

Instruções:

- Utilize o Linux para realizar a atividade. Sempre que estiver em dúvida sobre o funcionamento de uma função ou chamada, utilize o manual do Linux, disponível através do comando *man*.
- Submeta as respostas preenchendo o gabarito disponível no Moodle.

Exercício de laboratório 02 – IPC e Sincronização - Sinais

Introdução - Sinais

Sinais são um tipo de comunicação de eventos gerados pelo Sistema Operacional e enviados aos processos. Podem ser usados para suspender ou reiniciar processos, informar erros de hardware, etc. Os sinais são identificados por números (ex. SIGKILL corresponde ao número 9) e podem ser enviados aos processos pelo usuário por programação, utilizando a função *kill* em C, ou via terminal, utilizando o comando 'kill', ou, ainda, por alguma combinação de teclas, como, por exemplo, CTRL+C ou CTRL+Z.

Quando um processo recebe um sinal, sua execução é interrompida para o tratamento desse sinal. Há um tratamento padrão, dado pelo S.O., para cada sinal. Contudo, os sinais podem ser interceptados via código (exceto SIGKILL e SIGSTOP) para que o programa possa personalizar a ação padrão de cada sinal.

Para um tratamento personalizado do sinal é preciso associar seu número a uma função (normalmente chamada de *handler*). Vários sinais podem ser associados à mesma função, bem como podemos criar funções diferentes para tratamento de diferentes sinais. A função *handler* do sinal será chamada automaticamente pelo SO quando o sinal for recebido, sobrepondo o tratamento padrão. Após tratado o sinal, a execução do processo é retomada a partir do ponto em que se encontrava.

Para ver mais detalhes sobre sinais acesse o manual do Linux com o comando:

➤ *man 7 signal*

→ Os sinais SIGSTOP e SIGCONT são usados, respectivamente, para suspender e retomar um processo. Para ver os efeitos destes sinais, compile e execute o código do arquivo *lab02.c* e em seguida suspenda o processo enviando o sinal SIGSTOP (a contagem exibida na tela deve parar) e depois retome a execução do processo enviando o sinal SIGCONT.

→ Utilize o comando abaixo (em um terminal diferente) para enviar os sinais.

➤ *kill -s SIGSTOP <pid>*
➤ *kill -s SIGCONT <pid>*

Repita os passos anteriores, mas agora visualize o estado do processo após cada envio de sinal para confirmar se o processo está suspenso ou em execução.

→ Utilize o comando abaixo (em um terminal diferente) para visualizar o estado (coluna 'state') do processo.

➤ *ps -o pid,%cpu,state,command -p <pid>*

→ A interceptação de sinais pode ser feita em código C utilizando a função *signal* (ou *sigaction*). No programa de exemplo (*lab02.c*), temos a ação de imprimir o sinal interceptado (veja a função *signal_handler* no arquivo *lab02.c*).

a) Teste o envio de sinais diversos para o processo. Observe que apenas os sinais presentes na função *signal_handler_setup* são interceptados.

→ Utilize o comando abaixo (em um terminal diferente) para enviar os sinais.

➤ *kill -s <SINAL> <pid>*

Questões:

1) Sobre o tratamento de sinais no padrão POSIX, analise as afirmativas abaixo e marque a opção CORRETA.

- I. Podemos configurar a função handler de forma que ela utilize uma pilha diferente da pilha usada pelo processo. Isso pode ser feito usando as funções *sigaltstack* e *sigaction* em conjunto.
- II. Os números (valores) correspondentes a alguns sinais podem ser diferentes em algumas arquiteturas.
- III. Dentre os possíveis eventos indicados pelos sinais, temos tanto eventos relacionados ao S.O., quanto eventos relacionados ao processador.

- a) Somente a alternativa I está correta.
- b) Somente a alternativa II está correta.
- c) Somente a alternativa III está correta.
- d) Todas as alternativas estão corretas.
- e) Todas as alternativas estão erradas.

2) Marque a alternativa INCORRETA:

- a) O tratamento de um sinal irá ocorrer mesmo que o processo esteja dormindo devido à função *sleep*.
- b) O mesmo sinal pode ser enviado para mais um processo simultaneamente.
- c) Alguns sinais são, por padrão, ignorados.
- d) Por questões de portabilidade, a função *signal*, usada para configurar o *handler* do sinal, deve ser evitada.
- e) Apesar de não poderem ser interceptados para um tratamento personalizado, os sinais SIGKILL e SIGSTOP podem ser ignorados.

3) Sobre a função *sigwait*, indique a opção INCORRETA:

- a) É uma função usada para bloquear uma *thread* até que um determinado sinal seja recebido.
- b) Trata-se de uma função *thread-safe*, ou seja, não pode ser chamada por múltiplas threads simultaneamente.
- c) É equivalente, com algumas pequenas diferenças, às funções *sigwaitinfo* e *sigtimedwait*.
- d) Pode ser usada para aguardar mais de um sinal em uma lista, retornando caso algum seja recebido.
- e) Em caso de erro, *sigwait* retorna um número positivo.

4) Qual dos itens abaixo não corresponde a um possível tratamento padrão de sinais no POSIX:

- a) Suspender (ou bloquear) o processo.
- b) Continuar (ou retomar) o processo.
- c) Terminar o processo.
- d) Terminar o processo atual e criar uma cópia do processo.
- e) Terminar o processo e criar um arquivo de depuração com a imagem do processo no momento do término.

5) Modifique o tratamento dos sinais no arquivo *lab02.c* para que o processo tenha o seguinte comportamento (suba o código para a questão correspondente no questionário do moodle):

- Deve reverter a ordem de contagem caso receba o sinal SIGUSR1. Se estava contado de forma crescente, passa a contar de forma decrescente, e vice-versa. Não reinicie a contagem, apenas reverta a ordem.
- Deve reiniciar a contagem (iniciar do 0, em ordem crescente) caso receba o sinal SIGUSR2.

→ Utilize o comando abaixo (em um terminal diferente) para enviar os sinais.

➤ `kill -s SIGUSR1 <pid>`

➤ *kill -s SIGUSR2 <pid>*