



Disciplina: Sistemas Operacionais

Prof: Alba C. M. A. Melo

Descrição do Trabalho Prático (02/2024)

1. Título: Escalonador *Multicore* de Prioridades Estáticas Unix

2. Pré-requisitos

O aluno deverá conhecer a linguagem de programação C e chamadas de sistema Unix.

3. Especificação:

O presente trabalho consiste do projeto de um escalonador de prioridades estáticas para processos Unix (Linux ou MacOS) considerando 4 filas round-robin de prioridade (0, 1, 2, 3), onde a prioridade 3 é a menor prioridade. No lançamento do escalonador, será especificado o número de *cores* disponíveis e o *quantum*. O escalonador vai receber a descrição dos processos a serem escalonados, executando os processos nos *cores* disponíveis, respeitando o *quantum*, segundo as prioridades especificadas no arquivo de entrada.

4. Processo escalonador: Será executado via shell.

4.1) Sintaxe:

> escalona <numero_de_cores> <quantum>

<numero_de_cores>: numero de cores a serem usados pelo escalonador

<quantum>: valor do quantum em segundos

4.2) Arquivo de entrada:

O processo escalonador vai ler um arquivo no formato: <identificador> <arquivo executável> <tempo de início> <prioridade>, com separador “espaço em branco”, onde:

a) <identificador> é o número do processo numerado a partir de 1 (1, 2, etc)

b) <arquivo executável> é o nome do arquivo executável

c) <tempo de início> é o delay em relação ao início do escalonador, sendo que “0” indica que o processo está na fila de prontos no início da execução e, por exemplo, “5” indica que o processo estará na fila de prontos a partir do delay de 5 segundos.

d) <prioridade> é a prioridade do processo (0,1,2 ou 3).

4.3) Arquivos a serem executados (command):

Podem ser executados 3 arquivos (teste10, teste20 e teste30), onde os números 10, 20 e 30 especificam o número de segundos que o processo vai executar. O programa fonte deve ter um for com número de iterações suficiente para o tempo

especificado. Por exemplo, no meu computador, o processo gerado a partir do código abaixo dura 10s:

```
long i;  
for (i=0; i<60000000000; i++);
```

4.4) Exemplo de arquivo de entrada:

```
1 teste20 0 2  
2 teste10 0 0  
3 teste30 20 0  
4 teste10 15 1
```

4.5) Comportamento:

O escalonador lê o arquivo de entrada e coloca inicialmente na fila ready os processos com tempo de início = 0, executando primeiramente os processos que estão nas filas de alta prioridade. Processos de uma dada fila de prioridade somente serão executados se não houver processos nas filas de prioridade mais alta. Todas as filas são regidas segundo a política round-robin.

Ao final da execução, o escalonador imprime a ordem de execução dos processos, o tempo médio de turnaround e o tempo de execução de cada processo.

Atenção: para comunicação, os processos devem utilizar mecanismos de comunicação Unix (pipes, filas de mensagem IPC, memória compartilhada IPC, semáforos IPC ou sinais).

5. Documentação:

O código fonte do escalonador deve ser entregue, tendo como comentário, no início do programa, o nome dos alunos e matrícula, versão do compilador e versão do sistema operacional.

Bom trabalho !!!!!