Chapter 11
The Shell

We Shall be Covering ...

- The Linux shell
- Getting a shell
- Useful commands
- Input/output redirection
- Pipes
- Man and info pages

The Linux Shell

- A special program that accepts certain commands from the keyboard, executes the command, displays output on monitor
- Mainly text-based, understands limited number of commands
- Command-oriented text-based environment – Command Line Interface (CLI)

The CLI

```
$ cp userguide.sxw ~/userguide-bak.sxw
$ cd /etc
$ pwd
/etc
$ cat hosts
127.0.0.1   localhost.localdomain
192.168.1.1 garfield.mynet.net
192.168.1.9 fw.mynet.net
$
```

Getting a Shell

From the graphical desktop,

- Control-Alt-F1
  - F1 to F6 usually available
- Main Menu --> System Tools --> Terminal

Some Useful Commands

- ls – list files in the current directory.
- cd – change working directory. If your current path is /home/username/Trash for instance, typing “cd” will bring you back to /home/username.
- mkdir – make a new directory
- rmdir – delete a directory (has to be empty)
- cp – invoked such as “cp currentFile newFile”, and is used to copy files.
- mv – invoked such as “mv currentLocation newLocation”. This is used to either move or rename files.

Some Commands

- rm – invoked such as “rm myFile”; it is used to delete files permanently.
- pwd – prints the working (current) directory.
- cat – concatenate files (can be used to join them together), and prints its output to standard output (the terminal screen). Used like: “cat myFile”.
- less – allows for file viewing in the shell, and is most useful for text files; invoked like “less myFile”.
- find – can be used to find files via the command line. E.g. “find . -name toc”, which looks at the current directory (defined by “.”) for any files with the name “toc”.

Some Commands

- locate – picks entries from a database, that is updated regularly; invoked via “locate myFile”. Its much quicker than find (since it only searches a database), but might not be as quick to update as find (the update of the database might happen once every day only).
- date – display the current date! This can also be used to set the date of the system (but administrator privileges are required).
- history – built-in shell command for the BASH environment that shows the last run commands.

Input/Output Redirection

- By default for most commands,
  - input comes from the standard input device (stdin), the keyboard
  - output goes to the standard output device (stdout), the monitor
- I/O redirection – change the stdin and stdout, usually to a file
- E.g. the command “ls -l /usr/bin” will result in a 2133-line display! Redirect output to file

```
ls -l /usr/bin > dirbin
```

- To append to the file use “>>” instead of “>”

Input/Output Redirection

- The command “wc -l” can read from its input and print out on its output the number of lines

```
$ wc -l < dirbin
  2133
```

- Use both input and output redirection to files

```
$ wc -l < dirbin > dirbinlines
$ cat dirbinlines
  2133
```

Pipes

- A command usually perform one task (well!)
- Pipe - useful way to string together several commands to perform a bigger task
- The output of a command becomes the input to another command

```
command | command | command | command
```

Pipes

Example:

- To convert to uppercase the text in a file. Make use of the command “tr”,

```
$ cat /etc/hosts | tr [a-z] [A-Z]
```

- Save the result in a new file called hostcap

```
$ cat /etc/hosts | tr [a-z] [A-Z] > hostcap
```

Help

- Possible to create new command; a matter of creating an executable program using some programming language or script
- Very many commands available to the shell even without the user created commands
- To get help on how to use the commands installed the **man** and **info** commands may be used

Man

- There are **man** (manual) pages, for commands that reside on the system.

  Invoked by:

  ```
  man command-name
  ```

- E.g.

  ```
  $ man ls
  $ man cat
  $ man tr
  $ man info
  ```

- To view the manual pages for “man” itself:

  ```
  $ man man
  ```

Info

- The **info** help system provides more information. Available only for some commands. Invoked by:

```
info command-name
```

- E.g.

```
$ info ls
$ info cat
$ info tr
$ info man
```

- To view the info pages for “info” itself:

```
$ info info
```

End of Chapter 11