## Programming Assignment 2

### Objective

The objective of this assignment is to familiarize yourself with the notion of a process and the system call *fork( )* used to create a new process and *pipe( )* to communicate between processes.

### Assignment: Using the Fork System Call

Write a simple shell interface program that takes a simple shell command as user input from the parent process then sends the simple shell command to the child process via a pipe IPC which actually executes the shell command.

The idea is to write a C/C++ program using the *fork*( ) system call to create a child process that blocks on the pipe waiting for some simple shell commands sent by the parent process via the *pipe( )* IPC. The child process is then to execute the shell command read on the pipe IPC.

#### Simple Shell Program Implementation

The simple shell program (s***shell.c***) is a simple text-based program that takes user input from the parent then sends that input to the child process for execution. Both the parent and child process should loop waiting on either user input (parent) or pipe input (child). The program should run until the user types ‘quit’ or ‘q’ at the command prompt.

To start the simple shell program

*./****sshell***

*osh> <simple shell command>*  (where the user types in a shell command (ie *ls –l*))

The *<simple shell command>* is then sent to the child process via the pipe IPCwhich executes the command by invoking *execlp( ).* In the example above the directory contents would be displayed by the child process.

Basic Algorithm

1. Create the pipe for parent/child IPC
2. Fork the child which should start to block on the pipe waiting on shell commands
3. Parent should block waiting on user input
4. Once the shell command is input parent writes the command to the pipe
5. Child reads pipe, executes command, blocks on pipe waiting for next command.
6. Repeat steps 3-5 until ‘quit’ or ‘q’ is typed by user.

#### Error Handling

Perform the necessary error checking to ensure that a valid shell command was entered.

### Grading

The program will be graded on the basic functionality, error handling and how well the implementation description was followed. Be sure to name your program **sshell.c** (no extra characters, capitals) Note that documentation and style are worth 10% of the assignment's grade!

### Submission

The program source code should be posted to Blackboard.