# Introduction

The Department of Electrical and Computer Engineering The University of Alabama in Huntsville

CPE 435 Lab-01

Overview of OS model and Programming Tools

The goal of this lab is to familiarize you with your work environment for using the operating systems and to get started with linux processes.

For the most part these notes will apply to whichever environment you are using but you may have as yet unexpected issues with non windows10-wsl2-ubuntu configurations. Please contact the TA’s or myself if issues arise.

You will not have any physical access to UAH computers. You will have to use them remotely. The only computers authorized for this course are the special ones we are setting up for scheduled, remote access to win10-wsl2-ubuntu

DO NOT USE ANY OTHER MACHINES FOR THIS COURSE. USING BLACKHAWK WILL RESULT IN A 0 FOR THE ASSIGNMENT. IT IS THE CENTRAL SERVER AND I ASSURE YOU SOME OF YOU WILL CRASH THE MACHINE YOU ARE USING FOR THIS COURSE. IF YOU CRASH BLACKHAWK REMOVE MY EMAIL FROM YOUR CONTACTS LIST.AND SWITCH MAJORS.

# PartI: Entering, Compiling and Running a short program in Lab

For editing and preparing programs, the following notes will present an introduction to the *vi* text editor and the GNU C++ compiler. There is also a handout. There are several other editors available that you may wish to use instead of *vi*.

Type the following in terminal. *vi ﬁlename.cpp* (starts the editor) *i* (enter insert mode)

Following is the code that you will insert in your ﬁle:

# Demo Code

## Demo Code 1

using namespace std;

#include <iostream> #include <stdlib.h> #include <sys/types.h>

int main(int argc, char \* argv[]) {

pid\_t c\_pid;

cout << "The pid of the parent is “ << getpid() << endl;

c\_pid = fork();

if (c\_pid == 0){ /\* we are the child \*/

cout << "I am the child; my parents pid was " << getppid() << endl; exit(0);

}

cout << "I am the parent, the childs pid is " << c\_pid << endl;

exit(0);

}

**Press: [ESC]-colon-wq-[Enter]** after you are done typing the code. This will close the *vi* editor and save your work.

In terminal, **g++ ﬁlename.cpp -o ﬁlename** [Enter]

The above statement tells gnu compiler to compile your program into ﬁlename which is an executable

In terminal, **/ﬁlename** [Enter] should run the executable

Within *vi*, you can enter command mode at any time (escape-colon). You can move the cursor with either the j-down, l-left, h-right, k-up keys, or sometimes, but not always, with the cursor keys. A good thing to remember is escape-colon-quit-! which gets you out of the editor without saving the ﬁle (a good thing when something bad has happened).

To understand any command in linux, you can generally execute the *man* command, for example:

***man fork***

This should print the man page for fork,. If you are not sure of the command you want to use, you might try the command “apropos searchword”, which looks up commands which have searchword in the man page. Alternatively, you might want to buy a book on linux for a few dollars at the bookstore. Old books are generally ok for understanding commands.

# Extra Demo Codes

## Demo Code 2

The following code forks and prints the value of x from both the processes along with their process id.

/\*

This program illustrates the use of fork

-Prawar Poudel 7Jan2018

\*/

#include <stdio.h>

#include <iostream> #include <unistd.h> using namespace std; int main()

{

int x; x=0;

fork(); x++;

//this should be printed twice, once by parent and once by child cout<<"I am process "<<getpid()<<" and my x is "<<x<<endl; return 0;

}

### Assignment 1

Analyze the output for the demo code above in terms of order in which output statements are printed.

## Demo Code 3

/\*

This program illustrates multiple fork operations

-Prawar Poudel 7Jan2018

\*/

#include <stdio.h> #include <iostream> #include <stdlib.h> #include <unistd.h>

using namespace std;

int main()

{

printf("I am Parent\n"); fork();

printf("This by a parent and child\n"); fork();

printf("This should be 4 times\n"); return 0;

}

### Assignment 2

Hoe many processes are created from the above demo code? Explain

# Demo Code 4

/\*

This program illustrates the death of parent before the child is terminated The child hten becomes orphan since the parent is dead.

-Prawar Poudel 7Jan2018

\*/

#include <stdio.h> #include <stdlib.h> #include <unistd.h>

int main()

{

int pid;

pid = fork();

if(pid==0)

{

printf("I am child, my Id is %d\n",getpid()); printf("i am child, my parent is %d\n",getppid());

printf("Child is sleeping for 10 seconds until when the parent might be dead already\n");

sleep(10);

printf("I am the same child with Id %d, but my parent id is

%d",getpid(),getppid());

}else

{

printf("I am the parent with id %d. My parent is %d and child is

%d\n",getpid(),getppid(),pid); sleep(5);

}

}

## Assignment 3

What is orphan process? Is there any orphan process in above code? What is pid of orphan process if any is present?

# Part II (Expanding the program in Demo 1)

## Assignment 4

For homework, write a program that forks oﬀ 10 children, each of which will print out its pid and its serial number (1,2,3,4,5...10).

# Report Format

Background (as instructed)

Procedure and Results (what I did −− and what happened when I did it!) Go through the lab step by step and describe what happens as you progress through it. Go idea to have another window open where you can record what is going on as it happens. Make sure to answer any questions that are asked and perform any procedure that is asked of you. These are often underlined. You can capture any output from the program by using the UNIX redirect operator that sends data that normally goes to the screen to a text ﬁle. This can be merged into your report.

Conclusion: Brief statement on what you believe was the major item(s) that you were able to learn and/or demonstrate.