

Faculdade SENAC Porto Alegre

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Sistemas Operacionais

Trabalho sobre Escalonamento de Processos

1. Objetivo

O grupo deverá desenvolver um programa na linguagem que escolher que **simule** a execução de dos algoritmos de escalonamento

1. Fila de processos com a disciplina FIFO (First In First Out)
2. Menores Processos Primeiro (Shortest Job First)
3. Algoritmo baseado em prioridades
4. Algoritmo Round Robin ou Circular

O algoritmo de escalonamento é a parte de estratégia do gerenciador de processador parte integrante do sistema operacional.

O que se deve entregar?

- Um programa fonte escrito em Java.
- Um arquivo texto com um pequeno relatório do trabalho (max. 2 páginas) contendo uma descrição da execução do programa, dos parâmetros (argumentos) que o usuário deve digitar, os resultados que o usuário verá na tela e as funções do programa.

OBS: Não serão aceitos trabalhos via e-mail.

Demonstração: o trabalho só valerá como avaliação se a implementação for demonstrada e explicada. Somente a demonstração do programa, sem a entrega do arquivo fonte e do relatório, não valerá como avaliação.

Observações

- O trabalho pode ser feito em equipe de 4(quatro) alunos, no máximo. Não serão aceitos trabalhos individuais. Não serão aceitos trabalhos desenvolvidos por mais de 2 alunos, sob hipótese nenhuma.

Faculdade SENAC Porto Alegre

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

- Será sorteada uma pessoa para apresentação

2. Prazo de Entrega

O trabalho deve ser entregue e demonstrado no **dia 23 ou 25 de abril**.

3. Descrição do Trabalho

Uma das funções de um Sistema Operacional é distribuir ou escalonar os processos que estão no estado “pronto”, ou seja na fila dos processos prontos, para o estado de “execução”.

Os algoritmos de escalonamento, implementados na gerência de escalonamento de um S.O., têm basicamente a função de: (i) minimizar o tempo de resposta de cada processo, (ii) maximizar o número de processos por unidade de tempo (throughput); (iii) distribuir uniformemente o tempo de CPU.

Suponha que em uma fila de processos prontos estejam 3 processos, P1, P2 e P3 esperando ganhar o processador e com tempos de execução de 10, 6, 8 unidades de tempo, respectivamente. Suponha também que as prioridades dos processos são 2, 4 e 5, respectivamente, onde 5 representa a maior prioridade.

OBS: este algoritmo deve evitar que processos com alta prioridade executem indefinidamente. Portanto, o escalonador deve decrementar a prioridade do processo que está em execução, a cada duas unidades de tempo. Se esta ação fizer com que a prioridade do processo se torne menor do que a prioridade do processo que está em segundo lugar (2ª mais alta prioridade), então ocorrerá a troca de processos. Caso contrário o processo permanece em execução por mais duas unidades de tempo, quando então será novamente decrementada a prioridade dele.

O programa deve fazer a simulação do escalonamento através dos parâmetros digitados pelo usuário. Os parâmetros (argumentos do programa) são os seguintes: (i) quantidade de processos na fila dos “prontos”. Os processos devem ser identificados por P1, P2, ..., Pn; (ii) o tempo de execução de cada processo (tempo de *turnaround*); (iii) a prioridade de cada processo; (iv) a fatia de tempo que o S.O. dispõe para os

Faculdade SENAC Porto Alegre

**Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**

processos, (v) o tempo em que cada processo foi iniciado (startado). OBS: todos processos são inicialmente colocados na fila do processos “prontos”.

O programa deverá exibir na tela: (i) a seqüência de execução dos processos com seus tempos individuais; (ii) os tempos de espera de execução de cada processo; (iii) as médias dos tempos de espera de execução.