Operating System

Lab 06 Tasks

Name: Dua Amir

Sap ID: 47849

Batch: BSCS-5th semester

Lab Instructor:

Kausar Nasreen Khattak

Q1.

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

Answer:

To compile a C program in LINUX in online terminal, the process is as follows:

- First, we write a code in .c file.
- Then we use gcc command to compile that program.

```
gcc dua.c -o duaa
```

By above command, a file created. To execute that file, we just write following command.

```
./duaa
```

Q2.

What is the purpose of the -o option in the gcc command? Provide an example.

Answer:

By using -o option, we name our program's output file.

```
gcc dua.c -o duaa
```

Q3.

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

Answer:

- gcc is used for compiling C code.
- g++ is used for compiling C++ code.

g++ automatically handles the C++ codes. That's why we:

- Use gcc for C programs.
- Use g++ for C++ programs.

Q4.

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

Answer:

Compile a program:

```
g++ dua.cpp -o duaa
```

Run a program:

```
./duaa
```

Q5.

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

Answer:

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

```
  GNU nano 5.3                          program.cpp
# 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?

Answer:

File extensions tell us the format of our file, that we're working with which language compiler.

- .c is for C programs.
- .cpp is for C++ programs.

Q7.

What are the common errors that can occur when compiling C programs, and how can they be resolved?

Answer:

Common errors occurring while compiling C programs are:

i. **Syntax Errors:**
   Like missing semicolons, or unmatched brackets.
   Resolve: check and correct our code.
ii. **Linker Errors:**
   Like function is declared but not defined.
   Resolve: make sure all functions are defined.
iii. **Missing Libraries:**
   Like if I'm using math functions but not include maths library.
   Resolve: make sure we're including all required libraries.

Q8.

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

Answer:

We use following command to make a file executable:

```
chmod +x myfile
```

Q9.

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

Answer:

In the early days of LINUX, Linux apps were shared as tarballs, which are bundles of files. We have to unpack them to use the app, but it was hard to keep track of the version or where the files, making updates and removal tough. Also, if the app needed 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.

Answer:

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 struggles with handling dependencies.
