

Curso FFI0198 - Sistemas Operacionais

Projeto: SIMULADOR DE ALGORITMO DE ESCALONAMENTO

“The biggest speedup in a system is when it goes from *not working* to *working*. That is an infinite speedup. ” *John Ousterhout*

O objetivo do presente projeto é estudar o desempenho do algoritmo *Round-Robin* de escalonamento da CPU em função do quantum Q . O sistema apresenta escalonamento de tempos curtos e escalonamento de tempos longos. Para o escalonamento de tempos curtos é usado o algoritmo *Round-Robin* e são consideradas duas filas: uma fila *Ready*, para processos prontos para rodar na CPU, e uma fila *Blocked*, para processos bloqueados devido às operações de I/O. Quando o número de processos ativos ultrapassa um certo valor NP_{max} , os novos processos ficam em uma terceira fila de espera, chamada *New*. São considerados ativos os processos presentes na fila *Ready*, na fila *Blocked* e o processo na CPU. Um processo na fila *New* entra no final da fila *Ready* quando um processo ativo acaba de rodar e vai para a fila *Finished*. O algoritmo de escalonamento de tempos longos é o algoritmo FIFO (*First-In First-Out*).

O simulador deve ser organizado da seguinte forma:

1. O simulador lê os parâmetros em um arquivo de configuração (*simcon*) e gera os parâmetros necessários para a simulação; os parâmetros a serem considerados são:
 - o tempo total para a simulação T_{tot}
 - a semente I_{seed} usada para inicializar o gerador de números aleatórios
 - o quantum Q usado pelo algoritmo *Round-Robin*
 - a probabilidade P_{cr} de criar um novo processo (com número de processo P_{id}) a cada passo da simulação, considerando uma distribuição de probabilidade uniforme em $[0, 1]$
 - a probabilidade P_{tipo} de que o processo criado seja limitado por CPU ou limitado por I/O, considerando uma distribuição de probabilidade uniforme em $[0, 1]$
 - o tempo de processamento T_{proc} necessário para rodar o novo processo; este tempo de processamento é gerado usando uma distribuição gaussiana de média T_{pm} e variância T_{psigma} ; claramente T_{proc} deve ser inteiro e ≥ 1

- no caso de processos limitados por I/O: o número de vezes $Nbloq$ que o processo ficará bloqueado devido às operações de I/O; este número é gerado usando uma distribuição uniforme em $[1, \min(Nbmax, Tproc - 1)]$
- no caso de processos limitados por I/O: o tempo $Trun$ que o processo ficará na CPU antes de começar as operações de I/O; este tempo é gerado usando uma distribuição uniforme em $[1, Q]$
- no caso de processos limitados por I/O: o tempo $Tbloq$ que o processo ficará na fila de espera cada vez que for bloqueado devido às operações de I/O; este tempo é gerado usando uma distribuição uniforme em $[1, Tbmax]$
- o número máximo de processos ativos $NPmax$ que podem ser considerados pelo algoritmo de escalonamento de tempos curtos

Resumindo, as variáveis a serem definidas no arquivo *simcon* são: $Ttot, Iseed, Q, Pcr, Ptipo, Tpm, Tpsigma, Nbmax, Tbmax$ e $NPmax$. Para cada processo criado é necessário definir o número do processo Pid e o tempo de processamento $Tproc$. Para processos limitados por I/O é necessário definir as variáveis $Nbloq, Trun$ e $Tbloq$. Note que os tempos $Trun$ e $Tbloq$ devem ser inteiros.

2. O simulador simula o funcionamento da CPU; as operações a serem consideradas em cada passo da simulação são (na ordem elencada abaixo):
 - decidir se um novo processo será criado e, em caso positivo, quais são as características deste processo; o novo processo entrará no final da fila de espera *New* ou no final da fila de espera *Ready*, dependendo se o número de processos ativos NP é maior (ou não) do que $NPmax$
 - deixar rodar o processo que está na CPU por mais um passo
 - verificar se é preciso escalonar a CPU (escalonamento de tempos curtos), i.e.
 - se o processo na CPU terminou de rodar; neste caso o processo deixará a CPU para entrar na fila *Finished*
 - se o processo na CPU já usou o quantum Q ; neste caso o processo deixará a CPU para entrar no final da fila *Ready*
 - em caso de processo limitado por I/O, verificar se ele já foi bloqueado $Nbloq$ vezes; em caso contrário, verificar se ele já ficou na CPU por um tempo $Trun$; neste caso o processo deixará a CPU para entrar no final da fila *Blocked*
 - verificar se existem na fila *Blocked* processos bloqueados que acabaram as operações de I/O e podem ser movidos para o final da fila *Ready*; neste caso os processos deixam a fila *Blocked* para entrar no final da fila *Ready* usando um algoritmo de escalonamento FIFO

- se o processo que estava na CPU acabou de rodar e entrou na fila *Finished*, verificar se existem processos na fila *New* e, em caso positivo, acionar o escalonador de tempos longos
 - escrever cada ação do simulador, em cada passo da simulação, no arquivo *simlog*
3. No final da simulação, calcular o *throughput* e valor médio, mediana e desvio padrão para o tempo de *turnaround*; esses resultados são escritos no arquivo *simres*

Use os seguintes valores para os parâmetros de entrada:

$$\begin{aligned}
 T_{tot} &= 1000 \\
 P_{cr} &= 0.1 \\
 P_{tipo} &= 0.5 \\
 T_{pm} &= 20 \\
 T_{psigma} &= 50 \\
 N_{bmax} &= 20 \\
 T_{bmax} &= 10 \\
 N_{Pmax} &= 20 \\
 Q &= 5, 10, 15 \text{ e } 20
 \end{aligned}$$

Para cada valor do quantum Q a simulação deve ser repetida 5 vezes, mudando a semente $Iseed$. O estudante deverá fornecer o código fonte e, para cada simulação, os arquivos *simcon*, *simlog* e *simres*.