Useful Linux commands, creating, editing, compiling and running a C program

Here, we are concerned with creating a C program, compiling it, and running it. A C program file is called a source file. We compile a C program using **GCC**. A collection of compilers for many languages is the GCC, which stands for GNU Compiler Collection. This collection of compilers contains the compiler for C, C++, Fortran, and many more. It is a component of a GNU project developed by the Free Software Foundation. Make sure that GCC is installed in your system. To determine the installed version of GCC on your system, open a *terminal* (for Linux and macOS) or *command prompt* (for windows) and execute the following command:

```
$ gcc  --version↩
```

where ↩ denotes pressing *enter/return* key and $ denotes the user's current directory/folder. The appearance of $ depends on the operating system. The above command displays information about the installed GCC version and other details. If GCC is not installed in your system, you can follow the instructions on how to install gcc available on the internet. To create a new file or open an existing file, we use graphical text editor available in our system.

- To create a new file named "hello.c" or open an existing file named "hello.c" to edit in such a text editor in *windows* system, execute the following command in command prompt:

  ```
  $ notepad  hello.c↩
  ```

  This will create a new file "hello.c" or open an existing file "hello.c" in the notepad editor.

- If "hello.c" exists, then execute the following command in a terminal of *macOS* to edit:

  ```
  $ open -e  hello.c↩
  ```

  This will open an existing file "hello.c" in the TextEdit editor.

  On the other hand, if the file "hello.c" does not exist in the current directory, execute the following two commands in a terminal of *macOS*:

  ```
  $ touch  hello.c↩
  $ open -e  hello.c↩
  ```

  Here, the first command creates the file "hello.c".

- To create a new file named "hello.c" or open an existing file named "hello.c" to edit in such a text editor in *Linux* system, execute the following command in a terminal:

  ```
  $ gedit  hello.c  &↩
  ```

  Here, the symbol & allows you to run the gedit application in the background, freeing up your terminal for other commands.

Let us write a simple C program by typing the following in the newly created file "hello.c". We will discuss this program in detail later. Since C is *case-sensitive* language, be careful when typing a C program. All C functions (described later) are in lowercase lettering.

```
/* Simple C Program to print Hello Mars! */

#include <stdio.h>
int main(void)
{
    printf("Hello Mars!\n");
    return 0;
}
```

Listing 1: A simple C program

Once the program is typed, save it and then compile it. To compile it, execute the following command in the *terminal* (for Linux and macOS) or *command prompt* (for windows):

```
$ gcc  hello.c↩
```

If you have typed the program correctly, the compilation proceeds silently and the $ prompt appears in the next line. In this case, a default executable file will be created. Otherwise, you may get warning(s)/error(s). In the case of only warning(s), the default executable file will also be created. However, investigate the source(s) of warning, as they may indicate a serious problem in your code. On the other hand, the default executable file will not be created in the case of error(s). You need to remove the error(s) by editing the source file and compile again to generate the executable file. The default executable file will be named "a.out" (for Linux and macOS) or "a.exe" (for windows). To run the default executable file on Linux and macOS, execute the following command in the terminal:

```
$ ./a.out ↩
```

You will observe the following output in the terminal:

```
Hello Mars!
```

To run the default executable file on windows, execute the following command in the command prompt:

```
$ a.exe ↩
```

You will observe the same output that appears for Linux and macOS. To compile and run, make sure that you are in the same directory/folder where your source file "hello.c" was created.

Note that you can choose the name of the executable file instead of the default one. To get the name of the executable file as "xyz", you execute the following command:

```
$ gcc  hello.c  -o  xyz↩
```

However, we usually choose the name of the executable file such that it has some similarity with the source file. Thus, we may execute the following command:

```
$ gcc  hello.c  -o  hello↩
```

The new executable file "hello" can now be run in macOS and Linux systems by executing the following command in the terminal:

```
$ ./hello↩
```

It is better to compile without the "-o" option so that the executable file is the default one. Otherwise, you may create many executable files (for different source files) which are usually large in size. They may take lot of space in your system unless you remove them. However, you compile with the "-o" option if you want to run the corresponding executable file many times.

Return to Lecture Topics