

EP1

MAC0422 - Sistemas Operacionais

Breno Helfstein Lucas Daher

IME-USP, setembro 2017

Outline

ep1sh

- Aspectos gerais

- Comandos

ep1

- Escalonadores

Outline

ep1sh

Aspectos gerais

Comandos

ep1

Escalonadores

Implementação

- ▶ ler e parsear os comandos
- ▶ executar os comandos
- ▶ readline e history

Outline

ep1sh

Aspectos gerais

Comandos

ep1

Escalonadores

Comandos

- ▶ Chown e date: chamadas de função (ctime e chown)
- ▶ ping, cal e ep1: fork e execv
- ▶ exit: encerra o programa

Outline

ep1sh

Aspectos gerais

Comandos

ep1

Escalonadores

Camada comum

CPU

Os escalonadores foram simulados usando apenas uma CPU.

process

Os processos são parseados e armazenados em um array

threads

Todas as threads executam a mesma função em todos os escalonadores

fila de execução

Enquanto executam, os processos ficam armazenados em uma fila de prioridade (heap) com Shortest Job First e Highest Priority e numa fila (lista ligada) com o Round Robin

Round Robin

- ▶ Os processos que já podem ser executados ficam em uma fila
- ▶ Cada processo é executado por um quantum, se terminou, sai da fila, se não, vai pro fim
- ▶ O escalonador verifica se mais algum processo deve entrar

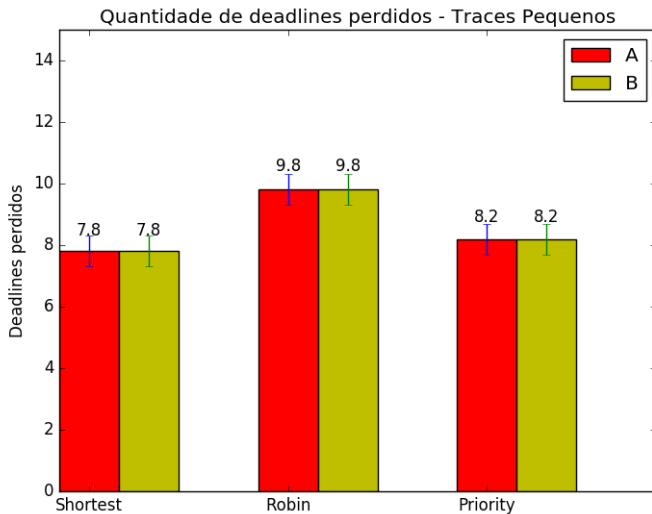
Highest Priority First

- ▶ Os processos que já podem ser executados ficam em uma fila de prioridade
- ▶ Cada processo é executado por um número pré determinado de quantaums, dependendo da prioridade e do tempo restante. Então se terminou, sai da fila, se não, volta pra fila
- ▶ O escalonador verifica se mais algum processo deve entrar

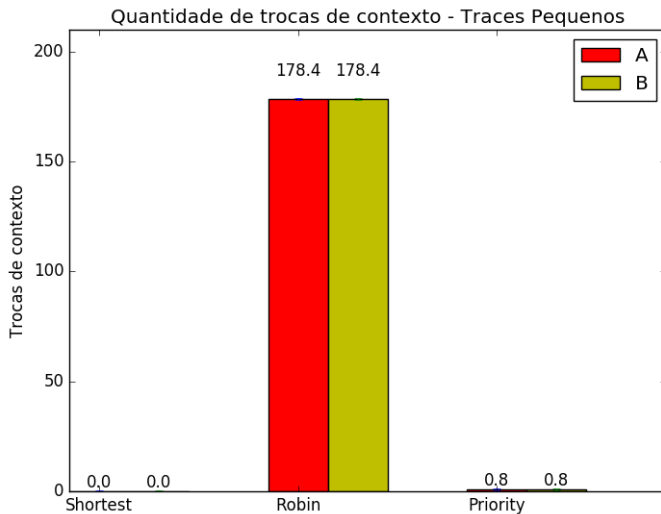
Shortest Job First

- ▶ Os processos que já podem ser executados ficam em uma fila de prioridade
- ▶ Cada processo é executado até o fim e sai da fila
- ▶ O escalonador verifica se mais algum processo deve entrar

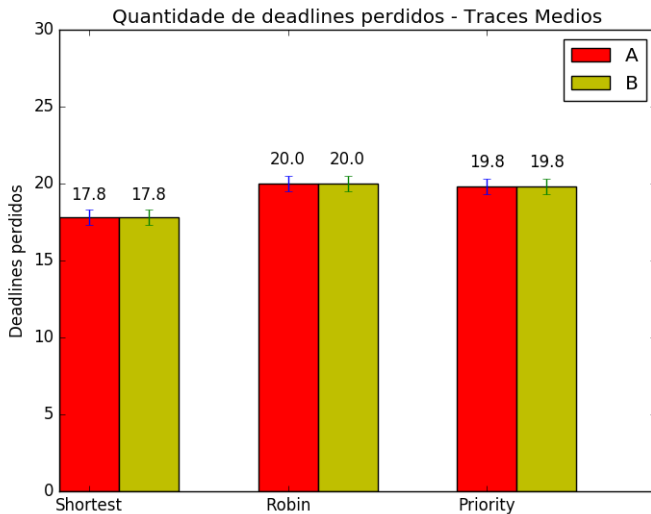
Gráficos traces pequenos - deadlines perdidos



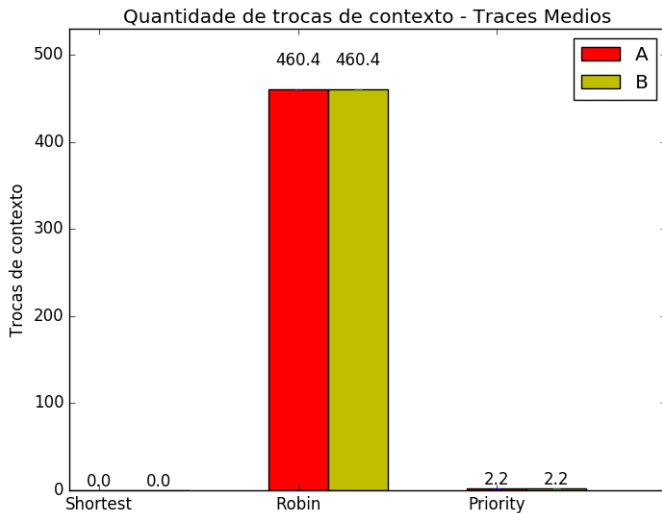
Gráficos traces pequenos - trocas de contexto



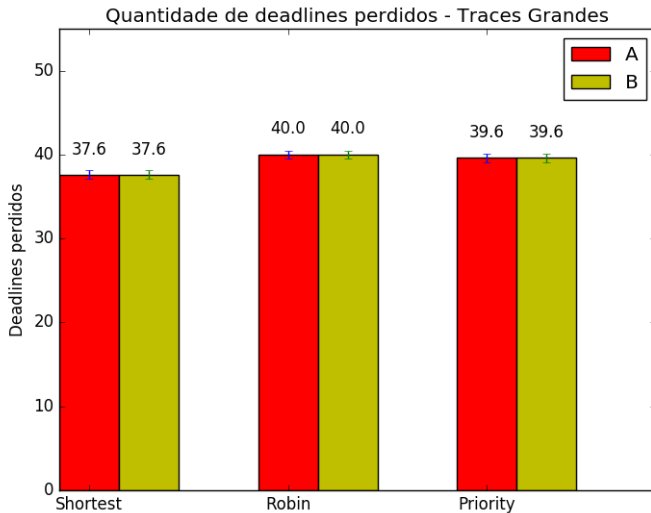
Gráficos traces médios - deadlines perdidos



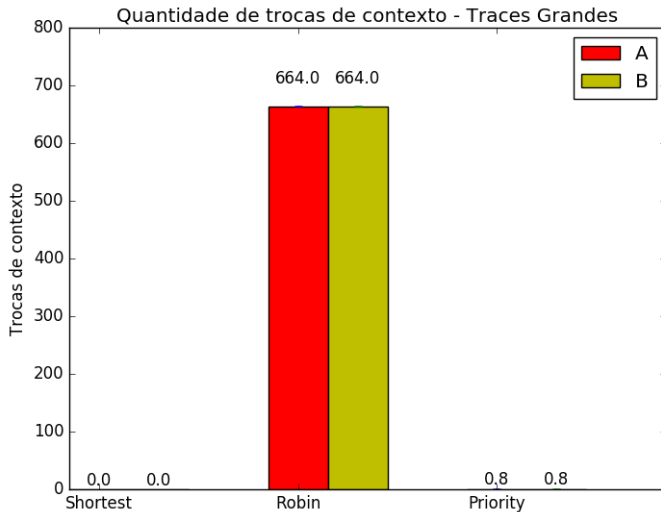
Gráficos traces médios - trocas de contexto



Gráficos traces grandes - deadlines perdidos



Gráficos traces grandes - trocas de contexto



Resultado dos testes

- ▶ Os testes foram realizados com traces de 5, 20 e 40 processos. Todos os traces foram gerados aleatoriamente
- ▶ Não houve diferença entre as máquinas A e B pois os escalonadores usam apenas 1 core, sendo o resultado produzido de forma determinística
- ▶ O escalonador de maior prioridade não conseguiu uma maior eficiência no cumprimento de deadlines devido a função que calcula quantos QUANTUMs cada trace deveria rodar. Não conseguimos fazer uma função que calculava um bom número de quantums.