Introduction to C and Linux

For the lab you are required to have a distribution of linux in order for you to compile the program as well as checked whether your program have run correctlly.

After this lab, students should be able to:

- Create, Edit, Save a file using Linux
- Compile a C program
- Check for open port and running program

Exercise 1:

1. Open up a terminal.
2. Create a new file by using the following command:

   ```
   touch helloworld.c
   ```

3. Edit the file by using nano text editor.

   ```
   nano helloworld.c
   ```

4. Type in the following program segment:

   ```
   #include <stdio.h>

   int main(void) {

   /* This is my first program in C */

   printf(“Hello World!”);
   printf(“I Love C”);

   return (0);
   }
   ```

5. Save the program by pressing ctrl O. When prompted for file name, press ENTER.
6. Exit from nano text editor by pressing ctrl X
7. The file should have been saved inside your home directory, to verify run the following command:

   ```
   ls –l
   ```

   You should see the file listed along with other files inside the directory.

Exercise 2: Compiling a C Program

In linux, the popular compiler are known as GNU C Compiler or gcc in short. For this course, we are going to use gcc to compile our C program.

1. Open up a terminal.
2. Go to your home directory or directory where you have saved your helloworld.c. To go to your home directory type *cd* or *cd /home/your_username.*
3. Compile the program by using the following command:

   ```
   gcc helloworld.c
   ```

4. If you get an error, edit the program again using nano and recompile. If you get no error (after running the above command, you are returned to the prompt) then congratulations you have succesfully compile your first C program.
5. A succesful compilation will create an object file called a.out. Check whether the file exist by running:

   ```
   ls –l
   ```

   You should see a file called a.out

6. To see the output of the program, you can run the file by running the following command:

   ```
   ./a.out
   ```

Exercise 3: Assigning names to the output file

One problem with having the output file name as a.out is that once we compile a new program, we are unable to run our previously compiled program without recompiling the program. In order to overcome this program, we could give a name to the output file during compilation.

1. Open up a terminal.
2. Go to your home directory or directory where you have saved your helloworld.c. To go to your home directory type *cd* or *cd /home/your_username.*
3. Compile the program by using the following command:

   ```
   gcc –o helloworld.out helloworld.c
   ```

   the option –o let us specify the name of the output file.

4. A succesful compilation will create an object file called helloworld.out. Check whether the file exist by running:

   ```
   ls –l
   ```

   You should see a file called helloworld.out instead of a.out.

5. To see the output of the program, you can run the file by running the following command:

   ```
   ./helloword.out
   ```

Exercise 4: Getting a variable from user

1. Open up a terminal.

2. Create a file called userinput.c by using nano text editor.

   ```
   nano userinput.c
   ```

3. Type in the following program segment:

   ```
   #include <stdio.h>

   int main(void) {

   /* This is my second program in C */

   int age;

   printf(“Hi, how old are you? > ”);
   scanf(“%d”, &age);

   printf(“You are %d years old”, age);

   return (0);
   }
   ```

4. Save the program by pressing ctrl O. When prompted for file name, press ENTER.
5. Exit from nano text editor by pressing ctrl X
6. The file should have been saved inside your home directory, to verify run the following command:

   ```
   ls –l
   ```

   You should see the file listed along with other files inside the directory.

7. Compile the program by using the following command:

   ```
   gcc –o userinpit.out userinput.c
   ```

8. If you get an error, edit the program again using nano and recompile.

Exercise 5: Checking for open port

There are times when we are required to check list of port open at our PC, to do this we can utilize the *netstat* command:

1. Open up a terminal.
2. Run the following command:

   ```
   netstat –an | grep <port no.>
   ```

   For example if you are interested in checking for port 80, you would run the command as:

   ```
   netstat –an | grep :80
   ```

3. If there is a result, it shows you that there are currently port 80 opened in your computer (or you are accessing port 80 at the remote location.

Exercise 6: Checking for running program

At times, we wanted to know if a program is running (for example a child program) then we can use the *ps* command to check.

1. Open up a terminal
2. Run the following command:

   ```
   ps waux | grep <command>
   ```

   replace command with the name of the program that you are interested in.

3. If you get a result, it shows that the program is still running.