COS LAB 1

**NAME:** Rahul Varma

**ROLL No:**  S20200010212

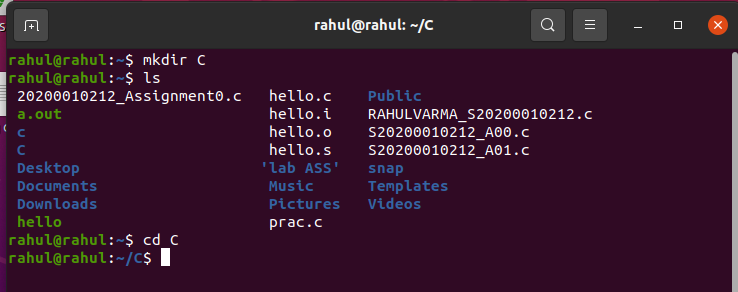
**Section : C**

**TASK** : How to Compile and Run “hello world” Program in Linux

Using gcc.

Complete procedure of analyzing a hello world program:

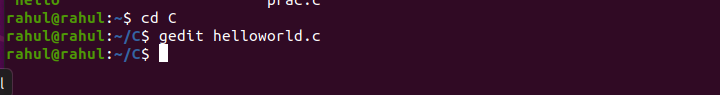
Making a directory in Linux subsystem ‘C’:



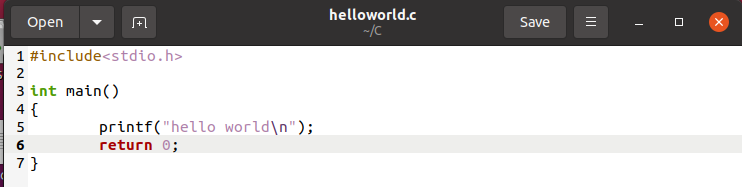
Now the directory named C has been selected.

Start by new .c file to write “hello world” code in.

I used **gedit** :



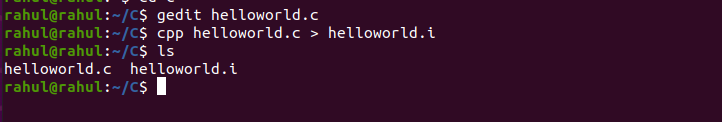
We can write the hello world C program in the opened editor:

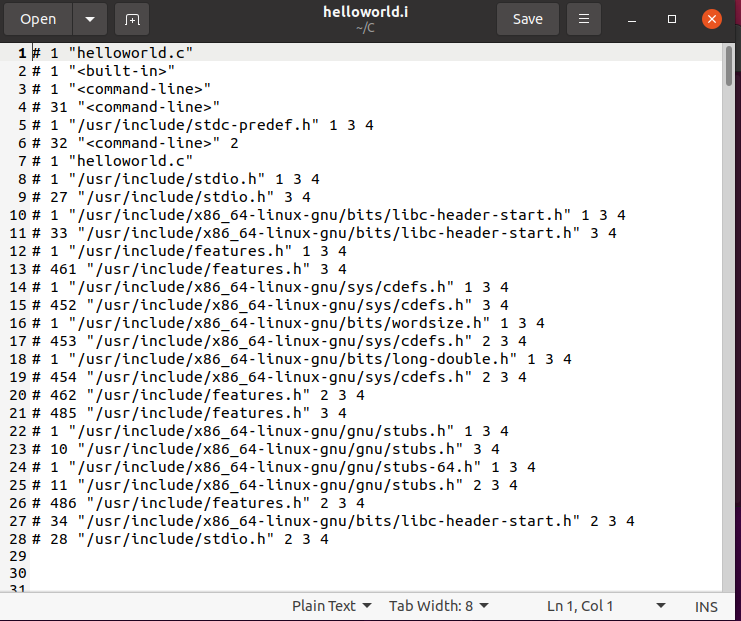


After writing the C code, we start the process of pre compilation

For the pre compilation, I used command:

cpp helloworld.c > helloworld.i



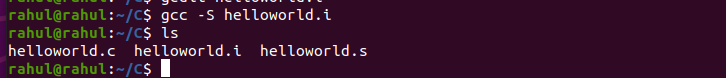
And when it is opened(using gedit helloworld.i), we can see a similar code as below (source file with all macros expanded): 

This is a preprocessor command and converts the C file as how it’ll be when it after done preprocessing.

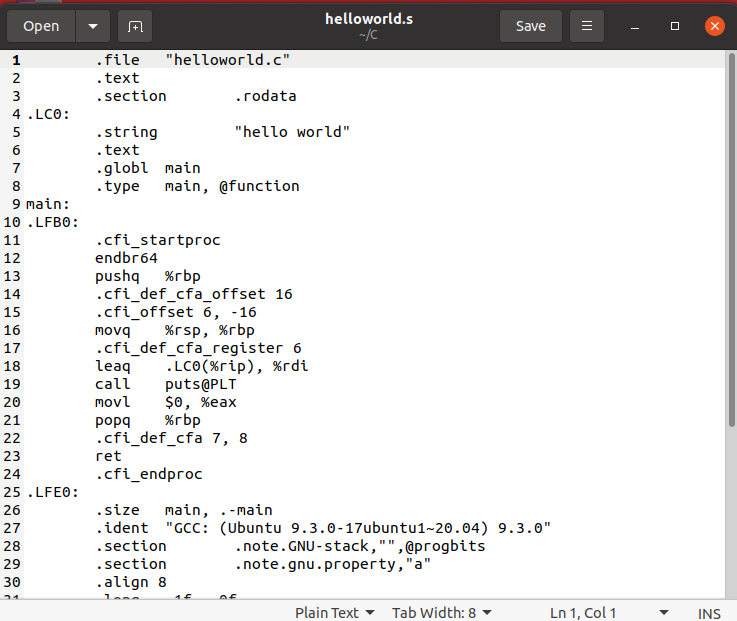
Now Compilation part :

And then comes the compilation part, for which I used as command :

gcc -S helloworld.i



And when it is opened(using gedit helloworld.s), we can see the assembled code as below:



Note: There’s more code,

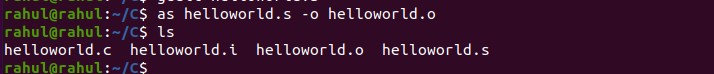
Only some part is taken screen shot

**Assembly Part:**

Now, we need to assemble helloworld.s to helloworld.o, for

this operation I used as operation:

Command used : as helloworld.s –o helloworld.o



And when it is opened, we can see that the assembler converts .s file to machine language and generates and object file (.o) named

Helloworld.o:



Note : here in this process, there is a undefined reference to printf

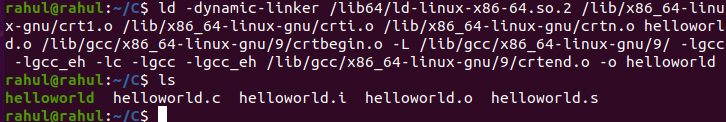
**Linking:**

In the final process, we need to link object files to executable file

which can be done by the following command:

ld -dynamic-linker /lib64/ld-linux-x86-64.so.2 /lib/x86\_64-linux-gnu/crt1.o /lib/x86\_64-linux-gnu/crti.o /lib/x86\_64-linux-gnu/crtn.o helloworld.o /lib/gcc/x86\_64-linux-gnu/9/crtbegin.o -L /lib/gcc/x86\_64-linux-gnu/9/ -lgcc -lgcc\_eh -lc -lgcc -lgcc\_eh /lib/gcc/x86\_64-linux-gnu/9/crtend.o -o helloworld

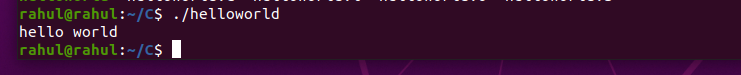
Execution of the above command:



Finally we are left with the final executable program helloworld,

we can use the command:

**./helloworld**



Conclusion :

I can conclude that the “hello world” program is executed by

(with out using gcc command)

Four steps:

1. [Preprocessing](https://cs-fundamentals.com/c-programming/how-to-compile-c-program-using-gcc#compile-c-program-preprocessing)

2.[Compilation](https://cs-fundamentals.com/c-programming/how-to-compile-c-program-using-gcc" \l "compile-c-program-compilation)

3.[Assembly](https://cs-fundamentals.com/c-programming/how-to-compile-c-program-using-gcc" \l "compile-c-program-assembly)

4.[Linking](https://cs-fundamentals.com/c-programming/how-to-compile-c-program-using-gcc#compile-c-program-linking)

Analysing each and every step in compiling C program to executable files ie. Pre compiling,assembling ,linking and running

Helloworld program with out using gcc command .

THANK YOU