

### Tarefa 3 de Sistemas Operacionais – INF 1316 – 2022-2

A tarefa consiste em programar em linguagem C programas (ou rotinas) que implementem um interpretador de comandos e escalonador de programas. O interpretador deverá solicitar ao escalonador a execução de programas (cada um a seu tempo). O escalonador por sua vez dispara a execução dos programas (um a um) de acordo com uma determinada política de escalonamento, que será:

LISTA DE PRIORIDADES (neste caso o interpretador deve indicar ao escalonador uma prioridade de execução que é um número de 1 a 7 sendo 1 a maior prioridade e 7 a menor) .

No caso da execução de tarefas com a mesma prioridade estas devem ser escalonadas com tempo compartilhado a cada 3UT (ROUND ROBIN).

O escalonador deve ter estruturas de dados capazes de possibilitar a execução desta política.

A comunicação entre os processos é feita da seguinte forma:

Interpretador de comandos -> Escalonador -> Sistema Operacional

O Sistema Operacional tem um escalonador próprio mas o escalonador da tarefa é quem vai coordenar a execução dos processos indicados pelo usuário via interpretador de comandos. Ou seja, o escalonador é quem vai indicar a ordem de disparo para a execução dos programas e vai influir na sua execução. É o escalonador que vai realizar a preempção, via comunicação entre processos (vai indicar a interrupção da execução de um processo – sinal SIGSTOP - e a continuidade da sua execução – sinal SIGCONT). Um processo com maior prioridade executa até que termine ou até que um outro, de maior prioridade, entre na fila de prontos. Se os processos mais prioritários tiverem a mesma prioridade, então eles serão executados com tempo compartilhado (ROUND ROBIN) entre eles, com fatia de tempo = 3UT's (unidades de tempo). Quando um processo terminar o seu tempo total de execução ele deve ser terminado pelo escalonador. Considere que 1 UT tem a duração de 1 segundo.

A linguagem a ser analisada pelo interpretador de comandos é a seguinte:

exec <nome\_programa>, prioridade=<numero inteiro, de 1 a 7> ,

inicio\_tempo\_execucao=<numero inteiro>, tempo\_total\_execução=<numero inteiro em UT's>

Os programas a serem executados devem ser programas com loop eterno elaborados por você. A entrada padrão e a saída padrão devem ser redirecionadas para os arquivos entrada.txt e saída.txt respectivamente.

Exemplo de entrada.txt:

exec p1, prioridade=3, inicio\_tempo\_execucao=0, tempo\_total\_execucao =8

exec p2, prioridade=2, inicio\_tempo\_execucao=2, tempo\_total\_execucao =9

exec p3, prioridade=1, inicio\_tempo\_execucao=3, tempo\_total\_execucao =11

exec p4, prioridade=2, inicio\_tempo\_execucao=4, tempo\_total\_execucao =13

exec p5, prioridade=1, inicio\_tempo\_execucao=5, tempo\_total\_execucao =14

Exemplo de saída.txt (linha do tempo):

Tempo 0: P1 executando – fila de prontos vazia

Tempo 1: P1 executando – fila de prontos vazia

Tempo 2: P2 executando – fila de prontos: P1

Tempo 3: P3 executando – fila de prontos: P2, P1

Etc...

O código do lab deve ser acompanhado de um relatório contendo o resultado da execução (os arquivos de entrada e de saída), seus comentários sobre o que ocorreu na execução dos processos e como pode ser validado através dos resultados, além de indicar o que funciona e o que não funciona no seu programa.

O trabalho pode ser feito de forma individual ou em dupla.