RIPHAH INTERNATIONAL UNIVERSITY, ISLAMABAD

Lab # 6

Bachelors of Computer Science – 6th Semester

Subject: Operating System

Submitted to: Ms. Kausar

Submitted by: Ayesha Noor _ 41379

Date of Submission: 29- Sep -2024

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

Solution: To compile a C program in Linux, you typically use the GCC (GNU Compiler Collection) compiler. **The basic command is: gcc filename.c**

Q2: What is the purpose of the `-o` option in the `gcc` command? Provide an example.

Solution: The **-o** option in the gcc command specifies the name of the output file. For example, to compile **filename.c** and name the output executable **my_program**,

you would use: **gcc filename.c -o my_program**

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

Solution:

- **gcc**: This is the GNU C Compiler, primarily used for compiling C programs.
- **g++**: This is the GNU C++ Compiler, used for compiling C++ programs. It automatically links C++ libraries that are necessary for the C++ program.

Use gcc for C programs and g++ for C++ programs due to the differences in how they handle linking and libraries.

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

Solution:

- To compile a C++ program from the terminal, you use the `g++` command:

g++ filename.cpp -o my_program

- To run the compiled program, use:

./my_program

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

Solution:

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

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

```
int main() {
   cout << add<int>(3, 4) << endl; // Outputs 7
   cout << add<double>(3.5, 2.5) << endl; // Outputs 6.0
   return 0;
}
```

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

Solution: File extensions like `.c` for C source files and `.cpp` for C++ source files help the compiler and development tools recognize the file type and apply the appropriate compilation rules. Using these conventions aids in maintaining organization and clarity within projects, especially in larger codebases.

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

Solution: Common compilation errors include:

- **Syntax errors**: Mistakes in code syntax that prevent compilation (e.g., missing semicolons).
- **Undefined references**: Occurs when a function or variable is declared but not defined.
- **Type mismatches**: Using incompatible data types in operations or function calls.

Resolution: Carefully read the error messages provided by the compiler, check the documentation, and revise the code accordingly.

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

Solution: To manage permissions for an executable file in Linux, you can use the chmod command. For example, to make a file executable for the All, you can use:

chmod 777 my_program

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.

Solution: A **tarball** is a compressed archive file created using the tar command, typically with a .tar.gz or .tgz extension. Advantages of tarballs include:

- Easy distribution of multiple files and directories.
- Reduces disk space and transfer time due to compression.

Limitations: Tarballs don't handle dependencies or installation processes automatically, making them less user-friendly for software installation compared to package managers.

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

Solution: The **RPM (Red Hat Package Manager)** package format is used for managing software installation in Linux distributions like Red Hat and Fedora. It addresses the shortcomings of tarballs by providing:

- Dependency management: Automatically installs required libraries and files.
- Version control: Keeps track of installed software versions.
- Easy uninstallation: Provides simple commands for removing software.

Using RPM packages enhances the user experience for software installation and management compared to tarballs.