

## Week 11 in-class exercise. Signals

The purpose of this exercise is to explore and play with signals and signal handlers.

1. The first task is to write a program called *greeting* that does nothing except run an infinite loop.

Work with someone in the classroom who has a computer so that you can compile and run your code.

2. Run your code in one window. Kill it by typing Control-C in the same window.

Run it again in the background. For that pause process (Ctrl+Z) and run *bg*.

It should look like it isn't running. Use *ps* to see that indeed it is still running.

Use *fg* to move it to the foreground.

3. Open another window (on the same machine) and run *ps* again. What is the *pid* of your infinite loop program?

Now that you know the *pid* use the *kill* command to kill it.

There are several possibilities that will work and the command parameters will depend on the *pid* of your running process.

Write the command you used here.

4. We are going to write a function in your program called *sing* that will eventually be used as a signal handler. What signature is required for that?

5. Write the function so that it prints the lines of "Happy Birthday" and then returns.

6. Change your program so that it expects one command-line argument that will hold the name of the birthday boy or birthday girl. Now change your *sing* function so that it sings using the actual name.

Hmmm... You can't change the signature. Why not? How to solve this problem?

7. When you run your *greeting* program, it doesn't sing the song because the *sing* function never gets called. Write the code to install *sing* as the handler for the *SIGUSR1* signal.

8. Now run your compiled *greeting* program from one window, look up the PID from another and send it a *SIGUSR1* signal. Did it sing? If not, go back and check your code.

9. Add a *sleep(10);* line to the middle of your singing to simulate taking longer to actually sing the song. Now compile, run and send your program a *SIGUSR1* signal from another window. Now, before it finishes the singing, send it a *SIGINT* signal from the other window. What happened? Why?

10. Change your program so that the *SIGINT* signal is not delivered to the program in the middle of the birthday song. Repeat the actions from above to confirm that the song is finished before the program is killed.

11. One more experiment: Run your signal greeting program in the foreground and kill it with Control-C from the same window. Run it again and from another window, send it the *USR1* signal to start the singing. While it is still singing, try to kill your greeting program using Control-C from its own window (like you just did already.) What happens? Why?