

INTRODUCTION

Your first activity is to write a simple program *test.c*. But first, you need to learn how to edit programs quickly and efficiently. That means using the keyboard exclusively, forgoing the mouse. So if you find yourself reaching for the mouse, stop!

An editor is a program that allows you to enter text into a file. Unlike word processors, such as Microsoft Word, editors are optimized for quickly and efficiently entering the text that makes up a computer program. Such program text is known as *source code*.

In this class, you will use the editor named *vim* to efficiently place C source code into a file. C is a programming language and it is used to write computer programs. Computer programs that are written using text that C understands are known as C programs.

One of the hardest things you have to do is to keep track of three kinds of commands when writing and running computer programs. Those commands are:

- system commands
- editor commands
- program commands

System commands are the commands you enter into the terminal window to start things off. Editor commands are the commands used to interact with the editor when placing the text of a computer program into a file. Program commands are the actual lines of text in the computer program. Complicated, no?

Don't worry. As with all things associated with writing computer programs, things become clearer and easier with practice. What you need to do now is, whenever you are instructed to issue a command, decide exactly what kind of command it is, system, editor, or program.

LEARNING *vim*

We'll start by learning the editor named *vim*. The easiest way to learn *vim* is to run the program *vimtutor*. To do so, boot a computer and open up a terminal window. Then, in the terminal, run the command:

```
vimtutor
```

Note that the command *vimtutor* is a system command. It instructs your computer to run the program named *vimtutor*. The *vimtutor* program is a computer program that teaches you how to use the *vim* editor. You must learn the *vim* editor as your first quiz will question you on how to efficiently enter source code into a file using *vim*.

Vim HIGH POINTS

You will have noticed that *vim*, unlike say Microsoft Word, has two modes, insert and edit. The reason is that most programmers spend a vast majority of their time editing old code as compared to inserting new

code. By separating inputting and editing, the edit commands can be made very efficient, in terms of the keystrokes needed to accomplish a task. Consider deleting a line in *vim* versus Word. In *vim*, a simple `dd`, anywhere on the line, suffices. In Word, you must select a line with the mouse (not always easy to do), the switch to the keyboard and type `control-x`. There are keyboard shortcuts for Word that speed up selecting a line, but it still takes considerable effort to delete a line in Word.

Remember these general rules:

- *c* means change lines and words
- *d* means delete lines and words
- *x* means delete character
- *r* means replace character
- *h* means go left (it is the leftmost of keys *h*, *j*, *k*, and *l*)
- *l* means go right (it is the rightmost of keys *h*, *j*, *k*, and *l*)
- *?* means search upwards (it is above */*)
- *.* repeats your last edit command
- *i*, *I*, *a*, *A*, *o*, *O* are insert commands
- the escape key gets you out of insert mode

Note that the arrow keys and keys like Page-Up also work as expected.

On the *LMS*, there are links to other *vim* tutorials and a *vim* command cheat sheet.

MODIFYING AN EXISTING PROGRAM

Here is an example C program:

```
#include<stdio.h>
int main()
{
    print("hello, world!");
    return 0;
}
```

You can tell, using simple logic, that this program when compiled into test executable and, when run, is likely going to say:

```
hello, world!
```

Your task is to enter this program into a file named *hello.c* and test it using the following gcc command:

```
gcc hello.c -o hello
```

Run your executable as shown below:

```
./hello
```

Next, you will modify it so that it prints the hello world ten times when you run the program.

SUBMISSION

To submit your activity, make sure you are in the */home/username/* directory. Run the command:

```
submit clab mr vim xxxx@iiitb.org
```

Replace *xxxx* with your iiitb email address.

You will be asked for a guest password. Your instructor will give you the correct password in class. Note that when you type your password, it will look as if nothing is happening. Don't worry, something is. Press the Enter key when you are finished typing in the password.