myShell

Learning Objectives

- Design and implement a C program using structured programming techniques
- Design and implement a C program using applying secure cyber design
- Design and implement a C program that creates a new process
- Design and implement a C program that monitors child processes
- Design and implement a C program that executes another program

Before Class

In preparation for in class activities, complete the following activities:

- None.

In Class

Assignment Information

Lab 0x06: myShell

Assignment Type:	Programming - Laboratory	Collaboration Policy:	Default
Assignment		Lab 0x06: myShell	
Due			
Section		Course Server Time / Date	
1121		Friday, 1800 07 Apr 2018	
3122		Friday, 1800 07 Apr 2018	
5123		Friday, 1800 07 Apr 2018	

General Comments:

- Make use of in-class time to the fullest extent possible
- Read the entire assignment before you begin
- You are given more time to complete programming assignments because they are expected to take you longer to complete
    - It is expected that you will need to work on programming assignments outside of scheduled class
    - Do not procrastinate on starting a programming assignment

Given Material:

- myShell.c

Test

The following are examples of how your program may be tested from the command line.

```
$ ./myShell Any Arguments  # Any arguments
NAME
        myShell - A simple shell

SYNOPSIS
        myShell

DESCRIPTION
        Tokenizes and parses single line command lines.

        ';' is used as a separator between commands on the same line.

$ ./myShell  # Normal usage
(PID)-#!>     # Awkward Blinking Cursor
(PID)-#!> ^C  # Ctrl-c to generate SIGINT
$  # myShell terminated, normal shell resumes

$ ./myShell
(PID)-#!>
(PID)-#!>     # 0x90
(PID)-#!> ^C

$ ./myShell
(6413)-#!> ls  # Part 1 - Single Command Line (No Arguments)
myShell  myShell.c
(6413)-#!> ps
  PID TTY          TIME CMD
 5742 pts/0    00:00:00 bash
 6413 pts/0    00:00:00 myShell
 6419 pts/0    00:00:00 ps
(6413)-#!> ^C

$ ./myShell
(6414)-#!> which ls  # Part 1 - Single Command Line (With Arguments)
/bin/ls
(6414)-#!> which gcc
/usr/bin/gcc
(6414)-#!> gcc myShell.c -o myShell2
(6414)-#!> ls
myShell  myShell2  myShell.c
(6414)-#!> ^C

$ ./myShell
(6415)-#!> badCmd  # Part 1 - Single Command Line (Bad command name)
myShell-ERROR 2-bad: No such file or directory
(6415)-#!>

$ cp /bin/ps ls  # Part 2 - Secure the PATH Setup
$ ls
ls  myShell  myShell2  myShell.c
$ unset PATH
$ ./ls
  PID TTY          TIME CMD
 5742 pts/0    00:00:00 bash
 6416 pts/0    00:00:00 ls
$ ./myShell
(6417)-#!> ls  # Part 3 - Secure the PATH
ls  myShell  myShell2  myShell.c
(6417)-#!> ^C

$ ./myShell
(6418)-#!> ls ; ps  # Part 3 - It's a Party Line
ls  myShell  myShell2  myShell.c
  PID TTY          TIME CMD
 5742 pts/0    00:00:00 bash
 6418 pts/0    00:00:00 myShell
 6420 pts/0    00:00:00 ps
(6418)-#!> ps ; ls
  PID TTY          TIME CMD
 5742 pts/0    00:00:00 bash
 6418 pts/0    00:00:00 myShell
 6421 pts/0    00:00:00 ps
```

```
ls  myShell  myShell2  myShell.c
(6418)-#!>
```

After Class

Activities

Furthering the in class activities, complete the following activities:

- Turn in the worksheet to your instructor after you have submitted your final source code.
- Submit your final source code in a single file named `myShell-#.c`, where `#` is the part of the implementation that you completed (e.g. `myShell-3.c`), following the submission directions on the course information web page.

  `submit` Assignment (project): *lab06*

Resources

Assignment Solutions:

- Compiled Solution: myShell (Debian x86-64)
- Worksheet Solution: myShell
- Source Code Solution: myShell.c