

Curso FFI0198 - Sistemas Operacionais

Projeto: SIMULADOR DE ALGORITMOS PARA GERENCIAMENTO DA MEMÓRIA

“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 dos algoritmos *Best-fit*, *First-fit* e *Worst-fit* para gerenciamento da memória. O sistema usa partição dinâmica da memória, sem paginação ou segmentação. Os novos processos ficam em uma fila de espera chamada *New*. Quando um processo é carregado na memória, permanece nela até acabar de rodar. Os processos carregados na memória pertencem à fila *Ready*. Os processos que já acabaram de rodar, ficam na fila *Finished*. O algoritmo de gerenciamento da memória verifica periodicamente a possibilidade de carregar processos da fila *New* na memória (e na fila *Ready*). Esta verificação é feita também quando um processo acaba de rodar e sai da memória, indo para a fila *Finished*. Os processos que ficam na memória são rodados na CPU usando um algoritmo de escalonamento *Round Robin* com quantum Q=2. Para simplificar o simulador, todos os processos considerados são limitados por CPU. O kernel do sistema operacional deve ser carregado na memória, na posição 0, no começo da simulação e deve sempre permanecer nela.

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 algoritmo para gerenciamento da memória *Algo*
- a probabilidade P_{cr} de criar um novo processo (com número de processo *Pid*) a cada passo da simulação, 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

- o tamanho do processo $Sproc$, i.e. a quantidade de memória necessária para rodar o novo processo; este tamanho do processo é gerado usando uma de probabilidade uniforme em $[1, Spmax]$; claramente $Sproc$ deve ser inteiro e ≥ 1
- a frequência $Tfreq$ com a qual o algoritmo de gerenciamento da memória verifica se é possível carregar novos processos na memória
- o tamanho da memória $Smem$
- o tamanho do kernel do sistema operacional $Sker$

Resumindo, as variáveis a serem definidas no arquivo *simcon* são: $Ttot, Iseed, Algo, Pcr, Tpm, Tpsigma, Spmax, Tfreq, Smem, Sker$. Para cada processo criado é necessário definir o número do processo Pid , o tempo de processamento $Tproc$ e o tamanho do processo $Sproc$. Note que os tempos $Ttot$ e $Tfreq$ devem ser inteiros.

2. O simulador simula o funcionamento da CPU e da memória; as operações a serem consideradas em cada passo da simulação são (na ordem listada 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*
- deixar rodar o processo que está na CPU por mais um passo
- verificar se é preciso escalaronar a CPU, 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*

nos dois casos, o primeiro processo da fila *Ready* é carregado na CPU

- se o processo que estava na CPU acabou de rodar e entrou na fila *Finished*, verificar se existem processos (um ou mais do que um) na fila *New* que possam ser carregados na memória (e na fila *Ready*)
- a cada $Tfreq$ passos da simulação, verificar se existem processos (um ou mais do que um) na fila *New* que possam ser carregados na memória (e na fila *Ready*)
- 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*; também calcular o valor médio, mediana e desvio padrão para a porcentagem de uso da memória ao longo da simulação; esses resultados são escritos no arquivo *simres*

Use os seguintes valores para os parâmetros de entrada:

```
Ttot = 1000
Pcr = 0.1
Tpm = 20
Tp sigma = 50
Sp max = 80
T freq = 20
S mem = 500
S ker = 10
```

Para cada algoritmo a simulação deve ser repetida 2 vezes, mudando a semente *Iseed*. O estudante deverá fornecer o código fonte e, para cada simulação, os arquivos *simcon*, *simlog* e *simres*.