

Relatório trabalho 1 Sistemas operacionais.

Aluno: João Gabriel da Cunha Vasconcellos.

Matrícula: 2211302.

Obs: Fiz os comentários e nomes de variável em inglês pois vou colocar o trabalho no meu portfólio no Github e isso o torna mais acessível.

Quais arquivos compõem o meu programa: Usei os conceitos de programação modular e fiz 3 arquivos, `kernel_sim.c`, `kernel_sim.h` e `main.c`.

Explicação geral sobre cada arquivo:

Kernel_sim.c: Aqui estão as funções que implementam o simulador em si os quais separei em um grupo de funções que implementam o kernel e outro grupo que manipulam os sinais. Entrarei em detalhe de cada função mais para frente no relatório.

Kernel_sim.h: Header contendo as funções usadas, estrutura de dados utilizada, definição de `time_slice`, mínimo e máximo de iterações além das probabilidades de interrupções de sistema.

Main.c: Aqui ocorre o fluxo principal do programa, recebe o número de processos, verifica se está entre 3 e 5, faz a alocação de memória, cria o Kernel, trava a execução do programa enquanto a função do kernel está em execução e no final libera a memória.

Detalhe sobre cada função (isso vai esclarecer a implementação).

Kernel_sim(): Inicia o nosso simulador, configura os manipuladores de sinais.

Depois ela entra em um loop para usar `fork` e criar cada processo, se o processo for filho, o loop executa o processo, caso seja pai, o loop inicia o processo configurando cada variável da estrutura de dados.

Em seguida, o primeiro processo é iniciado. Depois, entramos em outro loop para travar o programa enquanto houver processos rodando. No final finalizamos o `InterController` e damos `exit`.

Inter_controller_sim(): Essa função implementa as interrupções por tempo e também a interrupção pelos dispositivos 1 e 2 que vem de forma probabilística, não há um tempo determinado para elas.

Aplication_process() Essa função simula o comportamento de um processo de aplicação em um sistema operacional. O processo executa operações, faz chamadas de sistema (syscalls) que acessam dispositivos (D1 ou D2), e envia sinais para o KernelSim, indicando que está aguardando ser desbloqueado.

Handle_sigint() Essa função é responsável por exibir o estado atual de todos os processos de aplicação. Ela é chamada quando o sinal SIGINT (geralmente ativado ao pressionar Ctrl+C no terminal) é recebido. Esse tipo de função ajuda a visualizar o progresso dos processos, mostrando o estado e os acessos de cada um.

Handle_irq0() Essa função é responsável por tratar a interrupção gerada pelo time slice (temporizador). Esta função implementa o escalonamento dos processos, pausando o processo atual e alternando para o próximo.

Handle_irq1() Essa função manipula a interrupção associada ao dispositivo D1 (IRQ1). Ela percorre todos os processos em busca de algum que esteja bloqueado esperando por D1 (`process_states[i].status == 1`). Quando encontra um processo nessa condição, ela muda o estado do processo para "desbloqueado" (`process_states[i].status = 0`) e imprime uma mensagem informando que o processo foi desbloqueado. Se o processo não estiver em execução, ele é retomado com um sinal SIGCONT, e seu estado é atualizado para "executando" (`process_states[i].running = 1`).

Handle_irq2() Essa função é muito semelhante à anterior, mas lida com o dispositivo D2 (IRQ2). Ela também percorre todos os processos em busca de algum bloqueado esperando por D2 (`process_states[i].status == 2`). Quando encontra, faz o mesmo processo de desbloquear, imprimir a mensagem de status, e retomar a execução do processo se ele não estiver em execução.

```

L2]# Stopped ./main
cunha@cunha-desktop:~/inf1316/TrabalhoS0$ ^C
cunha@cunha-desktop:~/inf1316/TrabalhoS0$ gcc -o main kernel_sim.c main.c -I.
cunha@cunha-desktop:~/inf1316/TrabalhoS0$ ./main
Enter the number of application processes (between 3 and 5): 5
Starting the KernelSim...
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
Process: 5: syscall(D2)
KernelSim: Unlocked the process 5 that was waiting for D2
Process: 2: syscall(D1)
KernelSim: Unlocked the process 2 that was waiting for D1
Process: 3: syscall(D1)
KernelSim: Unlocked the process 3 that was waiting for D1
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2
KernelSim: Process 2 suspended, switching to process 3
KernelSim: Process 3 suspended, switching to process 4
KernelSim: Process 4 suspended, switching to process 5
KernelSim: Process 5 suspended, switching to process 1
KernelSim: Process 1 suspended, switching to process 2

```

Process status:

Process 1: PC = 0, status = Executing, Access D1: 0, Access D2: 0

Process 0: PC = 1, status = Executing, Access D1: 1, Access D2: 0

Process 0: PC = 0, status = Executing, Access D1: 0, Access D2: 0

Process 0: PC = 0, status = Executing, Access D1: 0, Access D2: 0

Process status:

Process 0: PC = 0, status = Executing, Access D1: 0, Access D2: 0

Process 1: PC = 11, status = Executing, Access D1: 0, Access D2: 0

Process 2: PC = 10, status = Executing, Access D1: 0, Access D2: 0

Process 3: PC = 10, status = Executing, Access D1: 0, Access D2: 0

Process 4: PC = 10, status = Executing, Access D1: 0, Access D2: 0

Process 5: PC = 10, status = Executing, Access D1: 0, Access D2: 0