

Primeiro Trabalho de Sistemas de Computação – INF 1019

O primeiro trabalho consiste em programar em linguagem C programas que implementem um interpretador de comandos e um escalonador de programas. O interpretador deverá solicitar ao escalonador a execução de programas. O escalonador por sua vez dispara a execução de programas de acordo com uma determinada política de escalonamento, que deverá ser:

POR PRIORIDADE (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) .

ROUND-ROBIN (com time-sharing e fatia de tempo de 0.5 segundos)¹

LOTTERY (nesse caso o interpretador deve indicar ao escalonador a quantidade de bilhetes dados a cada processo, e cada processo recebe pelo menos 1 bilhete)

No escalonamento por loteria cada programa ganha um certo conjunto de bilhetes aleatórios de loteria (de um total de 20 bilhetes) e a cada vez que um novo programa deve ser executado, o escalonador sorteia um bilhete e verifica se o mesmo está associado a um programa. Aquele programa que possuir o bilhete sorteado será o próximo a executar. Por exemplo, suponha que o programaA recebeu os bilhetes {15, 17, 8 e 2} e o programaB os bilhetes {16, 11, 6}. Então o escalonador vai repetidamente escolher aleatoriamente bilhetes no intervalo [0 e 19]. Digamos que o bilhete sorteado tenha sido 11, então o programaA será o próximo a executar. Depois, sorteia-se outro bilhete, e assim por diante.

O Unix tem um escalonador próprio mas o escalonador do trabalho é quem vai coordenar a execução dos processos indicados pelo usuário no 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 pois é o escalonador que vai realizar a preempção, via comunicação entre processos (vai indicar a interrupção da execução de um processo e a continuidade da sua execução).

Portanto, a comunicação entre os seus programas deve ser a seguinte forma:

Interpretador de comandos -> Escalonador -> chamadas ao Sistema Operacional (fork/exec, kill, signal, etc.)

O Escalonador deve ter estruturas de dados capazes de possibilitar a execução destas políticas. Você pode implementar três escalonadores próprios, cada um implementando uma das políticas acima. Ou então, caso você queira implementar um único escalonador para todas as políticas então considere a seguinte hierarquia de prioridades entre as 3 políticas:

Prioridade 1 >> Prioridade 7 >> ROUND-ROBIN >> LOTTERY

A linguagem a ser analisada pelo interpretador de comandos é a seguinte (cada comando em uma linha):

¹ Esse valor da fatia de tempo pode ser redefinido, caso o resultado do escalonamento não se mostre expressivo.

Exec <nome_programa> prioridade=<numero de 1 a 7>, no caso de esc. prioridades

Exec <nome_programa>, no caso de ROUND-ROBIN

Exec <nome_programa> numtickets = <numero de bilhetes a receber>

Os programas (“nome_programa”) a serem escalonados/executados devem ser de dois tipos: CPU-bound e IO-bound e devem ser criados por vocês. A entrada padrão e a saída padrão dos interpretador/escalonador devem ser redirecionadas para os arquivos entrada.txt e saída.txt, respectivamente. É da entrada.txt que o interpretador irá ler quais são os programas a serem executados. A saída deve ser clara o suficiente para mostrar como e porque ocorre a preempção na execução dos processos. Esses arquivos serão objetos de avaliação.

O trabalho pode ser feito de forma individual ou em dupla. O trabalho será avaliado em 28/04 na aula de laboratório, logo, deve ser enviado até 27/04 às 23:59:59” via e-mail para: jessicatsalmeida@gmail.com, e endler@inf.puc-rio.br

Deve ser entregue o código fonte e um relatório indicando que programas serão executados em seu teste, a ordem de entrada para o escalonador e a ordem de execução determinada pelo escalonador, juntamente com uma análise crítica sobre o que, de fato, ocorreu. Essa explicação também será objeto de avaliação.