# Lab #2:  Forking and Waiting

**Due Date:**

**September 28, 2016  2355 hours**

**A program is an executable file residing on disk. When you run it, it starts an executing instance known as a process.  Every UNIX process is guaranteed to have a unique numeric identifier called the process ID. The process ID or PID is always a nonnegative integer. The only way under UNIX (with a few exceptions) to create a new process is when an existing process calls the fork function.  You can find details about fork and wait in the manual pages (man fork).**

**The ideal solution is composed of a single parent process which simultaneously opens up multiple child processes which performs their function once and then quits.  Code to close up the child processes should be at the end of the program.**

**Also introduced in this lab are command-line options for UNIX programs. By convention, your program should start with the following two commands:**

```
void main(argc, argv)          //argc  "argument count" is an integer of the number of arguments
int argc; char *argv[];        //argv  "argument vector" is a character array of the arguments
```

**Write a simple program that uses the fork function. Take a look at pid.c and fork.c as sample code (lecture notes and moodle) and the digital notes about "C System Calls.pdf" which covers fork, exec, wait, and exit in more detail.**

**Program requirements:**

1. **The program should take either filenames or wildcards (*.c for example) as input**
2. **Create as many child processes as there are files on the command line.  Child processes should run simultaneously, not wait on each other.**
3. **For each child process, print the file name and the process ID of the child process**
4. **The parent process should wait for all the children to finish, show the output from system("ps -H"); and then print "Done" to the screen.  system("ps -H"); placed other locations of your program may be a helpful command to debug your code.**
5. **5% Extra Credit: instead of Moodle, submit your assignment via https://education.github.com/ (MUST be private, invite swirsz) Make sure your full name is included in the comments.  I'll send an email to all individuals who I have shared Github access 1 day before the assignment is due informing them I will retrieve their program from Github.**

**You must submit two things: program files and the output from running the program.**

```
bash-4.1$ ./p2 *.c > log.txt
bash-4.1$ cat log.txt
      Filename: program2.c          PID: 21755
      Filename: fork1.c             PID: 21754
      Filename: pid.c               PID: 21756
  PID TTY          TIME CMD
 3248 pts/19   00:00:00 ksh
 3359 pts/19   00:00:00   bash
28345 pts/19   00:00:00     p2
28360 pts/19   00:00:00       ps
Done!
bash-4.1$
```

Last Revised: September 29, 2016