

Primeiro Trabalho de Sistemas de Computação – INF 1316

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, cria e controla a execução dos processos correspondentes de acordo com uma determinada política de escalonamento, que deverá ser:

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

REAL-TIME (nesse caso o interpretador deve indicar ao escalonador em qual segundo e por quantos segundos o processo deverá executar)

Os programas escalonados na forma ROUND-ROBIN e REAL-TIME devem ter execução infinita ou muito longa, de pelo menos 5 minutos. Já os de PRIORIDADE devem ter um tempo máximo de execução e mais curto. Digamos que a duração máxima dos processos PRIO deve ser 20 segundos. por exemplo,

```
for (i=0; i<6; i++) {  
    printf("Processo PRIO %d executando\n", getpid());  
    sleep(0.5)  
}
```

Na política de escalonamento REAL-TIME cada processo deverá iniciar sua execução, periodicamente a cada minuto (sempre no mesmo instante inicial) e deve permanecer executando apenas durante um certo período de tempo (menor do que 60 segundos). Tanto o momento de início como a duração são indicados pelos seus parâmetros. Por exemplo,

P1 I=20 D=10, indica que programa de tempo real P1 deve ser executado a cada minuto e 20 segundos e deve executar por 5 segundos, ou seja, do HH:MM:20 até o HH:MM:30.

Atenção: Para esse escalonamento REAL-TIME o seu escalonador deverá verificar se um novo processo iniciado não possui parâmetros de escalonamento conflitantes com os dos demais processos já em execução. Por exemplo, se seu sistema já está executando P1 I=0 D=10 e P2 I=20 D=5, então não será possível executar um processo P3 I=11 D=20, pois não haveria como permitir a execução simultânea de P2 e P3. Você também deverá verificar se $I+D \leq 60$ segundos, pois o período de execução de um processo não deve ir além do início do próximo minuto.

O escalonador do Unix está em seu núcleo (kernel) mas esse escalonador que você ira implementar no trabalho (Escalonador) é um mero processo que vai coordenar a execução dos processos lidos pelo interpretador de comandos. Ou seja, o escalonador é quem vai controlar a ordem de execução dos programas utilizando sinais para controlar quando devem ser suspensos e quando devem prosseguir com a sua execução

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

Além disso, o seu Interpretador deverá usar algum tipo de comunicação inter-processo (shared memory, ou pipes) para passar os dados dos programas para o Escalonador:

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 deve implementar um único escalonador para todas as três políticas. Então considere a seguinte hierarquia de prioridades entre os 3 tipos de processos:

REAL-TIME >> PRIO 0 >> PRIO 7 >> ROUND-ROBIN

Ou seja, um novo processo REAL-TIME que entra no sistema causa a preempção de qualquer processo. Os PRIORIDADE causam a preempção de processos menos prioritários (p.ex. PRIORIDADE $n-1$ interrompe PRIORIDADE n). E os tanto os processos REAL-TIME como os de PRIORIDADE causam a preempção dos processos ROUND-ROBIN.

O interpretador irá ler do arquivo de entrada quais são os programas a serem executados e escalonados, e deverá inicia-los exatamente na ordem em que aparecem nesse arquivo, **com um intervalo de 1 segundo entre cada um deles**. Ou seja, o interpretador não deve ler toda a entrada e ordenar a lista por níveis de prioridade antes de passar para o escalonador. Os programas devem ser passados um-a-um para o escalonador assim como aparecem no arquivo de entrada. 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 arquivo de entrada do interpretador terá uma lista de sentenças que poderá ser uma das seguintes abaixo. abaixo, cada uma podendo ser uma das abaixo:

<nome_programa-PRIO> PR=< entre 0 e 7>, para escalonamento prioridades

<nome_programa-RR>, para o ROUND-ROBIN

<nome_programa-RT> I=<instante-inicial> D=<período-duração>, para REAL-TIME

Para finalizar toda a simulação do escalonamento, seu escalonador deve executar um `kill(SIGTERM, child)` para todos os processos ainda executando em loop infinito (RT e RR). E para tal implemente em seu sistema um tratador de um sinal (p.ex. SIGUSR1) de finalização.

O trabalho pode ser feito em dupla, e deve ser enviado por E-mail até **9/05 às 23:59:59** para o monitor e o professor. Além disso, será avaliado na aula prática seguinte à data de entrega. (com a presença da dupla). Cada dia de atraso no envio do trabalho acarreta um desconto de 1 ponto na nota máxima.

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.