



MAC0422 - EP1

Mariana Tiemi Silva Misu
12542842

Escalonamento com Prioridade

- Explicação do algoritmo:
 - Esse algoritmo é bem similar ao Round-Robin sendo que, ao contrário de um quantum constante, o quantum no Escalonamento com prioridade está determinado pela função $quant$, que encontra o valor do quantum por $\max(1, 5 - (dl - t)/5)$, dessa forma, o quantum é maior para processos com deadline menores e quanto mais passa o tempo.



Determinístico?

- O código do jeito que foi criado é determinístico, pois sempre deve produzir os mesmos resultados para os mesmos traces, e sendo assim, é possível deduzir os resultados antes de rodar o programa.

Determinístico?

- Para o a.txt: o Escalonamento com Processo será melhor pro cumprimento das deadlines, visto que ele vai priorizar a deadline do proc2;
- Para o b.txt: o Shortest Job First é melhor para o cumprimento das deadlines, pois os outros dois terminam realizando preempções e, sendo assim, não terminando os processos atrasando-os de forma a fazer com que alguns não cumpram com a deadline que seria cumprida se fossem finalizados sem a preempção;

Determinístico?

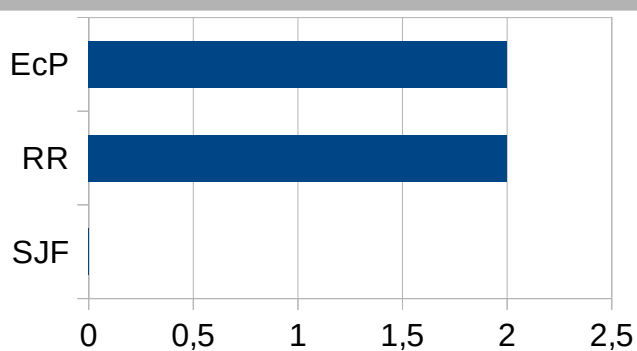
- Para o c.txt: o Round-Robin foi o melhor para o cumprimento das deadlines, pois com o Quantum=3 permite que ele realiza todas as tarefas dentro das suas deadlines, ao contrário dos outros escalonadores;
- OBS: todas essas observações sobre os testes foram feitas baseadas para um Quantum de 3 no Round-Robin;

Configuração das Máquinas para os testes

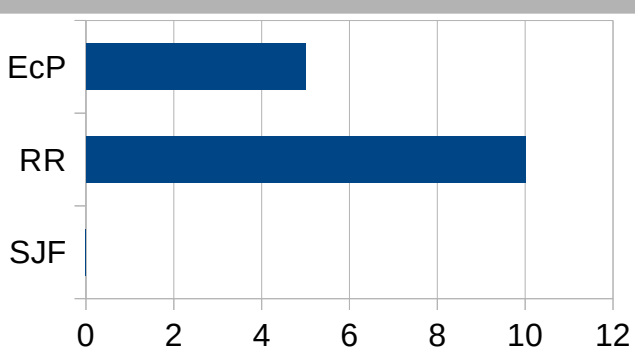
- Primeiro computador: x86_64 GNU/Linux, 11th Gen Intel(R) Core(TM) i7-11390H com 8 unidades de processamento;
- Segundo computador: com 4 núcleos;
 - Máquina principal: SO: macOS Sonoma 14.2.1 / Memória: 8GB / CPU: Apple M1 / GCC: gcc (Homebrew GCC 13.2.0) 13.2.0
 - VM: limactl version 0.20.1 / Linux: Linux version 6.1.0-18-cloud-amd64 (debian-kernel@lists.debian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils for Debian) 2.40) #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) / GCC: gcc (Debian 12.2.0-14) 12.2.0

Gráficos - Quantidade de mudanças de contexto

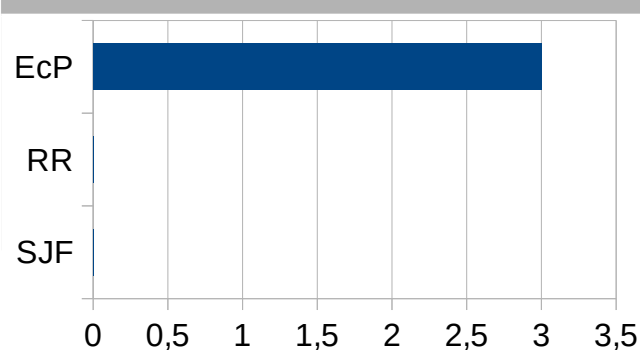
Máquina A- a.txt



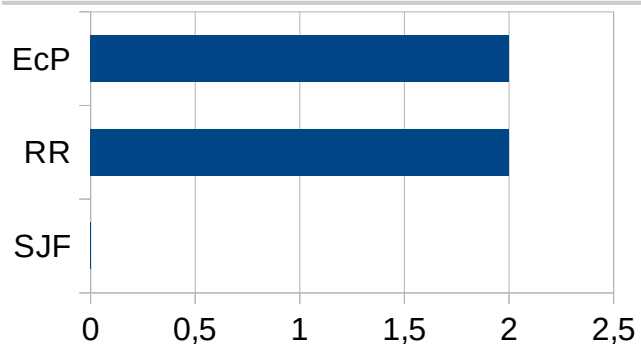
Máquina A- b.txt



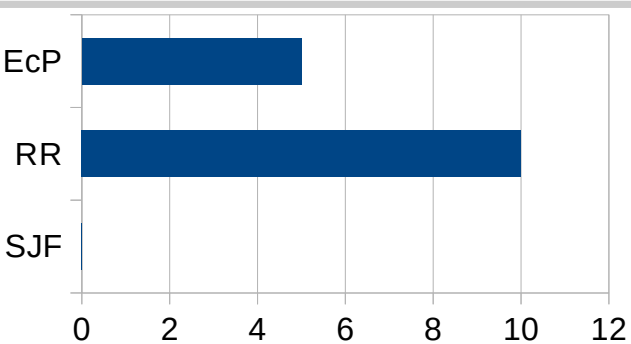
Máquina A- c.txt



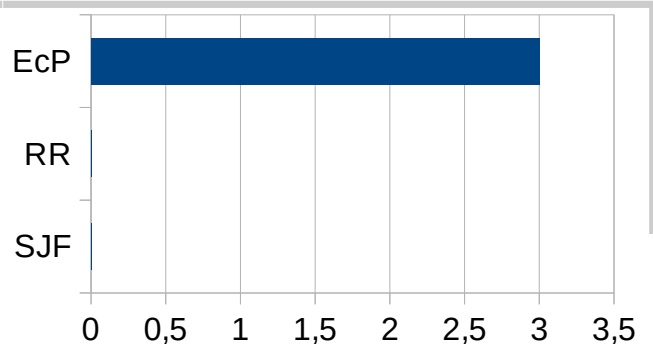
Máquina B- a.txt



Máquina B- b.txt

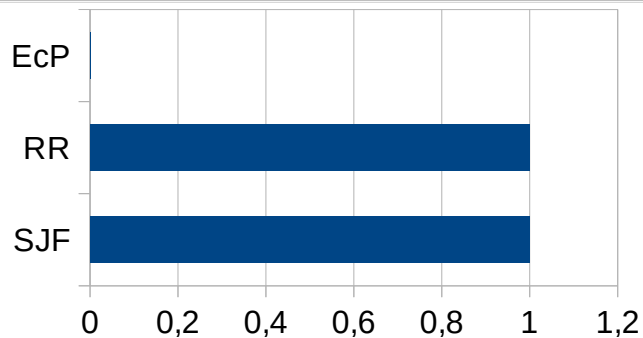


Máquina B- c.txt

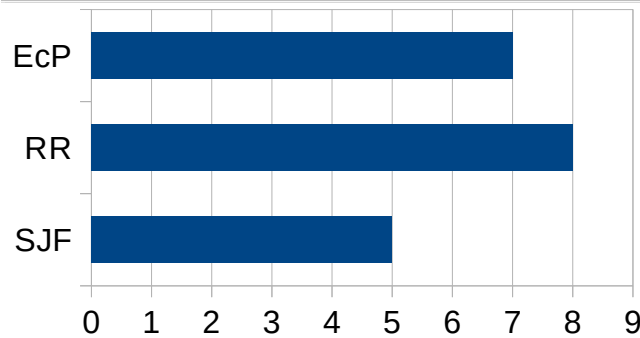


Gráficos - Cumprimento de deadline

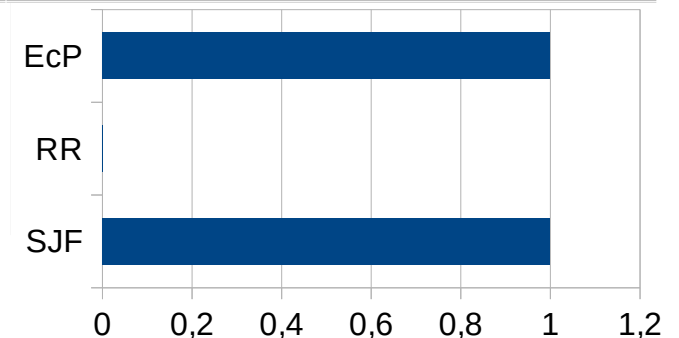
Máquina A- a.txt



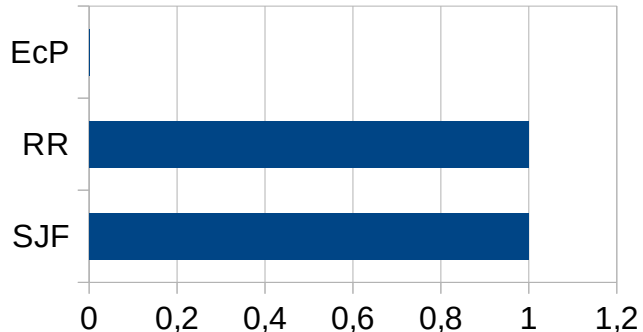
Máquina A- b.txt



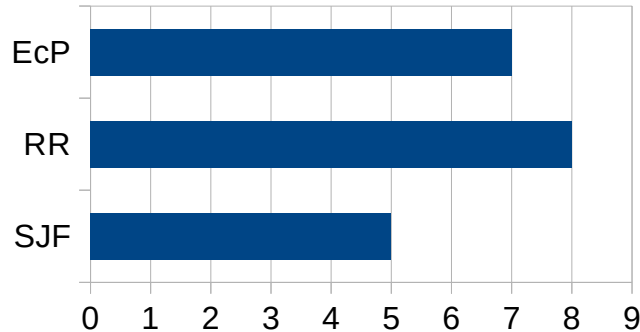
Máquina A- c.txt



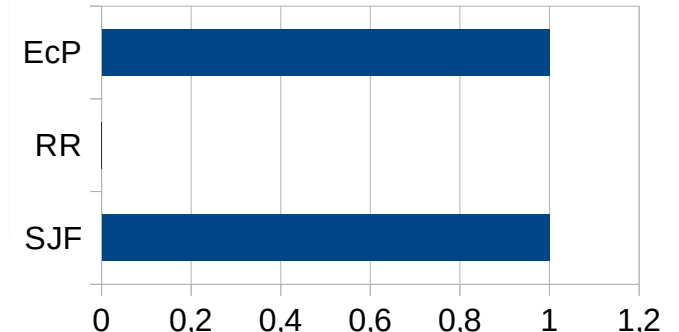
Máquina B- a.txt



Máquina B- b.txt



Máquina B- c.txt





Conclusão

- Como foi possível ver com o resultado igual dos gráficos nas duas máquinas, o código de fato demonstra-se ser um código determinístico, tendo que os resultados obtidos com o experimento eram os esperados dos 3 algoritmos de escalonamento.

Referências

- Os slides da disciplina (em especial os da Aula 5 (11/3))
- https://man7.org/linux/man-pages/man3/pthread_setaffinity_np.3.html
- <https://pt.wikipedia.org/wiki/Round-robin>