Linux Commands Reference Sheet by Sivaram.N			
Navigation Commands		***Informational Commands***	
cd or cd ~	Change directory to your home directory	**bc**	Used for calculator
cd dir	Change into the specified directory	**cal**	Shows the current month calendar
cd ..	Move up one directory (parent)	**date**	Shows the current date and time
cd -	Return to previous directory. An easy way to get back to your prevoius location.	**uptime**	Shows the current uptime
cd /	Change directory to root	**hostname**	Displays the hostname
!	To get the complete recent command	**who**	Display the users who are currently logged on
pwd	Prints working directory. Shows the current location in the file system Location	**W**	Shows who is logged on and what they are doing
ShortCuts: <something incomplete> TAB - Auto Completion up(down)_key - scrolls through command history		**whoami**	Displays who you are login as
File && Directory Commands		**free**	Shows memory and swap usage
mkdir	Create/Make a direcory	**id**	Displays your user-id and group-id
cd	Change into the specified directory	**df**	Report filesystem disk space usage
rmdir	Remove directory	**du**	Shows disk usage in particular directory. du -s (summary)
touch	Creates a file if not exists and updates the timestamp on the file if it already exists, without modifying its contents.	**uname -a** **uname -r**	Displays Kernel information Displays the Kernel version
cp **cp -i** **cp -r**	copies source file to target file. if target file exists, this will overwrite it!	**cat /proc/cpuinfo** **cat /proc/meminfo**	Displays information about your CPU Displays infomation about your memory usage
mv **mv -i (interactive)**	Renames the given file or directory. It is also used for move the content from source file to destination file.	**top**	Displays CPU process in a full-screen GUI.A great way to see the activity on your computer in real-time.
Rm **rm -i (interactive)** **rm -r (recursive)** **rm -rf**	Removes the file Always ask for user confirm Removes the given directories with all their contents Remove Forcefully	**history** **!!** **!1003** **!cat**	Displays the latest commands that you ran and their number. You can copy and paste command strings. call the latest command Recall a command by its number Recall the latest command matching a starting string
cat	Display the content of text file	**ls**	List all files & directories

Filter Commands	
more	*Displays a file output one page at a time*
less	*An imporved replacement for the "more" command. Allows you to scroll backward as well as forward*
head [<n>] <file>	Displays first <n> line of the given file (10 by default)
tail [<n>] <file>	Displays last <n> line of the given file (10 by default)
tail -f <file>	Display the last few lines of file, and then o/p appended data as the file grows(eg: tail -f var/log/syslog)
wc	To used to count the number of character in a file
wc -l	Prints the number of lines in a file.
wc -w	Prints the number of words in a file.
wc -c	Displays the count of bytes in a file.
wc -m	Prints the count of characters from a file.
sort <file>	Sorts the lines in the given file in character order and outputs them.
sort -r <file>	Same, but in reverse order.
sort –ru <file>	u: unique. Same, but just outputs identical lines once.

Soft and Hard Links

A soft(symbolic)link is a special file which is just a reference to the name of another one (file or directory):
Useful to reduce disk usage and complexity when 2 files have the same content.
eg:a.link > b.link
A hard link to a file is a regular file with exactly the same physical contents While they still save space, hard
links can't be distinguished from the original files. If you remove the original file, there is no impact on the hard
link contents. The contents are removed when there are no more files (hard links) to them.

ln –s file_name link_name
ln –s ../README.txt

ln file_name link_name - Create a hard link
Create a soft(symbolic) link
To create a link with to a file in another directory, with the same name

Searching Commands	
which	Shows full path of shell commands found in your path.
where is	Locates the program, source code, manual page for a command
find . -name name*	Find files starting with name in current directory.
grep <pattern> <files>	Scans the given files and displays the lines which match the given pattern.
grep error *.log	Displays all the lines containing error in the *.log files
grep –i error *.log	Same, but case insensitive.
grep –ri error	Same, but recursively in all the files in . and its subdirectories
grep –v info *.log	Outputs all the lines in the files except those containing info.
grep -o	Show matched part of file only
locate	A quickly way to search for files anywhere in file system.

Help

man <keyword>
man man
Info

Standard I/O, Redirection, Pipes

Standard Output
Standard output can be written (redirected) to a file using the > symbol, >> (append)
echo “README: No such file or directory” > README
Standard Input
Lots of commands, when not given input arguments, can take their input from standard input.
Standard Error
Error messages are usually output (if the program is well written) to standard error instead of standard output.
Standard error can be redirected through 2> or 2>>
Pipes
pipes are very useful to redirect the standard output of a command to the standard input of another one.