Using the Unix Operating System

The Unix operating system presents you with two different user interfaces: a graphical user interface that looks similar to the Windows or Apple Macintosh interface, and the non-graphical user interface, in which you type commands and are given typed responses to the commands. This introduction is intended to give you a summary of some of the major points about using the non-graphical user interface of the Unix machine.

The Unix operating system includes several hundred utility programs, which are often referred-to as *commands*. These commands are issued when you type them. They may have zero or more *parameters* to each command. A parameter is a character string that you type after the command that will modify the actions of the command. For example, as the following command is written, it has no (zero) parameters:

```
cd
```

Another version of the same command takes a single parameter:

```
cd   /usr
```

There are some conventions that are followed in Unix, and this document follows some conventions, as well. Conventions include:

1. Unix is case sensitive. This means that upper case and lower case characters (e.g., 'A' vs. 'a') are different on Unix. Therefore, you must always type a command as it is written. For example, you cannot type "**`CD`**" when the command is "**`cd`**".

2. Unix commands often produce no output. If a command is not normally one that is specifically used for displaying information to the screen, then "nothing happened" usually means that it worked.

3. When commands are written in this manual, it is implied that you must type the **`<Enter>`** key after the command.

4. If you see a command written with the "**`^`**" character, it means that you are to use the **`<Ctrl>`** key. For example, "**`^C`**" means that you simultaneously press the **`<Ctrl>`** key and the **`C`** key.

The tables below present commands and their descriptions, as well as examples of each command. The commands are organized according to several categories. In the example portion, what is written is what you would type at the "%" prompt on Unix.

For example, to read the manual page about the "ls" command, you type:

```
man ls
```

when the computer types the "**%**" prompt to you. It would look like this on your screen:

```
% man ls
```

Finding Help	**Examples of Usage**
man - online manual pages	`man ls`
whatis - Displays what a command does.	`whatis who`
apropos - Displays a list of commands that are related to the subject given by its parameter.	`apropos password`
Operations on Files	

ls - Displays a list of files in the current directory.	`ls`
	`ls -l`
	`ls -l test1.dat`

cp - Copies a file from one location to another.	`cp test1.dat test2.dat`
rm - Removes a file.	`rm test1.dat`
mv - Move a file from one location to another or rename a file.	`mv test1.dat Folder1` `mv test1.dat test1b.dat`
cat - Type the contents of a file to the screen all at once (does not page it to the screen)	`cat test1.dat`
more - Type the contents of a file to the screen a single page at a time.	`more test1.dat`
file - Determine what kind of data exists in a file.	`file test1.dat`

Operations on Directories	**Examples of Usage**
rmdir - remove an empty directory.	`rmdir Folder1`
rm - used to remove a directory and all of its contents (if the directory is not empty).	`rm -r Folder1`
mkdir - used to create a directory.	`mkdir Folder1`

Security

chmod - used to assign permission values to files.

```
chmod 644 test1.txt
chmod u+r test1.txt
chmod g-x test1.txt
```

Protection is expressed as a 3-digit number **XYZ** where **X = user**, **Y = group**, and **Z = other**.

Each digit is the sum of desired protection values, where **read = 4**, **write = 2**, and **execute = 1**.

ls - used to examine file protection.

```
ls -l test1.txt
```

Working Directory

cd - used to change the current directory

```
cd
cd /usr
cd ~
cd ..
```

pwd - find out the current working directory.

```
pwd
```

Miscellaneous	**Examples of Usage**
ps - determine your current running processes or all running processes on the system.	**`ps`** **`ps -ef`**
who - determine who is on the system.	**`who`**
u, users - give a short list of users on the system.	**`u`** **`users`**
history - display a history of recent commands that you've typed.	**`history`**
alias - create an alias of a command to customize your working environment.	**`alias h history`** (allows you to type '**`h`**' instead of '**`history`**')

Networking Issues

There are a number of special files and directories that related to the setting-up and usage of networking facilities on Unix. This information is geared specifically for Linux, especially the Red Hat Linux that is installed on the machines in our networking laboratory. The following tables give a summary of some of the various directories and files that you may need to examine/modify/use in performing assignments. The tables are in no particular order.

Directory or File	**What it is Used For**
`/etc`	Contains most configuration files in most versions of Unix & linux.
`/etc/services`	File containing list of services available at various ports.
`/etc/hosts`	File that performs local FQDN to IP translations.
`/etc/resolv.conf`	File containing the IP addresses of DNS servers.

Directory or File	What it is Used For
/etc/host.conf	File that shows the order of DNS resolution.
/etc/exports	Shows the file systems this host is willing to serve over the network.
/etc/inittab	A file that describes how the init process sets up the system for various run levels.
/etc/xinetd.conf	The file containing the top-level configuration of xinetd, the Internet super-server daemon.
/etc/xinetd.d	Directory containing other xinetd-related configuration files.
/etc/hosts.allow	List of hosts "allowed" to access services by tcpd.
/etc/hosts.deny	List of hosts "denied" services by tcpd.
/etc/rc.d	A directory containing a set of directories related to system configuration at various run levels.
/etc/rc.d/init.d or **/etc/init.d**	Directory containing startup scripts.
/etc/init.d/network	Probably the most important startup script, for your assignments. There are others in the same directory that you may want to examine and use.
/etc/sysconfig	A directory containing various specific system configurations.
/etc/sysconfig/network	Stores system-wide network configuration information.
/etc/sysconfig/network-scripts	A directory that stores some scripts specific to network configuration. An example of this is /etc/sysconfig/network-scripts/ifcfg-eth0, which contains the configuration information for Ethernet interface 0.

There are many commands specifically related to networking. You'll find that some of these commands are unique to Unix, and some of them have Windows/DOS counterparts. The following table gives a summary of most of these commands and is also in no particular order.

Command	**What it is Used For**
chkconfig	Shows the current run-level information for system services.
ifconfig	Shows interface configuration information.
ifuser	Used to identify destinations routed to a particular network interface.
mtr	A network diagnostic tool that combines the functionality of **traceroute** and **ping**.
netstat	Shows various network statistics. Dr. Turner mostly uses it for looking at routing tables, but it does a lot more.
ping	A simple utility to send **ICMP ECHO_REQUEST** packets to other network hosts.
traceroute	Used to print the route packets take to a network host.
tracepath	Used to trace a path to a network host and discover MTU along the path. MTU means *maximum transmission unit*, a link-layer restriction on the max number of bytes of data in a single transmission.
iptables	The linux firewall command.
nslookup	The original command-line DNS lookup utility.
dig	The new command-line DNS lookup utility.
host	A simple command-line DNS lookup utility.
nmap	Network exploration tool and security scanner.