**Operating System**

**Lab Report 2**

**Hafiz Ahmad**

**19l-1316**

**Section-6A2**

Creating, Compiling and Executing C/C++ programs using gcc/g++ Compilers and Make File

**INTRODUCTION:**

In this lab we learn about compiling and ececuting ths c /c++ programs we also introduce with many other options as we study Command line arguments are a way to pass data to the program. Command line arguments are passed to the main function. Suppose we want to pass two integer numbers to main function of an executable program called a.out. There are three compilation stages the first one is compiler stage in which cpp file is converted low language file which is assembly language file. The second stage is the Assembly stage in which assembly language code converted in object code which is fragments of code which computer understand directly. The last is the linker stage which links the object code to code libraries which contain certain built-in functions.

Compiling C/C++ program using g++ and gcc:

For C++: Command:

g++ source\_files… -o output\_file

For C: Command:

gcc source\_files… -o output\_file

**OBJECTIVES:**

• Learn the use of g++ and gcc compilers to compile and execute C++ and C programs

• To get familiarized with the working of Make File for C/C++ programs

**Application:**

By use of linux we easily do the same code as we done in windows on visual studio . It is quiet easier as just create file and code it and rum into the terminal for output.It makes codes more concise and clear to read and debug. No need to compile entire program every time whenever you make a change to a functionality or a class. Makefile will automatically compile only those files where change has occurred.02-Nov-2018

**Issues:**

No issue found while performing the lab.

**Conclusion:**

In this lab we learn how to Creating, Compiling and Executing C/C++ programs using gcc/g++ Compilers and Make File. We also familiar with the g++ /gcc working with make file concepts.

**Post lab:**

Problem 1:

Code:

Text

Description automatically generated

Input: 8 4 9 19 8 2



Problem 2:

Code:

Text

Description automatically generated

Input: 6 10 12 14 16 20



Problem 3

Code:

Text

Description automatically generated

Text, letter

Description automatically generated

Input:

1 2

2 4

3 6

4 10

5 12



Problem 4:

Output:

Make g++ -c letter.cpp

G++ -o Ot main.o word.o letter.o line.o

Text

Description automatically generated

Letter.h file



Letter.cpp file

Text

Description automatically generated

Word.h file



Word.cpp file

Text

Description automatically generated

Line.h file



Line.cpp file

Text

Description automatically generated with medium confidence

Main file

Graphical user interface, text, application, chat or text message

Description automatically generated

Makefile

