

Sistemas Operacionais (DCA0108)
Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Engenharia de Computação e Automação

Professor: Diogo Pinheiro Fernandes Pedrosa

Turma: 01

Local: 4A2, 35T34

Segunda Avaliação

Simulação de Escalonamento de CPU

Objetivo

O objetivo deste trabalho é o desenvolvimento de um programa que simule a execução dos algoritmos de escalonamento de processador (listados a seguir) para a avaliação dos seus desempenhos.

- *First Come, First Serve* (FCFS);
- *Shortest Job First* (SJF); e
- *Round Robin* (RR).

Características do Programa

Este projeto poderá ser feito em grupo, com até três componentes. A escolha da linguagem de programação é livre.

A modelagem do programa/simulação a ser desenvolvido deve respeitar o ciclo de vida de um processo. Considere que seu sistema hipotético somente mantém 10 processos carregados por vez na memória. Estes 10 processos têm que ser coletados do arquivo 'processos.dat' (arquivo anexo a este documento). Este arquivo contém uma sequência de processos que deverão ter seu ciclo de vida simulado pelo programa desenvolvido. Cada linha do arquivo contém informações necessárias para o trabalho, as quais aparecem na sequência:

- PID;
- Tempo de chegada;
- Quantidade de picos de CPU; e
- Sequência de picos de CPU.

Por exemplo, considere que em uma linha do arquivo esteja escrito:

11 10 6 40 44 56 77 18 30

Isso significa que este processo tem um identificador igual a 11, tempo de chegada no sistema igual a 10 unidades de tempo e tem 6 picos de CPU. Estes picos têm valores iguais a 40, 44, 56, 77, 18 e 30 unidades de tempo, respectivamente. Entre dois picos de CPU consecutivos, será considerado que há uma requisição de serviço de E/S, cujo tempo de execução é de 10 unidades de tempo (válido para todos os processos da simula-

ção). Logo, no exemplo acima há 5 requisições de E/S.

Para representação do processo, recomenda-se a implementação de um Bloco de Controle do Processo (PCB) que armazene, pelo menos, as seguintes informações:

- ID do processo;
- Estado (PRONTO, EXECUTANDO, ESPERA);
- Contador do programa (simulado), ou seja, o número acumulado de unidades de tempo de execução;
- Número de picos de CPU;
- Picos de CPU;
- Pico de CPU atual; e
- Tempo de finalização do atual serviço de E/S.

Sugere-se que o simulador mantenha uma fila de PRONTOS e também uma fila de ESPERA. Quando um “processo” é lido do arquivo, cria-se um PCB para ele e o insere na fila de PRONTO. Daí, o algoritmo de escalonamento tratado é quem vai definir qual processo entrará em execução. Processos que ocupam a fila de ESPERA devem responder a um padrão FCFS com a tarefa de E/S. Para efeitos de simulação, o tempo gasto com a troca de contexto não será considerado.

Saídas do Programa

Para cada simulação (cada algoritmo), o programa deve criar três arquivos LOG (formato texto simples) com informações e estatísticas da execução:

Primeiro arquivo LOG

Após cada processo ir para o estado FINALIZADO, escrever no arquivo:

- ID do processo;
- Tempo de chegada;
- Tempo de finalização (tempo que o processo deixa o sistema);
- Tempo de processamento (tempo que o processo ocupou a CPU, junto com os tempos de serviço de E/S);
- Tempo de espera; e
- Tempo de *turnaround*.

Segundo arquivo LOG

A cada 200 unidades de tempo, escrever no arquivo:

- Número de processos na fila de pronto;
- Número de processos na fila de bloqueados;
- Número de processos finalizados.

Terceiro arquivo LOG

Após a finalização de todos os processos, escrever no arquivo:

- Algoritmo de escalonamento usado;
- Valor atual do ciclo de CPU (unidade de tempo);
- Tempo médio de processamento;
- Tempo médio de espera;
- Tempo médio de *turnaround*.;
- Tempo total de utilização da CPU;
- Tempo que a CPU permaneceu ocupada;

- Taxa percentual da ocupação da CPU;
- Tempo que a CPU permaneceu ociosa; e
- Taxa percentual da ociosidade da CPU.

Como observação, é importante que cada arquivo LOG contenha também meios de identificar o grupo autor do trabalho e de qual simulação (de qual algoritmo) foi derivado.

Relatório

Para apresentação dos resultados, será necessária a escrita de um relatório (formato PDF via SIGAA) que contenha, pelo menos, os seguintes itens:

- Definição do problema tratado, da metodologia e ferramentas utilizadas para este projeto;
- Conceitos fundamentais relacionados com o objeto deste trabalho;
- Análise da implementação do programa;
- Análise dos resultados obtidos;
- Considerações finais;
- Referências consultadas; e
- Anexos (códigos dos programas, devidamente comentados).

O relatório deve estar bem formatado (estética adequada), com especial atenção à ortografia. Caso utilizem figuras e citações de terceiros, deve-se obrigatoriamente citar essas referências, dando-lhes o devido crédito. Desatenção nestes itens implicará desconto na nota do trabalho.

Observações Gerais

- Prazo para realização do trabalho: até a última semana do mês de novembro (28/11);
- Enviar, pelo SIGAA, arquivo compactado contendo relatório (em PDF) e os três arquivos LOG (devidamente identificados);
- Para a definição do *quantum* no algoritmo RR, pesquise tempos típicos e adote um que seja condizente com a simulação. Apresente isto no texto do relatório;
- Para o algoritmo SJF, o processo a entrar em execução será aquele que apresentar o próximo menor pico de CPU. Se dois processos da fila de pronto tiverem os mesmos próximos tempos de pico de CPU, deve-se escolher aquele que vier primeiro na fila (em um padrão FCFS);
- Sobre a formatação do relatório, sugere-se a consulta às normas ABNT para redação de trabalhos acadêmicos. Estas normas podem ser acessadas através do sítio www.abntcolecao.com.br, usando o nome UFRN para identificar empresa, usuário e senha (coloque UFRN nos três campos!). Importante: o acesso deve ser feito através de alguma rede da UFRN!

Agradecimentos

Este trabalho é uma cópia adaptada do trabalho do professor Laurence T. Yang, da *St. Francis Xavier University*, Canadá. O enunciado original pode ser acessado em:

<http://cs.stfx.ca/~ltyang/csci-375/projects/csci375-project.htm>