

## CSCI 2330 – Shell Exercises

Suppose you have a shell program with signal handlers for SIGINT and SIGCHLD. The SIGINT handler prints "**shell: SIGINT**" and then forwards the SIGINT to the foreground job as usual. The SIGCHLD handler just prints "**shell: SIGCHLD**".

Additionally, suppose that **prog** is some long-running (non-shell) program with signal handlers for SIGINT and SIGCHLD. The SIGINT handler prints "**pid N: SIGINT**" and the SIGCHLD handler prints "**pid N: SIGCHLD**", where **N** is the pid of the **prog** process.

Consider a shell process with **pid=10** and **pgid=10** in which the following sequence of events occurs. Assume that the OS assigns successive PIDs (11, 12, etc.) to each new process.

1. User executes **prog** in the background (i.e., **./prog &**)
  2. User executes **prog** in the foreground (i.e., **./prog**)
  3. The **prog** process created by step #1 calls **fork**.
  4. User types Control-C to send the shell a SIGINT.
  5. The process created by step #3 exits.
  6. The process created by step #1 exits.
  7. The process created by step #2 exits.
- a. Excluding the shell process, how many processes are created by the above steps? How many jobs are created?
- b. Draw a picture of the process tree just after step #3, including the shell process. Label each process with a PID and PGID.
- c. Write out the output (from all processes) when the above events occur.
- d. If step #7 happens very close to the same time as step #6, how might the output from part (c) be different? Hint: remember pending signals!