoritmo é preemptivo. Um processo em execução pode ser interrompido se outro processo mais urgente (com menor folga) se tornar pronto e não houver CPUs livres.

Limitação Principal: A principal limitação do LST vem do overhead que ele adiciona, pois, frequentemente calcular a folga de todos os processos pode se tornar inviável dependendo da quantidade de processos. Para a simulação, dada a limitação de processos

Deterministicidade

Sobre a não deterministicidade do código

Este simulador utiliza o tempo real do sistema para medir a passagem do tempo e a duração da execução dos processos. Embora a lógica dos escalonadores seja definida, o uso do tempo real introduz **não-determinismo**. Fatores externos como a carga do sistema, o escalonamento de threads pelo sistema operacional e variações na execução do processo que consome tempo real podem alterar ligeiramente os tempos medidos e a ordem exata dos eventos entre diferentes execuções, mesmo com a mesma entrada.

Um simulador puramente baseado em tempo lógico (simulado), onde o avanço do tempo é controlado internamente sem depender do relógio real, seria totalmente determinístico. Como resultado, pequenas variações nos tempos finais e no número de preempções podem ser observadas nos resultados, mas, serão variações pequenas.

Computadores usados para teste

Máquina A

AMD Ryzen 9 7900X 12-Core Processor (24 threads)

32 GB DDR5-6000 / PC5-48000 DDR5 SDRAM UDIMM

MSI GeForce RTX 5070 Ti VENTUS 3X OC

Para que os testes sejam viáveis, o código foi modificado para poder receber uma flag opcional --cpu que controla quantas cpus serão usadas pelo scheduler.

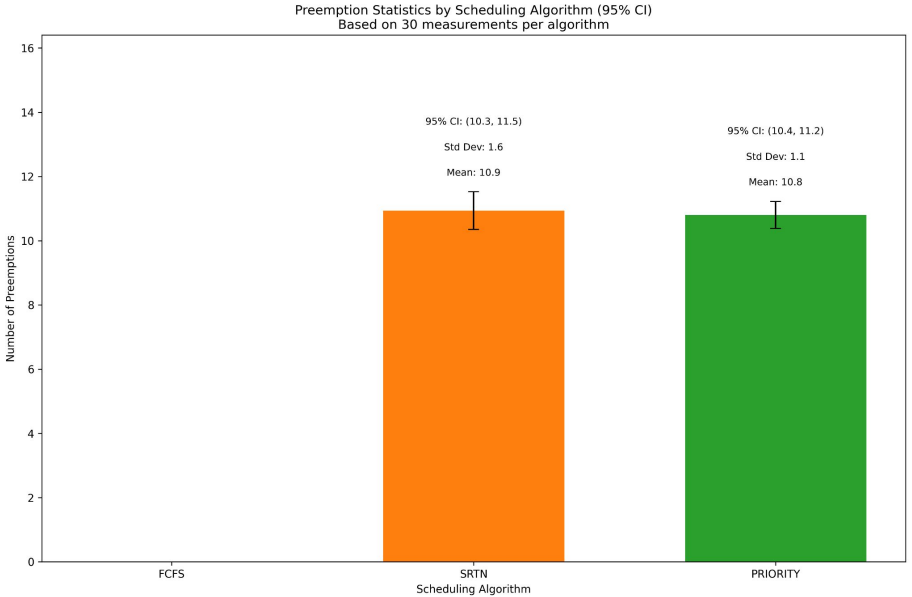
Para esta máquina, limitei em 6 para que houvesse competição pelas threads

Máquina B

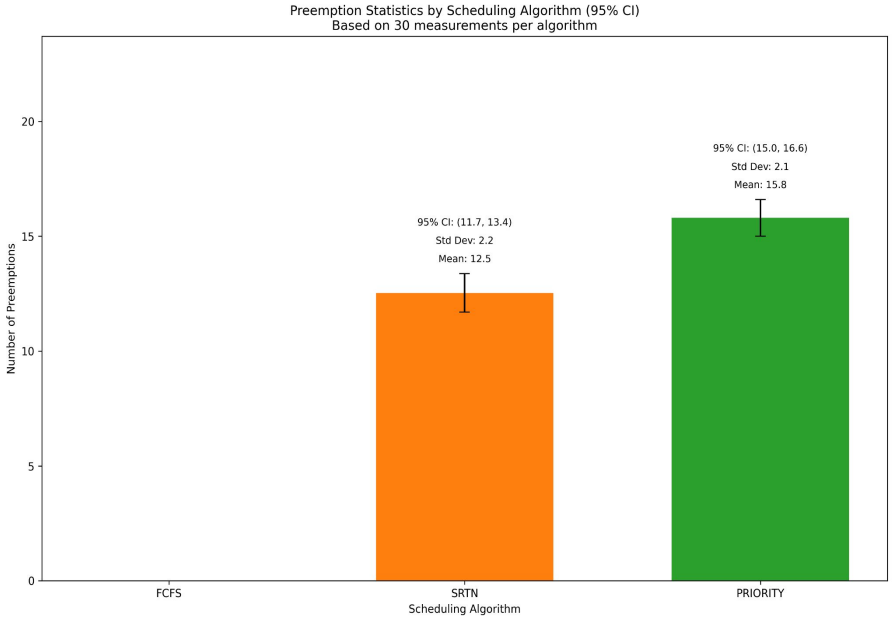
AMD EPYC™ 9634 (4 dedicated cores)

8 GB DDR5 RAM (ECC)

Preempção - Entrada Esperada

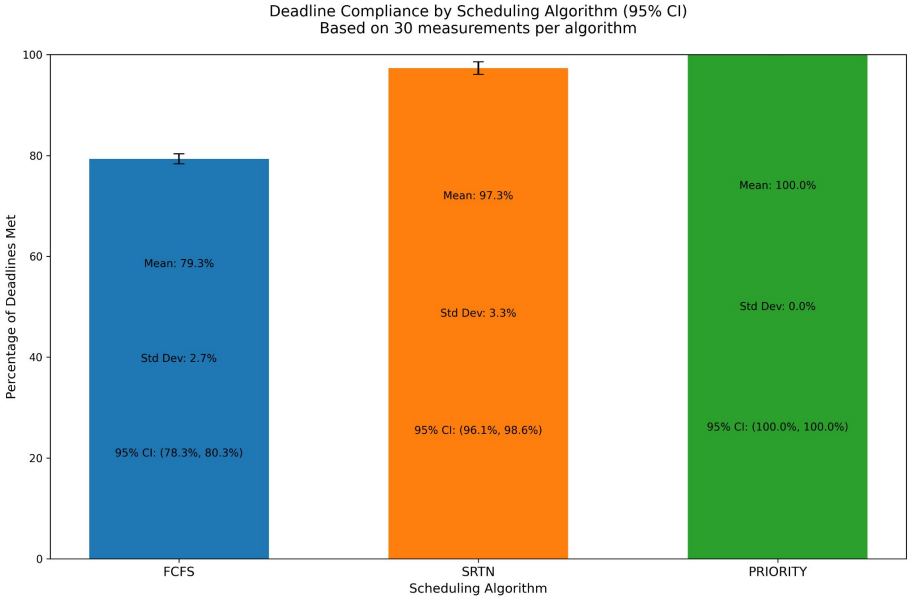


Máquina A

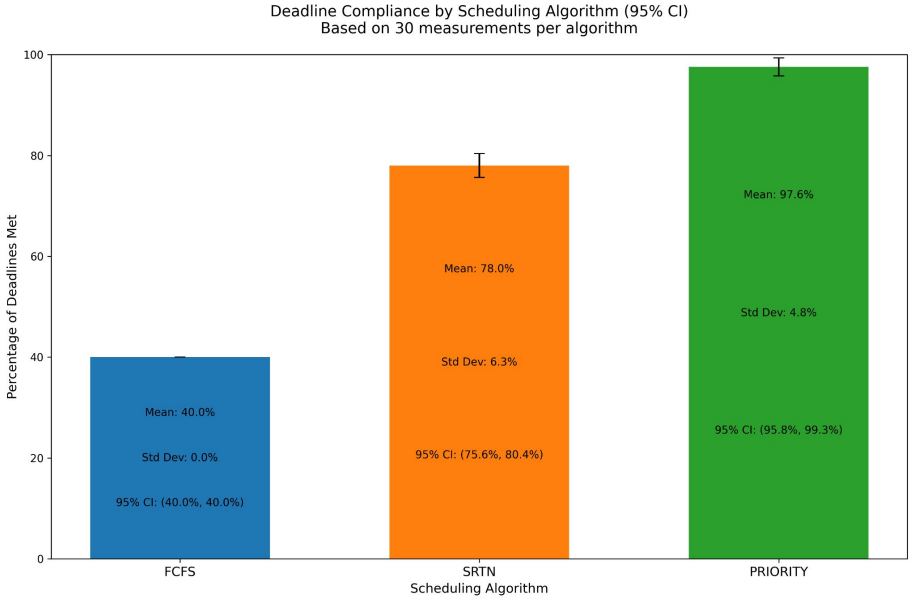


Máquina B

Deadlines - Entrada Esperada

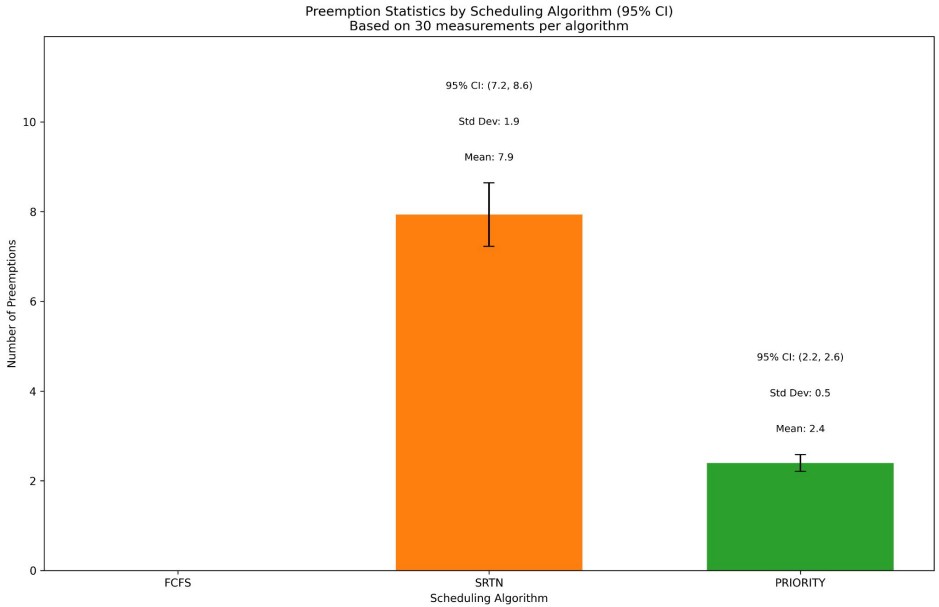


Máquina A

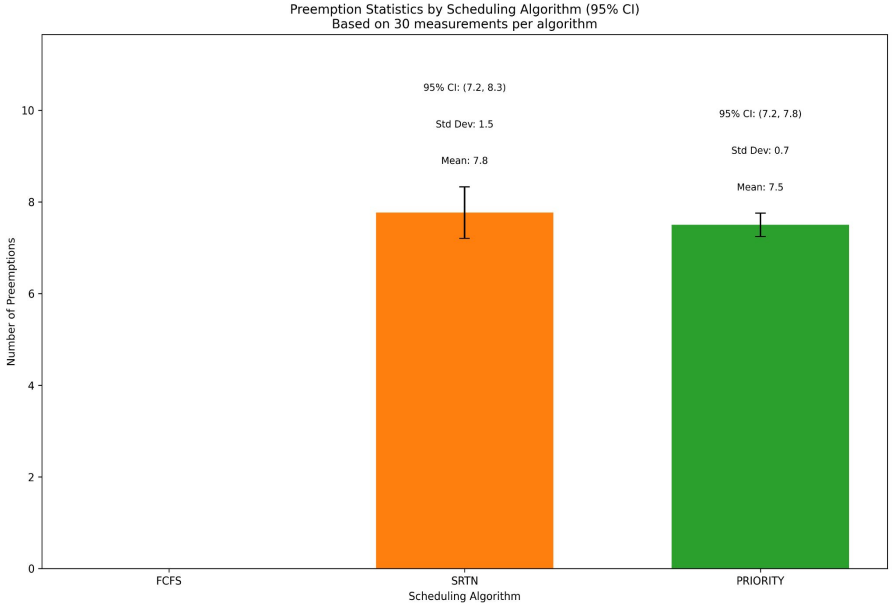


Máquina B

Preempção - Entrada Inesperada

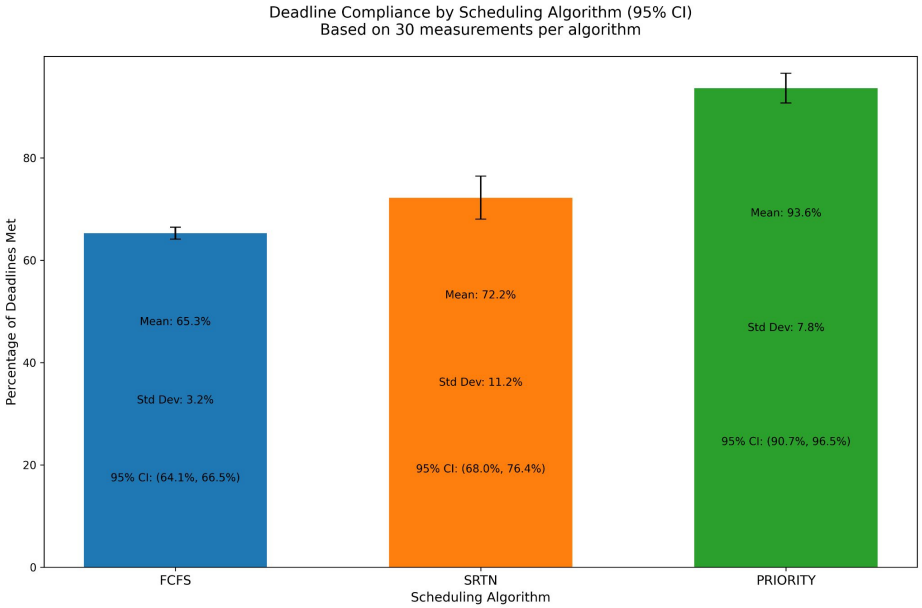


Máquina A

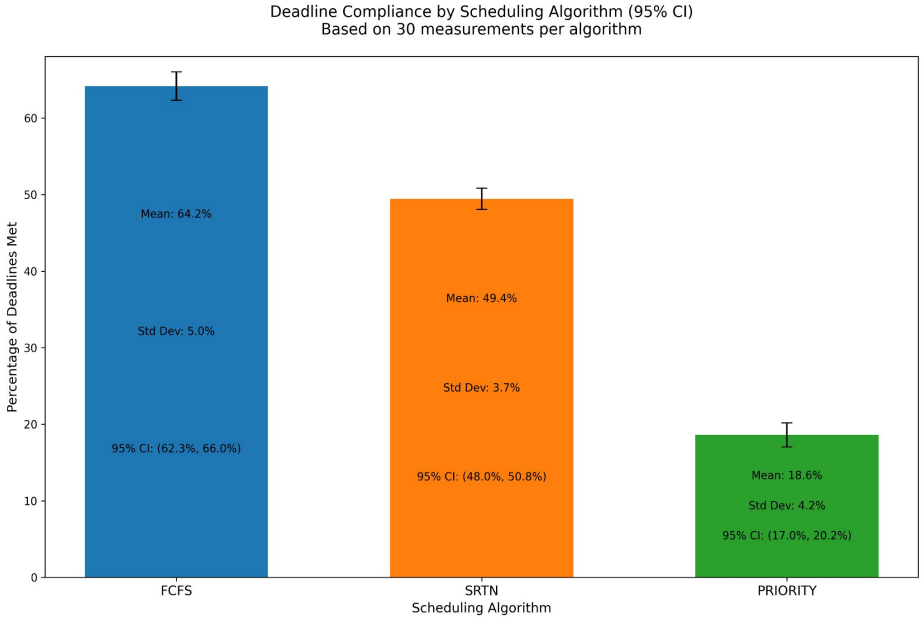


Máquina B

Deadlines - Entrada Inesperada



Máquina A



Máquina B