```
#!/bin/bash
# author: Surya Dev Singh Jamwal
# course: PG-DBDA
# user ID: sdevsinx
# date created: 2023 Mar 18
# title: Linux Module Day01
#++++++++++++++++++++++++++++++++++++



# 1. uname
# To check the version of your name
uname -r

# 2. uname
# To check OS
uname

# 3. man
# man command  in linux is used to display manual pages for any command.
# The manual page contains other system utilities information also
# The man page contains detailed information about how a command can be
used and what are the various arguments which that command can have
man uname
man mkdir

# 4. pwd
# pwd : prints the full path of  your current working directory
# This command is useful for navigating through your system file structure
pwd



# 5. version  check
# to check the version of pwd
/bin/pwd --version

#6. clear
# clears the terminal screen
# shortcut is ctrl + l
clear
```

```
#7. cal & ncal
# cal command is used to display the calendar for a given month & year
# only writing 'cal' will display the current month's calendar
# to enter desired month  enter "cal -m 2 2022"
cal     # prints calendar for current month for current year in horizontal
format
cal -m 2 2022   # prints 2nd month calendar for year 2022
cal -m 3    # prints 3rd month calendar for current year
ncal        # prints vertical calendar


#8. date
# date command displays the current date in linux
date


#9. whoami
# this command in linux is used to display the current user ID
# and user name of the user who is logged-in
# This will print user name of the user who is running the command
# This command is useful when number of users have logged in
# and your want to know which user account is being used
whoami

# 10. whatis
# whatis command is used to  get one line description of any command
# This can be used for quick reference
# When you don't know any command & don't want to go to manual page
whatis pwd

#11. w
# 'w' command is used to display the currently logged users and their
processes
# on terminal you'll get information such as user name, tty (terminal),
login session, the time when user logged in, and the current system load
average
# This command can be used when multiple users are logged in, what they're
doing, how long they've been logged in
# This command helps you to check login history and activities of a user
```

```
w

#12. sudo -i
# to login as root / admin user
# to create new user we need to login to root
sudo -i

#13. adduser cdac
# To add new user
adduser cdac

#14. su cdac
# to login to a  normal user account using user's user name
su cdac

#15. ps
# ps command will list all the running processes
ps


#16. ps aux
# This command displays list of processes that are running on your system, data is displayed with additional information such as CPU & memory usage
ps aux

#17 . history
# history command will list all the commands that you've previously used
history

#18. mkdir
# used to create a new directory  in file system
mkdir t1

#19. sudo
# use sudo before any command when you don't  have permission to execute a command
sudo mkdir

#20. ls
```

```
# lists out all the files & directories in the current directory or
specified directory
ls      # to show list of all the files, directories, links
ls --version    # to check version of ls
ls -l       # shows files, directories, links, size, modified date, time,
owner, permissions etc.
ls -a       # displays a list of all files including current dir, previous
dir, hidden files starting with '.'
ls -lh      # display all the information in human readable format
ls -lS      # displays all the information in sorted order
ls -S -n    # displays all the information in sorted order,, '-n' means
with uid, gid
ls -i       # check iNode # iNode stores information about files in linux
such as access mode, ownerships, file types, file size, group number,
number of links,
ls -R       # list files recursively, to list contents of directories &
its sub-directories
ls -lt      # displays file/dir listing with modified date in ascending
order
ls -d */    # '-d' lists only directories, '*/' lists contents of the
directories
ls ~        # displays list of files & dirs in home dir
ls *        # displays list of all the files, dirs & their sub-dirs
ls -S       # displays files & dirs, '-S' sorts in descending order by
date of creation
ls -n       # displays user ID (UID), group ID (GID)of file/dir
ls -G       # displays files & dirs which belong to same group

#21. mkdir t2 t3
#  to create multiple directories
mkdir t2 t3

#22. mkdir -p t4/t5/t6
# to create parent directory structure
mkdir t4/t5/t6

#23. cd
# to change directory as specified
cd t4
cd ~        # takes to home directory of currently logged in user
```

```
cd      # takes to home directory of currently logged in user
cd ..       # takes one directory level up
cd Desktop  # takes you to 'Desktop' directory
cd -        # takes you to the previous working directory
cd /        # takes you to the system's working directory
cd ~user    # takes you to the user's directory, works only with root

#24. mkdir -v t7 t8
# to print verbosing for mkdir, it'll print information that directory is
created
mkdir -v t7 t8

#25. PROBLEM STATEMENT : Create following directory structure
#
# t7 ->a
#     |   ->a1
#     |       ->a2
#     |           ->a3
#     ->b
#         ->b1
#             ->b2
#                 ->b3
cd t7
mkdir -p a/a1/a2/a3 b/b1/b2/b3
mkdir -p t77/{a/a1/a2/a3,b/b1/b2/b3} # with curly braces

#26. tree
# prints tree structure for files and directories in current directory
tree

#27.  rm (remove) command
# used to remove or delete files/dirs
touch filee.smmm
rm filee.smmm
rm *.smmm       # to remove files with a particular extension
rm -f filee.smmm    # to forcibly delete the file/dir, used to protected
dirs
rm -r dir22     # to recursively delete directory, to delete its sub-dirs
too
rm -i filee.smm     # to get prompt befor deleting file
```

```
rm -d dir29     # to remove a directory
rm -r dir31     # to delete dir & its sub-dirs
rm *.log        # to delete files using regex , for files with long names
and extension '.log'

#28. rmdir (remove dir) command
#  used to delete directory
rmdir dir29 # to delete directory
rmdir -p dir33  # to delete parent directory in structure
'dir31/dir32/dir33'

#29. cp <src_file> <dest_file>
# to copy a file from src to dest
cp src_file.txt dest_file.txt
cat src_file.txt            # to check contents of src_file
cat dest_file.txt           # to check contents of dest_file
cp -i src_file.txt dest_file.txt    # interactive copy, prompts before
overwriting
cp -r /ab/cd/src_dir_path /ef/gh/dest_dir_path  # recursive copy, for
copying directory, will copy src_dir along with its sub-dirs
cp -n src_dir dest_dir          # to avoid overwriting existing dir, use
'-n'

#30. rsync
# rsync is used to synchronize/transfer the data/files between two
locations
# This command is mostly used between two different machines
rsync -a src_fileName dest_fileName

#31. mv (move) command
# To move a file or a dir from one location to another
mv t1/filee.txt t2/.                # it will move 'filee.txt' in 't1' dir
to 't2' dir
mv src_path/*.txt dest_path/.           # to move all files with '.txt'
extension at src_dir to the dest_dir
mv -n src_path dest_path            # to avoid overwriting files, use '-n'

#32. creating backups
mv --backup -S 01 src_file_path dest_file_path  # to create backup using
mv
```

```
cp --backup -S 01 src_file_path dest_file_path  # to create backup using
cp
mv -b src dest                  # to create backup, similar to --backup,
but doesn't accept arguments
cp -b src dest                  # to create backup, similar to --backup,
but doesn't accept arguments

#33. difference between cp & mv command
```

cp	mv
used to copy files	used to move files
can't be used to rename files	can be used to rename files
will copy, but will not delete original file	will delete the original file while moving

```
#34. Renaming a file
# used to rename a file, need to install rename utility
# we can rename a file using mv command
mv old_fileName new_fileName    # renames file  with new name

#35. difference between rename &. mv command
```

rename command	mv command
more advanced, can use regex	mv command doesn't accept regex

```
#================================================================================
================

#1. How to create a file
# touch command will create a new file / overwrite the file creation time
touch f.txt # touches a file

# cat command method
cat > new1.txt  # uses redirection op to put empty contents in file.txt
```

```
        # creates empty file, you need to add contents in the text file
        # press ctrl + d to save the file
ls -l new1.txt  # To check if the file has been created or not, use 'ls -l
<fileName>'
cat new1.txt    # TO check the contents of the file, use 'cat <fileName>

#echo command method
echo "hello everyone" > new2.txt    # echo command will create a file in
the current dir , but we need to pass the contents of file as inline to
the command
cat new2.txt    # prints contents of file on terminal

# printf command method
print "hello everyone" > new3.txt # printf command will create a file in
current dir, but we need to pass the contents of file as inline to the
command

# nano editor  method
nano new4.txt   # nano command opens nano editor, type in the contents of
file, press 'ctrl+X' to save & exit the file

# vi editor method
vi new5.txt # vi command  opens vi editor, start insert mode using 'i'
key, type in the contents of file, press 'esc' to get to command mode, now
use ':wq' to save & quit vi editor

# vim editor method
vim new6.txt    # similar to vi , just an upgraded version of vi editor,
steps same as vi editor

#2. vi editor  commands
#3 modes in vi editor
#   a. insert mode 'i' to insert in file
#   b. command mode 'esc' to type commands
#   c.
#
#
# in command mode
# :q        to quit
# :w        to save
```

```
# :wq       to save & quit
# :q!       to forcibly quit
# :w!       to forcibly save
# dd        to delete/cut current line
# cc        to cut current line & open new line
# yy        to copy current line
# p     to paste
# o     to open new line in insert mode
# 'esc'u    to undo last changes
# gg        to go to 1st line
# G     to go to last line
# :12       to get to line no 12

# vi editor & vim editor difference
```

vi editor	vim editor
basic editor	advanced editor
only available in UNIX based OS	available on other OS e.g. windows
doesn't provide multiple levels of undo	provides multiple levels of undo