



EP1

Shell e Simulador de Processos

Isis Ardisson Logulo
7577410

SHELL

O newsh.c foi desenvolvido baseado na leitura de linhas da bibliotecas GNU readline e GNU history.

A configuração do prompt foi feita usando os syscalls:

- usuário → `getenv("USER")`.
- hora → `gettimeofday()`, usando também a `struct timeval`, fazendo uma transformação para o formato desejado.

Para a execução dos comandos, foram usadas as syscalls:

- `cd` → `chdir()`.
- `rm` → `unlink()`.
- `uname` → usando a `struct utsname` foi possível pegar os valores de `u.sysname`, `u.release`, `u.version` e `u.machine`.

Para a execução de bins e executáveis foi usado um `fork` com `execvp()`.

Simulador de Processos

Foram feitos 3 tipos de escalonadores:

- Shortest Job First

Usa pilha ordenada pelos dt's dos processos. O processo mais curto estará sempre no topo da pilha. Pega o processo com maior prioridade (topo da pilha) e executa ele até o fim, criando novas threads. Recebe todos os processos disponíveis no tempo atual. Sem cancelamentos dos processos.

- Round-Robin

Usa filas, os processos que chegam sempre entram no final da fila. Recebe todos os processos disponíveis no tempo atual e executa todas as threads na fila 1 vez. Tira o processo da fila e executa, criando uma nova thread com quantum. Passa para o processo para o fim da fila, caso não tenha terminado. Caso contrario, o processo é permanentemente removido e a execução continua. Um processo que venha a ser cancelado (por atingir seu quantum de execução) é reinserido ao final da fila.

Simulador de Processos

- Escalonamento com prioridade

Usa filas, este escalonador tem funcionamento similar ao Round Robin, com a diferença de que cada processo possui um tempo limite de execução dependente de sua prioridade. Processos de maior prioridade recebem um quantum maior.

Os aspectos que influenciam na prioridade de um processo se baseiam em deadlines impossíveis ou muito justas obrigando o processo a ser executado imediatamente por inteiro. Processos que estão estourando sua deadline. Processos que estão próximos da conclusão, sendo poupados do escalonamento. Processos que ainda faltam muito tempo para finalizar, recebendo menos prioridade.

Simulador de Processos

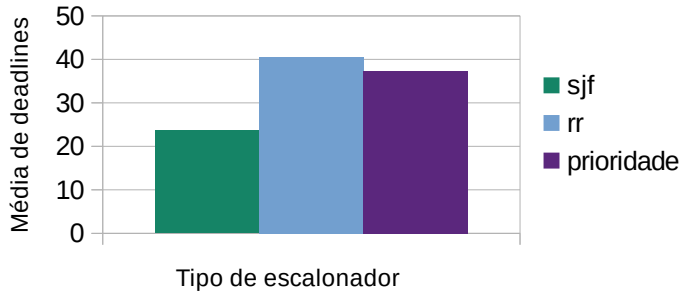
Todos os escalonadores recebem um vetor de linhas(struct com as informações do arquivo de trace → t0, dt, deadline e nome do processo). Essas linhas são convertidas em processos. Com infos do tempo real de execução restante para a finalização do processo e o índice de sua thread.

Foi determinado na estrutura do processo o quantum. Quando uma thread ultrapassa seu quantum, ela se cancela independentemente. O escalonador é responsável por definir o quantum de cada thread, já que seu cancelamento é automático.

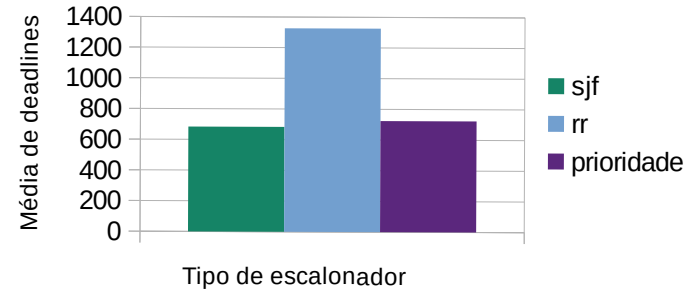
Testes - Deadlines

Máquina A e máquina B tiveram resultados muito semelhantes, então os gráficos foram unificados por serem parecidos.

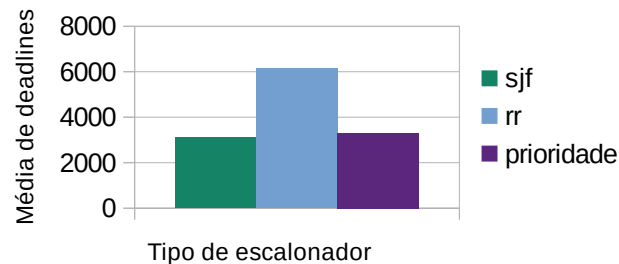
Deadline - 10 processos



Deadline - 50 processos



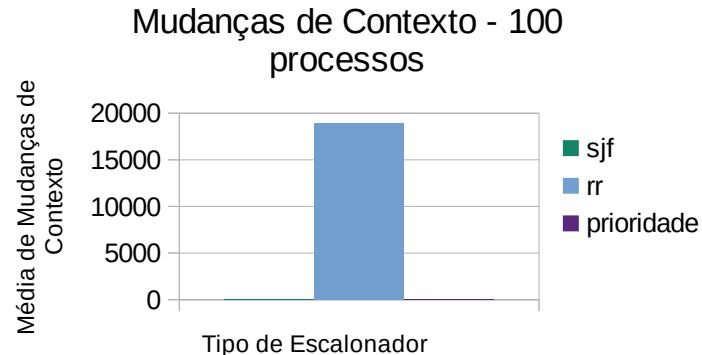
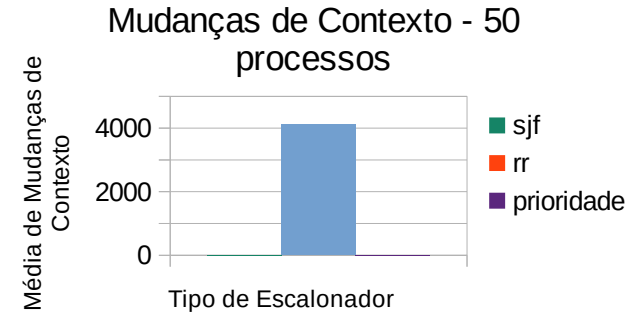
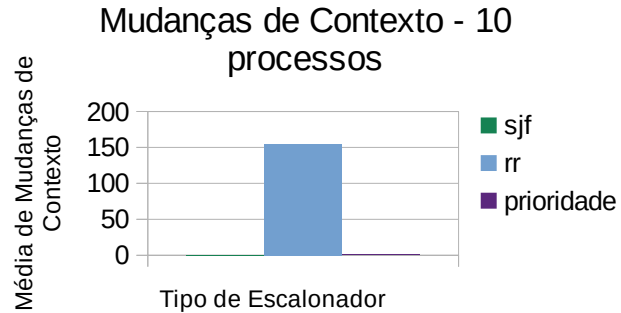
Deadline - 100 processos



Testes - Mudanças de Contexto

Configurações da maquina 1: 12th Gen Intel® Core™ i7-12700H

Configurações da maquina 2: 02th Gen Intel® Core™ i7 i7-2630QM



Conclusões

Os algoritmos de escalonamento se comportaram de maneira quase determinística, com resultados praticamente iguais em todos os 30 testes para cada escalonador. Apenas o Escalonamento com prioridade que se comportou igual ao Round-Robin quando tem poucos processos. Quanto às mudanças de contexto, o Round-Robin, pela sua característica de preempção a cada Quantum, possui maior quantidade de mudanças de contexto que os outros dois para um mesmo input.

Round-Robin possui maior quantidade de deadlines ultrapassadas para um mesmo input. Ele acabam acumulando muitos processos, causando o não cumprimento da deadline. Em contrapartida, o Shortest Job First e o Escalonamento com prioridade buscaram executar o processo com menor dt ou menor quantum, favorecendo o cumprimento da deadline.