

# LAB MANUAL Operating Systems

## Lab 2

### Execute a C/C++ program on terminal using gcc/g++ compiler

Follow these steps to run programs on terminal:

**Step 1.** Open terminal.

**Step 2.** Type command to install gcc or g++ compiler:

**\$ sudo apt-get install build-essential**

This will install the necessary C/C++ development libraries for your Ubuntu to create C/C++ programs.

To check gcc/g++ version type this command:

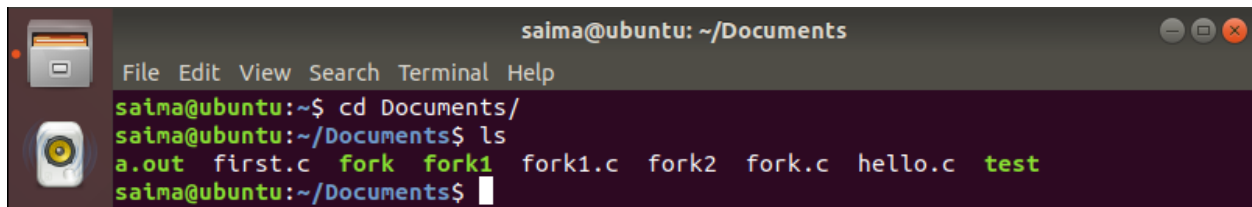
**\$ gcc --version or gcc -v**

**\$ g++ --version or g++ -v**

**Step 3.**

Now go to that folder where you will create C/C++ programs. I am creating my programs in Documents directory. Type these commands:

**\$ cd Documents/**

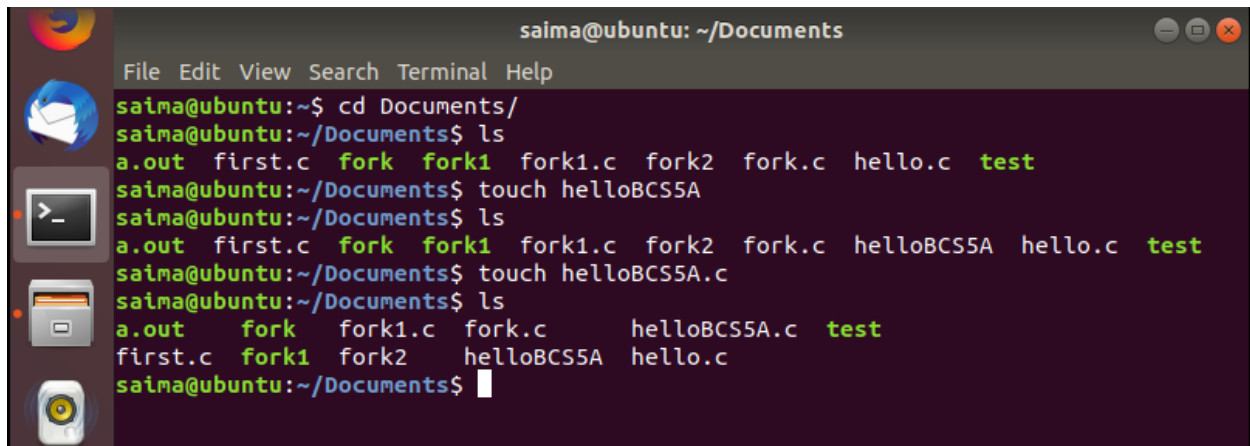
A screenshot of a terminal window titled 'saima@ubuntu: ~/Documents'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the following commands and output:

```
saima@ubuntu:~$ cd Documents/  
saima@ubuntu:~/Documents$ ls  
a.out first.c fork fork1 fork1.c fork2 fork.c hello.c test  
saima@ubuntu:~/Documents$
```

**Step 4.**

Creates a helloBCS5A.c file

**\$touch helloBCS5A.c (for C programs)**

A terminal window titled 'saima@ubuntu: ~/Documents' with a menu bar (File, Edit, View, Search, Terminal, Help). The user enters commands to create files and directories. The terminal output shows the creation of 'a.out', 'first.c', 'fork', 'fork1', 'fork2', 'fork.c', 'hello.c', and 'test'. Then, 'helloBCS5A' is created using 'touch'. Subsequent 'ls' commands show the updated file lists.

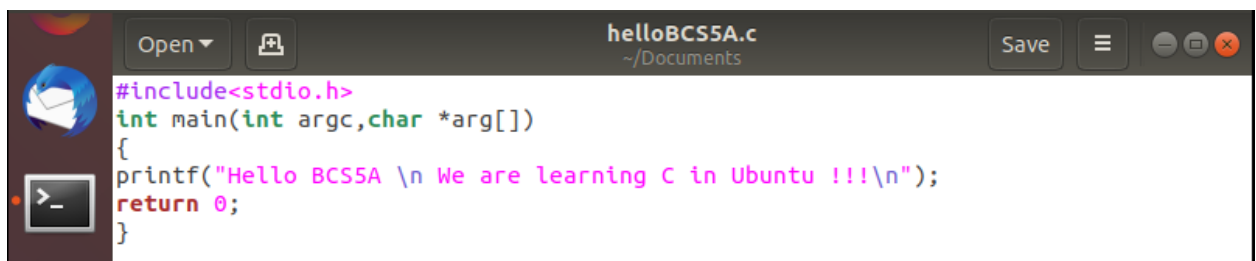
```
saima@ubuntu:~$ cd Documents/
saima@ubuntu:~/Documents$ ls
a.out first.c fork fork1 fork1.c fork2 fork.c hello.c test
saima@ubuntu:~/Documents$ touch helloBCS5A
saima@ubuntu:~/Documents$ ls
a.out first.c fork fork1 fork1.c fork2 fork.c helloBCS5A hello.c test
saima@ubuntu:~/Documents$ touch helloBCS5A.c
saima@ubuntu:~/Documents$ ls
a.out fork fork1.c fork.c helloBCS5A.c test
first.c fork1 fork2 helloBCS5A hello.c
saima@ubuntu:~/Documents$
```

### Step 5.

Add this code in the file:

C program code:

```
#include<stdio.h>
void main(int argc,char *arg[])
{
printf("Hello BCS5A \n We are learning C in Ubuntu !!!\n");
return 0;
}
```

A code editor window titled 'helloBCS5A.c' with a menu bar (Open, Save, and icons). The code is pasted into the editor, showing the same C program as in Step 5.

```
#include<stdio.h>
int main(int argc,char *arg[])
{
printf("Hello BCS5A \n We are learning C in Ubuntu !!!\n");
return 0;
}
```

Step 6. Save the file and exit.

Step 7. Compiling C program.

```
$ gcc helloBCS5A.c -o helloBCS5A
```

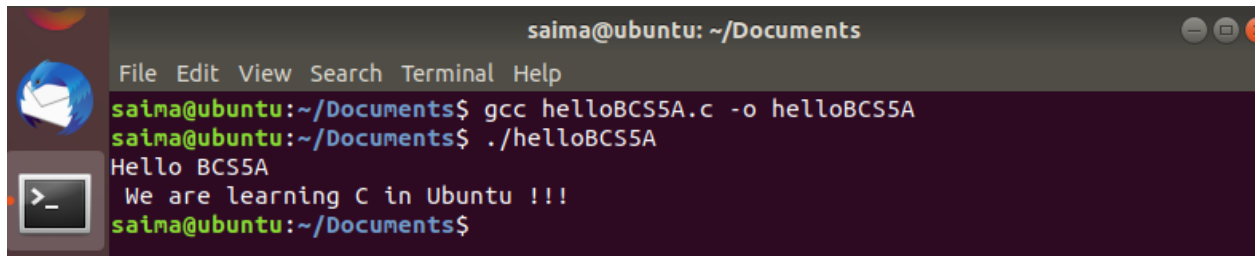
It will create an executable file

Step 8. To run this program, type this command:

```
$ ./helloBCS5A
```

It will show output on the terminal.

Step 9. OUTPUT



```
saima@ubuntu: ~/Documents
File Edit View Search Terminal Help
saima@ubuntu:~/Documents$ gcc helloBCS5A.c -o helloBCS5A
saima@ubuntu:~/Documents$ ./helloBCS5A
Hello BCS5A
We are learning C in Ubuntu !!!
saima@ubuntu:~/Documents$
```

To clear the screen

\$ clear

## Lab Tasks

1. Install g++ in your VM, learn how to create, save, and code a C file. How to compile and run using hello world program.
2. Execute the following commands on the command line and explain what each command does.
  - 1) date
  - 2) date -d "today"
  - 3) date -d "yesterday"
  - 4) timedatectl
  - 5) date --set "15 Aug 2020"
  - 6) man date
  - 7) dpkg-reconfigure tzdata
  - 8) cal
  - 9) date +%D
  - 10) date +%d%m%y