Job Control Basics

Kevin Schmidt, Susan Lindsey, Charlie Dey

Processes

Process: an application the computer is running, e.g.: `vim`, the bash shell, window manager.

- Use the `top` command to view **all** processes currently running.
- Use the `ps` command to view just your own processes.
- Each process has its own id, the `pid`
- You can manage processes using job control commands

```
$ ps
  PID TTY          TIME CMD
 6146 pts/7    00:00:00 bash
10137 pts/7    00:00:00 vim
11189 pts/7    00:00:00 ps
```

Jobs

Job: a group of processes started from the current shell.

- Jobs can contain processes, but processes don't contain jobs.
- Use the `jobs` command to display a list of active background processes.
- Each job has its own job id, the `jobid`
- You can manage jobs using job control commands

```
$ jobs
[1]   Stopped                 vim myprogram.cpp
[2]-  Stopped                 vim xer
[3]+  Stopped                 more pascal.cpp
```

Job Control Commands

Command	Description	Example
`ctrl-z`	Suspend the current/foreground job	within `vim`
`ctrl-c`	Kill the current/foreground job	an `a.out` that's hanging
`fg <jobid>`	resume a background job	`$ fg %1`
`bg <jobid>`	place a job in the background	`$ bg %2`
`kill <pid | jobid>`	kill a job or process	`$ kill 10137`
`ps`	display processes - `man ps`	`$ ps`
`jobs`	display jobs	`$ jobs`

Workflows

Development without job control

```
$ <322> vi myprogram.cpp
$ <323> icpc myprogram.cpp
$ <324> ./a.out
$ <325> !322
$ <326> !ic
```

repeat forever

Development with job control

```
$ <363> vi myprogram.cpp
[1]+  Stopped                 vim myprogram.cpp
$ <364> !ic
icpc myprogram.cpp
compilation aborted for myprogram.cpp    #compilation error
$ <365> fg                                #resume editor
vim myprogram.cpp
[1]+  Stopped                 vim myprogram.cpp
$ <366> !ic; ./a.out
```

3 Minute Exercise

Practice some job control. Create and manage multiple edit sessions.

- Edit one of your source files
- Save your edits and suspend the session
- Edit a brand new file, save, and suspend the session
- List your jobs
- Resume editing the first file

Another Example Workflow

```
$ <335> vi myprogram.cpp          #suspend the vi session
[1]+  Stopped                     vim myprogram.cpp
$ <336> icpc myprogram.cpp
$ <337> ./a.out
Hello, world that is beautiful!
Hello, world that is beautiful!
^Z                                #suspend program execution
[2]+  Stopped                     ./a.out
$ <338> jobs                      #list current jobs
[1]-  Stopped                     vim myprogram.cpp
[2]+  Stopped                     ./a.out
$ <339> %1                        #resume the vi session
vim myprogram.cpp
[1]+  Stopped                     vim myprogram.cpp
$ <340> %2                        #resume the program execution
./a.out
Hello, world that is beautiful!
Hello, world that is beautiful!
^C                                #kill the program
$ <341> fg                        #resume the vi session
```