

## Homework Assignment

---

### HW1: Getting Started in UNIX/LINUX and C

**Assigned:** 29 May

**Due:** 4 June

The primary objective of this assignment is to get you familiar with all of the basic tools that you will need as quickly as possible. When you have finished this assignment you will have used the UNIX/LINUX commands **ls**, **cd**, **mkdir**, **cp**, **man**, **gcc**, and **nano**, **emacs** or **vi**. You will also have written and executed your first C program.

For this assignment, you can read: K&R 1.1 -> 1.9, Glass & Ables Ch. 3 (this will help you with some of the work below), and read through the Basic UNIX Guide, concentrating on the commands and how to use them. You probably want to try the commands in UNIX as you read about them. Sit at a terminal to do this. Directions for turning in this homework and all future ones are given at the end of this assignment.

#### 1. Get a UNIX account and a directory for this course

Apply for this course account remotely or at a terminal in the UNIX/PC computer room (S-3-157) on the 3<sup>rd</sup> floor of the Science Building. Basically, you will login in with the account name, apply, with a password apply and follow the instructions.

After your application for a UNIX account has been processed, you should log in using your home PC and look around. After downloading the SSH package and installing it on your PC, run the Secure Shell Client and log on to server: **users.cs.umb.edu** (or **users2.cs.umb.edu** or **users3.cs.umb.edu**) using your username and password. After logging on, you should see the command prompt:

```
user_name@VM71:~$
```

Try the **ls** command to see the contents of your home directory and the **pwd** command to see its full name. Try

```
user_name@VM71:~$ ls -al
```

to see even more of the files you have. To understand the output, you may want to try

```
user_name@VM71:~$ man ls
```

The output may be pretty overwhelming. It is the same sort of thing you will see in Glass and Ables' LINUX Text, Chapter 3.

One of the things you will find in your home directory is a cs240 subdirectory. When you learn about links you will understand that the cs240 subdirectory is a link to something like:

```
/courses/cs240/sum18/ramin/your_login_name
```

Use your cs240 subdirectory for all your work in this class. Its protections have been set so I can read what's in it but no one else can. (The default UNIX file protection is to allow anyone to read anything.)

Use the **mkdir** command to create a subdirectory named "hw1" in your cs240 directory. I will refer to that directory as your project directory. Change to the hw1 directory using the **cd** command. Use that directory for all your work on this assignment.

```
user_name@VM71:~$ mkdir hw1
user_name@VM71:~$ cd hw1
```

## Homework Assignment

---

My project directory for this assignment is `/courses/cs240/sum18/ramin/GROUP/hw1`. Any files needed for a homework assignment are in my project directory unless otherwise specified. Use the command:

```
user_name@VM71:~$ ln -s /courses/cs240/sum18/ramin/GROUP/hw1 cs240_hw1
```

to set up a logical link `cs240_hw1` in your `hw1` directory to link to my `hw1` project directory. From this point on, if you want to refer to my `hw1` project directory, you can use the name `cs240_hw1`. Copy the files from my directory to your `hw1` using `cp`.

Space

```
user_name@VM71:~$ cp -r cs240_hw1/ .
```

### 2. Learn to use a UNIX/LINUX editor.

The standard UNIX/LINUX editor is `vi`. `vi` is described in your Glass UNIX book and in other places, such as <http://www.cs.colostate.edu/helpdocs/vi.html>.

There some alternatives: the emacs editor or the nano editor. emacs also has a good on-line tutorial. To learn how to use emacs, type

```
user_name@VM71:~$ emacs
```

To learn how to use nano, type

```
user_name@VM71:~$ nano
```

And then type CTRL + G

You will see a screen with the GNU copyright message. The last line on that screen tells you how to invoke the tutorial. This information and lots of other useful stuff is available on line. You may find this command helpful

```
user_name@VM71:~$ help
```

### 3. Show line numbers in vi sessions or nano

In order for `vi` to display line numbers by default, create a `.exrc` file in your home directory that has the following line:

```
set nu
```

Then save the `.exrc` file and exit. Logout and log back in. Your new `vi` session should show line numbers.

For nano it's simple. You should just type ALT+C in nano.

### 4. Write and run a C program

```
"Hello, world!"
```

(ending with a newline), on your terminal. Use `vi` or emacs or nano to create a source code file named `hello.c` in your `hw1` subdirectory. See the texts and the lecture notes for details on coding this program.

To compile your program, type:

## Homework Assignment

---

```
user_name@VM71:~$ gcc -m32 -o hello hello.c
```

When your program compiles successfully, you can list the contents of the project subdirectory (**ls** command). You will find a new file named "hello". That file is the executable program. To run it from the current directory, at the UNIX prompt type:

```
user_name@VM71:~$ ./hello
```

### 5. Turn in homework

These directions will hold for all homework you turn in for cs240. You are expected to turn in homework as hardcopy at the beginning of class when due. No late homework will be accepted. (Accommodations will be made for this first assignment for people who sign up for the course late.)

The procedure to follow in making up your homework solutions is:

- (1) use the **script** command to start a typescript.
- (2) use **ls -l** to list files in your directory and write date/time
- (3) use **cat** to print out all solution files (usually programs)
- (4) use **gcc -m32** to compile all programs
- (5) execute all programs, demonstrating that they work with test input
- (6) use the **exit** command to end **typescript** file

Step (4) is the most challenging part. You are expected to demonstrate that the program you've written works, and make up test data of your own if none is given in the assignment. The first few assignments will have tests specified.