
MAC 422

Sistemas Operacionais

Jonas Arilho
Pedro Bortolli

11/09/2017

Shell

- Biblioteca *readline*
 - função *readline()*
 - History
- Biblioteca *unistd*
 - Chamadas de sistema com *execv()*
 - Criação dos processos com *fork()*

Shell

- Biblioteca *grp*
 - Definir o grupo do usuário para o *chown*
 - Biblioteca *time*
 - Função *localtime()* para o *date*
-

Escalonamento de Processos

Escalonamento de Processos

Shortest Job First

- Heap binária
- Ordena por dt crescente

Round Robin

- Fila implementada com lista ligada
- Quantum fixo

Priority

- Quantum varia
- Quanto mais próximo ao seu deadline, mais quantum um processo recebe

Detalhes

- Uso da função gettimeofday
- Uso da função usleep
- Single Core x Multi Core

Resultados (gráficos)

Termologia dos Resultados

Mudanças de Contexto

- Processos que, durante sua execução, são forçados a parar para dar lugar a outro
- Mais mudanças de contexto implica em um maior uso da CPU

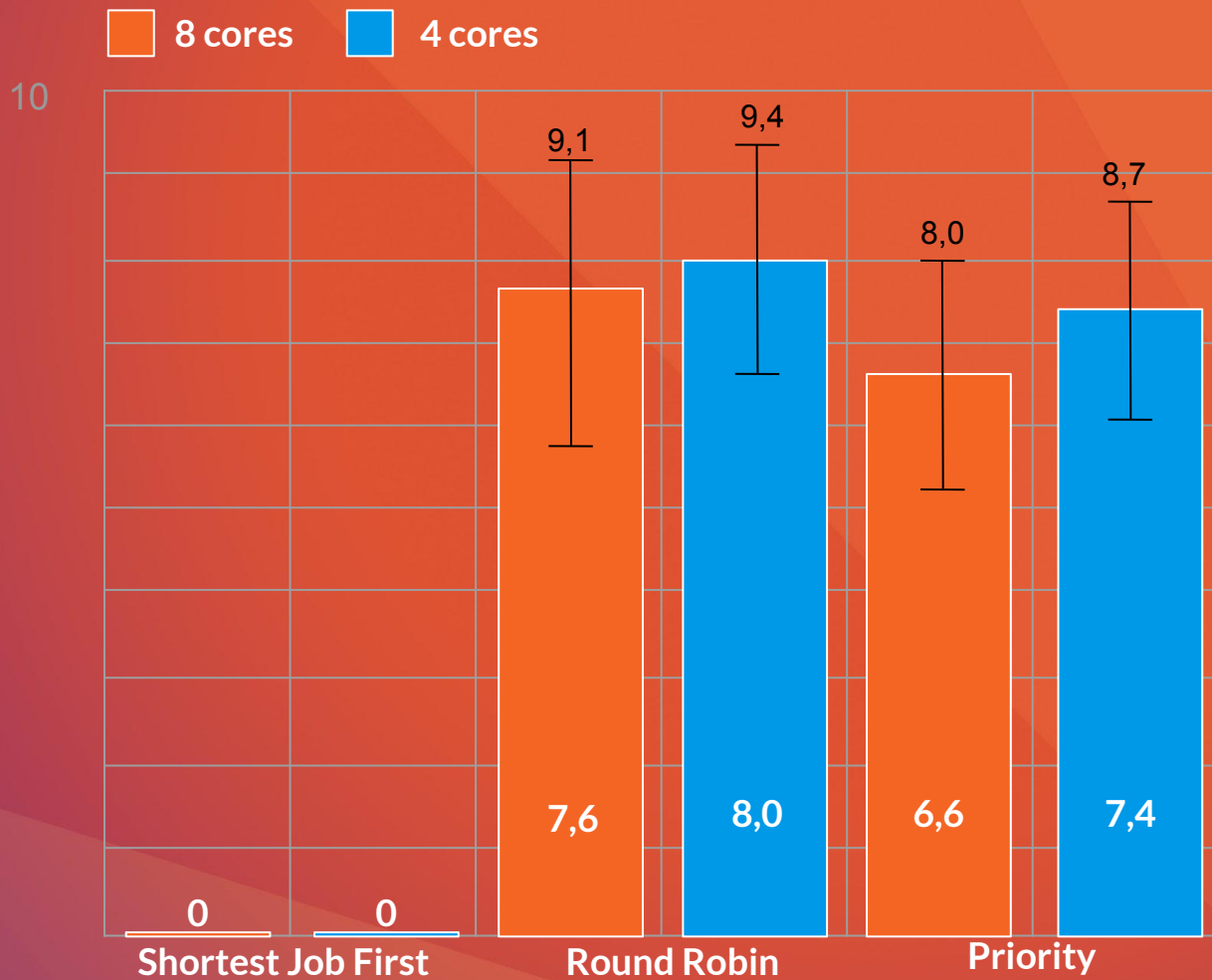
Deadlines perdidas

- Processos que foram terminados após sua “deadline”
 - Implementação multi core resulta em menos deadlines perdidas
-

Gráficos

Mudanças de Contexto

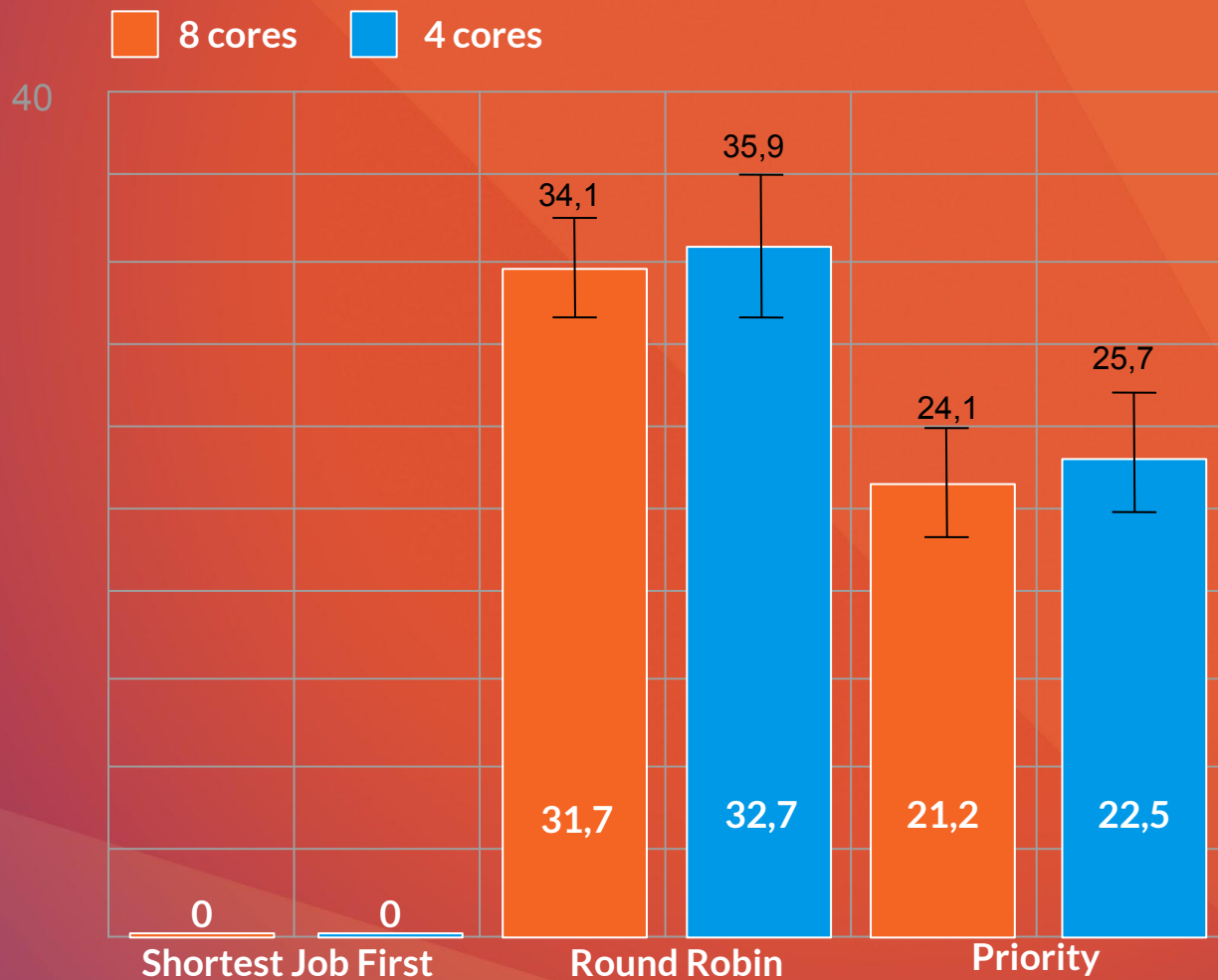
10 processos



Gráficos

Mudanças de Contexto

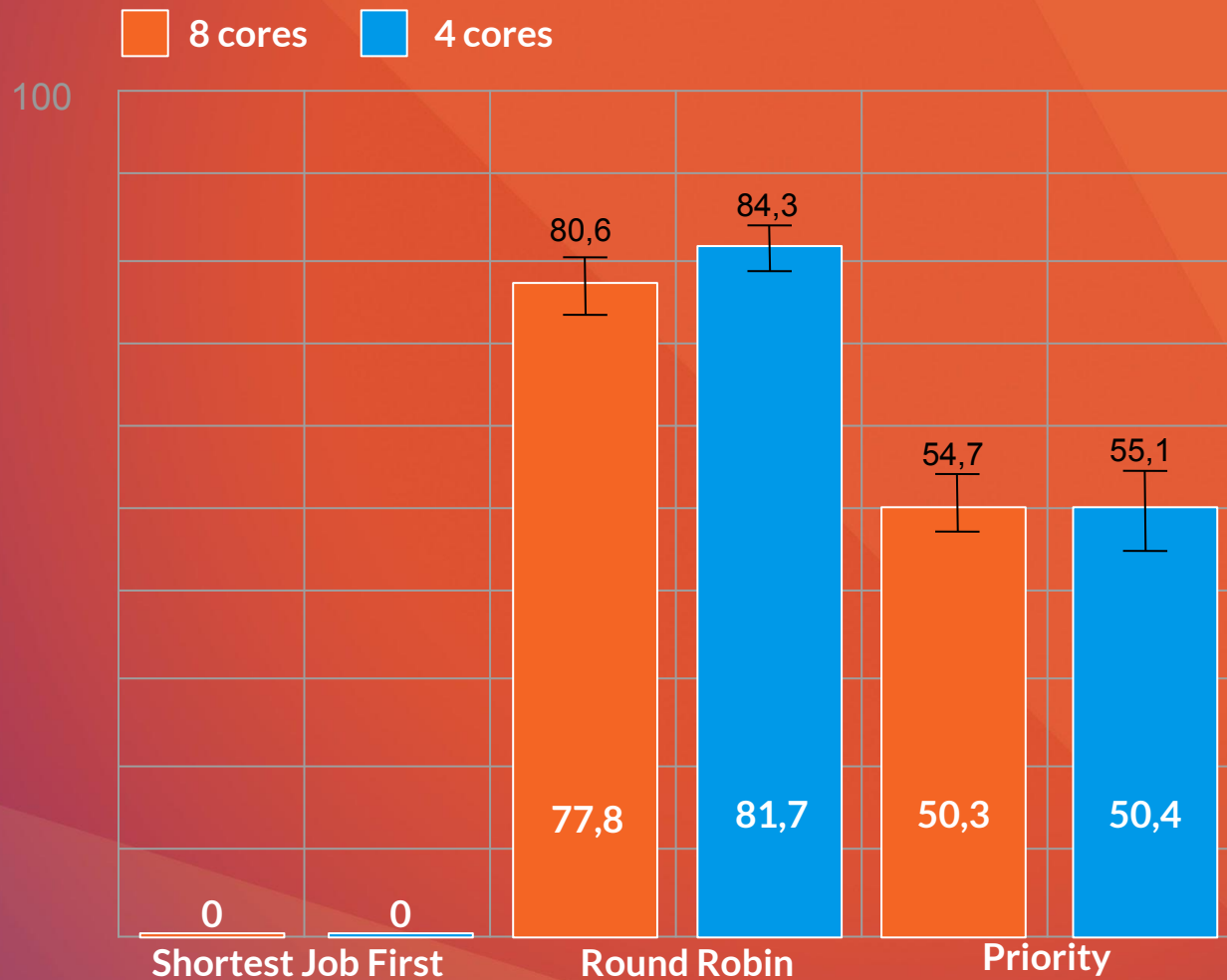
40 processos



Gráficos

Mudanças de Contexto

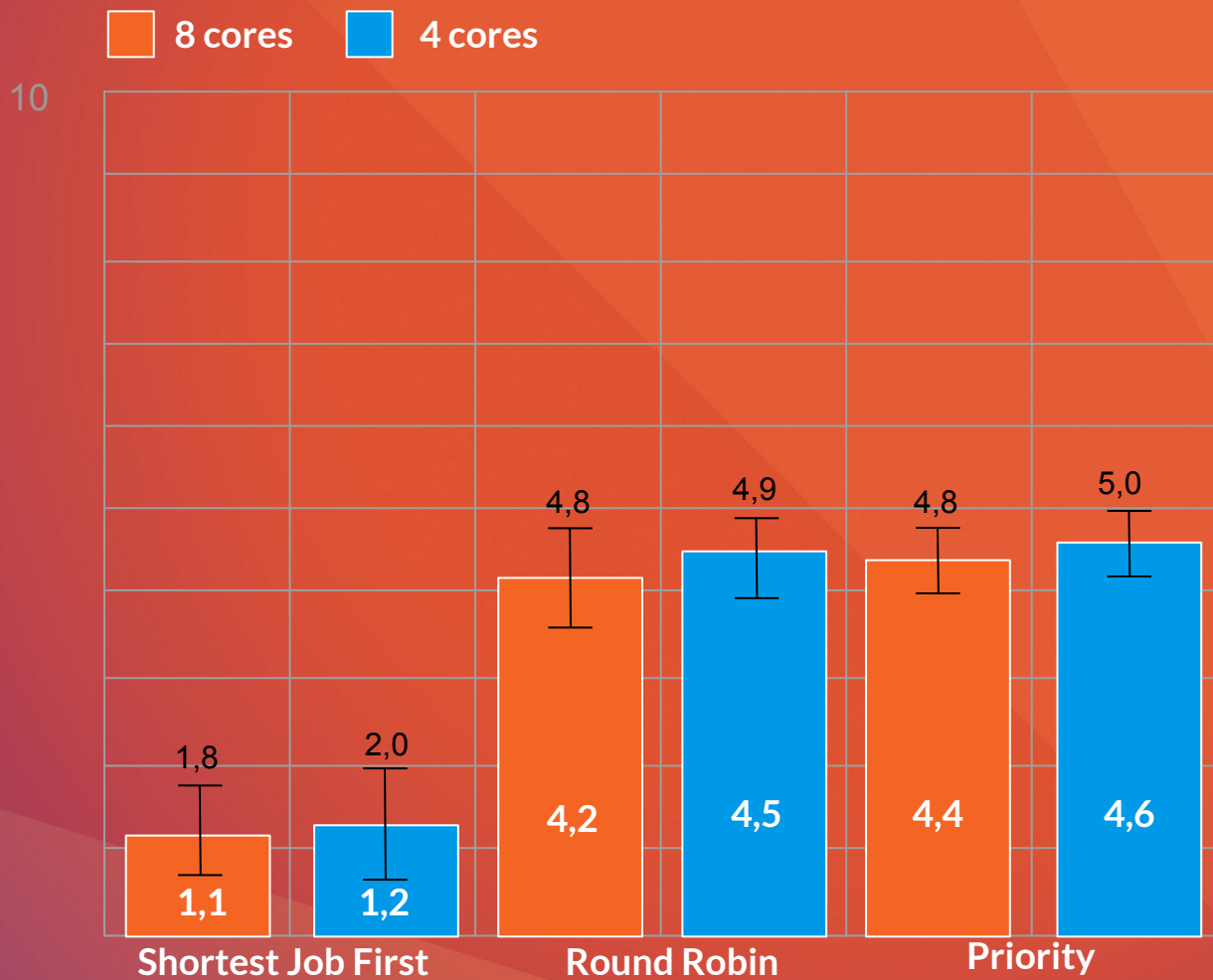
100 processos



Gráficos

Deadlines perdidas

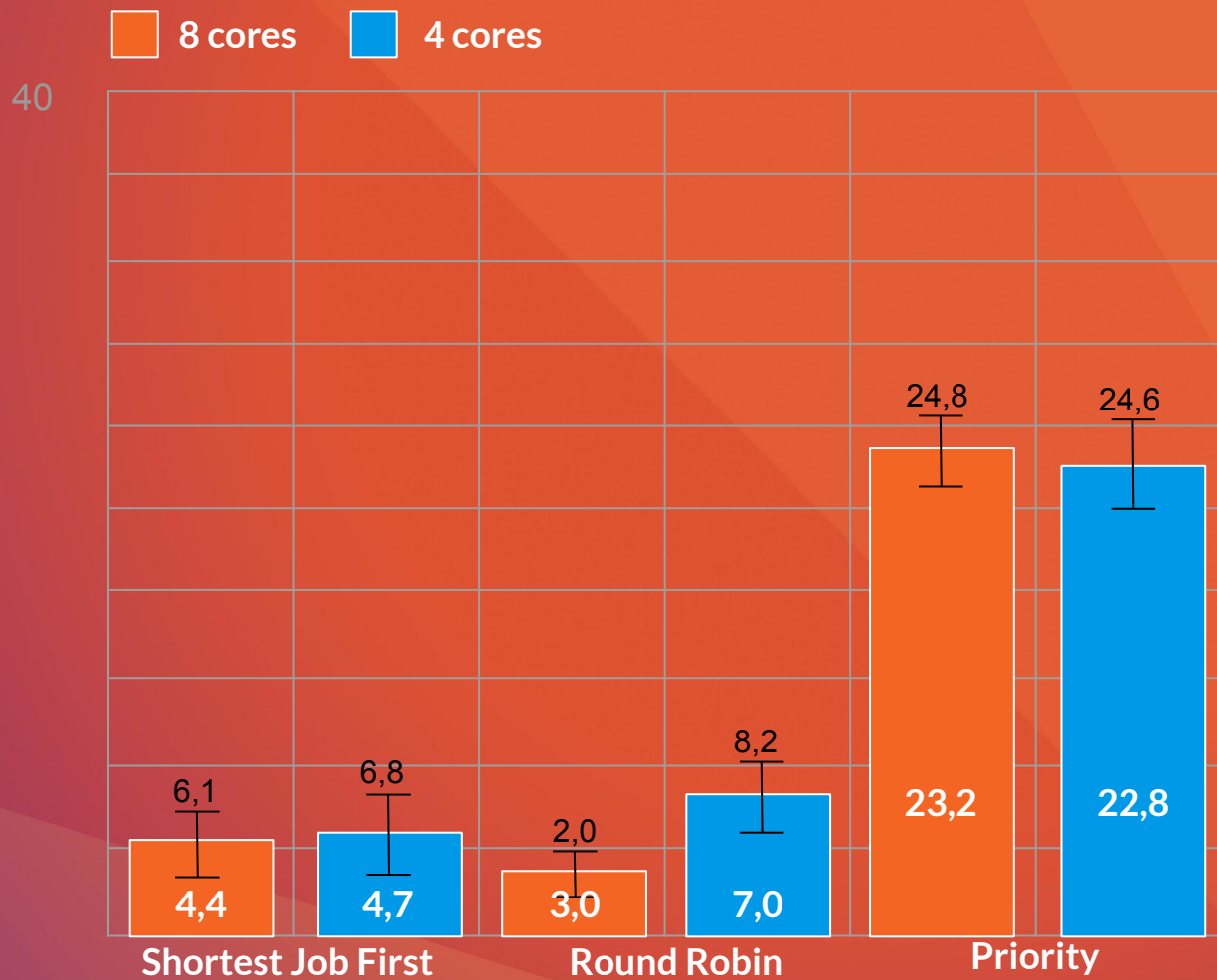
10 processos



Gráficos

Deadlines perdidas

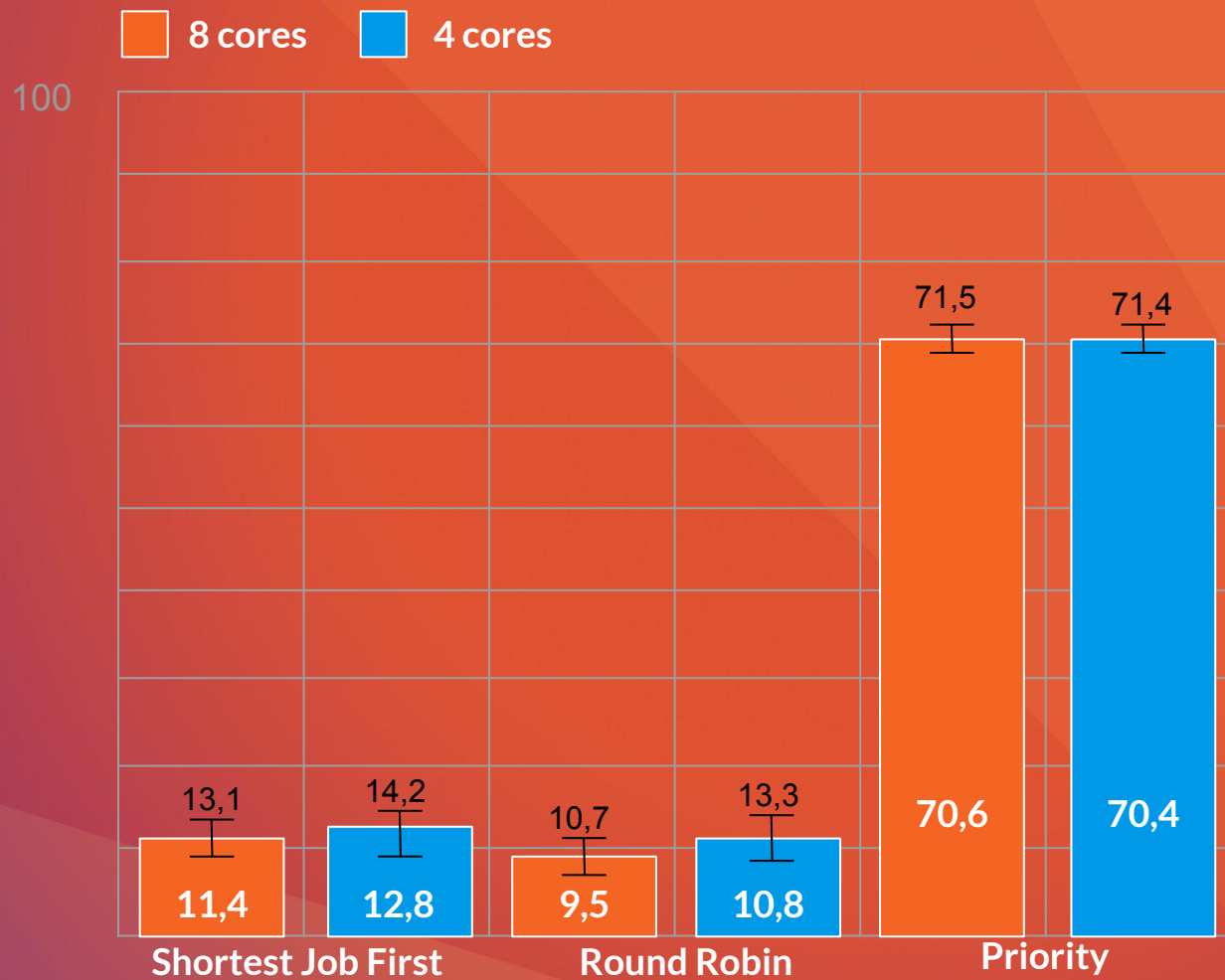
40 processos



Gráficos

Deadlines perdidas

100 processos



Conclusões

Shortest Job First

- Poucas deadlines perdidas (multi core)
- Sem mudanças de contexto, já que o menor processo obrigatoriamente roda até o fim

Round Robin

- Poucas deadlines perdidas, mas pior que SJF
- Mais trocas de contexto, pois não adapta o quantum às condições atuais de um processo

Priority

- Muitas deadlines perdidas (single core)
- Menos trocas de contexto do que o RR já que dá quantum de acordo com a necessidade

Conclusões

- Resultados práticos justificam os esperados
 - Cada processo tem seu lado bom e lado ruim
 - Shortest Job First: mais rápido, porém injusto com processos maiores
 - Round Robin: mais justo, porém consome muita CPU ao realizar inúmeras mudanças de contexto
 - Priority: tenta equilibrar justiça com eficiência
-