

Simulando o escalonamento preemptivo de processos com chamdas de sistema

Primeiro Trabalho de Sistemas Operacionais – INF 1316

O primeiro trabalho consiste em programar (em linguagem C e usando as primitivas fork/exec, sinais e comunicação inter processos do Unix/Linux) um simulador de um kernel/núcleo de sistema operacional (KernelSim) que gerencia de 3 a 6 processos de aplicação (A1, A2, A3, etc.) e intercala as suas execuções a depender se estão esperando pelo término de uma operação de leitura ou escrita em um dispositivo de E/S simulado (D1), ou o sinal de aviso do término de sua falta de tempo (time slice). Para isso você deverá implementar também um processo adicional que emula o controlador de interrupções (InterController Sim), que gera as interrupções referentes ao relógio e ao término da operação de I/O no dispositivo D1. São eles respectivamente o IRQ0 (TimeSlice) e IRQ1 (dispositivo D1). A Figura 1 mostra os elementos que devem fazer parte de seu sistema.

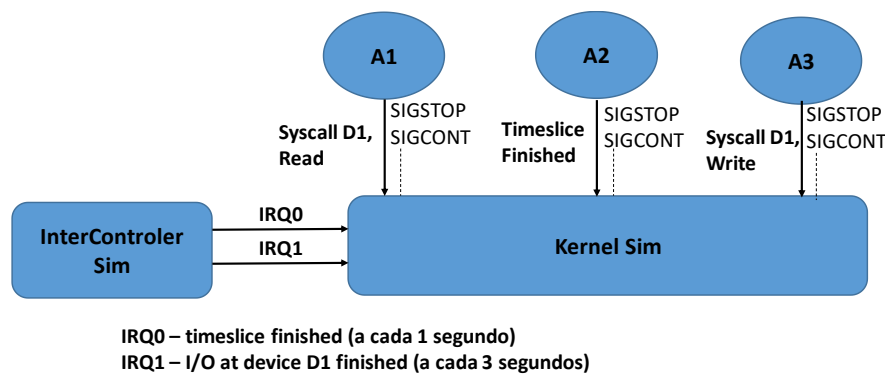


Figura 1: principais processos do seu sistema.

No seu programa, cada um dos elementos em azul da Fig. 1 deve ser um processo Unix. Tanto InterController Sim para KernelSim devem ser processos infinitos que executam em paralelo.

Uso de Sinais UNIX

Você pode usar sinais Unix para emular as interrupções de InterController Sim para KernelSim, e também deve usar os sinais SIGSTOP e SIGCONT para interromper e reiniciar, respectivamente, a execução dos processos A1, A2, A3, ..., A6.

Time-sharing

Qualquer processo A_i pode executar ininterruptamente no máximo durante TimeSlice segundos. Portanto, quando chega um IRQ0, o KernelSim, envia um SIGSTOP para o processo que estava executando, escolhe outro processo de aplicação e o ativa usando o sinal SIGCONT, contanto que este processo não esteja esperando pelo término de um syscall para o dispositivo de I/O, D1. Mais ainda, se dois processos A1 e A2 tiverem executado uma syscall para I/O para o dispositivo, então o primeiro IRQ2 indicará o término do primeiro I/O (e irá desbloquear um dos processos A_i) e o segundo IRQ2 sinalizará o término do segundo syscall (e irá desbloquear outro processo A_i), independente a qual chamada de sistema ele corresponde.

Chamada de Sistema (syscall)

Portanto, outra possibilidade é um processo de aplicação, digamos A1, executar um `syscall(D1,W)`, o que faz com que seja imediatamente interrompido pelo Kernel Sim através de um sinal SIGSTOP. E esse mesmo KernelSim coloca então o processo A1 na fila interna de processos bloqueados pelo dispositivo D1. Eventualmente, InterControllerSim irá gerar um novo IRQ1, o que causará a liberação do primeiro processo que estava na fila dos bloqueados, digamos que seja A1, e, assim, o KernelSim irá poder desbloquear A1 usando SIGCONT.

O dispositivo D1 precisa de 3 segundos para executar as operações R/W solicitadas pela `syscall()`, assim, faremos o InterConntollerSim gerar interrupções para o dispositivo, que indicam o término de um pedido de I/O no dispositivo.

Para isso, implemente o InterConntollerSim para gerar:

- Um IRQ0 a cada 1s indicando o fim do timeslice dos processos (use `sleep()` dentro do corpo do loop)
- Um IRQ1 a cada 3s após o pedido de I/O de cada processo indicando o final da operação de I/O

Obs: Se você achar que essas frequências estão dificultando visualizar a execução, você pode alterar a duração do Time-slice e do tempo de atendimento da operação de I/O.

Processos de Aplicação

Cada processo de aplicação Ax deve conter um laço (loop) de até MAX iterações e ter um contador de interações chamado de PC.

No corpo do loop deverá haver um `sleep(1)`, e deverão ser definidos os tempos, após início, que serão executadas cada `syscall(D1, R-or-W)`.

Troca de Contexto

No seu sistema a troca de contexto acontecerá a cada vez que há o chaveamento de um processo de aplicação para um outro. E no caso específico do seu programa esse contexto deve ser o contador PC do processo. Então, ao reativar um dos processos, antes suspenso, o seu PC anteriormente guardado deve ser restaurado, para garantir que todos os processos de aplicação executem exatamente MAX interações. Além disso, se a interrupção de um processo se deveu a uma `syscall`, então os parâmetros desse `syscall` também devem fazer parte do contexto a ser salvo e restaurado do processo.

Observações Finais

Como acontece a avaliação?

O trabalho pode ser feito de forma individual ou em dupla. Deverá ser enviado na Data de Entrega (vide EAD) e deverá ser apresentado e explicado na Data de Apresentação na aula de laboratório (necessariamente com a presença da dupla). Cada dia de atraso acarreta um desconto de 1 ponto na nota máxima.

O que deve ser enviado/entregue?

Deve ser entregue o código fonte e um relatório indicando que programas fazem parte do seu trabalho incluindo eventuais programas de teste e uma explicação do que deve ocorrer (sugestão: linha do tempo da execução dos processos, cada um iniciando a sua execução com intervalo de 1s entre os processos). Essa explicação também será objeto de avaliação.