HW 9

Mike Hanling

02 APR 18

Questions

1. (5 points) What does it mean that signals arrive asynchronously?

   They can come from different processes, things do not run one after another, signals can come from anywhere at anytime.

2. (10 points) What signal is generated from the following keyboard-shortcut/command?

   (a) Ctrl-c

   SIGINT

   (b) Ctrl-z

   SIGSTOP

   (c) fg/bg

   SIGCONT

   (d) Ctrl-

   SIGQUIT

3. (5 points) Run the command kill -l to list all the signals and their signal numbers. Find either the matching signal-number/signal-name for the following values below. Additionally, for each signal, use man 7 signal to describe the default action of each.

   (a) SIGKILL

   9: Terminate

   (b) 14

   SIGALRM: Terminate

   (c) SIGALRM

   14: Terminate

   (d) SIGABRT

6: Terminate and core dump

(e) 21

SIGTTIN: Stop

(f) 1

SIGHUP: Terminate

(g) SIGCHLD

17: Ignore

(h) SIGTTOU

22: Stop

4. (15 points) Provide a brief plain-English explanation, e.g., reference which signal is sent and to which process(es), for each the kill/killall commands. (hint: You should look in the man pages)

   (a) killall -17 sleep

   SIGCHLD signal sent to all sleep processes running

   (b) kill -9 2237

   SIGKILL sent to process 2237

   (c) killall -SIGUSR1 a.out

   Signal 10 is sent to all a.out processes

   (d) kill -SIGABRT -1

   Signal 6 sent to all process excluding kill and init

   (e) killall sleep

   SIGTERM sent to all sleep processes

   (f) killall -u

   Kill only processes owned by this user

5. On a lab machine, run the following program in the background

   aviv/ic221-hw/hw09/ic221-signaler &

   From the same (or another) terminal on the same machine, send the program the following two signals below and describe the results.

   (a) (3 points) SIGUSR1

”Ha Ha, missed me!”
An ASCII art cat.

(b) (3 points) SIGUSR2

”You shot me!”
ASCII art skull and crossed bones.

6. Consider the program below and answer the associated questions:

```
int count = 0;

void handler(int signum){

  printf("You Shot Me!\n");
  count++;

  if(count > 3){
    printf("I'm dead!\n");
    exit(1);
  }

}

int main(){

  //establish signal hander for SIGTINT and SIGSTOP
  signal(SIGTINT,handler);
  signal(SIGTSTP,handler);

  //loop forever
  while(1);

}
```

(a) (3 points) What is the output of the program if the user hits Ctrl-c only once? Explain.

You Shot Me!
SIGINT is handler and count is less than 4, so it prints the above once.

(b) (3 points) What is the output of the program if the user hits Ctrl-c four times? Explain.

You Shot Me!
You Shot Me!
You Shot Me!
You Shot Me!
I’m dead!
SIGINT is handled four times, and then count is greater than 3, so the last line is printed and the program exits.

(c) (3 points) What is the output of the program if the user hits Ctrl-c three times and Ctrl-Z once? Explain.

You Shot Me!
You Shot Me!
You Shot Me!
You Shot Me!
I’m dead!

SIGINT and SIGSTOP are handled with the same function, so the explanation from the quesion above still applies.

7. (5 points) What does the system call pause() do? Yes, it pauses the program, of course, but until when does the program stay paused and why is it a useful command?

It says paused until there is a signal received to terminate, stop, or one that has a way to be handled. It is useful with alarms, as it will ahve the program sleep until the alarm has gone off, sending the SIGALRM so that it can be handled.

8. (10 points) How many times does the program below print alarm? Explain

```
int count = 10;

void handler(int sugnum){

  printf("Alarm!\n");
  count /= 2;
  alarm(count);

}

int main(){

  signal(SIGALRM, handler);
  alarm(count);

  while(1) pause();
}
```

4 because it will print after 10 seconds, then 5, seconds, then 2, seconds, then 1 second, and then the alarm is for zero seconds, so it will never go off again.

9. (10 points) Convert the following use of signal() below to a sigaction() call.

```
signal(SIGUSR1, handler);
```

```
struct sigaction action;
action.sa_handler = handler;
sigaction( SIGUSR1, &action, NULL);
```

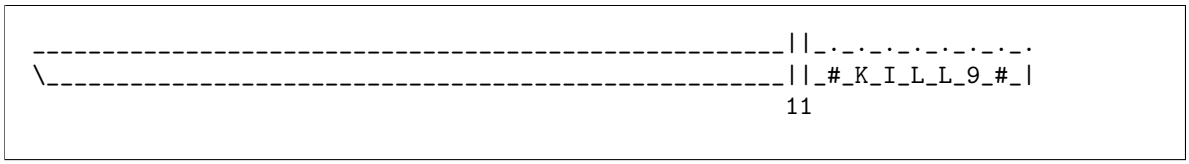
10. (5 points) What sigaction flag is used to ensure that system calls will be restarted when interrupted?

SA_RESTART

11. (5 point) Provide an example of why the read() system call would need to be restarted due to a signal delivery.

If the siganl comes from alarm, like making sure the user is not taking too long to put in an input, then the read() will not restart, so the user will never be able to input text even if it appears that way in a program.

12. (5 points) Look in man 7 signal and kill -l and draw a picture of your favorite signal. Be sure to clearly identify it. (You will not use LaTex bonus points for hand drawing your picture.)


_____||_._._._._._._._._.
_____||_#_K_I_L_L_9_#_|
11