

CS/240/Working Environment

1 Unix/Linux

Linux is a UNIX-like operating system, you will be working on top of it for the entire semester in this course. We provide a brief tutorial below showing only the most basic commands. To learn more about working on a UNIX system you can find several more comprehensive tutorials. Two good ones are:

- UNIX Tutorial for Beginners (<http://www.ee.surrey.ac.uk/Teaching/Unix/>)
- Intro to UNIX (<http://www.cs.bu.edu/teaching/unix/intro/>)

1.1 Introduction

This is a very basic tutorial aiming at the students new to the UNIX command-line environment. This might be appropriate for you if you have never (or rarely) worked under command-line environment of any variations of UNIX, Solaris, Linux, or Mac OS X.

1.2 Shell

At the present time, most of UNIX-like systems you can get access to are equipped with various graphic user interfaces, like the X-Window system. To work under the command-line environment, you need to first start up a terminal. On our lab machine, you can open it following the menu Applications → Accessories → Terminal. You should see something that looks like:

```
<hostname>:~$
```

It is called “command prompt” or “prompt” for short. The prompt waits for you to type a command. The commands you type are read in and interpreted by a program called a shell. The shell lets you communicate with the operating system. Within the shell you can tell the operating system to run various programs, feed the result of one program into another, look at what is going on in the operating system, and change the system settings. A variety of shells can be available in the system, our current default shell is “bash”.

For the rest of the tutorial, the commands you need to type are preceded by “\$”. The output is listed on the next line below (always without a “\$”). Instructions to type the command “cmd” into the prompt, press the Enter key, and get the output “output” would look like:

```
$ cmd
output
```

Remember, exclude the dollar sign in your typing; it indicates the existence of the prompt.

1.3 Basic Commands

The most common UNIX command is “ls”. Type:

```
$ ls
```

The command “ls” with no arguments lists the contents of the current directory. If you want to see the details of the files, type:

```
$ ls -l
```

This lists the files, who has permission to read, write, or execute each file, who owns the file, the time and date the file was last written and the file name.

The command “pwd” prints the directory path of the current directory. This current directory is the directory where the shell is currently in. Type:

```
$ pwd
/homes/<your user id>
```

The directory path printed is a list of the directories necessary to reach the current directory from the root directory. You can create a new directory, for example **cs240**, under current directory by:

```
$ mkdir cs240
```

To change the current directory to the **cs240** directory, type:

```
$ cd cs240
```

By **ls**, You will notice there is nothing in there. But when typing:

```
$ ls -al
```

You will see two directories: “.” and “..”. One dot represents the current directory and “..” represents the parent directory. You can now go back to the parent directory by:

```
$ cd ..
```

The home directory is the directory where all your personal files are stored and is where the current directory points to when you login. To change the current directory to your home directory type “cd” with no arguments. Try:

```
$ cd
$ pwd
$ ls
```

Sometimes you would like to go back to the previous commands. To do so, type the <up arrow> (the key that has the arrow pointing upwards):

```
$ <up arrow>
```

Continuing to press the up arrow will continue to step back through the history of commands you have typed. To execute the current command displayed type <ENTER>. You may also type the <down arrow> to go back to the more recent commands.

When typing a command, you don’t always have to type all the characters. You may use <tab> to complete it. For example, type:

```
$ ls /h<tab>
```

This will complete to “ls /home/”. If the prefix matches several commands, files, or directories, you may see a list of those matched by typing <tab> twice:

```
$ ls /b<tab><tab>
```

This will show all the files and directories that start with “/b”. Also you can refer to a list of files or directories by using the character “*”. Using the “*” character in a file or directory name is called a wildcard. It represents one or more characters. For example:

```
$ ls /usr/lib/*.a
```

Lists all the files that end in “.a” in the directory /usr/lib. Here is a summary of the commands you have learned and some more:

Table 1: Summary of UNIX commands

ls	List the contents of the current directory
ls -al	List all the contents of the current directory in detail
pwd	Print the path of the current directory
mkdir new-directory	Create a new directory
cd directory-path	Change to a directory
cd	Change to home directory
cp old-file new-file	Copy the file old-file into a file called new-file
mv old-file new-file	Rename a file from old-file to new-file
rm file	Remove a file
rm -r directory	Remove a directory
echo "message"	Output the string "message"
cat file	Display the contents of the file
<up-arrow> / <down-arrow>	Get previous / next command
<tab>	Command completion

1.4 File system

The UNIX file system is a hierarchy of directories that start with the root directory “/” You may list the contents of this directory typing:

```
$ ls /
```

There are some directories you can find under root directory:

Table 2: Directories under root

/boot	the Linux kernel and boot loader files
/etc	the configuration files
/bin, /usr/bin	most of the programs for the system
/sbin, /usr/sbin	the programs for system administration
/usr	a variety of things that support user applications
/var	the files that change as the system is running
/lib	the shared libraries
/home	users' personal work
/tmp	temporary files
/dev	the device files of the machine

1.5 Home Directory

Every student with an account has a home directory. The home directory is a directory reserved for you. It is where you will store your programs and homework. When the special character “~” is at the beginning of an argument of a command, it will mean your home directory. Try:

```
$ ls ~ - Lists your home directory
$ ls ~/cs240 - Lists the cs240 directory under your home
$ ls ~cs240 - Lists the user cs240's home directory
```

1.6 Wildcards

When the special character “*” appears in the argument of an UNIX command, it will be expanded to all the files and directories that match that argument. For example, type:

```
$ cd /bin - Go to /bin directory
$ ls - Lists all the files in /bin
$ ls m* - Lists the commands that start with m
$ ls *ma* - Lists all the files that contain ma
$ ls *ma* *ta - Lists all the files that contain ‘ma’ or end with ‘ta’
```

In UNIX, any file or directory started with a “.” is hidden. Wildcards do not match hidden items. If you want to see those them, you will have to either use “ls -al” or match those files or directories explicitly. Try:

```
$ cd - Goes to home directory
$ ls - Lists home directory. Hidden item do not show up
$ ls -al - Lists all items, including the hidden ones and also prints timestamp
$ ls .* - Lists all items that start with ‘.’
$ echo .* - Prints to the screen the names of the items that match .*
```

1.7 Other Useful Commands

- **date** - prints current system date and time
- **who am i** - prints the current user’s details
- **w** - display information about currently logged-in users
- **man** - find and display reference manual pages eg: **man ls** shows manual page for the **ls** command. Look for the man pages for **getchar** and **putchar**.
- **clear** - clear the terminal screen
- **history** - show command history list

1.8 I/O Redirection

Usually input for a program comes from keyboard while the output is printed on the screen. This standard methods for input and output are called standard input (stdin) and standard output (stdout). But if you want to redirect the input from a file or redirect the output to a file, you can use the I/O redirection operators.

To redirect standard output to a file instead of the screen, we use “>” operator:

```
$ echo hello
hello
$ echo hello > result
$ cat result
hello
```

At the second execution of echo, “hello” is redirected to the file “result”. In this case, the content of “result” will be overwritten if it already exists. If instead we want to append the output of echo to the result file, we should use “>>” operator:

```
$ echo world >> result
$ cat result
hello
world
```

To redirect standard input we use “<” operator:

```
$ cat < result
hello
world
```

You can combine input redirection with output redirection, but be careful not to use the same filename in both places.

```
$ cat < result > result2
```

2 GCC

GCC (the GNU Compiler Collection) is a collection of compiler front end for a variety of languages including C, C++, Java, etc. In this class, we will only use the C compiler, also shortened as gcc (the GNU C Compiler). Following shows the command to compile a list of source files:

```
gcc -o <binary_name> <source_file_list>
```

For example, the following command will compile two source files under current directory, and create an executable program named “prog”:

```
gcc -o prog source1.c source2.c
```

To run the program “prog”, type:

```
./prog
```

There are many C standards co-existing. The gcc default is “gnu89”. However, “c99” is more modern, supporting neat syntax. The C code on the lecture slides and in the solutions are written against “c99” standard. To enable the support when compiling using gcc, add `-std=c99` option:

```
gcc -o prog -std=c99 source1.c source2.c
```

For more information, visit:

- GCC official website: <http://gcc.gnu.org/>
- An Introduction to GCC: <http://www.network-theory.co.uk/docs/gccintro/>

3 VMWare

You are required to attend the lab sessions you signed for and encouraged to work in the lab during its free hours. The lab provides a programming environment that is guaranteed to work. However, in case you want to work on a personal computers, we provide a lightweight VMware Linux virtual machine. It runs on the free VMware Player.

- VMware Player downloaded from <http://www.vmware.com/products/player/>
- VM image downloaded from http://www.cs.purdue.edu/homes/cs240/CS240_image.zip

To boot the virtual machine, unzip `CS240_image.zip`, locate the file `Ubuntu12.04.vmx`, and double click it.

- default user: cs240; password: cs240
- root password: cs240

Tools you need:

- Gedit: simple text editor;
- Emacs: advanced text editor;
- LXTerminal: the terminal;
- Chromium: the web browser.

Dropbox is installed for you to synchronize the files on the virtual machine with other machines. If you don't have an account, you can apply for one, which gives you 2GB space for free.