# Window Commands:

\ => the path for any directory is denoted by backslash.

ls => list all the directory.

ls path => eg. ls c:\ => list all the directory for the given path.

ls -Force path => list the hidden directory as well.

Get-Help ls -Full => get help about specific command.

~ => this represent user’s home directory like “C:\User\sachin sav”.

cd absolute/relative path => change directory to given path.

cd .. => go one level up from current directory.

pwd => print working directory.

` => this is escape character in windows.

mkdir directory\_name => make directory of given name i.e. “directory\_name”.

mkdir directory` name => mkdir “directory name” => make directory of given name i.e. “directory name”.

history => shows all previous entered commands.

CTRL+R => opens search option here one can type text and it will auto complete written text based on history commands(new in power shell)

#start\_typing => write # and start typing it will auto complete typed text based on history commands.

Arrow\_Up/Down => to navigate up and down in history.

clear => cls => clear the screen, does not delete history.

cp source destination => copy file/folder from source to destination.(if destination is absent it will copy the file into present directory).

cp source\_pattern destination => cp .\temp2\\* => it will copy all the file present in temp2 folder into current directory.

cp source\_folder destination => it will only copy the source folder at the destination not the content of inside it for that we have to call “-Recurse”, it goes through every child of source folder and copy it into destination along with source directory.

cp source destination -Verbose => it will show information while copying the file/folder.

cp folder1 folder2 -Recurse -Verbose => copy folder1 with all of its content into folder2.

mv source destination => move file/folder from source to destination(Can Use wild card like \*).

mv folder1 folder2 => Rename folder1 to folder2 if folder2 doesn’t exist before, else it will move folder1 into folder 2.

rm file/folder -Recurse => remove file or folder. without -Recurse also it will work, it will just ask for one extra conformation.

rm file/folder -Force -Recurse => To delete system file or folder.

rm file\_name -Confirm => delete file, it will give prompt before each deletion.

cat file\_name => print whole file in the console.

cat file\_name -Head n => print first n line in the console.

cat file\_name -Tail n => print last n line in the console.

more file\_name => print the all text till console is not full then it will pause, then press ENTER to go forward by a line at a time OR press SPACE to go next page. press q to quite.

start notepad file\_name => to open/create file of given file\_name and open it in notepad for editing.

Get-Alias alias\_name => will show to Powershell commands for given alias\_name. eg. get-alias ls-> Get-ChildItem.

PowerShell contains Powershell command, Aliases and cmd.exe commands, like Get-ChildItem is powershell command, ls is alias, dir is cmd.exe command.

By default window search function searches string in file on the bases of file property not by content of the file, but we can enable it using indexing, open “Indexing Option” by searching then select “Users” and press “Advance” button then choose “File Type” tab and then choose “indexing by property and content” radio button.

sls word\_search file\_name => Using CLI it is very easy to search word in the file.

sls word\_search \*.txt => it will search “word\_search” in all the .txt file and give the result. Note- sls is alias for Select-String.

ls -Recurse -Filter \*.exe -Depth 2 => this will list all file of current and sub folder till depth 2 which are ending with .exe.

echo text => to print the text on shell.

echo text > file\_name => to write the text in file.

echo text >> file\_name => to append the text in file.

cat h2.txt | sls sachin => it will print all the matching string of file h2 which contain sachin.

Note : pipe(|) character is used for passing output of one command as input to other command.

Redirection operator:

1. > : it will redirect stdout stream.
2. 2> : it will redirect stderr stream.

$null => it is a variable which contain definition for nothing (like black hole) use for redirection into it.

### Week-2

Computer Mangaement- system tool- localuser and gruoup =>gui option to manage user(in window 10 home this is not available)

Get-LocalUser(glu) => shows all user account with other info like enable or not.

Get-LocalGroup(glg) => shows all group with other info.

Get-LocalGroupMember(glgm) group\_name => shows all user present in group group\_name.

net user user\_name passwd\_x => to change the passwd of user user\_name to passwd\_x.

net user user\_name \* => this will ask passwd in next line and shows that in \* form to be secure.

net user user\_name /logonpasswordchg:yes => force user named user\_name to change password on next login.

net user user\_name user\_pwd /add => to add user, we can write \* in place of user\_pwd, then it will ask for pwd in next line.

net user user\_name /del => to remove user named user\_name.

Remove-LocalUser(rlu) user\_name => to remove user named user\_name.

icacls dir\_name =>  It is capable of displaying and modifying the security descriptors on folders and files. An access control list is a list of permissions for securable object, such as a file or folder, that controls who can access it.

[ Other way to see this, go to direcotry right click-> properties -> security. ]

icacls dir\_name /grant “Everyone:(OI)(CI)(R)” => set Read permission for Everyone group.

icacls dir\_name /remove Everyone => remove permission for Everyone group.

# Linux Commands:

/ => the path for any directory is refer by forward slash and it also shows the root directory.

ls --help => for help.

man ls => manual information, more detail information than help.

ls -l => ls -l / => -l flag is use for long listing format(detail file information).

ls -a / => show hidden file as well.

ls -l -a / => ls -la / => show hidden file in long listing format.

pwd, cd, ~, .. => are same as windows.

~ => eg. \home\sachin I.e. refer to user’s home directory.

/ => eg . \ I.e. refer to root directory.

\ => Escape character in Linux.

mkdir dir\_name => work same like window.

history, CTRL+R,Arrow\_Up/Down,clear => Work same as window.

Note: #start\_typing => this doesn’t work here.

cp source destination => same as windows except “-Recurse” and “-Verbose” which has been replaced by “-r” and “-v” respectively.

cp folder1 folder2 -rv => copy folder1 with all its content into folder2.

mv source destination => same as windows.

rm file/folder -r => to delete file or folder recursively.

cat file\_name => more file\_name => same as window.

less file\_name => launch an interactive shell can go up and down by pressing up and down arrow key,pageUp and pageDown, g for navigate to beginning of page,G for end of page, /word\_search to search word or phrase in the file,q for quite the interactive shell.

head -n file\_name => print first n line.(by default 10)

tail -n file\_name => print last n line.(by default 10)

nano file\_name => open/create file of given file\_name and open it in interactive shell for editing.

vim file\_name => to open/create file of given file\_name and open it in interactive shell in command mode, :q to quite to interactive shell, :w for save, :wq for save and quite, :q! for quit without saving, “:set number” for displaying text with line number, “i” to go in insert mode then one can edit the file in command mode one can’t, press ESC to come back in command mode from insert mode, “dd” to delete a line, “[n]dd” to delete n lines,”u” for undo, “CTRL+r” for redo, /word\_search to search word or phrase in a file and “n” to go to next match and “N” for previous match.

:%s/yourtext/replacetext/gc => use for replace yourtext with replacetext, g flag shows replace all occurrence and c flag will ask for conformation before each replacement.

grep pattern file\_name => stands for “global regular expression print”, it will search given pattern in the given file\_name.

Redirection Operator :

1. < : used to redirect stdin stream.
2. > : used to redirect stdout stream.
3. 2> : used to redirect stderro stream.

/dev/null => a directory for storing definition of nothing(like black hole).

| => operator to pass output of one command to input to other(same as window).

ls | grep hel => to find directory of name which match string hel from all the current directory.

### Week-2

ubuntu --default-user root => execute this command in cmd, this will change default user of ubuntu as root, then you can change passwd of anyone simpy by command *passwd user\_name* in ubuntu.

root => this is the first user that get created while installing linux, it has permission of all things.

sudo => superuser do , something which require root user permission.

sudo su - => su(substitude user) by default it change to root.

exit => to came back to normal user from root user.

cat /etc/group => shows all the groups.(eg. sudo:x:27:sachin represent group\_name:engrpted\_password:group\_id:listOfUserInTheGroup respectively).

cat /etc/passwd => shows all the users.(eg. root:x:0:0:root:/root:/bin/bash represent user\_name:encrypted\_passwd:user\_id: other info are irrelevent for now).

passwd user\_name => it will ask for old\_pwd then new\_pwd, used to change the passwd.

sudo passwd -e user\_name => it will expire the current passwd of user named user\_name and tell to set new password when user login next time. (similar to windows /logonpasswordchg:yes

sudo useradd user\_name => add user named user\_name.

sudo userdel user\_name => delete user named user\_name.

ls -l dir\_name => it lists all dir and file in long listing format. eg. shown below

[ -rwxrwxrwx 1 root root 17230 Sep 16 21:15 Commands.docx

drwxrwxrwx 1 root root 4096 Sep 16 21:22 temp

1st char is shows info about is it normal file(-) or directory(d) next 3 char showing permission for owner user,next 3 for permission of groups which owner belong to,next 3 shows permission for other groups. read(r), write(w), executable(e), disabled(-), in our case 1st root is owner name, 2nd root is showing owner belong to group named root. ]

ls -l file\_name => to see file in long listing format.

ls -ld dir\_name => to see directory in long listing format.

u-owner, g-group, o-other.

4-read, 2-write, 1-execute.

sudo chmod [ugo][-+][rwx] dir\_name => + to add permission , - to remove permission.

sudo chmod 754 dir\_name => 7=4+2+1 i.e rwx, 5=4+1 i.e rx, 4=4 i.e r, respectively for ugo users.

sudo chown user\_name dir\_name => change owner of dir named dir\_name to user named user\_name.

sudo chgrp grp\_name dir\_name => change group of file belong to.