

We are going to simulate a set of three friends who go for a walk, and get lost from one another. The three friends are a skunk called Skunky, a frog called Froggy, and a rabbit called Arnold. Each animal will be represented by a process, and you will need to write a separate program for each process. Arnold is the leader, so he will create the other two processes. The Arnold process will first call *fork* to create the processes, and then the new process will call *execv* to execute the new program. You use the return value from the fork function to determine whether you are the Arnold process or the new process, like this:

```
int x = fork( );
if (x == 0)
{
    // I am the new process
}
else
{
    // I am the Arnold process
}
```

For the Arnold process, the return value from *fork* is the process ID of the new process. You need to keep track of this so you can use it to signal the child processes. Eventually, you will need to send a signal to the Skunky process using the following system call.

```
kill(Skunky_Process_ID, SIGUSR1)
```

You will signal the Froggy process using the following system call.

```
kill(Froggy_Process_ID, SIGUSR2)
```

But first, the Arnold process must wait for a SIGUSR1 signal from Skunky and a SIGUSR2 signal from Froggy. Arnold will wait using the *pause* function. Skunky and Froggy will find out Arnold's process ID by executing the *getppid* function. They will wander around for a while, and then signal Arnold. After signaling Arnold, they will pause and await a reply. To simulate wandering around, generate a random number from 1 billion to ten billion and loop that many times. (Crude, but it will work.) Then use the *kill* function to signal Arnold.

To intercept and process the signal, use the *signal* system call. Define your handler function as follows:

```
void *Handler(int x)
```

Then call this for SIGUSR1, SIGUSR2, or both. (Arnold will need two handlers.)

```
sighandler_t signal(SIGUSR1,Handler)
```

When Arnold gets Skunky's signal he prints out "I can smell you, Skunky!".

When he gets Froggy's signal he prints out "I can hear you croaking, Froggy!".

When Skunky gets Arnold's signal he prints out "I can see your ears, Arnold!".

When Froggy gets Arnold's signal he prints out "I can see your fur, Arnold!".

Once all processes have sent their signals and printed the above messages they print a final message:

"Arnold going home now." (change the name for the other two)

After printing the final message they quit.