Lab Tasks

By

Hajira Imran(44594)

Subject: Operating System

Date:9/29/2024

BSCS SEMESTER – 5

RIPHAH INTERNATIONAL UNIVERSITY

ISLAMABAD, PAKISTAN

Lab Tasks

Note: Include screenshots, required to illustrate your explanation for all Questions.

Q1: Explain the process of compiling a C program in Linux. What command is used to compile the program?

1. Sudo apt install.
2. Sudo apt install gcc.
3. Sudo apt install g++.
4. Sudo apt update.
5. Create a file for example nano tap.c
6. When GNU screen appears write C or C++ program.
7. Press ctrl X.
8. Press Y for yes. And then press Enter.
9. Write chmod 777 tap. cpp (for c++) and (for C tap.c)and give all commands to the file.
10. Write gcc tap.c -o tap
11. Write ./tap to see the output.

Q2: What is the purpose of the `-o` option in the `gcc` command? Provide an example. By using -o option, we name our program's output file.

```
[root@localhost ~]# gcc tap.c -o tap
```

Q3: What is the difference between `g++` and `gcc`? When would you use each?

1. gcc is used for compiling C programs.
2. g++ is used for compiling C++ programs.
3. g++ automatically handles the C++ codes. That's why we:
4. Use gcc for C programs.
5. Use g++ for C++ programs.

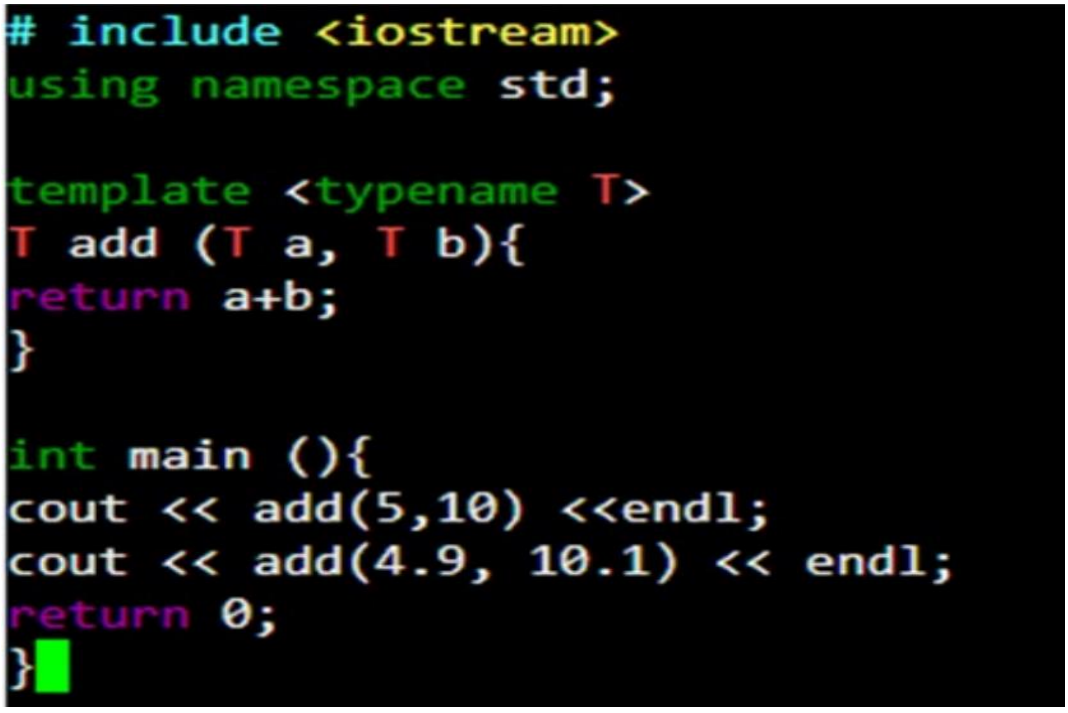
Q4: How do you compile and run a C++ program from the terminal? Provide the necessary commands.

- Chmod 777 play.cpp
- g++ play.cpp -o play
- ./play

```
[root@localhost ~]# chmod 777 play.cpp
[root@localhost ~]# g++ play.cpp -o play
./play
```

Q5: What are templates in C++ in Linux? Write a simple example of a function template.

Templates in C++ can create functions or classes that can work with any types of datatypes. So, we don't need to rewrite the same code for different types like int, float, etc. In simple, we can say it makes our code flexible.



```
# include <iostream>
using namespace std;

template <typename T>
T add (T a, T b){
return a+b;
}

int main (){
cout << add(5,10) <<endl;
cout << add(4.9, 10.1) << endl;
return 0;
}
```

Q6: Discuss the significance of file extensions in C programming. Why should source files be saved with `.c` or `.cpp` extensions?

In C programming file extensions .c is for C programs. And .cpp is for C++ programs.

Q7: What common errors can occur when compiling C programs, and how can they be resolved?

Syntax Error:

Foe Missing semicolons, unmatched brackets.

Solution:

Recheck the code properly and correct it.

Missing Libraries:

Like if I'm adding cin and cout but not enter iostream library

Solution:

Make sure we include all the required libraries.

Linker Error:

Functions are declared but not defined.

Solution:

Make sure all functions are defined.

Q8: Explain how you can manage permissions for an executable file in Linux. What command is used for this purpose?
We use following command

```
chmod +x myfile
```

Q9: What is tarball, and what advantages does it offer for distributing software on Linux? Discuss the limitations of using tarballs for software installation and management.

In the early days of LINUX, Linux apps were shared as tarballs, that are bundles of files. users have to unpack them to use the app, but it was hard to keep track of the version of files or where the files located in the sytem, making updates and removal tough. Also, if the app needed by other software, we had to install and update that manually.

Q10: Explain the purpose of the RPM package format and how it addresses the shortcomings of tarballs.

RPM, or Red Hat Package Manager, is used to install software on Red Hat Linux. It bundles everything needed for an app into one file with a `.rpm` extension. This package includes details like the version, a list of files, and any other software it needs to run. The `rpm` command helps install these packages, but it can struggle with handling dependencies.