

**Department of ICT  
Faculty of Technology  
University of Ruhuna**

**Programming Practicum – ICT1142**

**Level 1- Semester 1**

**Lab Sheet 01**

**| 2022**

**Objectives:**

- To understand the execution environment of a C program and to write, edit, compile and execute a basic C program in Linux (Ubuntu) environment.
- To familiarize with basic Linux commands.

**Equipment**

- A PC running the Ubuntu Linux operating system (OS).
- The GCC 'C' compiler – it is installed by default on Ubuntu.
- A text editor

**The development environment and the development cycle**

C programs go through 3 main phases during development:

- ✓ **Editing** (writing the program)
- ✓ **Compiling** (i.e., translating the program in to an executable code and detecting syntax errors)
- ✓ **Running** the program and checking for logical errors

**1. Edit**

Editing a file by typing the C program in a text editor and making corrections if necessary. The program is stored as a text file on the disk, usually with the file extension **.c** to indicate that it is a C program (e.g. helloworld.c ).

**2. Compile**

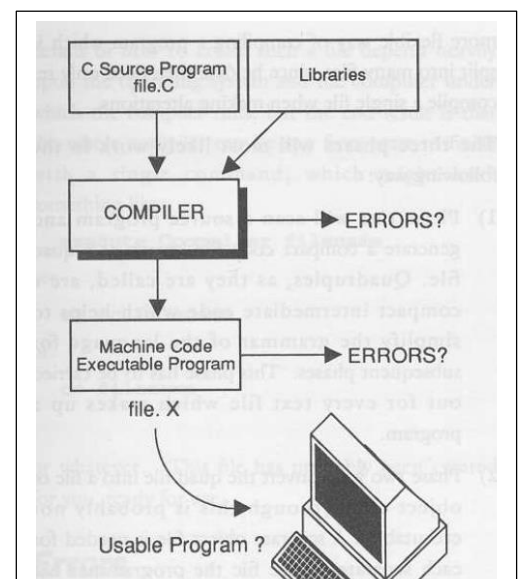
A compiler translates the C program (e.g. helloworld.c) (**source code**) into machine language code (**object code**), which stores on the disk as a file with the extension **.o** (e.g. helloworld.o). A **linker** then links the object code with **standard library** routines that the program may use and creates an **executable image**, usually as a file with the file name without any extension (e.g. helloworld ).

In C program uses “**gcc**” compiler to compile the source code. The gcc is known as command line compiler.

*General form in Linux platforms (Ubuntu).*

**gcc [source file name] -o [executable filename]**

Ex: - **gcc helloworld.c -o helloworld**



The **-o** option informs the compiler of the desired name for the executable file that it produces. The name used in this example is helloworld. If the **-o** option is omitted; the compiler will give the name **a.out** to the executable file.

### 3. Execute

Once the executable file has been created by the compiler (which means no any errors), it is ready to run/execute. The executable file is loaded from the disk to memory and the computer's processing unit (Central Processing Unit) executes the program one instruction at a time.

To run, proceed the executable file name with a dot and a forward slash and then press the ENTER Key.

Ex: - **./helloworld**

---

## Study on Linux commands

### Exercise 01

**In this part, you will learn about some basic Linux commands.**

1. Login to the Linux machine
2. Open a terminal window

Open a Terminal by doing the one the following.

Ubuntu icon → Type "Terminal"

or

Press .Ctrl+Alt+T. This will launch the Terminal.

3. To see which directory you are in (current working directory), use the command, **pwd**  
In your login you will be able see your home directory Ex: - **/home/ict**

**\$ pwd**

Note that, **pwd** stands for "print working directory".

4. To list the files and other directories (or folders) in your current directory, type the command **ls**

*Note that, because you don't have any files or directories you will not see files or directories created by the user.*

Some extensions to the **ls** command that provides additional details of files and directories are as follows. List down all the files/directories including hidden files/directories. Try them also.

**ls -al**

**ls -as**

**ls -las**

Now, you will see a number of files, directories that have been created for you by the LINUX OS.

The way to distinguish between a file and a directory is that, on the second column from the left in the output of **ls -las** has the character “**d**” for all the Directories. Others have the character “**-**”.

5. To make a directory/folder use the command **mkdir** <directory name>

To make a directory with the Student Reg.Number “TG\_2020\_123” type the following.

```
$ mkdir TG_2020_123
```

Type **ls** command to check whether directory is created or not.

6. To remove a directory/folder use the command **rmdir** <directory name>

```
$ rmdir TG_2020_123
```

7. To go to a different directory/folder (or change the directory), use the command, **cd** <directory name>

First create the same directory again

```
$ mkdir TG_2020_123
```

Let’s go into the directory that we created above. Type the command,

```
$ cd TG_2020_123
```

Now type **pwd** to see your current directory path.

8. To go one directory backwards (or to the current directory’s parent directory), type the following command.

```
$ cd ..
```

Now again type **pwd** to see your current directory path.

9. To clear the terminal type **clear** and it will clear your terminal.

10. Create the C source code by using text editor. Use the following commands.

First you have to go into the directory that we created above (You can create your own directory with any name).

```
$ cd TG_2020_123
```

Open the text editor and create a C source file named myfile.c

```
$ gedit myfile.c
```

Now text editor is opened and you can create your source code. After creating your source code, save your works and close the editor window.

11. Compile the program.

**\$ gcc [source file name] -o [executable filename]**

**Ex: -**

**\$ gcc myfile.c -o myfile**

If it shows some errors, open the text editor again and correct them.

12. Execute the program.

**\$ ./[executable filename]**

**Ex: -**

**\$ ./myfile**

## Exercise 02

### Learn first C program

- Open a new source file to create a program called *helloworld.c*

```
//My first program
#include <stdio.h>

int main()
{
    printf("Hello, World! \n");
    return 0;
}
```

- Compile and execute the program.  
➤ If there are errors, correct them and recompile and execute.  
➤ The result will be to write the text “Hello, world!” on the monitor screen.

## Exercise 03

Write a simple C program called “myhobbies.c”, which print the output as bellow. Insert inline comment at top of the program as “My register number is XXXXX”. Print your hobbies in the three lines.

Hello, I am <your name>  
My Hobbies are:  
<Hobby 1>  
<Hobby 2>  
<Hobby 3>

#### **Exercise 04**

Write a simple C Program to display Your Name, School and Subjects you learned in A/Ls using three (03) display statements as follows.

Name: Suresh Perera  
School: Central College Matara  
Subjects: BST, ICT, ET

#### **Exercise 05**

Modify the Program in Exercise 04, to get the same output using a single display statement.