APPENDIX: COMMONLY USED SHELL COMMANDS CATEGORIES

LINUX COMMANDS

Commands tell the operating system to perform set of operations.
The general syntax of the Linux commands is:
Command [-options] <argument1, argument2, …, argument >

We can divide Linux commands into following categories

- File Handling
- Text Processing
- System Administration
- Process Management
- Archival
- Network
- File Systems
- Text Editor Commands

LINUX COMMANDS

- Primary – man(manual) pages.
  - man <command>
    - shows all information w.r.t. online manual pages related to the command
  - Info <command>
    - Reads info documents about the command
  - <command> --help
    - shows the available options for that command
- Secondary – Books and Internet

FILE HANDLING (1/4)

ls:
list directory contents
Usage: ls [OPTION]... [FILE]…
eg. ls, ls -l, ls LHC_School

pwd:
prints the name of current working directories
Usage: pwd
eg. mkdir LHC_School

cd:
changes directories
Usage: cd [DIRECTORY]
eg. cd LHC_School

FILE HANDLING (2/4)

mkdir:
make directories
Usage: mkdir [OPTION] DIRECTORY...
eg. mkdir LHC_School

vim:
Vi Improved, a programmers text editor Usage: vim [OPTION] [file]...
eg. vim file1.txt

cp:
copy files and directories
Usage: cp [OPTION]... SOURCE / DEST
eg. cp sample.txt sample_copy.txt cp sample_copy.txt target_dir

FILE HANDLING (3/4)

mv:
move (rename) files
Usage: mv [OPTION]... SOURCE / DEST
eg. mv source.txt target_dir mv old.txt new.txt

rm:
remove files or directories Usage: rm [OPTION]… [file]...
eg. rm file1.txt ,
rm -rf some_dir

FILE HANDLING (4/4)

find:
search for files in a directory hierarchy
Usage: find [OPTION] [path] [pattern]
eg. find file1.txt
find -name file1.txt

history:
prints recently used commands Usage: history

PATTERN

A Pattern is an expression that describes a set of strings which is used to give a concise description of a set, without having to list all elements.

Example:
ab*cd matches anything that starts with ab and ends with cd etc.

ls *.txt – prints all text files

rm *.obj – removes all object files

TEXT PROCESSING (1/3)

cat:
concatenate files and print on the standard output…
Usage: cat [OPTION] [FILE]… eg. cat file1.txt file2.txt
cat -n file1.txt

echo:
display a line of text
Usage: echo [OPTION] [string] … eg. echo I love Pakistan
echo $HOME

TEXT PROCESSING (2/3)

grep:
print lines matching a pattern
Usage: grep [OPTION] PATTERN [FILE]...
eg. grep -i apple sample.txt

wc:
print the number of newlines, words, and bytes in files…
Usage: wc [OPTION]... [FILE]...
eg. wc file1.txt wc -L file1.txt

TEXT PROCESSING (3/3)

sort:
sort lines of text files
Usage: sort [OPTION]... [FILE]...
eg. sort file1.txt sort -r file1.txt

LINUX FILE PERMISSIONS

- 3 types of file permissions – read, write, execute
- 10-bit format from 'ls -l' command

1	2 3 4	5 6 7	8 9 10
file type	owner	group	others

E.g. drwxrw-r-- : means **d** is directory, **owner** has all three permissions, **group** has read and write, **others** have only read permission

- read permission = 4
- write permission = 2
- execute permission = 1
- E.g. rwxrw-r-- = 764
- E.g. 673 = rw-rwx-wx

SYSTEM ADMINISTRATION (1/2)

chmod:
change file access permissions
Usage: chmod [OPTION] [MODE] [FILE]
eg. chmod 744 calculate.sh
chown:
change file owner and group
Usage: chown [OPTION]... OWNER[:[GROUP]] FILE...
eg. chown remo myfile.txt
su:
change user ID or become super-user Usage: su [OPTION] [LOGIN]
eg. su remo, su

SYSTEM ADMINISTRATION (2/2)

passwd:
update a user's authentication tokens(s) Usage: passwd [OPTION]
eg. passwd
who:
show who is logged on Usage: who [OPTION]
eg. who , who -b , who -q

PROCESS MANAGEMENT

ps:
report a snapshot of the current processes Usage: ps [OPTION]
eg. ps,ps -el
kill:
to kill a process(using signal mechanism)
Usage: kill [OPTION] pid eg. kill -9 2275

ARCHIVAL

tar:
to archive a file
Usage: tar [OPTION] DEST SOURCE
tar -cvf /home/archive.tar /home/original tar -xvf /home/archive.tar
zip:
package and compress (archive) files Usage: zip [OPTION] DEST SOURSE
eg. zip original.zip original
unzip:
list, test and extract compressed files in a ZIP archive
Usage: unzip filename
eg. unzip original.zip

NETWORK

SSH:
SSH client (remote login program)
"ssh is a program for logging into a remote mach ine and forexecuting commands on a remote machine"
Usage: ssh [options] [user]@hostname eg. ssh -Y guest@10.105.11.20
scp:
secure copy (remote file copy program)
"scp copies files between hosts on a network"
Usage: scp [options] [[user]@host1:file1] [[user]@host2:file2]
eg. scp file1.txt guest@10.105.11.20:~/Desktop/

FILE SYSTEM (1/2)

fdisk:
partition manipulator eg. sudo fdisk -l
mount:
mount a file system
Usage: mount -t type device dir
eg. mount /dev/sda5 /media/target
umount:
unmount file systems
Usage: umount [OPTIONS] dir | device... eg.
umount /media/target

FILE SYSTEM (2/2)

du:
estimate file space usage Usage: du [OPTION]... [FILE]...
eg. du
df:
report file system disk space usage
Usage: df [OPTION]... [FILE]...
eg. df
quota:
display disk usage and limits Usage: quota [OPTION]
eg. quota -v

EDITOR COMMANDS

vi:
Vi Improved, a programmers text editor Usage: vim [OPTION] [file]...
e.g. vi hello.c
gedit:
A text Editor. Used to create and edit files. Usage: gedit [OPTION] [FILE]...
eg. gedit
pico:
Simple and very easy to use text editor
Usage: pico [OPTION] eg. pico

USING ALIASES

Aliases provide command-substitution functionality. They can be used to create new commands or modify the default behaviour of existing commands

Syntax: alias <string entered by user>=<string substituted by the shell>

The *alias command is used to view and create aliases*

- called with no arguments, it prints out the current aliases
- alias name=value creates a new alias
- custom user aliases are stored in *.bashrc or .cshrc*

Examples:

alias rm = 'rm -i' (change the behaviour of *rm to confirm deletes)*
alias ll = 'ls -l | more' (create a new command for friendly file listings)

ENVIRONMENT VARIABLES (1/2)

Environment variables refers to global settings that control the function of the shell and other Linux programs. They are sometimes called global shell variables
or in simple words
They define the user environment and are read from initialization files each time a user logs in…

To view the value of a variable, type:
printenv VARNAME OR *echo $VARNAME*
To check your environment, type
printenv OR *env*
To Set Environment Variable
setenv EDITOR "vim"

Some common environment variables:

HOME:	Your home directory (often be abbreviated as "~")
PWD:	Current working directory
EDITOR:	User's preferred text editor

ENVIRONMENT VARIABLES (2/2)

Some common environment variables:

EDITOR:	Sets the editor to be used by programs such as mail clients
PATH:	Specifies directories to be searched for executables.
SHELL:	The default login shell
USER:	Current loggedin user's name
TERM:	The type of terminal you are running (e.g. vt100,xterm, & ANSI)
LD_LIBRARY_PATH:	It is a colon seperated set of directories where libraries should be searched for

To reload any initialization file without having to logout and login again, type
source <filename>
e.g. source ~/.bashrc source ~/.tcshrc

Profile files: session startup files are hidden files, find them via command "ls –a"….

.tcshrc .cshrc .bashrc etc…

USEFUL vi EDITOR COMMANDS

Arrow keys	Move cursor	^G	show current file and line
hjkl	Same as arrow keys	^F	forward screen backward screen
insert / i x	Insert text	^B	
dw dd	delete a character delete a word delete a line	^D	scroll down half screen
		^U	scroll up half screen
		nG	go to the beginning of the specified line (end default), where n is a line number
3dd	deletes 3 lines	/pat	search pat (next line matching pat)
u	undo previous change	^E	scroll window down one line
ZZ	exit vi , saving changes	^Y	scroll window up one line top line on screen
ESC	end insert or incomplete command (delete or rubout) interrupts	H	last line on screen
DEL	erase last character (backspace)	L	middle line on screen
^H	erase last word	M	
^W			