CS 361 – Lab 4

MONDAY, FEBRUARY 18TH, 2019.

Signals

- to list all signals use **kill -l**

```
$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGEMT       8) SIGFPE       9) SIGKILL     10) SIGBUS
11) SIGSEGV     12) SIGSYS      13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGURG      17) SIGSTOP     18) SIGTSTP     19) SIGCONT     20) SIGCHLD
21) SIGTTIN     22) SIGTTOU     23) SIGIO       24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGPWR      30) SIGUSR1
31) SIGUSR2     32) SIGRTMIN    33) SIGRTMIN+1  34) SIGRTMIN+2  35) SIGRTMIN+3
36) SIGRTMIN+4  37) SIGRTMIN+5  38) SIGRTMIN+6  39) SIGRTMIN+7  40) SIGRTMIN+8
41) SIGRTMIN+9  42) SIGRTMIN+10 43) SIGRTMIN+11 44) SIGRTMIN+12 45) SIGRTMIN+13
46) SIGRTMIN+14 47) SIGRTMIN+15 48) SIGRTMIN+16 49) SIGRTMAX-15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9
56) SIGRTMAX-8  57) SIGRTMAX-7  58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4
61) SIGRTMAX-3  62) SIGRTMAX-2  63) SIGRTMAX-1  64) SIGRTMAX
```

Signals

- each signal has a name, value, and default action
- the name starts with SIG and ends with a description
- value of the signal is to identify the signal with a number

Signal	Value	Action	Comment
SIGHUP	1	Term	Hangup detected on controlling terminal or death of controlling process
SIGINT	2	Term	Interrupt from keyboard
SIGQUIT	3	Core	Quit from keyboard
SIGILL	4	Core	Illegal Instruction
SIGABRT	6	Core	Abort signal from abort(3)
SIGFPE	8	Core	Floating point exception
SIGKILL	9	Term	Kill signal
SIGSEGV	11	Core	Invalid memory reference
SIGPIPE	13	Term	Broken pipe: write to pipe with no readers
SIGALRM	14	Term	Timer signal from alarm(2)
SIGTERM	15	Term	Termination signal
SIGUSR1	30,10,16	Term	User-defined signal 1
SIGUSR2	31,12,17	Term	User-defined signal 2
SIGCHLD	20,17,18	Ign	Child stopped or terminated
SIGCONT	19,18,25	Cont	Continue if stopped
SIGSTOP	17,19,23	Stop	Stop process
SIGTSTP	18,20,24	Stop	Stop typed at tty
SIGTTIN	21,21,26	Stop	tty input for background process
SIGTTOU	22,22,27	Stop	tty output for background process

- `Term` : The process will terminate
- `Core` : The process will terminate and produce a core dump file that traces the process state at the time of termination.
- `Ign` : The process will ignore the signal
- `Stop` : The process will stop, like with a `Ctrl-Z`
- `Cont` : The process will continue from being stopped

Handling Signals

- the signal() function is used to catch/handle a signal (can also ignore signals)
- int signal(int signum, void (*handler)(int))
  - the first is an integer (i.e., int), a signal name
  - the second is a function that accepts an int argument and returns nothing, the signal handler
  - e.g., set second parameter to SIG-IGN to ignore a signal, to SIG_DFL to handle it in the default way
  - pass SIGUSR1 or SIGUSR2 as first argument for user defined signals
- raise(int signum) allows a process to send a signal to itself

Signal Example 1

```c
#include <stdlib.h>
#include <stdio.h>

#include <signal.h> /*for signal() and raise()*/
#include <unistd.h>

/* hello_loop.c*/
void hello(int signum){
  write(1, "Hello World!\n", 13);
}

int main(){

  //Handle SIGINT with hello
  signal(SIGINT, hello);

  //loop forever!
  while(1);

}
```

```
#> ./hello_loop
^CHello World!
^CHello World!
^CHello World!
^CHello World!
^CHello World!
^CHello World!
^CHello World!
^\Quit: 3
```

Signal Example 2

```c
#include <stdlib.h>
#include <stdio.h>

#include <signal.h> /*for signal() and raise()*/
#include <unistd.h>

void hello(int signum){
  write(1, "Hello World!\n", 13);
}

int main(){

  //execute hello() when receiving signal SIGUSR1
  signal(SIGUSR1, hello);

  //send SIGUSR1 to the calling process
  raise(SIGUSR1);
}
```

```
#> ./hello_signal
Hello World!
```

Inter- Process Communication

- a process can send a signal to another process using kill()
- int kill(pid_t pid, int signum)
  - sends signum signal to the process with ID pid
- Unix kill command can also be used
  - kill –XXX pid1 pid ..... Pid
  - XXX is the signal name without the SIG
  - e.g., kill –INT 6421sends a SIGINT to process 6421
- killall is used to send a signal to all processes of a given name or running the specified command
  - e.g., killall cat

Task

- a. Create signal handlers for any three signals of your choice, reporting those signals to which it didn't succeed.
- b. Fork off a process for each signal and send the signal to it. If the signal handler succeeds it should print a message giving the signal number. The parent should then kill the process and wait on it before forking the next.