

UNIX Commands

1. The login prompt indicates that the terminal is available for someone to log in. This message also indicates that previous user has logged out. Suppose you have an account named ‘sita’ , enter this string at the prompt. Then press the [enter] key after the string:

login: sita [enter]

Password:

The system now requests you to enter the secret code that handed to you by administrator.

2. The system now shows the \$ as the prompt.

3.

\$ date

Displaying both date and time

4.

\$ cal 7 2008

Calendar command

5.

\$ who

Who are the people using the system with me ?

6.view processes

\$ps

7.

\$ ls

Listing files

8.

\$ ls > file1

Directing output to a file

9. cat file1

10. wc file1 (how many lines are there in file1)

11. ls | wc

Output of ls to input of wc

12. x = 5 ; echo \$x

13.

Kernel is the core of the operating system - a collection of routines mostly written in C. It is loaded into memory when the system is booted and communicates directly with the hardware. User programs (the applications) that need to access the hardware (like the hard disk or terminal) use the services of the kernel, which performs the job on the user's behalf. These programs access kernel through a set of functions called system calls.

14.

Two entities support the UNIX system - **file** and **process**

A **file** is an array of bytes that can contain virtually anything. It is also related to another file by being part of a single hierarchical structure.

The second entity is the **process**, which is the name given to a file when it is executed as a program.

15. A single user can also run multiple tasks concurrently; UNIX is a multitasking system.

16. The sequence of directories that the shell searches to look for a command is specified in its own **PATH** variable.

\$ echo \$PATH

/bin: /usr/bin: /usr/local/bin:.

17.

(wc file1; ls -l file1) > file2

18.

Browsing the manual pages online : **man**

\$ man wc

19.

\$ date +%m

\$ date +%h

\$ date +"%h %m"

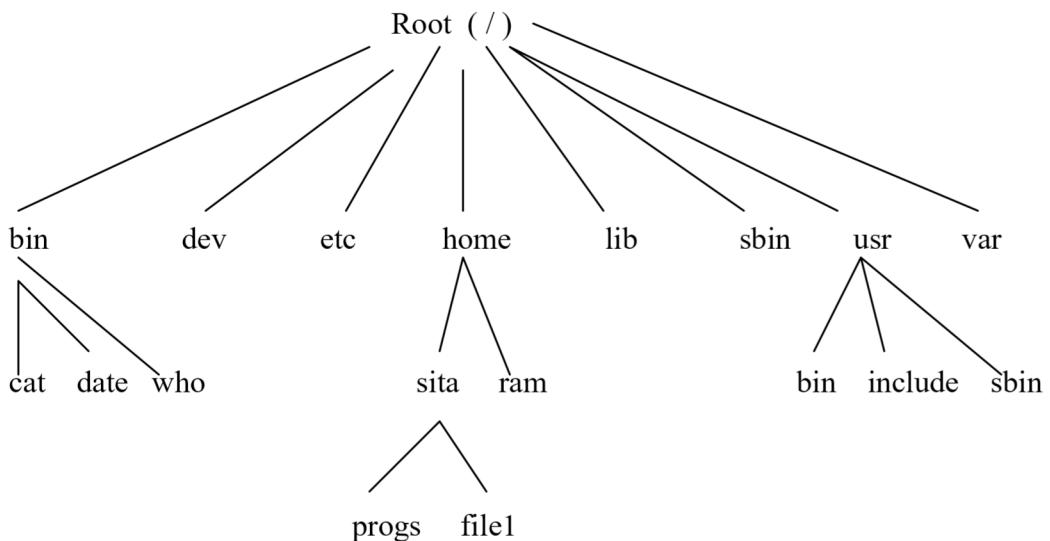
20.
Calculator : bc
\$ bc
12 + 5
17

21.
The mailer : mailx

22.
Change passwd
\$ passwd
Passwd: Changing password for sita
Enter login password:
New password:
Re-enter new password:
passwd(SYSTEM): passwd successfully changed for sita

23.
Knowing your machine's characteristics
\$ uname

24. The UNIX File System Tree



When you log on to the system. UNIX automatically places you in a directory called home directory. It is created by the system when a user account is opened. If you log in using the login name sita, you'll land up in a directory that could have pathname /home/sita .

```
$ echo $HOME  
/home/sita
```

```
$ pwd  
/home/sita  
Checking your current directory
```

```
$ cd progs  
Changing the current directory : cd
```

```
$ pwd  
/home/sita/progs
```

25.
Making directory : mkdir
\$ mkdir patch

26.
Using cat to create a file
\$ cat > foo
Hello world !
[cntrl-d]

```
$ cat foo
```

27.
Copying a file : cp

```
$ cp file1 file2
```

28. cp options
Interactive copying (-i), copying directory structures (-R)

29.
Deleting files : rm
Options

30.

Renaming files : mv

```
$ mv file1 file2
```

The mv command has two distinct functions:

- It renames a file (or directory)
- It moves a group of files to a different directory.

31.

Compressing and Uncompressing files : gzip and gunzip

32. The archival program : tar

- c create an archive
- x extract files from archive
- t display files in archive

33.

Compressing and archiving together : zip and unzip

34.

File permissions

```
$ ls -l dateval.sh  
-rwxr-xr-x    1      sita    research    890    Jan 31 23:17  dateval.sh
```

35.

Changing file permissions

```
$ ls -l file1  
-rw-r--r--    1      sita    research    890    Jan 31 23:17  file1
```

```
$ chmod u+x file1 ; ls -l file1  
-rwxr--r--    1      sita    research    890    Jan 31 23:17  file1
```

```
$ chmod ugo+x file1 ; ls -l file1  
-rwxr-xr-x    1      sita    research    890    Jan 31 23:17  file1
```

```
$ chmod go-r file1 ; ls -l file1  
-rwx--x--x    1      sita    research    890    Jan 31 23:17  file1
```

36.

Absolute Permissions

Read permissions	- 4 (Octal 100)
Write permissions	- 2 (Octal 010)
Execute permissions	- 1 (Octal 001)

37.

```
$chmod 666 file1 ; ls -l file1
-rw-rw-rw- 1      sita    research    890   Jan 31 23:17  file1
```

Ms. Shoma Chatterjee

USEFUL UNIX COMMANDS

cancel	cancel print requested with lp
cat file	Display the file
cat file1 file2 > files	Combine file1 and file2 into files
cat file1 >> file2	Append file1 at the end of file2
chgrp [options] newgroup files	<p>change the group of one or more files. New group is either a group ID number or a group name located in /etc/group.</p> <ul style="list-style-type: none"> -f quiet (do not print error messages) -h change the group on symbolic links (If not used, chgrp will act in the file referenced by the symbolic link) -R recursively (descending through the directory, including subdirectories and symbolic links.)
chmod [options] mode files	<p>Change the access mode of one or more files. Only the owner or privileged user may change its mode. Create mode by concatenating <i>who</i>, <i>opcode</i> and <i>permission</i>.</p> <p><i>who</i> (u- user)(g-group)(o-other)(a-all) <i>opcode</i> (+ add permission)(-remove permission)(= assign permission, and remove from the not assigned) <i>permission</i> (r or 4 – read)(w or 2 – write)(x or 1 – execute)</p> <p>Examples: chmod u+x file (add permission to the user to execute the file) chmod 751 file (chmod u=rwx, g=rx, o=x file) They are the same command.</p> <ul style="list-style-type: none"> -f quiet -R recursive
chown [options] newowner files	<p>Change the ownership of files to newowner (either a user ID or a login name from /etc/passwd).</p> <ul style="list-style-type: none"> -f quiet -h change the symbolic link, not the referenced file. -R recursive -c print information about the files changed.
clear	clear the terminal display
cmp file1 file2	Compare two files (0-identical)(1-different)(2-inaccessible)
comm [options]	Compare file1 and file2. Outputs three columns: lines unique to file1, lines

file1 file2	unique to file2 and lines common to both file. -1 suppress printing column 1 -13 suppress printing columns 1 and 3
cp [options] file1 file2 cp [options] files directory	Make a copy of file1 with the name file2 Copy the files into directory with same name files. -f force (remove existing files in the destination) -i interactively (prompt for confirmation) -p preserve the original file's permissions and ownership in the new file. -r, -R recursively (copy directories recursively) -L used with -R. if symbolic link points to a directory, recursively traverse the directory. -L used with -R. -s make symbolic link instead of copying.
df [[options]]	Report the number of free disk blocks and inodes on all mounted file systems. -h human readable -k kilobytes -m megabytes
diff [options] file1 file2	Reports the differences between file1 and file2. Prints file1 text flagged (<) and file2(>). -b ignore blank spaces. -e produce a script of commands to re-create file2 from file1, using ed editor. -D merge file1 and file2 into a single file containing conditional C preprocessors (#ifdef). Defining symbol and then compiling yields file2. Not defining symbol, yields file1.
du [options]	Print disk usage. Options like df command.
echo [options] [string]	Echo arguments to standard outputs. Often used for producing prompts from shell scripts. \a alert \n new line \r carriage return \t tab

egrep [options] [regexp] [files]	<p>Search files that match a regular expression [regexp]. Support metacharacters +, ?, (), etc. But must be enclosed with quotes.</p> <p>-n print the line matching. -v print lines that don't match [regexp].</p> <p>Example: egrep '(Victor Victoria)' file Seach Victor or Victoria in file.</p>
fgrep [options] [pattern] [files]	Fast grep. Seach files for lines that match a literal. Options are the same as for egrep.
find [options] pathname condition	<p>Find and locate all the files inside the pathname specified that matches some condition (-name, -type, -size, -user).</p> <p>Examples:</p> <pre>find /home -name chapter1 find /home /usr -name 'memo*' -user ann find / -mtime -2 (find files modified within the last two days) find / -size 10k (find files with 10 kB)</pre>
gcc	<p>GNU C compiler.</p> <p>-c creates linkable object file (.o) for each source file but do not call the linker. -o file specify output file as file. Default is a.out -Wall enable almost all possible warnings. -lm when using the math.h function.</p>
gftp	<p>Tranfer files to and from a remote network site hostname. Below is presented an example of gftp screen and login configuration necessary to access OMG computers /homes/username directory.</p> 
gunzip [gzip options] [files]	Restore the file.gz.

gzip [options] [files]	Reduce the size of one or more files and move to file.gz. -f force -q quite (suppress warnings) -r recursively walk the current directory tree and compress all files found. -v display the name and percentage of reduction.
head [options] [files]	Print the first few lines of files (default is 10 lines). -n 50 print the first 50 lines.
hostname [option]	Print name of current host system. -i display the IP address of the host.
id [options] [username]	display information about yourself or another user: used ID, group ID, etc.
kill [options] IDs	Terminate one or more processes IDs. You must own the process or be a privileged user. -l list the types of signals. -s 9 send signal 9 (complete kill).
locate [pattern]	Search files matching the pattern.
lp [options] [files]	Send files to a printer.
ls [options] [names]	List the files in the current directory. With names is specified, list the files inside the directory names or that match a file names. -a list all files, including the hidden . files. -d list only directory information, not its contents. Used with -l. -h human readable, using abbreviations for kilobyte, megabyte, etc. -k size in kilobytes -l long format list (includes permissions, owner, size, modification time, etc). -r list files in reverse order (by name or by time). -R recursively list subdirectories. -s list size in blocks (1block = 512 KB) -t list files according to modification time (newest first) -u list files according to the file access time.
make [options] [targets]	Update one or more targets according to dependency instructions in a description file (makefile) in the current directory.
man subject	Display information about a subject (name of a command).
mkdir [options]	Create one or more directories. You must have write permission in the parent

directories	directory in order to create a directory. -m set the access mode (ex. mkdir -m file1) (to read-execute only directory).																		
more [options] [file]	Display file content on a terminal. One screenful at a time. To change the next screenful use SPACEBAR and to scroll the lines use ENTER. h for help, / to search, or :n to go to the next file.																		
mount [options] [[device] directories]	The file structure on device is mounted on directory. If no device specified, mount looks for an entry in /etc/fstab to find what device is associated with the given directory. Directory which must exist and should be empty, become the root of the newly mounted file structure. If mount is invoked with no arguments, it displays the name of each mounted device, the directory on which it is mounted, its filesystem type, and any mount options associated with the device. -a mount all filesystems listed in /etc/fstab. Use -t to limit this to all filesystems of a particular type.																		
mv [options] sources targets	Move files and directories around on the system and to rename them. <table border="1" data-bbox="426 945 1432 1298"> <thead> <tr> <th>Source</th><th>Target</th><th>Result</th></tr> </thead> <tbody> <tr> <td>file</td><td>name</td><td>rename file as name</td></tr> <tr> <td>file</td><td>existing file</td><td>overwrite existing file with source file</td></tr> <tr> <td>directory</td><td>name</td><td>rename directory as name</td></tr> <tr> <td>directory</td><td>existing directory</td><td>move directory to be a subdirectory of existing directory</td></tr> <tr> <td>one or more files</td><td>existing directory</td><td>move files to directory</td></tr> </tbody> </table> - f force (force the move even if target file exists) - i interactive (prompt for a y (yes) response before overwriting an existing target. - u Do not remove a file or link if its modification date is the same as or newer than that of its replacement.	Source	Target	Result	file	name	rename file as name	file	existing file	overwrite existing file with source file	directory	name	rename directory as name	directory	existing directory	move directory to be a subdirectory of existing directory	one or more files	existing directory	move files to directory
Source	Target	Result																	
file	name	rename file as name																	
file	existing file	overwrite existing file with source file																	
directory	name	rename directory as name																	
directory	existing directory	move directory to be a subdirectory of existing directory																	
one or more files	existing directory	move files to directory																	
passwd [options] [user]	create or change a password associated with a user name. Only the owner or a privileged user may change a password.																		

ps [options]	Report on active processes. When needed to force one process to finish, <i>ps</i> shows the processes and <i>kill</i> allows to finish it. -u [user] dispaly processes running by user. If not specified the user, displays for the actual user. -r show processes currently running.
pwd	Printing Working Directory (pwd). Print the full pathname of the current directory
rm [options] files	Delete one or more files. To remove a file, you must have write permission on the directory that contains the file, but you need not have permission on the file itself. If you do not have write permission on the file, you are prompted (y or n) to overwrite. -f Force removing write-protected files without prompting. -i Promp for y or n to remove the file.
rmdir [options] files	Delete the named directories (the directory itself, not the contents). Directories are deleted from the parent directory and must be empty. -r remove directories and the files inside.
script [options] [file]	Create a record of your login session, storing in file everything that displays on the screen. -a append the script record to file.
sed [options] [files]	Stream editor. Edit one or more files without user interaction. -e apply the instructions to the files. -f apply the set of instructions from the editing script.
sort file	Sort the lines of file in alphabetical order.
ssh [options] hostname [command]	Securaly log a user into a remote system and run commands on that system. HOSTANME: user@hostname (ex. aluizio@peel.omg.unb.ca). -X enable X11 forwarding.

stty [options] [modes]	<p>Set terminal I/O options to the current device. As a privileged user, you can set or read settings from another device using the syntax: stty [options] [modes] < device</p> <p>-F device read or change settings of device terminal instead of the current terminal.</p> <p>control modes (n (set terminal baud rate to n (e.g. 19200)), input modes, output modes, etc.)</p>
su [options] [user] [shell-args] sudo (useful to install).	<p>Login as user. If no user specified, become superuser. Enter EOF to terminate.</p> <ul style="list-style-type: none"> - change the user (eg. su - user)
tail [options] file	<p>Print the last 10 lines of a file.</p> <p>-n 20 show the last 20 lines.</p> <p>Eg. grep '<title>' file tail -n 20 (show the last 20 lines containing instances of title).</p>
tar [options] [files]	<p>Copy files or restore files from tape (tape archive). If any files are directories, tar acts on the entire subtree. Very used (-cvf to archive and -xvf to extract).</p> <ul style="list-style-type: none"> -c create a new archive. -x extract files from archive. -v print function letter (x for extraction and a for archive) -f arch store files in or extract files from archive arch. -t list the contents like ls -l. <p>Examples:</p> <p>tar -cvf /dev/rmt/0 /bin /usr/bin (Create an archive of /bin and /usr/bin, show the command working and write the tape in /dev/rmt/0).</p> <p>tar -tvf /dev/rmt/0 (list the archives content in a format like ls -l).</p> <p>tar -xvf /dev/rmt/0 /bin (Extract the /bin directory)</p>
tty [options]	Print the device name of your terminal.

stty [options] [modes]	<p>Set terminal I/O options to the current device. As a privileged user, you can set or read settings from another device using the syntax: stty [options] [modes] < device</p> <p>-F device read or change settings of device terminal instead of the current terminal.</p> <p>control modes (n (set terminal baud rate to n (e.g. 19200)), input modes, output modes, etc.)</p>
umount [options] [arguments]	<p>Unmount a filesystem. Announces to the system that a removable file structure previously mounted on the specified directory is to be removed. A busy filesystem (with open files or with a directory that is some processes's currently directory) cannot be unmounted.</p> <p>-a ummount all filesystems specified in /etc/mtab. -f force the ummount.</p>
users [file]	Display the currently logged-in users as a spaced separated list.
which [options] [files]	<p>List which files are executed if the named commands are run as a command. <i>which</i> reads the user's .cshrc file (using the source built-in command), checking aliases and searching the path variable.</p> <p>-a print all matches, not just the first</p> <p>example: \$ which file ls /usr/bin/file :ls aliased to ls -sFC</p>
who [options] [file]	<p>With no options, list the names of users currently logged in to the system.</p> <p>am i print user name of the invoking user (similar to id) -u report terminal usage. Present idle time (HOURS:MINUTES, . (dot-less than one minute), or old (more than 24h)). -H print headings.</p>