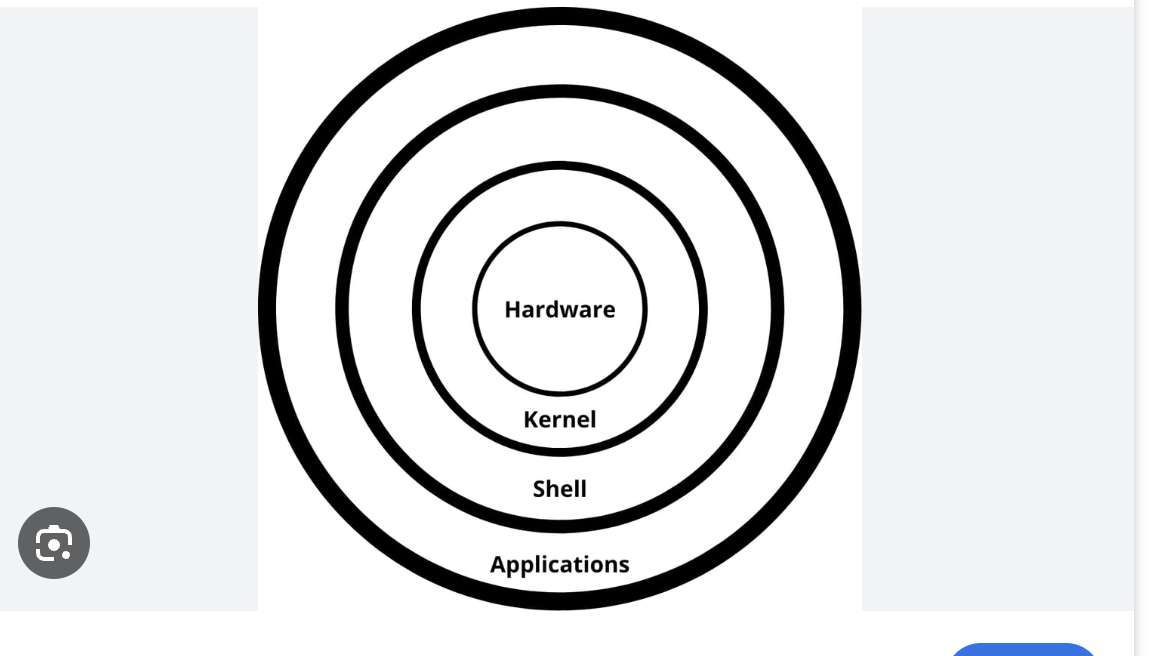
# DevOps

## Day 1: Introduction

**Day 2:** Introduction to DevOps and Unix, Created AWS account (created an EC2 Ubuntu instance )



**Day 3: Unix Commands**

clear -- clear the screen

pwd -- present working directory

ls -- list the files

mkdir dir -- Make directory/ create directory

ls dir -- List the files under dir folder/directory

cd .. -- get out of the current directory

ls \dr -- this won’t work, because it will search root directory. We shouldn't use / (/ refers to root)

ls dr -- this will work

ls a – this will display all the files, this will show the hidden files as well(Hidden files starts with .)

mkdir -p dir -- create a directory if it doesn't exist (-p Creates parent directories if they don't exist.)

mkdir -p dir/dir1 -- this will create a new directory under dir directory.

mkdir -p NPR -- hers -p stands for parent directory, if the parent directory NPR already exist, the command won't throw any error.

if the parent directory doesn't exist then it will create a new directory with the name NPR

-----------------------------

touch filename -- this will create a file under the current directory.

touch devops/class1/filename -- this will create a file under the class1 directory

touch file1 file2 – this will create multiple files under the current directory

touch file1

touch file1

The touch command won’t throw the error if we already have the file with the same name. In mkdir we do get the file exist error and to overcome it we use the -p and we don’t have this option under the touch command.

A screenshot of a computer

Description automatically generated

**ls:**

ls -- list  
ls -l – longlist(directory, file, permissions, memory usage)

ls -lt – longlist with time(most recent file will be displayed first)

ls -lrt – this will be reverse the list by displaying the oldest file first

ls -a -- used to display hidden files (hidden files starts with ".filename")

* if it starts with "d" - directory
* if it starts with "-" – file
* if it starts with “l” link

**mkdir:**

mkdir dir1

mdir -p dir1

mkdir dir1 dir2

mkdir -p dir1 dir2

mkdir -p Test/Test1/Test2 – This will create all the folders (Subfolders)

**touch:**

touch filename

touch filename1 filename2

**Day 4: AWS and Unix Commands**

**CloudFormation:**

Created a stack with the help of the json file

**Commands:**

vi – Visual editor

[esc]I - insert

[esc]:wq – write and quit

[esc]:q – quit

[esc]:wq! - ! to forcefully quit the file

cat

[esc]:set nu – to set the line numbers

[esc]:set nonu – to remove the line numbers

[esc]:2 -- to move to the specific line number

[esc]dd -- to remove a specific line(first move to the specific line)

[esc]:3,5d – to delete the multiple lines

* this will delete the line number 3 to 5 (including 3rd and 5th lines)

[esc]:5d – delete the line 5

[esc]:d – this will delete the line where the cursor is present

[esc]u -- undo

[esc]/searchstring -- To search for a specific word on the file

We can use n to look at the next matching search string

[esc]:%s/currentstring/newstring/g – To replace the string

* will replace the pattern with new pattern in all the lines of a file
* g stands for global, s stands for search
* Search string will be case sensitive

[esc]:3s/pattern/newpattern/g – to replace the pattern – it will replace all the same pattern in 3rd line of file

[esc]:3s/pattern/newpattern/ -- if we don’t use the g, then it will replace the first occurrence in the specified line number

[esc]:3,5s/pattern/newpattern/g – to replace the pattern from line 3 to line 5

[esc]:3,$s/pattern/newpattern/g – it will replace the pattern from line 3 to end of the file

[esc]:$s/pattern/newpattern/ -- it will replace the last line

[esc]:%s/pattern/newpattern/g – will replace the pattern with new pattern in all the lines of a file

[esc]:s/pattern/newpattern/g – it replace the pattern in current line of a file

tac filename: This will display the content in the reverse order

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

**Day 5: Unix Commands**

**Move/Rename files or folders**

mv - cmd used to rename file or directory and also used to move the file or directories from source to destination path.

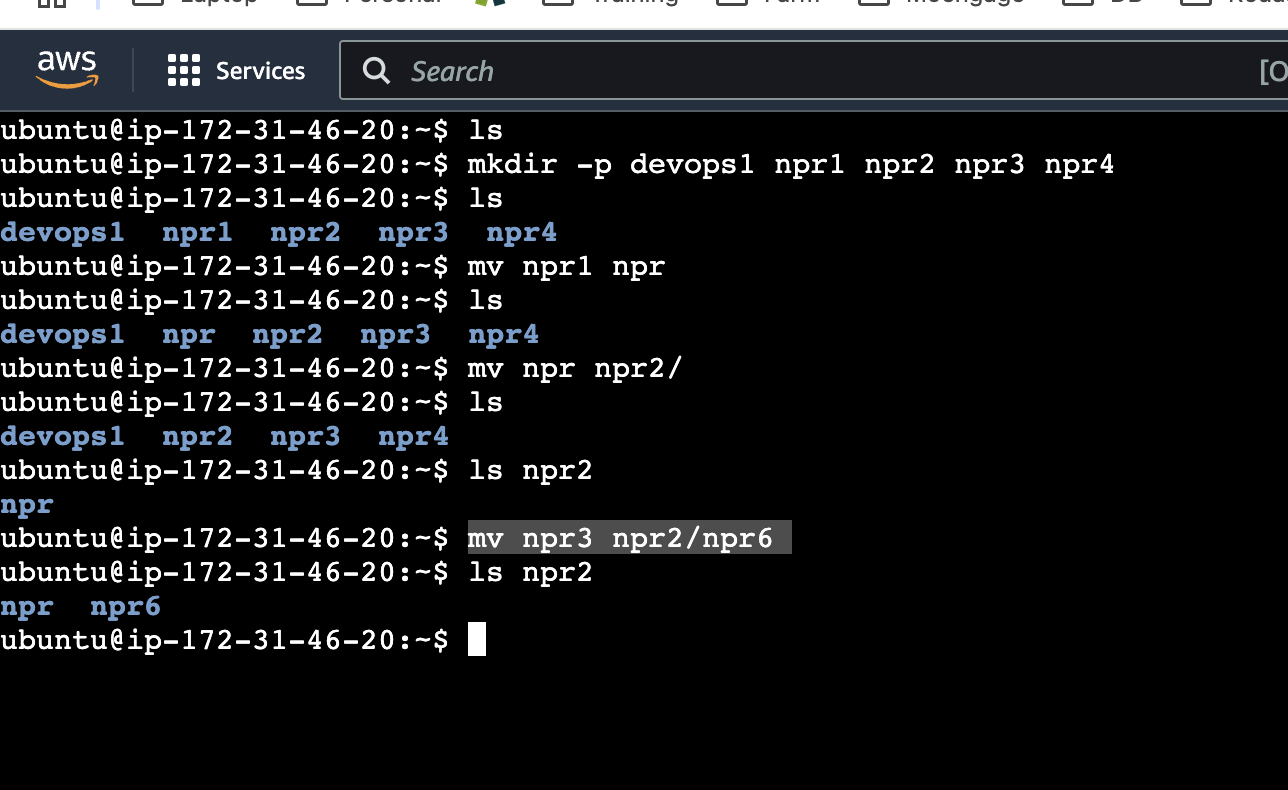
mv oldfilenmae newfilename - renaming a file

mv olddirectoryname newdirectoryname - renaming a directory

mv directory pathofdestination/ : to move the files or directories to destination

mv npr1 npr – This will rename the file/directory from npr1 to npr

mv npr npr2/ -- This will move the file/directory from the current path to npr2 directory

mv npr3 npr2/npr6 – This will rename the file/directory from npr3 to npr6 and will move to the npr2 directory 

**Copy files/folder:**

cp : used to copy a file/folder from source to destination r - recursive

cp pathofsource pathofdetination - used to copy the file from source to destination

cp -r pathfoldername pathofdestination- used to copy the folder from source to destination

cp pathofsource samepathassourcewithnewname – it will create duplicate file with new name,

note-since we are copying a file to same path if file already exist then it will over write, if file doesnot exists it will create duplicate files

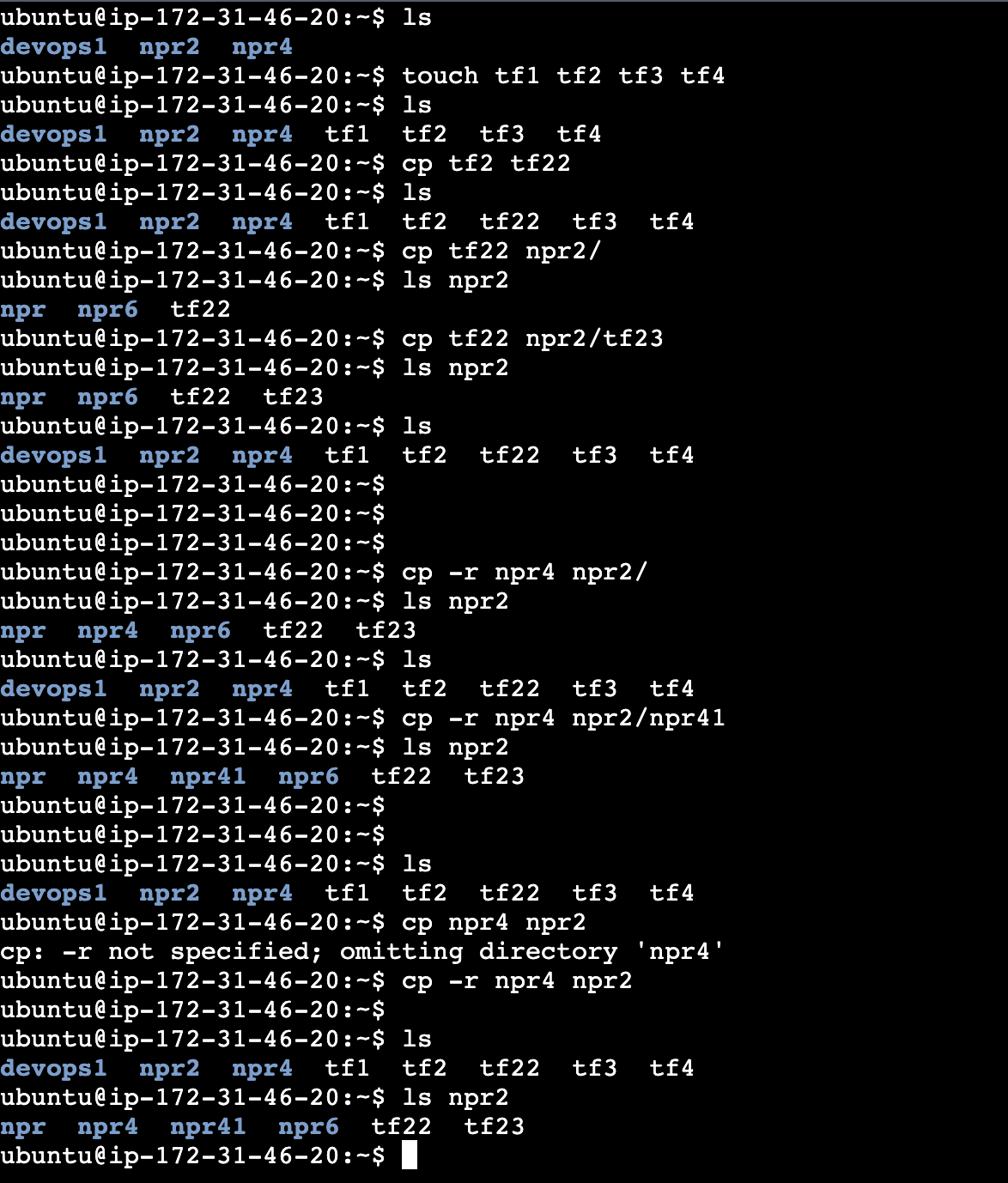
cp tf2 tf22

cp tf22 npr2/

cp tf22 npr2/tf23

cp -r npr4 npr2/

cp -r npr4 npr2/npr41



**Day 6: Unix Commands**

**Change permissions**

User : Group : others

R-4 W-2 X-1

r-read; w-write; x-executable; a-all

chmod : used to change the permission of a file or directory

chmod permission filename

ex: chmod 764 filename

chmod -R permission directoryname

ex: chmod -R 761 directoryname

example:

1)change the permission for the directory as read write for user, read execute for group and write for other

chmod -R 654 directory

2) read permission to the user for a file

chmod u+r file name

3) Add execute for user, write for group and remove write for others

chmod u+x, g+w, o-w filename

4) Add execute for all for a file and directory

chmod a+x filename

chmod -R a+x directory

**Disk Usage:**

du : used to display the size of the file

du -sh folder name : used to display size of the file/ folder(current file)

du -sh \* -used to display size of all the directories/ files

**du - disk usage,**

**s –summarize,**

**h- human readable format**

df – disk free/size -used to check the size of the disk drive

df -h : used to check the size/memory of the system

df -h . : used to check size /memory of the current drive

df - disk free

free – To check the memory

lsblk

The ' lsblk' command, short for “list block devices,” is a powerful tool that provides detailed information about block devices such as hard drives, solid-state drives, and other storage-related devices.

**To delete multiple files/directories**

Folder: rm -rf foldername1 foldername2 ….

File: rm filename1 filename2 ….

Deleting files through cases

Case 1 : by names – rm filename

Case 2 : by extensions – rm \*.ext (.html , .xls etc)

r – recursive

f – forcefully

**Day 7: Unix Commands**

**head: command used to display first n lines of a file**

ex: head filename (default n-10)

head -5 filename: first 5 lines

head -3 filename: first 3 lines

**tail: to display last n lines of a file**

ex: tail filename (default n-10))

tail -2 filename: last 2 lines

head -5 testfile1 | tail -1 – this will give us the 5th line of the file

head -7 testfile | tail -2 – this will give us the content from 5th to 7th line

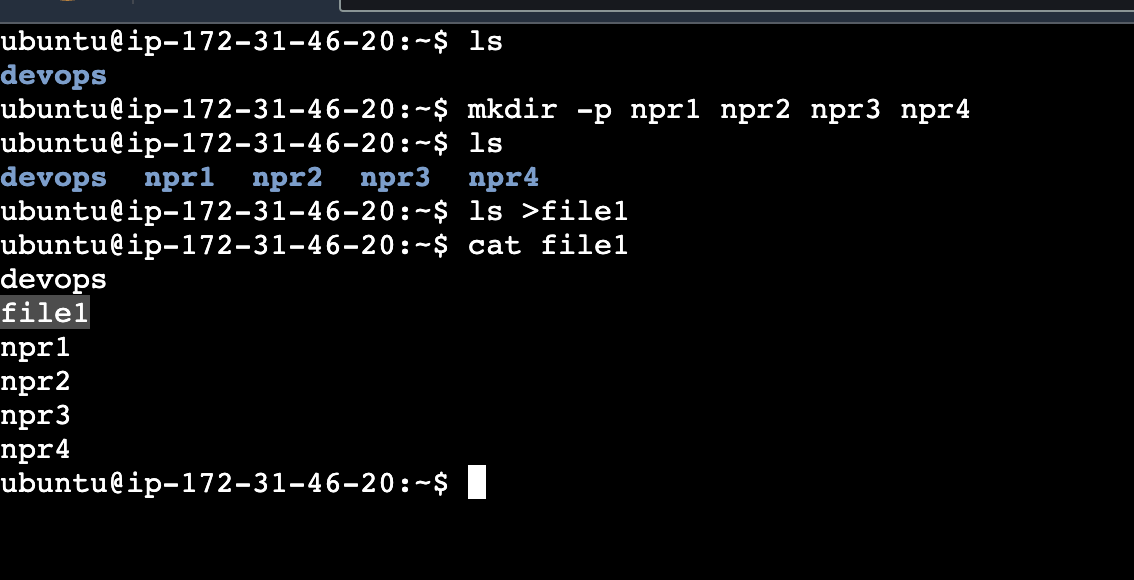
**>: redirect operator (>) - used to copy output of one command to a new file.**

If the file already exists it will overwrite.

ex: head -5 filename>filename2

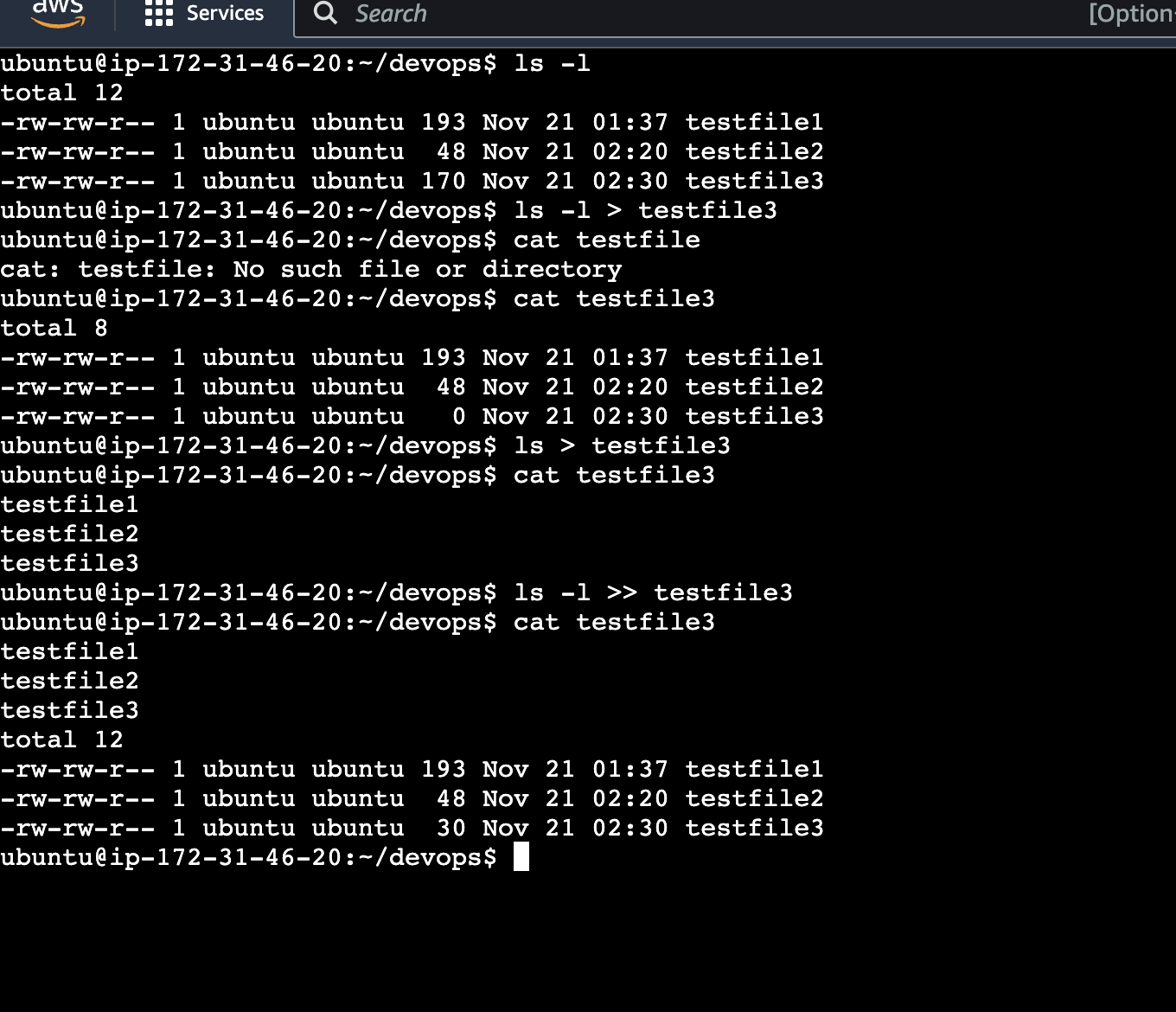
ls -l >testfile3

ls >file1



**>>: append symbol Append (>>) - used to append the output of command to the end of a file.**

ls -l >> testfile3

****

**To print the statements:**

echo “Welcome”

echo \* : this command does the same job as "ls" command do

echo : used to print the statements

syntax : echo "welcome" ;

output = welcome

echo -e : used to print the multiple statements.

Example echo -e "good morning \n weclome"

\n : will display the statements in two lines: output will be displayed as

good morning

welcome

ex: echo -e "good morning \t welcome"

\t : t => tab - will display the statements in with space(tab)

output: good morning welcome

**Notes:**

head - It is used to display first n number of lines

head filename – it will display the first 10 lines of a file.

head -7 filename – it will display the first 7 lines of a file

head -3 filename – it will display the first 3 lines of a file

head -1 filename – it will display the first line of a file

tail – It is used to display the last n number of lines

tail filename – it will display the last 10 lines of a file

tail -1 filename -- it will display the last 1 line of a file

tail -4 filename -- it will display the last 4 lines of a file

pipe (|) – it is used to pass output of one command as an input to another command

head -5 filename | tail -1 – this will display the 5th line of a file

How to display the content from line 16th to 20th?  
head -20 filename | tail -5

How to display 47th line of a file?

head -47 filename |tail -1

Redirect (>) – it is used to write the output of a command to the file.

If the file exists, it will override the content

If the file doesn’t exist, it will create a new file and add the content

head -3 file2>file3

Append(>>) – it is used to append the output of a command to the end of the file.

If the file exists, it will add the content to the end of the file.

If the file doesn’t exist, it will create a new file and add the content

echo – this is used to display the statements

echo “Welcome” -- output will be Welcome

echo -e "Good morning \nwelcome"

echo -e : used to print the multiple statements

\n : will display the statements in two lines

Output will be

**Good Morning**

**welcome**

Example: echo "Hi \n Good Morning”

Output will be

Hi \n Good Morning

We need to use the -e to print multiple statements

echo -e “Hi \n Good morning”

echo -e "hi\t how"

**Day 8: Unix Commands**

**grep : is used to search a pattern in a file**

syntax: grep pattern filename or grep "pattern" filename

recommended to use pattern within "" ie, 2nd syntax

It is used to search specific pattern in a file (case sensitive)

ex: grep "pattern" filename

grep “npr” file2 – case sensitive

grep -i “npr” file2 – case insensitive

grep -w “npr” file2 – this will match the exact word

grep -i -w “npr” file2- this will match the exact word with case insensitive

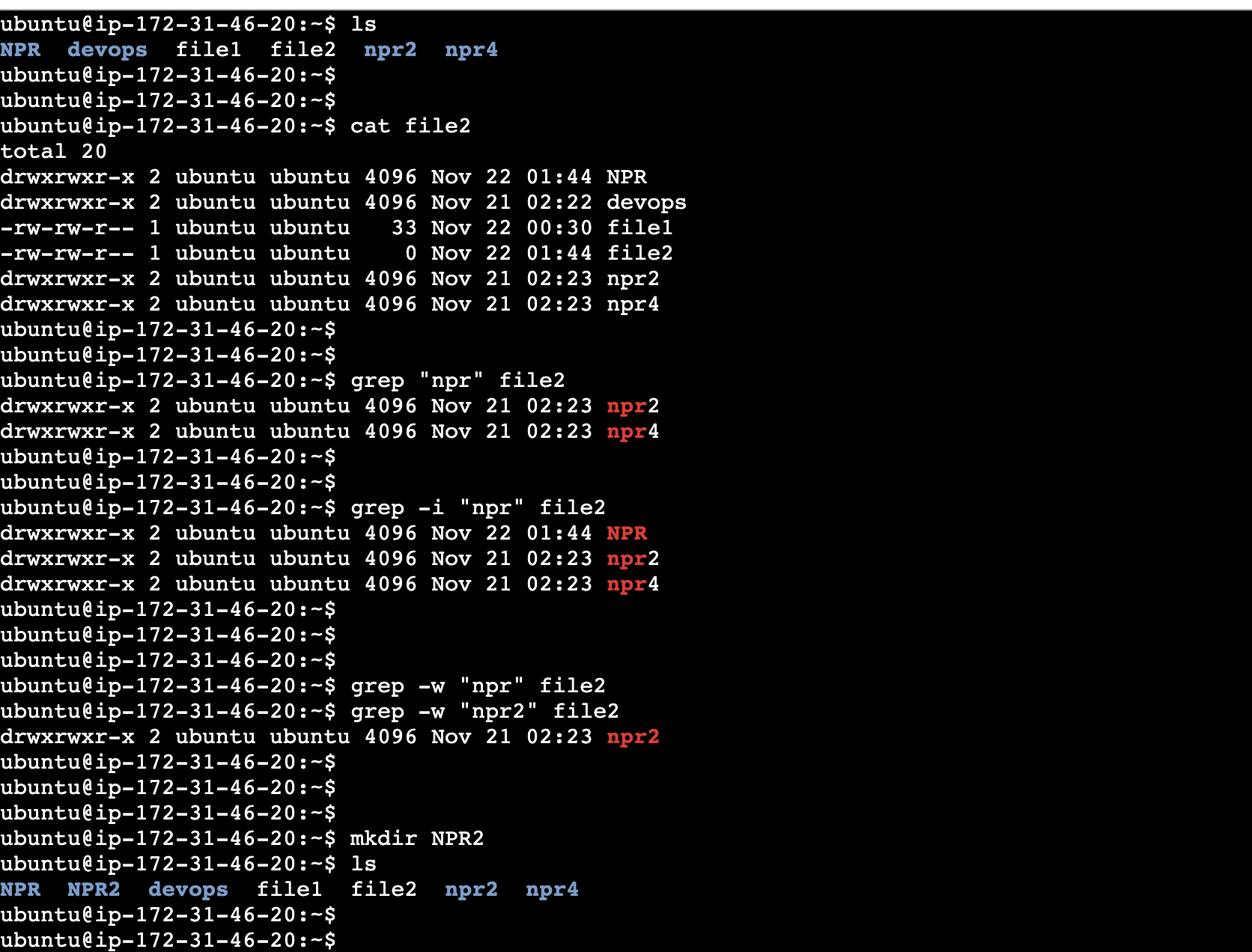
grep -c “npr” file2 – this will display the number of lines which contain the matching patterns

grep -l “npr” \* -- this will list the filenames that matches the pattern

grep -L “npr” \* -- this will list the filenames that doesn’t matches the pattern

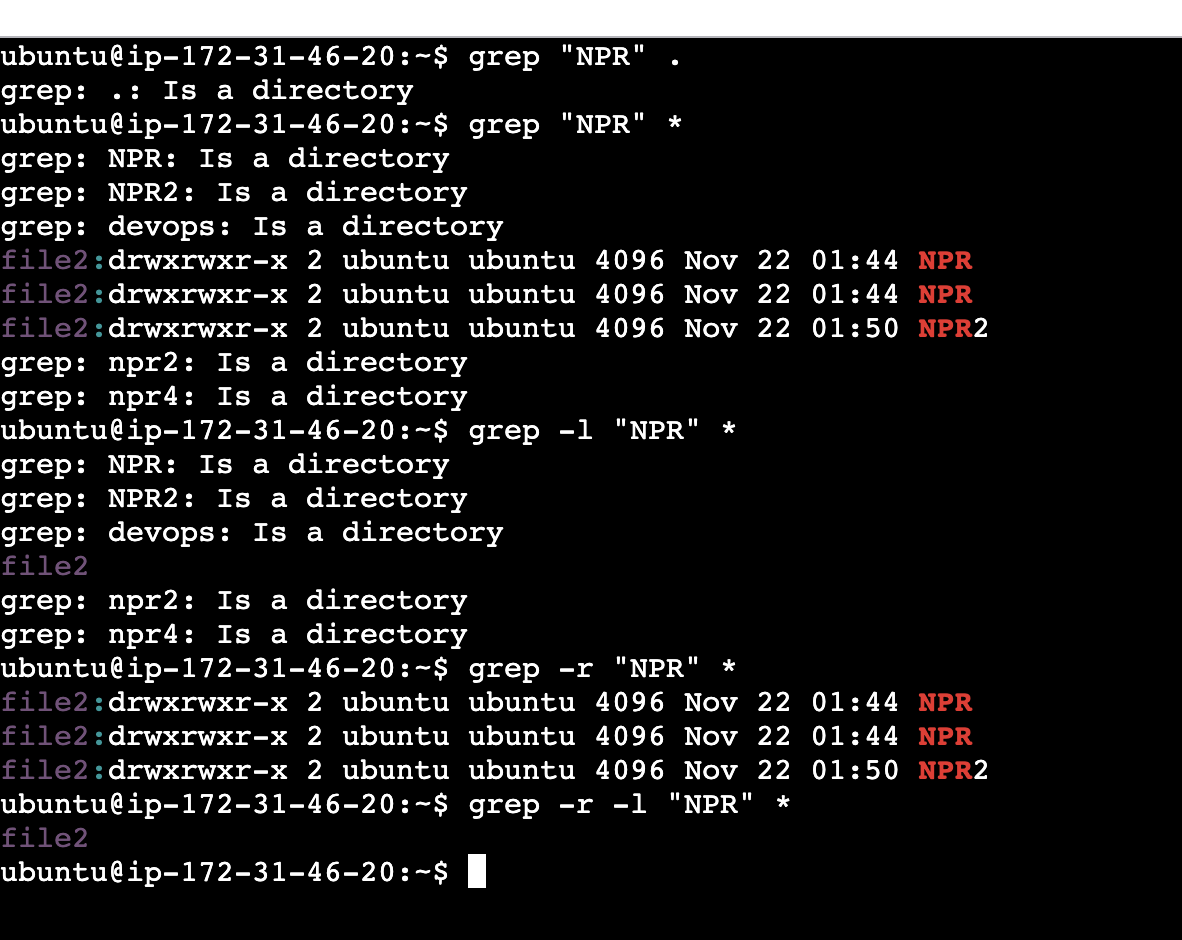
grep “npr” \* -- this will search the pattern in the current directory

grep -r “npr” \* -- this will search the pattern in the current directory and sub directories as well



A screenshot of a computer

Description automatically generated



**A screenshot of a computer

AI-generated content may be incorrect.**

**wc : word count;**

**used to count no. of lines, no of words, no. of characters in a file**

l- lines

w - words

c - characters

wc filename: no of lines, words, characters

wc -l filename: no of lines

wc -w filename: no of words

wc -c filename: no of characters

to count no. of characters in a 7th line of a file

head -7 file | tail -1 | wc -c

to count no. of words in 5th &6th line of a file

head -6 file | tail -2 | wc -w

to count no. of characters in a 8th line of a file and copy to a new file

head -8 file | tail -1 | wc -c >file1

**Notes:**

grep – it is used to search the pattern in a file

syntax: grep “npr” file2

This will display the line which contain the mentioned pattern(npr)

By default, this is case sensitive

grep -i “npr” file2

i – case insensitive

This will display the line which contain the mentioned pattern by ignoring the cases

grep -i -w “npr2” file2

This will display the line which matches the exact word and it’s case insensitive

grep -c “npr” file2

This will display the number of lines that contain the pattern npr

grep -c -w “npr” file2

grep -l “npr” \*

this will display the filenames which contain the pattern in the current directory

l- list the filenames that contains the matching pattern

\*- current directory

grep -r -l “npr” \*

this will display the files which contain a pattern from current directory and the sub directory

grep -i “^d” file2

this will display the lines that starts with the matching pattern(d)

grep -i “d$” file2

grep -i “2$” file2

This will display the lines that ends with the matching pattern (d or 2)

How to display the empty line from the file

grep “^$” file2

How to display the count the empty lines in a file

grep -c “^$” file2

grep -v “test” file2

this will display the lines, that doesn’t match the mentioned pattern

grep -n “npr” file2

this will display the line number which contain the mentioned pattern

grep -e “npr” -e “test” file2

this will display the lines that matches mentioned patterns (npr or test)

this is called egrep(multiple grep)

**wc: word count**

This is used to display the number of lines, words and the characters of the file

Syntax: wc filename

This will display the number of lines, words and the characters

wc -l filename

This will display the number of lines

wc -w filename

This will display the number of words

wc -c filename

This will display the number of characters

Find the number of characters in 14th line of a file

head -14 filename | tail -1 | wc -c

Find the number of words from 23rd to 26th line of a file

head -26 filename | tail -4 | wc -w

**Day 9: Unix Commands**

**SED command (stream editor)**

It is used to replace a string by another string in a file, delete the specific lines from a file & to display the specific line.

sed ‘s/pattern/pattern2/g’ filename : replace a string only in a console o/p not in the file

**This command case sensitive**

S indicates substitute –

sed ‘s/pattern/pattern2/Ig’ filename: it will replace the string irrespective of cases

Capital I indicates case insensitive

sed -i ‘s/pattern/pattern2/g’ filename : it will replace a pattern in a string in a file

i indicates insert

sed -i ‘s/pattern/pattern2/Ig’ filename: it will replace the string irrespective of cases

Capital I indicates case insensitive

sed -i ‘$s/pattern/pattern2/Ig’ filename – to change the pattern at the last line of the file

sed -i ‘2s/pattern/pattern2/Ig’ filename - 2S : it will replace only in a second line

sed -i ‘2, 5s/pattern/pattern2/Ig’ filename - 2, 5S means it will replace in 2 thru 5 lines

Sed -i ‘1s/pattern/pattern2/2g’ filename: It will replace from second occurrence of a file in the 1st line

Sed -i ‘1s/pattern/pattern2/2’ filename: It will replace only the second occurrence of a file in the 1st line

Sed -i ‘s/pattern//g’ filename: to delete the pattern

**How to print the lines using sed command**

sed -n ‘4p’ filename : to print the 4th line of a file

sed -n ‘2,5p’ filename : to print the 2 to 5th line of a file

sed -n ‘3p;5p’ filename : to print 3rd and 5th line of a file

sed -n ‘$p’ filename : to print the last line of a file

**How to delete a line using sed commands**

Sed -i ‘4d’ file name : it will delete 4th line of a file

sed -i ‘2,5d’ file name : it will delete lines from 2 to 5 of a file

sed -i ‘2d;5d’ filename : it will delete 2nd and 5th line of a file

sed -i ‘/^$/d’ filename : it will delete a empty line in a file

sed -i ‘/patternname/d’ filename : it will delete a line which contains a pattern

**Cut is used to cut the files (display) in column wise**

cut -d “ “ -f1 filename : display the first column of a file

**d : delimiter f :column 1: first column**

cut -d “ “ -f1-3 filename : display from 1st through 3rd column of a file

cut -d “ “ -f1,3 filename ; display 1st and 3rd column of a file

Disadvantages,

1. we can’t display in row wise

2. If we leave more space between the columns, then we will not get desired output.

**Notes:**

**sed(Stream editor):**

This is used to replace the pattern in a file

This is used to display the specific lines from the file

This is used to delete the specific lines from the file

Example:

sed ‘s/unix/linux/g’ testfile1

s stands for substitute

g stands for global in that line

This will replace a string unix with linux only in the console output, not on the file.

By default, this is case sensitive

sed ‘s/unix/linux/Ig’ testfile1

capital I stands for the case insensitive

This will replace a string irrespective of cases(upper or lower)

sed ‘7s/unix/linux/g’ testfile1

This will replace a string on the 7th line

sed ‘7s/unix/linux/2’ testfile1

This will replace a string unix with linux only in the second match on the 7th line

sed '1s/unix/linux/1I' testfile1

case insensitive

sed '7,15s/unix/linux/g' testfile1

This will replace a string from the 7th line to 15th line

sed 's/^$/replacing the blank lines/g' testfile1

This will replace a empty line with the provided string

sed -i ‘7s/unix/linux/g’ testfile1

Here i stands for insert

This will replace a string on the 7th line on the file

sed -i ‘$s/pattern/pattern2/Ig’ filename –

This will change the pattern at the last line of the file

sed ‘s/unix//g’ testfile1

this will delete the word unix

sed -n ‘7p’ testfile1

This will print the 7th line of a file

sed -n '3,7p' testfile1

This will print from 3rd to the 7th line on the console

sed -n '3p;7p' testfile1

This will print the 3rd and the 7th line on the console

sed -i ‘7d’ testfile1

this will delete the 7th line from the file

sed ‘7d’ testfile1

this will delete the 7th line on the console

sed -i ‘7,11d’ testfile1

This will delete the content from 7th to 11th line on the file

sed -i ‘7d:11d’ testfile1

This will delete the content on the 7th and the 11th line on the file

sed ‘unix/d’ testfile1

This will delete the line which contains the mentioned pattern

**Cut is used to cut the files (display) in column wise**

cut command is used the cut the file in the column wise

cut -d “ “ -f1 filename

**d : delimiter f :field 1: first column**

This will display the first column of a file

cut -d “ “ -f1,3 excelfile

This will display the first and the 3rd column

cut -d “ “ -f1-3 excelfile

This will display the content from 1st to the 3rd column

**Day 10: Unix Commands**

**awk** : this command is used to display the files in both column and row wise

This command is used for multiple purpose

awk -F “ “ ‘{print $2}’ filename

This will display the 2nd column of a file

**-F : field separator**

awk -F “ “ ‘{print $2, $4}’ filename

This will display the 2nd and 4th column of a file

awk ‘{print $2, $4}’ filename

This will display the 2nd and 4th column of a file

awk -F “ “ ‘{print $1, $2, $3}’ filename

This will display the 1st, 2nd and 3rd column of a file

awk -F “ “ ‘{print $0}’ filename

This will print all the columns from a file

awk -F “ “ ‘{print $NF}’ filename

This will display the last column of a file

**NF- number of fields**

awk ‘{print $NF}’ filename

This will also display the last column of a file

awk -F “ “ ‘{print $(NF -1)}’ filename

awk ‘{print $(NF -1)}’ filename

This will display the last but one column of a file

awk -F “ “ ‘/Roy/ {print}’ filename

awk ‘/Roy/ {print}’ filename

This will print the row/line which contains the pattern Roy.

This is case sensitive

awk -F “ “ ‘(NR>1) {print $2}’ filename

awk ‘(NR>1) {print $2}’ filename :

This will display from 2nd row and print 2nd column in a file

awk -F “ “ ‘(NR>1) {print $2, $5}’ filename

awk ‘(NR>1) {print $2, $5}’ filename :

This will display from 2nd row and print 2nd & 5th column in a file

awk -F “ “ ‘NR==4 {print}’ filename

This will print the 4th row of a file

awk -F “ “ ‘NR==2, NR == 4 {print}’ filename

This will print from the 2nd row to the 4th row of a file

awk -F “ “ ‘NR==2 || NR == 4 {print}’ filename

This will print the 2nd row and the 4th row of a file

--

awk -F “ “ ‘NR==2 || NR == 4 {print$1}’ filename

This will print the 2nd row and the 4th row of a file and only the content of the 1st column

awk -F “ “ ‘NR==2 || NR == 4 {print$1, $2}’ filename

This will print the 2nd row and the 4th row of a file and only the content of the 1st and 2nd column.

Assignment:

I need to retrieve the recent 2 filename created

awk ‘ NR==2, NR==3 {print $NF}’ filename : it will display rows 2 to 3 of last column

awk ‘ NR==2; NR==4 {print $0}’ filename: it will display only 2nd and 4th row of a file

awk -F " " 'NR==3 | | NR==5 {print $2}' filename : it will print column 2 from line 3 and 5 of a file

A computer screen with white text

AI-generated content may be incorrect.

**umask**

it is used to set the default permission for the newly created files and directories

The full permission for the file is 666

The full permission for the directory is 777

**Default values**

777 for directories

666 for files

**syntax:**

umask 022

This will set the permission for the file and directories as below

**File:**

666-022 = 644(RW:R:R)

Read write for User

Read for Group

Read for Others

**Directory:**

777-022 = 755 (RWX:RX:RX)

Read write and execute for User

Read execute for Group

Read execute for Others

umask 000 – gives full permission for the newly created file

**To display the OS version:**

lsb\_release -a

cat /etc/os-release

uname -v

This will disoplay the os name and when was this OS created

uname -v : check linux version

uname -a : to dispay all the details of linux

hostname

This will display the server/hostname

**Day 11: Unix Commands**

**chown (Change ownership)**

**This is used to change the ownership of a file or directory(username and groupname)**

Syntax:

chown username:groupname filename

chown ubuntu:ubuntu filename

This will change the ownership of a file

chown ubuntu filename

This will change the username of a file

chown :ubuntu filename

This will change the group name of a file

chown -R ubuntu:ubuntu directoryname

This will change the ownership of the directory

chown username:groupname filename - used to change the username group name of a file

chown -R usernmae:groupname dircetoryname- used to chnage ownerhsip of folder name

chown usernmae:groupname filename - used to chnage ownerhsip of file name

**sudo (Super user does)**

This is used to run the command with root permissions.

sudo chown ubuntu:ubuntu filename

**su (super user/switch user)**

This is used to login as another user or root user

sudo su –

This will login as root user

su - username

This will login as normal user

**find -it is used to find the location of a file or directory**

/ -- search from rootpath

. – from current directory/current path

path(/home/ubuntu/devops) – from the specificed path

find . -name file10

This will search for the file/directory ‘file10’ from the current directory

This is case sensitive

find . -iname File10

This will search for the file/directory ‘file10’ from the current directory

This is case insensitive

find . -type f -iname File10

This will search for the file ‘file10’ from the current directory

This is case insensitive

f stands for file

find . -type d -iname File10

This will search for the directory ‘file10’ from the current directory

This is case insensitive

d stands for directory

find /home/ubuntu/devops/ -type d -iname File10

This will search for the directory ‘file10’ from the /home/ubuntu/devops/ path

find /home/ubuntu/devops/ -type f -iname File10

This will search for the file ‘file10’ from the /home/ubuntu/devops/ path

find . -type f -mtime +7

This will display the files that were modified before 7 days

**m- modified**

**Time - days**

find . -type f -mtime -7

This will display the files that were modified in the last 7 days

How to list the files which are created or modified within 3 months

Find . -type f -mtime -90

How to list the files which are created or modified 3 months ago

Find . -type f -mtime +90

find . -type f -mmin -20

This will display the files that were modified in the last 20 minutes

**m- modified**

**min - minutes**

find . -type f -mmin +20

This will display the files that were modified before 20 minutes

find . -type f -prem 644

This will display the files that are having specified permissions (644)

find . -type f ! -prem 644

This will display the files that are not having specified permissions (644)

**! – negation**

find . -type f -empty

This will display the empty files

find . -type f -size +4k

This will display the files which are in size 4k

find . -type f -size +4M

This will display the files which are in size 4 mb

How to find the files which is modified within 3 months and 2 months back?

find . -type f -mtime -90 | find . -type f -mtime +60

find . -type f -mtime -90 ! -mtime +60

Find the files which were modified 3 months ago and delete

find . -type -f -mtime +90| xargs rm -rf

**xargs**

This is used to pass the arguments/values in the sequence to the other command

**Day 12: Unix Commands**

By default, the find command will check the sub folders as well(Auto recursive).

We can restrict this with the help of maxdepth option.

To restrict the automatic recursive, we use maxdepth

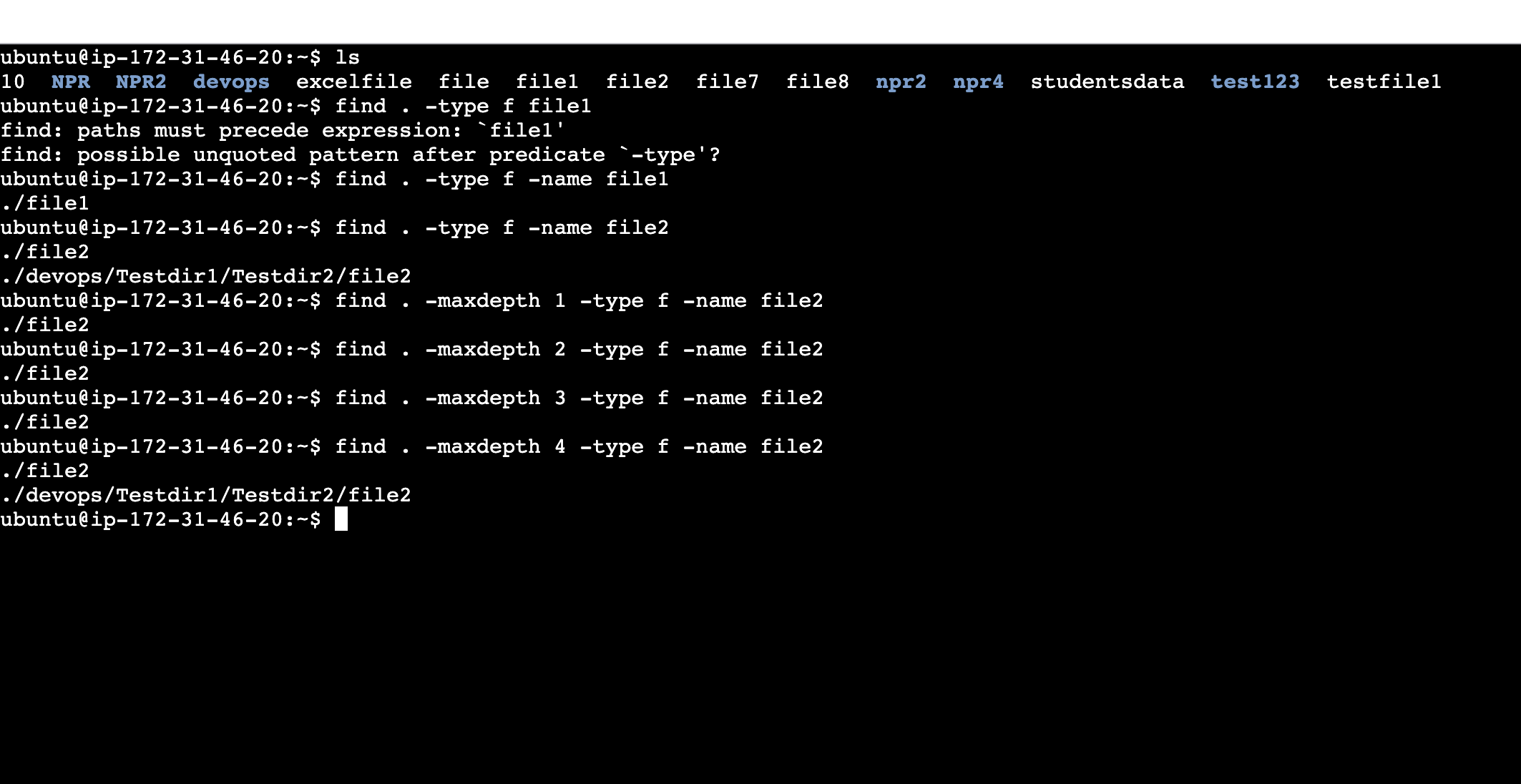
Max depth (number) indicates it will search the number of level you gave.

find . -maxdepth 1 -type f -iname filename

This will search in the current directory

find . -maxdepth 2 -type f -iname filename

This will search in the 1st sub directory



find -maxdepth 1 -mtime +2

this will list the files which were modified 2 days ago in the current directory

**-maxdepth 1**

This will search in the current directory

**-maxdepth 2**

This will search from the current directory and 1st level sub directory

**-maxdepth 3**

This will search from the current directory and 1st , 2nd level sub directory

**Link**

**It is a shortcut to the file.**

There will be two types of links

**1. Soft link (symbolic link)** :

It is short cut to a file, if we make any changes to original file it will reflect in link.

If the file is deleted, then link doesn’t work

Syntax: ln -s filename linkname

**2. Hard link**:

It is short cut to a file, if we make any changes to original file it will reflect in link.

If the file is deleted, then link will work

Syntax: ln filename linkname

**Diff b/n Soft link and hard link**

1. If I delete the actual file, soft link will not work

2. If I delete the actual file, hard link will work, because it points to the inode of a file. Inode is the memory address (unique identification number for file)

**List the files which are started with “t”**

1. ls t\*

2. ls |grep -i “^t”

**Ps (process status):** ps is a command used to check current process running on a system in a background

syntax:

ps : print all the process for current user and terminal

ps -e: print all the process within the system

ps -ef : it will print all running process with more detailed output

How do you check/list wheather specific process running or not

ps -ef | grep -i “processname”

ps -C processname (This will check if the specific process in running or not. This is used in scripting)

**Kill: kill is used to stop the process forcefully**

**syntax**

kill -9 pid

Pid – process id

How do you list process started by a specific user

ps -u username

**How do you check from how long the server was up and running?**

**uptime**: it is used to check load average and also to check

the server is up & running

Using uptime we can display load average, for the past 1, 5 and 15 min respectively

Load average: it is the average number of process that are either in runnable (using CPU, waiting to use CPU) or uninterruptible (waiting for IO access)

IO : Input output

**Zero (0) : is no load and 1 : is fully loaded**

Load average displays average for the past 1, 5 and 15 min respectively

uptime: this is used to check how long the system was up en running

**Note:**

Linux flavour

1. centos/ Redhat
2. ubuntu/Debian

Package manager is used for

1. centos/ Redhat – yum
2. ubuntu/Debian – apt

Port

1. ssh and scp – uses a port number 22
2. telnet – uses a port number 23
3. http -- uses a port number 80
4. https -- uses a port number 443
5. dns -- uses a port number 53

**Day 13: Unix Commands**

**telnet:**

It is used to check the port number of the remote server is available or not.

syntax:

telnet ipaddress portnumber

telnet 172.32.42.2 22

**netstat:**

This is used to check the listing/used ports and available ports on the current server

syntax:

netstat -ntlpu

This will display all the used ports

netstat -t

This will display only the tcp ports

netstat -u

This will display only the udp ports

How do we check a specific port is open or not on the current server?

netstat -ntlpu | grep “portnumber”

**How do we setup password less connection between server1 and server 2?**

* run ssh-keygen command in server 1and copy the public key.
* goto the server 2 and paste the public copy in authorized\_keys file which is under /.ssh folder(cd .ssh/ --- vi authorized\_keys)
* Next time when we login, it will login without the password

**How do you login to the remote server**

using ssh command

syntax:

ssh username@servername

ssh [ubuntu@172.42.32.1](mailto:ubuntu@172.42.32.1)

**scp (Securecopy)**  
This is used to copy a file from one server to another

syntax:

scp filename username:ipaddress:path

scp file10 [ubuntu@172.42.32.1:/home/ubuntu](mailto:ubuntu@172.42.32.1:/home/ubuntu)

for directory:

syntax:

scp -r directoryname username:ipaddress:path

scp -r dir1 [ubuntu@172.42.32.1:/home/ubuntu](mailto:ubuntu@172.42.32.1:/home/ubuntu)

**rsync**: it is also used to copy a file/directory from one server to another server

syntax:

rsync file10 [ubuntu@172.42.32.1:/home/ubuntu](mailto:ubuntu@172.42.32.1:/home/ubuntu)

rsync -r dir1 [ubuntu@172.42.32.1:/home/ubuntu](mailto:ubuntu@172.42.32.1:/home/ubuntu)

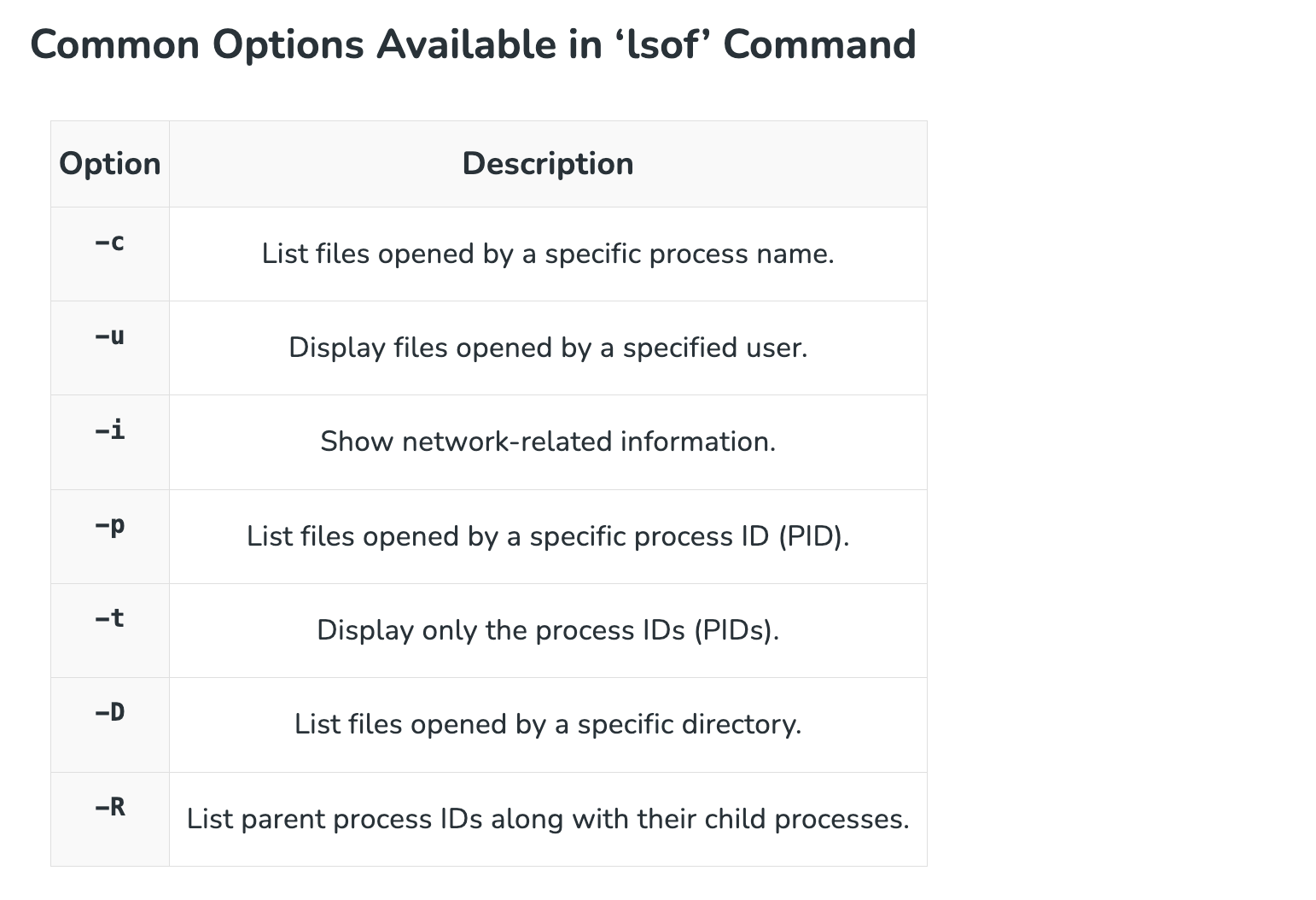
**Differecne between scp and rsync**

While copying a data from one server to another server, if copying stopped in between due to some network issue once system is back if I use scp command it will start copying from beginning

If I use rsync command it will start copying from where it was stopped Uptime:

**lsof:**

lsof command stands for List Open Files



List the files used by a specific process

lsof -p pid(process id)

List the open files for a specific user

lsof -i -u usernames

**i is not not case insensitive option here**

whoami:

This is used to check the user who is logged in

who:

This will display the users logged into the system

**Whereis** :

it will display the executable files and source files of the command where it is present

Example:

whereis ls

whereis cp

**How do we check number of users created in the server?**

cat /etc/passwd

cat /etc/passwd | wc -c

**date:**

This will display the current system date and time

syntax:

date

date ‘+%D’: it will display date in mm/dd/yy

date ‘+%d’: it will display only date

date ‘+%A’: it will display day

date ‘+%a’: it will display the first 3 letters of specific day

date ‘+%M’: it will display minutes

date ‘+%m’: it will display month in number(11 for november)

date ‘+%h’: it will display month name

date ‘+%H’: it will display hours

date ‘+%S’: it will display Seconds

date ‘+%T’: it will display time in the 24h format

**Day 14: Unix Commands**

**Gracefull stop:**

sudo service servicename stop

sudo service servicename start

sudo service servicename restart

sudo service servicename status

sudo systemctl stop servicename

sudo systemctl start servicename

sudo systemctl status servicename

sudo systemctl restart servicename

**how to do you execute the command in background?**

using & at the end of the command

example:

ls &

**adduser:**

This is used to create a new user

syntx: sudo addduser username

sudo adduser Phani

password: aws

**how do we delete the user?**

sudo deluser username

**how do you set a password for the user?**

sudo passwd username

**how to remove the password**

sudo passed -d username

how do you check which process is taking more cpu usage?

top

htop

**nohup (no hang up)**

While running a script or command, if executions stop due to some network issues, once the network is back it resumes the executions from where it was stopped.

syntax:

nohup commandname &

example:

nohup ls &

**how do you download the file from the url?**

using wget

syntax:

wget url

A screenshot of a computer screen

AI-generated content may be incorrect.

**Installing Packages**

**How to we install pacakges?**

Syntax

Packagemanager install package name

For ubuntu = apt install packagename

For centos/redhat = yum m install packagename

To install packages we require admin access or sudo permissions

Ex: sudo apt install netstat

sudo apt list –installed

This will show all the list of packages installed on the server

**How do you check whether a specific package is installed or not?**

sudo apt list –installed| grep -i “name”

sudo apt list --installed| grep -i "maven"

**how do we remove the package?**

syntax: pacakagemanager remove packagename

example:

sudo apt remove maven

**how to unzip the file?**

tar -xvf zipfilename

example:

tar -xvf apache-tomcat-9.0.97.tar.gz

how to create a zip file?

tar -cvf zipfilename filename

example:

tar -cvf devops.tar.gz devops/

tar -cvf student.tar.gz studentsdata

How do you check server/host is reachable or not?

**ping command**

syntax:

ping servername

ping ipaddress

**example:**

ping google.com

**curl**

This is used to fetch the data from the URL

**syntax:**

curl url

**example:**

curl google.com

This is also used to download the files from the url

syntax:

curl -o filename url

example

curl -o apache-tomact.tar.gz <https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.97/bin/apache-tomcat-9.0.97.tar.gz>

This is also used to upload the files

syntax

curl -T filename url

Here filename is uploaded to the URL provided in the curl command

**sleep**

this is used to pause the execution for some time

syntax:

sleep NUMBER[SUFFIX]

* **“NUMBER”** represents the time duration for which the command should sleep.
* **“SUFFIX”** can be used to specify the unit of time (s for seconds, m for minutes, h for hours, etc.).
* **Note:**If no suffix is provided, the default unit is seconds.

example:

sleep 10

sleep 3m

sleep 3h

sleep 3.5

A screen shot of a computer

Description automatically generated

**nslookup**

This is used to fetch the server details using the domain the name

syntax:

nslookup domianname

A screenshot of a computer

AI-generated content may be incorrect.