UNIVERSIDADE FEDERAL DE SANTA CATARINA
Curso de Engenharia Mecatrônica
EMB5632 – Sistemas Operacionais
Prof. Dr. Giovani Gracioli
2020/02

Simulação de Algoritmos de Escalonamento

O presente trabalho tem por objetivo escrever um programa para simular o escalonamento de um conjunto de processos/threads através do uso de algoritmos de escalonamento de processos conhecidos na literatura. O programa deve suportar os seguintes algoritmos de escalonamento de processos:

- FCFS (First Come, First Served)
- Shortest Job First
- Por prioridade, sem preempção
- Por prioridade, com preempção por prioridade
- Round-Robin com quantum = 2s, sem prioridade
- Round-Robin com prioridade e envelhecimento (tq=2, α=1)

No caso dos escalonamentos round-robin, o envelhecimento deve ocorrer a cada quantum e não há preempção por prioridade.

O programa deve ler os dados dos processos através de um arquivo de entrada, onde cada linha do arquivo corresponde a um processo, com as seguintes informações inteiras separadas por um espaço em braco:

- data da criação do processo
- duração em segundos
- prioridade estática (escala de prioridades positiva, ex: prioridade 3 > 2).

Um exemplo de arquivo de entrada para o programa seria:

```
0 5 2
0 2 3
1 4 1
3 3 4
```

Neste exemplo, o processo P1 tem data de criação 0, sua execução dura 5 segundos e sua prioridade é definida como 2. Esse formato deve ser respeitado e não deve ser modificado em nenhuma hipótese (a avaliação seguirá esse formato e caso haja erro de leitura o programa não executará e consequentemente terá sua nota zerada). Note que a ordem dos processos não precisa estar ordenada por data de criação de cada processo.

Para cada algoritmo, o programa de simulação de escalonamento de processos deverá produzir as

seguintes informações em sua saída padrão (stdout):

- *Turnaround time* (tempo transcorrido desde o momento em que o processo é criado até o instante em que termina sua execução)
- Tempo média de espera (soma dos períodos em que o processo estava no seu estado pronto)
- Número de trocas de contexto
- Diagrama de tempo de execução

Para simplificar, o diagrama de tempo de cada execução pode ser gerado na vertical, de cima para baixo (uma linha por segundo), conforme mostra o exemplo a seguir:

```
tempo   P1  P2  P3  P4
 0- 1    ##  --
 1- 2    ##  --   --
 2- 3    --  ##   --
 3- 4    --  ##   --   --
 4- 5    --       ##   --
 5- 6    --       ##   --
 6- 7    ##        --   --
 7- 8    ##        --   --
 8- 9    --        --   ##
 9-10    --        --   ##
10-11    --       ##    --
11-12    --       ##    --
12-13    ##             --
13-14                   ##
```

Sendo que “##” significa que o processo em questão está em execução, “--” significa que o processo já foi criado e está esperando, “ “ (espaço em branco) o processo ainda não iniciou ou já acabou, ou seja, não está na fila de prontos.

Sugere-se definir para cada processo:

- identificador
- datas de inicio e de conclusão
- duração (tempo necessário no processador)
- prioridades estática e dinâmica (se houver)
- estado atual (novo, pronto, executando, terminado)
- tempo já executado (total e no quantum atual)
- … (outros campos podem ser necessários para algumas políticas de escalonamento)

O trabalho deve ser escrito em C++ e portante deve apresentar as abstrações da simulação em classes apropriadas e seus relacionamentos. A solução deve conter também uma estrutura de classes usadas para implementar os algoritmos de escalonamento a fim de reaproveitar o código e facilitar a adição de novas políticas no simulador (engenharia de software).

Formato de Entrega e Avaliação

O trabalho em duplas deverá ser entregue no moodle no dia especificado pela tarefa. Todos os arquivos contendo o código do trabalho, bem como Makefile e um **relatório** apresentando

sucintamente a solução e seu projeto (i.e., projeto OO, com diagramas UML), deverão ser submetidos pelo moodle. Não serão aceitos trabalhos entregues fora do prazo. Trabalhos que não compilam ou que não executam receberão nota ZERO, bem como trabalhos que sejam considerados como plágio. Erros de português terão 0.2 de desconto para cada um (ortografia e concordância).

Os itens para avaliação são: (i) funcionamento do programa; (ii) saída do programa (conforme especificação); (iii) clareza do código (utilização de comentários e nomes de variáveis adequadas); (iv) qualidade do relatório; (v) compilação sem warnings; (vi) sem vazamento de memória e (vii) modelagem do software desenvolvido com diagramas UML.

Caso a dupla apresente a saída da simulação utilizando uma interface gráfica, terá um bônus de até 2 pontos na nota do trabalho.

Referências

- Disciplina de Sistemas Operacionais do curso de Ciência da Computação da UFPR. Professor Carlos Maziero.
- Sistemas Operacionais Modernos. 3ª edição. Andrew S. Tanembaum. Pearson. 2010.