

Relatório de Trabalho Prático 1

Trabalho feito por:

Daniel Espadinha a22407006

Rodrigo Beleza a22407392

Link repositório: <https://github.com/RodrigoBeleza/Aula3-2025.git>

Cenário	Métrica (segs)	FIFO	SJF	RR	MLFQ
1	Tempo Médio Execução	27,22	26,67	37,94	37,89
1	Tempo Médio Resposta	12,22	11,67	11,67	22,89
2	Tempo Médio Execução	22,28	15,5	24,06	24,03
2	Tempo Médio Resposta	15,78	9	9	17,53
3	Tempo Médio Execução	n.d	n.d	n.d	3,78
3	Tempo Médio Resposta	n.d	n.d	n.d	0
4	Tempo Médio Execução	n.d	n.d	n.d	3,72
4	Tempo Médio Resposta	n.d	n.d	n.d	0

Tabela 1 - Tempos médios de execução e de resposta para cada algoritmo e cenário

Cenário 1 (Processos com tempos relativamente longos: 10, 15 e 20 segundos):

FIFO

Processos são executados na ordem de chegada. O tempo médio de execução é relativamente alto, e a resposta inicial é decente, mas processos longos atrasam os seguintes.

SJF

Executa primeiro os processos mais curtos. Observa-se uma leve melhoria na execução média, porque os processos curtos terminam mais rápido, reduzindo a espera dos outros. A resposta média também é ligeiramente menor.

RR

O *time slice* fixo aumenta o tempo médio de execução porque parte do tempo da CPU é gasto a alternar entre processos. A resposta inicial de cada processo é semelhante à SJF porque são todos escalonados rapidamente, mas o total demora mais.

MLFQ

Aproxima-se de RR, mas organiza os processos que bloqueiam, por prioridade (ex.: I/O). O tempo de resposta aumenta bastante (22,89 s), provavelmente porque os processos da CPU longo são “rebaixados” na fila.

Resumo do Cenário 1: SJF é o algoritmo mais eficiente em termos de execução média, FIFO tem tempos decentes, mas sofre com processos longos, RR e MLFQ aumentam a execução devido à alternância e prioridade dinâmica.

Cenário 2 (Processos de tempos variados: 2–15 segundos):

FIFO

Executa na ordem de chegada. Média de execução decente, mas tempo de resposta pior do que SJF e RR para processos curtos, porque estes esperam mais tempo atrás de processos longos.

SJF

Ganha destaque aqui. Como os processos mais curtos são escalonados primeiro, a execução média cai significativamente e a resposta média para todos os processos melhora (9seg). Isto demonstra claramente a vantagem do SJF em processos de diferentes tamanhos.

RR

Tempo de execução um pouco maior do que o FIFO, mas tempo de resposta muito parecido com SJF. RR dá uma boa resposta inicial, mas penaliza a execução total devido a trocas de contexto.

MLFQ

Aproxima-se de RR em execução média, mas com tempo de resposta mais alto (17,53 s). Isso indica que os processos que não são interativos foram rebaixados, aguardando mais tempo antes de voltar à CPU.

Resumo do Cenário 2:

SJF é claramente superior em execução e resposta para estes cenários diferentes. FIFO e RR têm execuções mais altas, e MLFQ privilegia interatividade, aumentando a resposta para processos de CPU longo.

Cenários 3 e 4 (processos I/O intensivos):

Apenas **MLFQ** é relevante, porque estes cenários envolvem I/O e bloqueios. FIFO, SJF e RR não tiveram dados válidos ou não foram aplicáveis (por isso "n.d.").

MLFQ mostra tempos de execução médios muito baixos (3,78 e 3,72 s), e tempo de resposta praticamente zero. Isso ocorre porque MLFQ prioriza processos interativos e I/O, permitindo que estes processos que bloqueiam rapidamente voltem a ser executados, maximizando a responsividade.

Resumo dos Cenários 3 e 4:

O comportamento demonstra claramente a vantagem de escalonadores multilevel com prioridades dinâmicas em tarefas que misturam CPU e I/O ou em processos interativos. MLFQ é altamente eficiente para processos que alternam entre CPU e I/O, garantindo respostas instantâneas e execução rápida de pequenos blocos da CPU.

Conclusão:

- **SJF** é ótimo para execução média em processos CPU-bound com tempos variados;
- **FIFO** é simples, mas pode penalizar processos curtos se houver um longo à frente;
- **RR** é justo e garante boa resposta, mas aumenta o tempo total de execução devido à alternância;
- **MLFQ** brilha em cenários interativos ou com I/O, garantindo uma resposta quase instantânea, mas com execução média de CPU total ligeiramente maior.