University of Massachusetts Dartmouth
CIS 370, Fall 2013
Lab 3 – Process Management
09/24/2012, 09/26/2012

Objective

This lab is designed to focus on the different system calls used to create, synchronize, and execute processes in a UNIX environment. Upon completion of this lab, you should have a thorough understanding of the behavior of `fork` and `exec` calls.

Description

This lab consists of two programming assignments. Please submit two separate C files (one for each part of the assignment). It would behoove you to review the lecture slides on processes before you begin.

1. Fibonacci's Child

The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, …. Formally it can be expressed as:

$fib_0 = 0$
$fib_1 = 1$
$fib_n = fib_{n-1} + fib_{n-2}$

Write a C program which creates two processes (a parent and a child) that generates the Fibonacci sequence on the *CHILD* process. The number of terms to which the sequence should be calculated will be passed in via the command line. Perform necessary error checking to ensure that a non-negative number is passed in. The child should print its PID before starting the sequence. The parent should wait for the child to finish printing the sequence, and then print its own PID before exiting.

Example:

Entered at the command line	Output
`$ ./lastnameFibonacci 2`	`0, 1`
`$ ./lastnameFibonacci 5`	`0, 1, 1, 2, 3`
`$ ./lastnameFibonacci 10`	`0, 1, 1, 2, 3, 5, 8, 13, 21, 34`

Hint: Use the function `atoi()` to convert an argument passed in through the command line into an integer.

For example:

```
atoi(argv[4]);
```

2. Execute Copy

Write a C program called `lastnameExecute` which is able to execute the `sampleCP` program included with the lab materials online. The `sampleCP` program is a simplified version of your own `lastnameCP` program from Lab 2, which takes two filepaths as parameters. Upon successful completion of the copy, `lastnameExecute` (NOT the `sampleCP` program) should

print out a confirmation message. The input to `lastnameExecute` will be the same two filepaths passed to `sampleCP`. **You should not modify `sampleCP` however you MUST compile it before being able to execute it. Please call the executable "sampleCP".**

Example:
Running the following from the command line:

```
$  ./lastnameExecute /tmp/file1 /tmp/file2
```

Should result in `sampleCP` performing the copy from *file1* to *file2* and produce terminal output similar to this:

```
Beginning execution of sampleCP.
tmp/file1 successfully copied to tmp/file2…Exiting.
```

Deadline

Section 01 – Tuesday, 10/01/13
Section 02 – Thursday, 10/03/13

Useful Links: Linux/Unix process management

- **Chapter 5**, *Unix System Programming*, Haviland and Gray, Prentice Hall (Lab Book)
- **Chapter 3**, *Operating System Concepts*, Silberschatz, Galvin, Gagne (Course Textbook)
- http://www.advancedlinuxprogramming.com/alp-folder/alp-ch03-processes.pdf