Aim and to do(s)

Tuesday, January 10, 2023 12:45 PM

link
Link to lab
TEMPLATE FILE PROVIDED-

```
#!/bin/bash

# This checks if the number of arguments is correct
# If the number of arguments is incorrect ( $# != 2) print error message and
exit
if [[ $# != 2 ]]
then
  echo "backup.sh target_directory_name destination_directory_name"
  exit
fi

# This checks if argument 1 and argument 2 are valid directory paths
if [[ ! -d $1 ]] || [[ ! -d $2 ]]
then
  echo "Invalid directory path provided"
  exit
fi

# [TASK 1]
targetDirectory=
destinationDirectory=

# [TASK 2]
echo ""
echo ""

# [TASK 3]
currentTS=``

# [TASK 4]
backupFileName=""

# We're going to:
  # 1: Go into the target directory
  # 2: Create the backup file
  # 3: Move the backup file to the destination directory

# To make things easier, we will define some useful variables...

# [TASK 5]
origAbsPath=``

# [TASK 6]
cd # <-
destDirAbsPath=``

# [TASK 7]
```

```
cd # <-
cd # <-

# [TASK 8]
yesterdayTS=

declare -a toBackup

for file in $() # [TASK 9]
do
  # [TASK 10]
  if (())
  then
    # [TASK 11]
  fi
done

# [TASK 12]

# [TASK 13]

# Congratulations! You completed the final project for this course!
```

1. **TASK 0**
   ~$ wget `https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/Final%20Project/backup.sh`

2. **Calling script at end**
   ~$ ./backup.sh important-documents .
   Argument 1 is important-documents folder
   Argument 2 is current working directory"."
3. Schedule to run the main program script every 2 minutes
   Write this cronjob in crontab file
   */2 * * * * **/usr/local/bin/backup.sh /home/project/important-documents /home/project**
4. Using `crontab`, schedule your `/usr/local/bin/backup.sh` script to backup the `important-documents` folder every 24 hours to the directory (`/home/project`)
   .tar.gz is equivalent to .tgz
   0 0 * * * tar –c –z –v –f /home/project/iportant-ocuents.tgz important-documents
   Using script we created
   0 0 * * * /usr/local/bin/backup.sh important-documents /home/project

conditionals

Tuesday, January 10, 2023 12:48 PM

Shell Script: Conditionals

This reading will get you sufficiently familiar with bash *conditionals* for the final project.

Conditionals are ways of telling a script to do something *under specific condition(s)*.

In this reading, you will learn about shell script conditionals using `if else`.

If

Syntax:

```
if [ condition ]
then
    statement
fi
```

You must always put spaces around your conditions in the `[ ]`.

Every `if` condition block must be paired with a `fi`.

Example

```
$ cat if_example.sh
a=1
b=2
if [ $a -lt $b ]
then
    echo "a is less than b"
fi
```

```
$ sh if_example.sh  # sh tells the terminal to run the script if_example.sh using the default shell
a is less than b
```

If-Else

Syntax:

```
if [ condition ]
then
    statement_1
else
    statement_2
fi
```

You don't use `then` for `else` cases.

Example

```
$ cat if_else_example.sh
a=3
```

```
b=2
if [ $a -lt $b ]
then
    echo "a is less than b"
else
    echo "a is greater than or equal to b"
fi
```

```
$ sh if_else_example.sh
a is greater than or equal to b
```

Elif

The statement `elif` means "else if":

Syntax:

```
if [ condition_1 ]
then
    statement_1
elif [ condition_2 ]
then
    statement_2
fi
```

Example

```
$ cat elif_example.sh
a=2
b=2
if [ $a -lt $b ]
then
    echo "a is less than b"
elif [ $a == $b ]
then
    echo "a is equal to b"
else # Here a is not <= b, so a > b
    echo "a is greater than b"
fi
```

```
$ sh elif_example.sh
a is equal to b
```

Nested Ifs

As in other prgramming languages, it's also possible to nest if-statements.

Syntax:

```
if [ condition_1 ]
then
    statement_1
elif [ condition_2 ]
    statement_2
    if [ condition_2.1 ]
    then
        statement_2.1
```

```
    fi
else
    statement_3
fi
```

Example

```
$ cat nested_ifs_example.sh
a=3
b=3
c=3
if [ $a == $b ]
then
    if [ $a == $c ]
    then
        if [ $b == $c ]
        then
            echo "a, b, and c are equal"
        fi
    fi
else
    echo "the three variables are not equal"
fi

$ sh nested_ifs_example.sh
a, b, and c are equal
```

Alternatively, this example could have been simplified to a single if-statement:

```
a=3
b=3
c=3
if [ $a == $b ] && [ $a == $c ] && [ $b == $c ]
then
    echo "a, b, and c are equal"
else
    echo "the three variables are not equal"
fi
```

`&&` means "and"

Bonus: "test"

Sometimes, instead of using brackets around conditions, you'll see the `test` command in use:

Example

```
$ cat test_example.sh
a=1
b=2
if test $a -lt $b
then
    echo "a is less than b"
fi

$ sh test_example.sh
a is less than b
```

`test` and `[ ]` are the same command. We encourage using `[ ]` instead as it's more readable.

Practice and testing

Tuesday, January 10, 2023 12:55 PM

1. Write a script which checks if 1st value given is less or not than 2nd value
   ~$ echo '#1/bin/sh' > file.sh
   ~$ echo 'if [$1 –lt $2]' >> file.sh
   ~$ echo "then" >> file.sh
   ~$ echo 'echo "$1 is less than $2"' >> file.sh
   ~$ echo 'else' >> file.sh
   ~$ echo 'echo "$1 is greater than $2"' >> file.sh
   ~$ echo 'fi' >> file.sh
   Run the script
   ~$ sh file.sh 4 7
   Or ~$ ./file.sh 7 3
2. Find if 1st number is greatest out of 3 numbers using elif
   echo '#1/bin/sh' > file.sh
   ~$ echo 'if [$2 –lt $1]' >> file.sh
   ~$ echo "then" >> file.sh
   ~$ echo 'echo "$2 is less than $1"' >> file.sh
   ~$ echo 'elif [$3 –lt $1]' >> file.sh
   ~$ echo 'then' >> file.sh
   `$ echo 'echo "$3 is less than $1"' >> file.sh
   ~$ echo 'else' >> file.sh
   ~$ echo 'echo "$1 is the smallest"' >> file.sh
   ~$ echo 'fi' >> file.sh
3. Find greatest out of 3 numbers using && operator
   ~$ echo '#!/bin/sh' > file.sh
   ~$ echo 'if [$1 –lt $2] && [$3 –lt $2]' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$2 is the greatest"' >> file.sh
   ~$ echo 'elif [$2 –lt $1] && [$3 –lt $1]' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$1 is the greatest"' >> file.sh
   ~$ echo 'else' >> file.sh
   ~$ echo 'echo "$3 is the greatest"' >> file.sh
   ~$ echo 'fi' >> file.sh
4. Write the same above code using test command
   ~$ echo '#!/bin/sh' > file.sh
   ~$ echo 'if test $1 –lt $2 && test $3 –lt $2 ' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$2 is the greatest"' >> file.sh
   ~$ echo 'elif test $2 –lt $1 && test $3 –lt $1 ' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$1 is the greatest"' >> file.sh
   ~$ echo 'else' >> file.sh
   ~$ echo 'echo "$3 is the greatest"' >> file.sh
   ~$ echo 'fi' >> file.sh
5. Write same code using exit after statements of every if statement-exit is like "break" in java
   ~$ echo '#!/bin/sh' > file.sh
   ~$ echo 'if test $1 –lt $2 && test $3 –lt $2 ' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$2 is the greatest"' >> file.sh
   ~$ echo 'exit' >> file.sh
   ~$ echo 'elif test $2 –lt $1 && test $3 –lt $1 ' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$1 is the greatest"' >> file.sh
   ~$ echo 'exit' >> file.sh
   ~$ echo 'else' >> file.sh

~$ echo 'echo "$3 is the greatest"' >> file.sh
~$ echo 'exit' >> file.sh
~$ echo 'fi' >> file.sh

6. Find greatest out of 3 numbers using nested if statement
   ~$ echo '#!/bin/sh' > file.sh
   ~$ echo 'if [$1 –lt $2]' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'if [$3 –lt $2]' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$2 is the greatest"' >> file.sh
   ~$ echo 'else' >> file.sh
   ~$ echo 'echo "$3 is the greatest"' >> file.sh
   ~$ echo 'fi' >> file.sh
   ~$ echo 'else' >> file.sh
   ~$ echo 'if [$3 –lt $1]' >> file.sh
   ~$ echo 'then' >> file.sh
   ~$ echo 'echo "$1 is the greatest"' >> file.sh
   ~$ echo 'else' >> file.sh
   ~$ echo 'echo "$3 is the greatest"' >> file.sh
   ~$ echo 'fi' >> file.sh
   ~$ echo 'fi' >> file.sh

7. CHECK IF NUMBER OF ARGUMENTS TYPED BY COMMAND LINE ARGUMENT PROMPT IS 2 OR 3
   ~$ if [$# ==2] || [$# ==3]
8. Check if number of arguments is not equal to 3 and less than 7
   ~$ if [$# !=3] && [$# < 7]

9. Check if 2 variables contain valid directory paths
   ~$ if [-d $1] && [-d $2]
10. Check if one variable contains a valid directory path and another contains an invalid file path
    ~$ if [-d $1] && [! -f $2]
11. Get current timestamp in seconds
    ~$ date "+%s"
12. Store current timestamp in a variable and store in a script
    ~$ echo 'current=$(date "+%T")' >> file.sh
13. **Concat a variable's value to a string and store the new string in another variable(in script)**
    ~$ echo 'new_var='welcome to [$current] th anniversary'' >> file.sh
14. Define a variable called `backupFileName` to store the name of the archived and compressed backup file that the script will create.
    The variable `backupFileName` should have the value `"backup-[$currentTS].tar.gz"`
    a. For example, if `currentTS` has the value `1634571345`, then `backupFileName` should have the value `backup-1634571345.tar.gz`.

    ~$ echo 'backupFileName='backup-$currentTS.tar.gz'' >> file.sh
15. Define a variable called `origAbsPath` with the absolute path of the current directory as the variable' s value.
    ~$ echo "origAbsPath=$(pwd)" >> file.sh
16. Define a variable called `destAbsPath` with value equal to the absolute path of the destination directory.
    Click here for Hint

    *First use `cd` to go to `destinationDirectory`, and then use the same method you used in **Task 5***

    PreviousNext
    ~$ echo "destAbsPath=\$(cd $destinationDirectory;pwd)" >> file.sh
17. Change directories from the current working directory to the target directory `targetDirectory`.
    Click here for Hint

    `cd` into the original directory `origAbsPath` and then `cd` into `targetDirectory`.

    PreviousNext
    ~$ echo "\$(cd $origAbsPath;cd $targetDirectory)" >> file.sh
18. Define a numerical variable called `yesterdayTS` as the timestamp (in seconds) 24 hours prior to the current

timestamp, `currentTS`
Click here for Hint

Math can be done using `$(())`; for example: `zero=$((3 * 5 - 6 - 9))`

Thus, to get the timestamp in seconds of 24 hours in the future, you would use:

a. `tomorrowTS=$(($currentTS + 24 * 60 * 60))`

PreviousNext

~$ echo "yesterdayTS=\$(($currentTS-24*60*60))" >> file.sh

19. **Declare an array in bash**
    ~$ declare –a my_array1
    Here my_array1 is an array we have declared
20. **Add value 15 and variable hello1 to this array**
    ~$ my_array1+=15
    ~$ my_array1+=$hello1
21. # ARRAY RULES
    Lets declare array
    ~$ declare –a myarray

- **When we add an element to array without encapsulating it in parenthesis,then it is appended as a string to the first element of the array**
  ~$ myarray+=18
  ~$myarray+=17
  ~$myarray+=23
  Contents of myarray-
  181723 at index 0
- **When we add element after encapsulating it in parenthesis,then it is added to the next available index in the array(it is like appending in arraylist of java)**
  ~$myarray+=("is")
  ~$myarray+=(66)
  ~$myarray+=88
  Contents of myarray-
  18172388 is 66
  Index =0 1 2
- **When printing array using echo, if we just write $<arrayname> or ${arrayname} then only 1st element of array is printed**
  ~$ echo $myarray or ~$ echo ${myarray[0]} or ~$ echo ${myarray}
  Output:
  18172388
- **To print the whole array with spaces b/w consecutive elements, we write ${arrayname[@]}**
  ~$ echo ${myarray[@]}
  Output:
  18172388 is 66
- **To print n th element of array, we write ${arrayname[n]}**
  ~$ echo ${myarray[2]}
  Output:
  66

22. # For and for each loop rules
    ~$ for File in $(get many files)
    ~$ do
    ~$ run many things done on one file we get
    ~$ done
    For <one thing> in <many things of same category>
    Do
    Statements
    Done
23. Store last modified date of a file in a variable
    ~$ date –r ./backup.sh
24. Store last modified date of a file in seconds in one variable whose path is stored in another variable
    ~$ another_variable=$(date –r $one_variable +%s)
25. Inside the for loop, you want to check whether the `$file` was modified within the last 24 hours.
    To get the last-modified date of a file in seconds, use `date -r $file +%s`

Then compare the value to `yesterdayTS`

a. Idea: `if [[ $file_last_modified_date > $yesterdayTS ]]` then the file was updated within the last 24 hours! In the `if-then` statement, add the `$file` that was updated in the past 24-hours to the `toBackup` array.

```
for var1 in (ls)
do
if [ $yesterdayTS -lt $(date -r $var1 +%s) ] or write if(($(date –r $var1 +%s) > $yesterdayTS))
then
toBackup+=($var1)
fi
done
```

26. Archive and compress an array containing files to another file
    Lets say the name of final file exists in a variable called new_file
    New_file="archivecompress.tar.gz"
    Lets say array name is new_array
    -v is for verbosely telling us which files are getting compressed and archived
    ~$ tar –c –z –v –f $new_file ${new_array[@]}
27. Unzip a zipped folder/file and not modify the timestamp on it
    ~$ unzip –DDo <filename>.zip
28. Modify the timestamp after unzipping the file
    ~$ unzip <filename>.zip
29. Update timestamp of all files which are in folder my_f
    ~$ touch my_f/*
    /* means that go in my_f folder and update timestamp of all files there
30. Update all files' timestamps in my_f folder which end with .page extension
    ~$ touch my_f/*.page
31. Copy a file to /usr/local/bin directory using sudo command(when we don't have administrator permissions,we can't use sudo-otherwise we can)
    ~$ sudo cp backup.sh /usr/local/bin
32. Explicitly start crontab service in a virtual lab environment like theia
    ~$ sudo service cron start
33. Stop cron service
    ~$ sudo service cron stop
34. run a backup of all your user accounts
    # at 5 a.m every week on monday
    In crontab-
    0 5 * * 1 tar /var/backups/home.tgz /home/

applications

Tuesday, January 10, 2023 1:09 PM

```
theia@theia-vardaan12345:/home/project$ a=5;
theia@theia-vardaan12345:/home/project$ b=6;
theia@theia-vardaan12345:/home/project$ echo "$a"
5
theia@theia-vardaan12345:/home/project$ echo '#1/bin/sh' > file.sh
theia@theia-vardaan12345:/home/project$ echo 'if [ $1 -lt $2 ]' >> file.sh
theia@theia-vardaan12345:/home/project$ echo "then" >> file.sh
theia@theia-vardaan12345:/home/project$ cat file.sh
#1/bin/sh
if [ $1 -lt $2 ]
then
theia@theia-vardaan12345:/home/project$ echo 'echo "$1 is less than $2"' >> file.sh
theia@theia-vardaan12345:/home/project$ echo 'else' >> file.sh
theia@theia-vardaan12345:/home/project$ echo 'echo "$1 is greater than $2"' >> file.sh
theia@theia-vardaan12345:/home/project$ echo 'fi' >> file.sh
theia@theia-vardaan12345:/home/project$ cat file.sh
#1/bin/sh
if [ $1 -lt $2 ]
then
echo "$1 is less than $2"
else
echo "$1 is greater than $2"
fi
theia@theia-vardaan12345:/home/project$ ./file.sh 4 7
bash: ./file.sh: Permission denied
theia@theia-vardaan12345:/home/project$ chmod +x file.sh
theia@theia-vardaan12345:/home/project$ ./file.sh 4 7
./file.sh: line 2: [: -lt: binary operator expected
4 is greater than 7
theia@theia-vardaan12345:/home/project$ sh file.sh 3 1
file.sh: 2: [: 3: unexpected operator
3 is greater than 1
theia@theia-vardaan12345:/home/project$
```

```
theia@theiadocker-vardaan12345:/home/project$ sudo service cron start
 * Starting periodic command scheduler cron
   ...done.
theia@theiadocker-vardaan12345:/home/project$
```

Intro to linux,unix

Tuesday, January 3, 2023 4:35 PM

Unix-family of OS
Examples of unix based OS-
Oracle solaris,freBSD,HP-UX,IBM AIX,Apple macOS
Linux-family of Unix like OS
Linux Features-

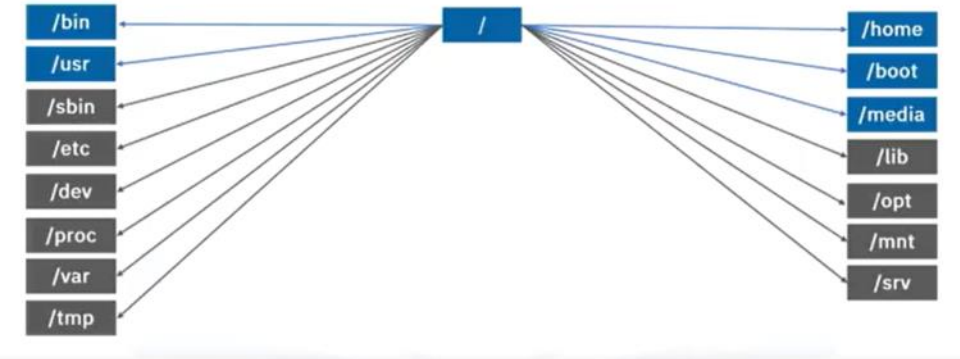
- Free,open source
- Secure
- Multitasking
- Multi-user
- Portability

Kernel-core component of OS which enables components to communicate with OS's hardware
LINUX ARCHITECTURE

- **User**-communicates with **application** via system daemons,shells,user apps,and tools
- **Application**:works with **OS(**applications like programming languages,system tools,shells,user apps like browsers,text editors
- **OS-**built on top of linux **kernel**
- **Kernel-**interacts with **hardware** layer(kernel is lowest level software in linux-it has complete control over linux
- Kernel starts when computer is booted and remains in computer's memory

JOBS OF KERNEL

LINUX FILESYSTEM-contains all files needed to run linux and contains all data
It begins at root directory**(/)**
Below root directory is the tree like structure of all files and folders
/bin directory-HOLDS USER BINARY FILES(DIRECTLY BELOW ROOT DIRECTORY)
Other directories-/usr and /home and /boot(contains system/s boot files) and /media(contains temperory media files like CD or USB files)



LINUX DISTRIBUTIONS-

Also called distro-they are specific flavors of linux OS
All distro(s) use the linux kernel

Different distro(s) examples-
DEBIAN-open source,stable,reliable[it is core-not derived from another linux distros]
UBUNTU(based or built on top of debian) [it is not a core distro]
3 versions of ubuntu-

- Ubuntu desktop
- Ubuntu server
- Ubuntu core(for IOT)

RED HAT –core distro(shipped as red hat enterprise linux)
FEDORA –supports many architectures
SUSE LINUX ENTERPRISE(SLE)-supports many architectures like arm and raspberry pi
2 versions-

- **Server[SLES]**
- **Desktop[SLED]**

It contains SUSE PACKAGE HUB-maintained by company called SUSE
ARCH LINUX-highly configurable

3. Which layer of the Linux system assigns software to users, helps detect errors, and performs file management tasks? **1 point**

- ○ Kernel
- ○ Application
- ◉ Operating system
- ○ Hardware

4. Which layer of the Linux system is responsible for memory management, process management, device driver management, and system calls and security? **1 point**

- ◉ Kernel
- ○ Operating system
- ○ Application
- ○ Hardware

6. Which of the following is a GUI-based text editor? 1 point

- (●) GNU nano
- ( ) vim
- ( ) vi
- ( ) gedit

7. Which of the following is a path that you can use to navigate to a user's home directory? 1 point

- ( ) \
- ( ) \myhome
- (●) \home
- ( ) ~

Linux terminal overview

Tuesday, January 3, 2023 6:21 PM

Linux shell is an OS level application which interprets linux commands
Shell versions-

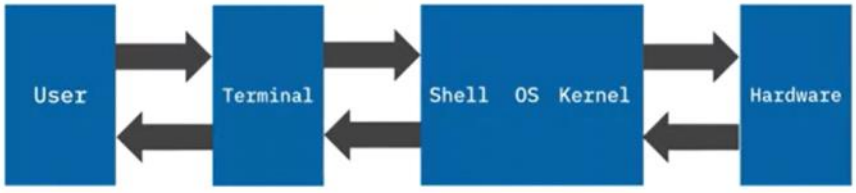
- Bash
- Zsh

TERMINAL-any application we can use to interact with linux shell-we enter commands in it and receive output from it

Special paths-

1. Home directory ~
2. Root directory /
3. Parent of current directory ..
4. Current directory .

Communicating with Linux system



Vertical line represents the command prompt-that's where we type the next character from keyboard
[note:we work on terminal,not on shell-we implement shell via a terminal,so shell is an abstract thing]
[in windows,shell is implemented by command prompt
In mac,it is implemented by terminal]

- Run a python command
  python program_name.py
- We get output in the terminal
  /home/me/Documents/ $

Represents that current working directory is Documents directory which is in me directory which lies in home directory
Current working directory-the location where shell will look for any commands which we run in the terminal
We write commands after $ sign

- Go to root directory
  cd /
  Previous state: /home $
  Here current working directory is home but we wanna go to root directory
  Updated state: / $

Now current working directory is root directory

- Go to bin directory
  cd bin
  Previous state: /home $
  Updated state: /bin $ now current working directory is bin directory
- Run program ls inside current directory-this command displays names of all files within the current directory
  ./ls
- Going to home directory
  cd ~
  Previous state: /home $
  Updated state: /home/me $
  Now running ls command from this current working directory
  /home/me $ ls
  [note :we didn't write ./ls this time as we already were at home directory]
- **CHANGING THE CURRENT WORKING DIRECTORY**
  Start at /home $
  Change current working directory to parent of current working directory
  /home $ cd..
  Updated state:home directory's parent is the root directory
  / $
- Going to media directory inside root directory
  / $ cd/media
- Going to my-usb-drive directory inside media directory
  / $ cd/media/my-usb-drive
- Navigating to the media directory and then to root directory from my-usb-drive directory
  /media/my-usb-drive $ cd../..
  States to go one level up and then go to root directory
- Going to Documents directory right from my-usb-drive directory
  /media/my-usb-drive $ cd../../home/me/Documents
  Current state: /home/me/Documents $
- If some python program exists in /home/me directory,then we can run it from documents directory by-
  /home/me/Documents $ cd..
  Going one level up to /home/me directory
  Current state: /home/me $
  Now run the python file
  /home/me $ python ./program_name.py
  ./program_name.py calls and runs the python file from the current directory as "." Represents current working directory

LINUX COMMAND PRACTICE

Tuesday, January 3, 2023 8:19 PM

Link to code

If current directory is home ~
You can run a previous command, by pressing the up arrow key
If u press UP arrow 3 times,it would place the command u entered 3 commands before the current command

1. Go to pictures directory
   ~ $ cd Pictures/
2. Go to documents python-examples folder
   ~ $ cd Documents/python-examples/
3. Going to home directory from home directory-nothing actually happens
   ~ $ cd ~ or write ~ $ cd
4. Going to python-examples folder by first going to home directory
   ~ $ cd ~/Documents/python-examples
5. 2 ways of running python program file from python-examples folder

- ~/Documents/python-examples $ python program_name.py
  or write ~/Documents/python-examples $ python3 program_name.py
- ~/Docuements/python-examples $ python ./program_name.py
  Or write ~/Documents/python-examples $ python3 ./program_name.py

6. Going to root directory(not home directory) from current working directory(which is ~/Documents/python-examples
   ~/Documents/python-examples $ cd /
   Updated status: / $

TEXT EDITORS

Tuesday, January 3, 2023 8:56 PM

2 categories-

- Command line text editors
1. GNU nano
2. Vi
3. vim

- GUI based text editors
  Gedit
- Both modes
  Emacs

  Python package managers use package managers such as pip and conda

COMMANDS-------

OPENING A TEXT FILE IN nano
nano <filename>
This opens a new nano text editor window in which we can edit
OPEN VI TEXT EDITOR APP VIA COMMAND LINE
vim
OPEN VIM TO OPEN A FILE
vim <filename>

2 modes of vim-

- Insert mode:where we enter text----after opening vim, type -
  i to enter insert mode
  Press esc key to exit insert mode and go into command mode
  Then the text we typed is written to the buffer at the cursor's current location
- Command mode:where we do everything else

I. Saving file in command mode
:sav <filename.txt>
By doing this the buffer is written to the file and it is shown that a file has been saved with some lines of code in it

II. Write and make changes to an existing file
:w <filename.txt>

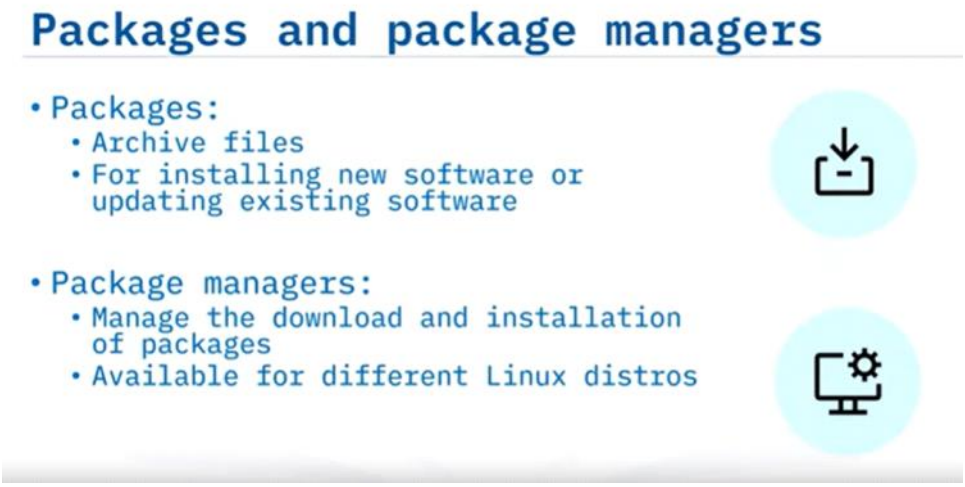
III. QUIT VIM SESSION AFTER SAVING FILE
:q

IV. Quit vim session without saving file after the last saved code
:q!

V. Convert a package from one format to another using alien

- Convert .deb to .rpm
  alien –r <package_name> .deb
- Convert .rpm to .deb
  alien <package_name>.rpm
- Update in apt update manager
  sudo apt update
  It checks for any updates and if there are any,it tells us but doesn't update automatically
- Manually update/upgrade
  sudo apt upgrade

- Update in yum update manager
  sudo yum update
- Installing new package/software on a deb based system
  Use apt
  sudo apt install <package_name>
- Installing new package in rpm based system
  Use yum
  sudo yum install <package_name>
- Installing pandas library in python using pip
  pip install pandas



EXAMPLES OF GUI BASED LINUX DISTRO PACKAGE MANAGERS

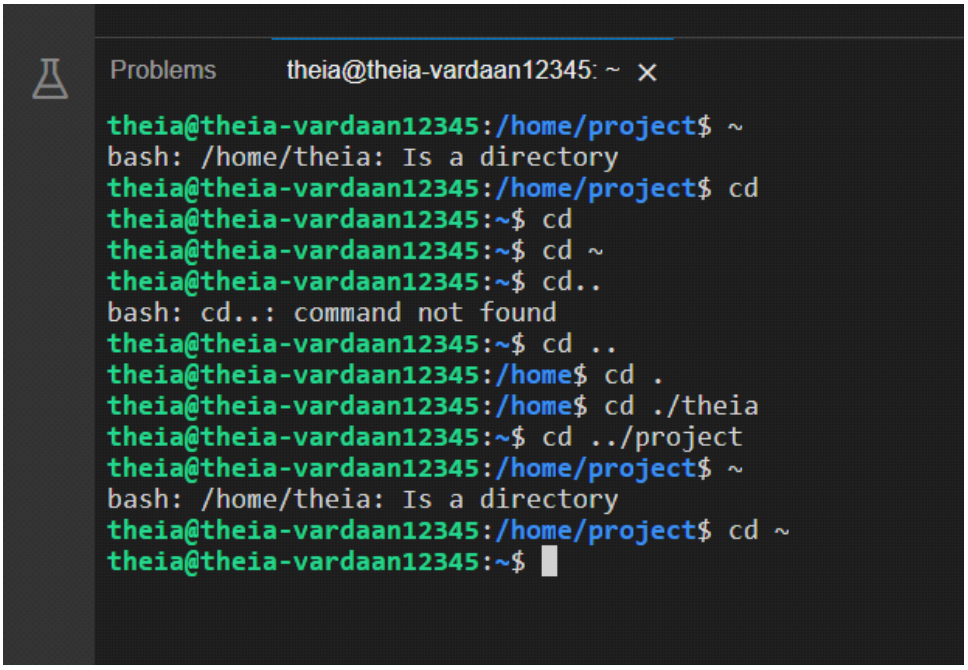
- **PackageKit**
- **Update Manager(**for updating deb based linux systems)
  Command line tool for updating deb based linux systems:apt
  UPDATE RPM BASED LINUX SYSTEMS-USE PackageKit
  Command line tool for updating RPM BASED LINUX SYSTEMS:yum
  Fullform is yellowdog updater modified

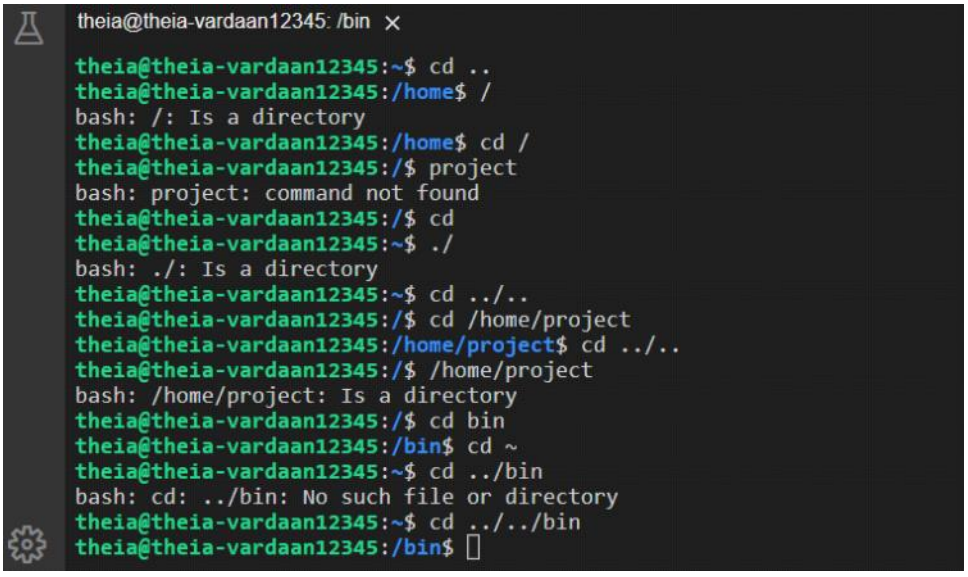
Linux labnweek 1

Tuesday, January 3, 2023 9:36 PM

instructions
Link to lab(login first)
Rule: there is no space between current working directory's location and $ sign
Correct: /home/theia$
Wrong: /home/theia $
We are using theia------an IDE which works on cloud as well as desktop
Lets say our current working directory is /home/project
User's home directory is /home/theia == ~

1. Change current working directory to home directory
   /home/project $ cd ~
   Method 2---
   /home/project $ cd
2. Find the home directory's location-write location of directory u want to find without preceeding it with cd
   /home/project $ ~
   Output:
   bash: /home/theia: Is a directory
3. If we are at /home/theia and want to go to /home
   /home/theia $ cd ..
4. Now go to /home/project
   /home $ cd project
5. Now go to root directory
   /home/project $ cd /
6. Find what is root directory from /home/project-write location of directory u want to find without preceeding it with cd
   /home/project $ /
7. Go to home directory from root directory
   / $ cd
8. Go to project directory from /home/theia
   /home/theia $ cd ../project or ~ $ cd ../project
9. Go to root directory from /home/theia or home directory
   ~$ cd ../..
   First go one level up and then again go one level up
10. Go to bin directory
    /$ bin
11. Go to home directory
    /bin$ cd ../home/theia or /bin$ ~
12. Go to bin again from home directory
    ~$ cd ../../bin
    Go one level up to /home
    Then go one level up till root
    Then go one level down till bin





BROWSING DIRECTORIES USING ls command

Lets say we are in /bin directory

1. See all files in current directory
   /bin$ ls
2. Go to /home/project and use ls there-we can't go to a location and use ls there in one command
   First in one command go to location,then in another command use ls on that current working directory or use ls on a specific location without actually going there like in point 3
   /bin$ cd ../home/project
   /home/project$ ls
   Nothing is returned as /home/project is empty directory

3. See contents of root directory from /home/project directory
   /home/project$ ls /
   We just see contents of root directory-our current working directory still remains /home/project
4. Go to sbin directory and see its contents
   /home/project$ cd /sbin or /home/project$ cd ../../sbin
   Go directly to root directory and then to sbin inside it
   /sbin$ ls
5. Find contents of /bin
   /sbin$ ls /bin
   One of the files is "ls" as "ls" runs by /bin/ls

UPDATING AND INSTALLING SOFTWARE/PACKAGE-use sudo command
Nano command-allows us to use terminal as a simple text editor
Vim text editor also does the same

- Find if any updates are there to be done
  ~$ sudo apt update
  Tells how many packages need updates
- See list of packages which need updates
  ~$ apt list –upgradable or ~$ sudo apt list --upgradable
- Actually update the packages
  ~$ sudo apt upgrade
- Upgrade nano
  ~$ sudo apt upgrade nano
  It will ask [Y/n]
  We use capital Y and not small y because Y is default-for saying no we have to press n and then enter but for saying yes we can either press y and then enter or just press enter
  That's why Y is capital as to enable it we don't need to explicitly press y before pressing enter
- Install nano
  ~$ sudo apt install nano
- Install vim
  ~$ sudo apt install vim

CREATING AND EDITING FILES USING NANO

a. Creating a new python file using nano and storing it
   We can't go to a particular working directory and run a python file in it in one command just the same rule for using ls
   ~$ nano <filename>.py
   This creates a different window of nano text editor in which we can write python code
   To exit,press ctrl+X
   Confirm file name
   Now look again at updated contents of root directory-we will find the python file now there
b. Running a file inside current working directory
   ~$ python <filename>.py or ~$ python3 <filename>.py or ~$ python2 <filename>.py
   Output of file is shown in next line

QUESTIONS
U are at /home directory

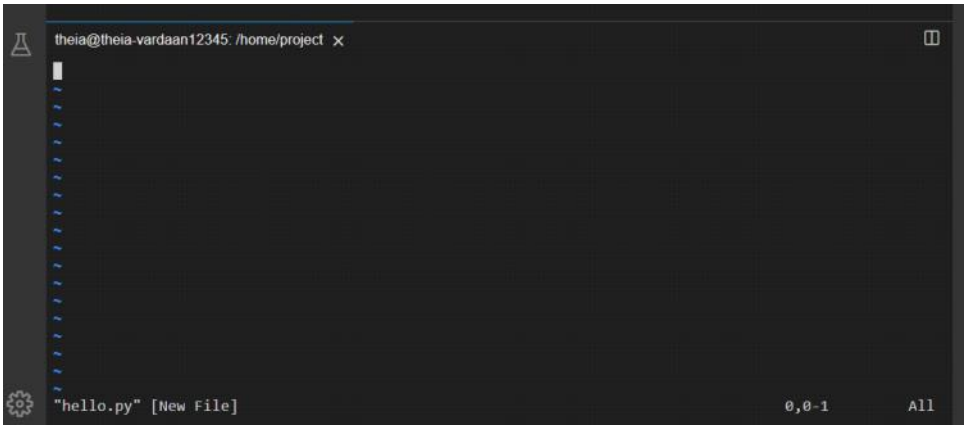
a. FIND CONTENTS OF /usr
   /home$ ls ../usr
b. Go to /media
   /home$ cd ../media or /home$ cd /media

c. Go to /home/project directory
/media$ cd ~/../project
First go to home directory which is /home/theia
Then go to /home
Then go to project folder in it

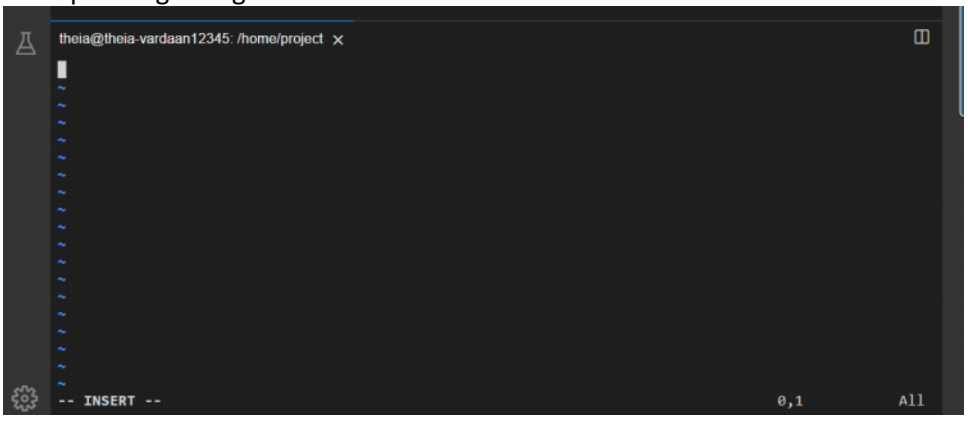
d. Now open pr.py file again
/home/project$ nano pr.py
Again the text editor with previously saved code is opened but we can't edit the previously saved text-we can only add extra text but can't change anything which was previously there

e. Use vim to create a python file,write on it,save it
In terminal
~$ vi <filename>.py
Then in vim text editor
Type "i" to enter insert mode
Then type the python code
Then press esc to enter command mode
Then write this to save the file
:wq
Press enter
Now in terminal run the file
~$ python3 <filename>.py

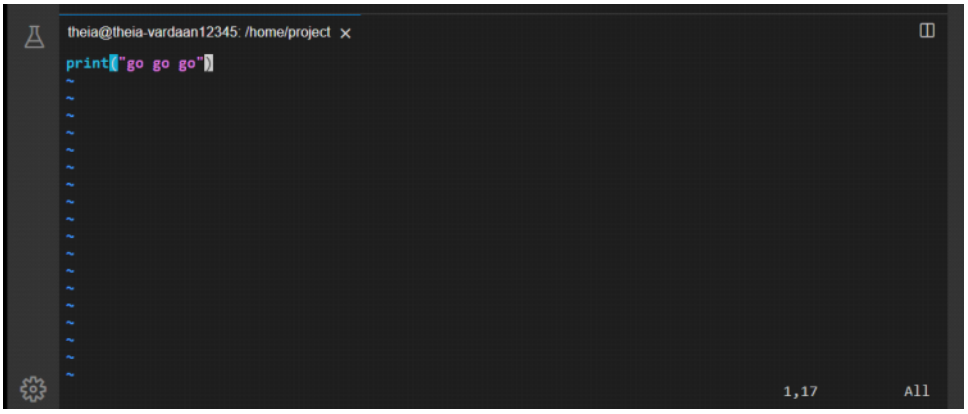
f. Find version of nano
~$ nano --version



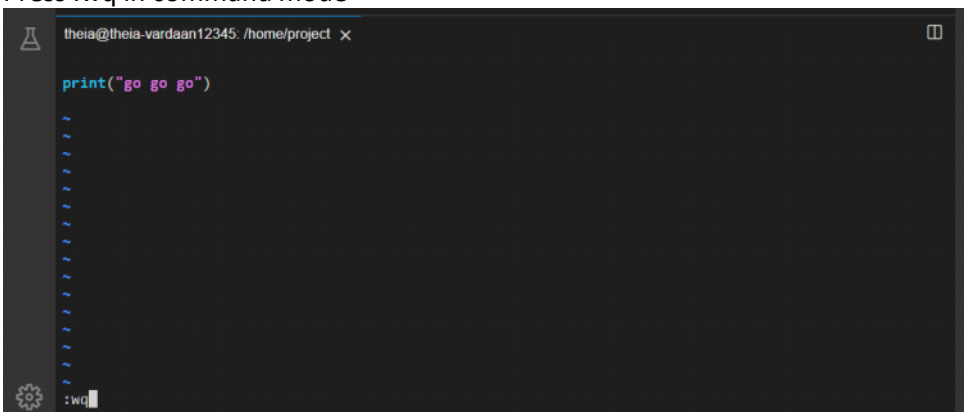
After pressing I we go into insert mode



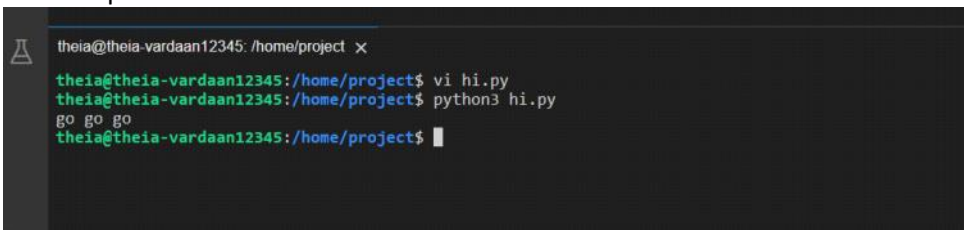
Type python code
Then press escape to go into command mode



Press :wq in command mode



Final output



Intro to scripting languages

Sunday, January 8, 2023 5:33 PM

1. Form of shebang directive
   #!interpreter [optional argument]
2. Invoke bourne shell(called sh shell) from bin directory
   #!/bin/sh
3. Invoke bourne again shell(bash) from bin directory
   #!/bin/bash
4. Create a python directive
   #!/usr/bin/env python3
   Or #!/usr/bin/env python2
5. Run a simple program using shell script
   From the command prompt or terminal,create a simple shell script
   ".sh" is an extension which tells that a file is a shell script
   ~$ touch file1.sh
   Here file1 is a shell script
6. Turn a sh file(a text file) into a bash script
   We do this by echoing a bash shebang and appending that bash shebang text to our file(sh file) by using ">>"
   >> is bash output redirection operator used for appending output to a file
   ~$ echo '#! /bin/bash' >> file1.sh
   '#! /bin/bash' is called bash shebang
   '#! /bin/sh' is called sh shebang
   Now 1st line of file1.sh becomes the bash or sh shebang
   This line was not printed but rather was put or stored or written after last previously written line in file1.sh by use of ">>"
   Then use echo command to print a statement and then redirect this statement to our bash script
   ~$ echo 'hi hello bye' >> file1.sh
   Now 'hi hello bye ' line is written after previously last written line(the bash shebang) in file1.sh
   Before running bash script,we have to make it executable file
   Change executable permission by using +x

   ~$ chmod +x file1.sh
   Now run the bash script
   ~$ ./file1.sh or ~$ bash <relative path of script file to run>
7. Print a line and then ask for next command in same line
   ~$ echo –n <line>
8. Print a line and ask for next command via command prompt in next line
   Not writing –n after echo is like writing println() in java-the next line comes in next line
   ~$ echo <line>
9. Read user input and store it in a variable
   ~$ read name11
   Here name11 variable is created and it stores the string input the user gives
10. Print a variable with some words before and after it in one statement using echo and get next statement in same line(use –n after echo)

Use double quotes only(not single quotes)
~$ echo –n "welcome $name11 to the world"
We used $variable_name in b/w some words

11. Add some comments to a script
~$ echo –n '# hi hello' >> file.sh
~$ echo –n '#hi bye' >> file.sh
~$ echo 'toto' >> file.sh
What we see in file--
Hi hellohi byetoto
12. When editing script using echo command and giving another echo command with sentence in it,then use single quotes after 1st echo and use double quotes after second echo(the actual command we are putting in the file)
~$ echo 'echo "welcome to $name1 and his $number11 birthday"' >> file.sh
13. Add w permission for user and remove execute permission for other group members for a file
~$ chmod u+x o-x file.sh
14. Find out path of bash command and path of sh command
~$ which bash or ~$ which sh
15. Create a full bash script which takes a user's name and age and year of birth and print all 3
~$ which bash
~$ which sh
~$ #!/bin/bash
~$ #!/bin/sh
~$ echo '#!/bin/bash' >> file.sh
~$ echo '# my script' >> file.sh
~$ echo 'echo "give name"' >> file.sh
~$ echo 'read name1' >> file.sh
~$ echo 'echo "welcome $name1 to world"' >> file.sh
~$ echo '# next statement' >> file.sh
~$ echo 'echo "give age"' >> file.sh
~$ echo 'read age12' >> file.sh
~$ echo 'echo "your age is $age12"' >> file.sh
~$ echo '# 3rd statement' >> file.sh
~$ echo 'echo "give dob"' >> file.sh
~$ echo 'read doby' >> file.sh
~$ echo 'echo "your dob is $doby"' >> file.sh
16. If bash shebang is not the first line of script
Use ~$ bash <filemname> to run file
Otherwise use ~$ <path of file to run>
17. Take many inputs in a script during running it
~$ bash file.sh <input1> <input2>
18. Take many inputs in script(and store them in variables)
All this is written in script and not in terminal or command prompt
Name1=$1;
Age1=$3;
Date=$7;
It means store the 1st given value by user during calling of script in name1 variable and likewise store 7th given value(during calling of script) in date variable
Note:the order of given values starts from 1 and not from 0
We use $1 to access those values given by user
19. Read and store many inputs in different variables during running time of script(not at starting of script)
We write it in script
read name1 name2 name3

When user inputs some values separated by delimiter space then automatically they are stored in these variables by order
If 4 words or values are enetered,then last variable receives all the excess values as well
If we give only 2 values, then last variable stays at null value
Default delimiter which is used to separate values given by user is space

20. Chain 2 commands using pipe (shorthand for pipeline command) command
    ~$ command1 | command2
    Both commands should be filter commands
    ~$ cat file.txt | wc –l file.txt ----------correct as both wc and cat are filter commands
    ~$ cd .. | cat file.txt --------incorrect as cd command is not filter command and so we can't go to parent directory and then see contents of cat file
21. Sort the output of ls command in descending order of directories/files names
    ~$ ls | sort –r
22. See all variable definitions of a particular shell
    ~$ set
23. See just first 4 variable definitions
    ~$ set | head –n 4 or ~$ set | head –4
24. Define a new shell variable(declare and initialize)
    Write in shell script
    No spaces around "=" sign
    <name of variable>=<value of variable>;
25. Clear a variable's value(totally garbage collect the variable-we can't access it again)
    Use unset command
    unset <variable name>
26. Extend a shell variable to environment variable by extending its scope

    ~$export <shell variable name>
27. List all environment variables
    ~$ env
28. Find all environment variables which start with "Gre"
    ~$ env | grep 'gre*'
29. Find all environment variables which end with the word "Fre"
    ~$ env | grep '*fre'

PIPELINE ACTIVITIES

30. Display all unique lines of a file in reverse sorted order alphanumerically
    ~$ sort –r file.txt | uniq
    First we combine all names consecutively in reverse order and then we remove the consecutive names
31. replace all vowels in a string with underscores
    ~$ echo 'how are you my friend' | tr 'aeiou' '_'
    Ini this we give tr command a string using output from echo command and then we ask tr to replace aeiou with _ character
32. replace all consonants with an underscore
    ~$ echo 'how are u my firend' | tr –c 'aeiou' '_'
33. change all of the text to upper case of a file's contents
    ~$ cat file.txt | tr '[a-z]' '[A-Z]'
34. Sort all contents of a file in reverse order and print only unique contents with every letter changed to lowercase
    ~$ cat file.txt | sort –r | uniq | tr '[A-Z]' '[a-z]'
35. Find current usd price from a web url
    Url is https://api.coinstats.app/public/v1/coins/bitcoin\?currency\=USD

Step 1:convert contents from this file(written In json) to json object
~$ curl -s --location –request GET https://api.coinstats.app/public/v1/coins/bitcoin\?currency\=USD
Or simply ~$ curl https://api.coinstats.app/public/v1/coins/bitcoin\?currency\=USD
Step 2:we want to search for
"price": decimal value1.decimal value 2
Or "price": integer value
There can be any number of spaces b/w "price": and the first digit
For integer value,we have no "."
Digits can be b/w 0 to 9
Now find this line in the json object we created
grep -oE "\"price\"\s*:\s*[0-9]*?\.[0-9]*"

grep -oE "\"price\"\s*:\s*[0-9]*?\.[09]*"

Let's break down the details of this statement:

- `-o` tells `grep` to *only* return the matching portion
- `-E` tells `grep` to be able to use extended regex symbols such as `?`
- `\"price\"` matches the string `"price"`
- `\s*` matches any number (including 0) of whitespace (`\s`) characters
- \s means only one space and \s* means any number of spaces
- `:` matches `:`
- `[0-9]*` matches any number of digits (from 0 to 9)
- `?\.` optionally matches a `.` (this is in case price were an integer)

What if "." was not an optional argument to match(it was necessary to be there)-no ? used

36. Use pipelining for above commands
    ~$ curl -s –location –request GET https://api.coinstats.app/public/v1/coins/bitcoin\?currency\=USD | grep -oE "\"price\"\s*:\s*[0-9]*?\.[0-9]*"
    Or
    $ curl -s --location --request GET https://api.coinstats.app/public/v1/coins/bitcoin\?currency\=USD |\ grep -oE "\"price\":\s*[0-9]*?\.[0-9]*"
37. get *only* the value in the price field, and drop the "price" label, you can use chaining to pipe the same output to another `grep`:
    ~$ curl -s --location --request GET https://api.coinstats.app/public/v1/coins/bitcoin\?currency\=USD |\ grep -oE "\"price\":\s*[0-9]*?\.[0-9]*" |\ grep -oE "[0-9]*?\.[0-9]*"
38. List all files in bin directory with names starting with "bath"
    ~$ ls /bin/bath*
39. Find all files in bin directory with names starting with any character and "ash"
    So name should be of 4 characters and there can be only 26 possibilities as there is only one character which we are changing
    ~$ ls /bin/?ash
40. Print "$name hi"
    Here $name is not printing variable name's value-rather only print $name directly
    ~$ echo "\$name hi" used double quotes
    or ~$ echo '$name hi' used single quotes
    We use \$ to specify that $ is to be used as an escape character rather than a meta character which is used to print value of variable name
41. Print variable name's value
    ~$ echo "$name" use only double quotes
    If u use single quotes,then $name will be printed rather than name's value
42. Overwrite contents of a file with any error message
    ~$ garbage 2>> file.txt
    NOTE:WE CAN'T USE PIPELINING IN THIS AS garbage is not a valid command and so shell won't go to the next command until the previous command runs successfully
    ~$ garbage | 2>> file.txt ----------incorrect

Garbage is not a command and so will give an error

43. Store output of pwd command in a variable called here1
    ~$ here1=$(pwd)
    Or ~$ here1=`pwd`
44. Pass arguments while running a bash script
    ~$ ./file.bash <Argument1> <argument2>
45. Run 2 commands in bash script sequentially in batch mode
    Command1;command2;command3
    echo 'hi bye';cat file.txt
46. Run 2 commands in parallel to each other simultaneously in concurrent mode
    Command1&command2&command3
    echo 'hi bye'&cat file.txt&sort -r

ADVANCED SCRIPTING LAB

47. Find current working directory and then make a new file in batch mode
    ~$ pwd;touch file.txt
48. Show every file in bin directory
    "*" by itself means "everything"
    /bin/* means every file
    ~$ ls –ltr /bin/*
    Or ~$ ls –ltr /bin
49. lists all files whose name ends with a '.conf' in the /etc directory.

    ~$ ls –ltr /etc/*.conf
    Give full relative path of etc directory
    If we used etc/*.conf then it would not work
    We have to write /etc/*.conf
50. lists all files whose name starts with any single character followed by 'grep'.

    ~$ ls –ltr /etc/?grep
51. Use single quotes to print a statement with double quotes-we can do this as double quotes within single quotes are interpreted as double quotes(directly printed)
    `$ echo 'hi "how are " you'
    Console output:
    Hi "how are " you
52. Use single quotes to print a statement with single quotes-we can't do this as single quotes inside single quotes are not printed
    ~$ echo 'hi 'how are ' you'
    Console output:
    Hi how are you
53. Use double quotes to print a statement with double quotes-we can't do this as double quotes within double quotes are not interpreted
    ~$ echo "how "are" you"
    Console output:
    How are you
54. Use double quotes to print a statement with single quotes-we can do this as single quotes within double quotes are interpreted as single quotes(directly printed)

    ~$ echo "how 'are' you"
    Console output:
    How 'are' you
55. Print username with variable USERNAME(automatic variable like PATH)
    ~$ echo "$USERNAME"

If we write echo '$USERNAME' then only $USERNAME will be printed
So use double quotes when printing a variable's name

56. Create a variable with a value
~$ name1=1000;
57. Declare a variable directly as environment variable-we can't do this
First we have to declare it as a shell variable and then only we can export and convert it to environment variable
~$ export fire1 ---------incorrect
Correct-
~$ fire1
~$export fire1
Right now,fire1 has null value(it is only declared and then changed to environment variable)
58. See an environment variable which starts with "name1"
~$ env | grep 'name1*'
59. Store the output of the command `hostname -i` in a variable named $myip
~$ myip=`hostname -i`
Or ~$ myip=$(hostname -i)
60. **Print the following message on screen:**

"Running on host : *host_name*" ,

Where 'host_name' should be your current hostname.

Use double quotes only for outside echo command as we have to print variable

- **METHOD 1:store hostname in another variable and then use batch mode**
~$ myip3=`hostname` | echo "\"Running on host\" : $myip3"
- **METHOD 2:don't store in another variable-just use variable substitution**
~$ echo "\"Running on host\" : $(hostname)"

61. Find location/path of a command
~$ which <command>
~$ which cat
~$ which bash
~$ which ls
62. see the permissions for the file `cat` in the output.
~$ ls –ltr `which cat`
63. **Dipslay the contents of file 'newoutput.txt' in all uppercase.**

Redirect contents of this file as input to tr command
2 methods-

- Use pipelining(not batch mode where each command runs individually without the previous command's output becoming next command's input)
~$ cat newoutput.txt;tr '[a-z]' '[A-Z]' ------incorrect as we used batch mode instead of pipe
~$ cat newoutput.txt | tr '[a-z]' '[A-Z]'-------correct
- USE REDIRECTION USING < operator
~$ tr '[a-z]' '[A-Z]' < newoutput.txt
File is sent to tr command as an argument

64. Find names of all files and directories in a directory and all its subdirectories
~$ find *
65. Find relative paths of all files and directories in a directory and its subdirectories
~$ find
66. **Count the total number of files in your current directory.**
Step 1:get names of all files
Then count number of lines

Methods:use ls with wc –l
Or use find with wc –l
~$ ls | wc –l
Or ~$ find * | wc –l

67. Print the 23rd column of a file where delimiter is not just one space but can be many b/w every different column
Consider this scenario where b/w 2 columns,the number of spaces is not same

```
theia@theia-vardaan12345:/home/project$ df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay          98G   40G   54G  43% /
tmpfs            64M     0   64M   0% /dev
tmpfs            16G     0   16G   0% /sys/fs/cgroup
/dev/vda2        98G   40G   54G  43% /etc/hosts
shm              64M     0   64M   0% /dev/shm
tmpfs            28G   16K   28G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs            16G     0   16G   0% /proc/acpi
tmpfs            16G     0   16G   0% /proc/scsi
tmpfs            16G     0   16G   0% /sys/firmware
```

Here we can't us cut –fn method
~$ cut –f 3 file.txt ----------won't work
We have to use awk command when number of a particular delimiter is not 1 but many b/w 2 columns
~$ awk '{print $23}' file.txt

68. **Find the total disk space usage**
Steps-
Use df to get space info
Then find the number after "overlay" tag
we have to search for this----

```
theia@theia-vardaan12345:/home/project$ df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay          98G   40G   54G  43% /
tmpfs            64M     0   64M   0% /dev
tmpfs            16G     0   16G   0% /sys/fs/cgroup
/dev/vda2        98G   40G   54G  43% /etc/hosts
shm              64M     0   64M   0% /dev/shm
tmpfs            28G   16K   28G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs            16G     0   16G   0% /proc/acpi
tmpfs            16G     0   16G   0% /proc/scsi
tmpfs            16G     0   16G   0% /sys/firmware
```

There can be many spaces b/w "overlay" and 98G and then we want 40G value
Use pipelining
~$ df -h | grep overlay | awk '{print $3}'

69. Print details of 3 columns using awk command
NOTE:any characters we use inside print() method of awk won't be printed
Only the $n columns we try to take are taken out
~$ awk '{print$1 and $2 and $4}'
This is same as
~$ awk '{print $1 $2 $4}'
Even spaces b/w $1 and $2 are not printed on the console

70. CREATING A FULL BASH SCRIPT TO DO THE SAME THING AS ABOVE WITH COMMAND LINE ARGUMENTS and pipelining
STEP 1:CREATE SCRIPT USING BATCH MODE
Use \$ inside quotes to use escape meaning instead of special meaning of $ sign
~$ which bash
~$ which sh
~$ #adding shebang directive
~$ echo '# 1 line coming' > file.sh

~$ echo '#!/bin/bash' >>file.sh
~$ #now adding script
~$ echo '#coming script' >> file.sh
~$ echo "col1=\$1" >> file.sh
~$ echo "col2=\$2" >> file.sh
~$ echo "col3=\$3" >> file.sh
~$ echo 'echo "column 1 is $col1"' >> file.sh
~$ echo 'echo "column 2 is $col2"' >> file.sh
~$ echo 'echo "column 3 is $col3"' >> file.sh
STEP 2:USE PIPELINING
~$ df –h | grep overlay | ./file.sh
This calling won't work 5

71. **List five largest files**
    METHOD 1:
    Use ls with sort and then use head –n on it
    ~$ ls -ltr | sort | head -n 5
    METHOD 2:
    Use ls –S feature of ls command(it won't work on ls –ltr) to automatically sort and then use head
    ~$ ls -s | head -5
72. Let us create a bash script named `dirinfo.sh` that takes the directory name as an argument and prints the total number of the the directories and the number of files in it.
    Steps:
    Make a file which takes in the name-store using $1 argument of command line arguments
    Then use find –d and find –f commands to store in another 2 variables the wc –l result
    Then print those 2 variables
    ~$ which bash
    ~$ which sh
    ~$ echo '#!/bin/bash' > dirinfo.sh
    ~$ echo "count1=\$(find \$1 –type -d | wc -l)" >> dirinfo.sh
    ~$ echo "count2=\$(find \$1 –type -f | wc –l)" >> dirinfo.sh
    ~$ echo "echo "\$count1"" >> dirinfo.sh
    ~$ echo "echo "\$count2"" >> dirinfo.sh
    Running the script-give relative path of a directory
    ~$ ./dirinfo.sh /tmp
    tmp is a directory name we gave
73. **Create a script and declare a variable in it with a string stored in it**
    Find all files and directories in "name" directory
    Return 2 separate counts-make a new script for it which takes this input as argument
    1st file
    ~$ echo '#!/bin/bash' > file.sh
    ~$ echo 'store=$1' >> file.sh
    ~$ echo 'echo $store' >> file.sh
    2nd file
    ~$ echo '#!/bin/bash' > file2.sh
    ~$ echo 'countd=$(find $1 –type –d | wc –l)' >> file2.sh

~$ echo 'countf=$(find $1 –type –d | wc –l)' >> file2.sh
~$ echo 'echo $countd' >> file2.sh
~$ echo 'echo $countf' >> file2sh
Calling the files using pipes
~$ ./file.sh /var/log | ./file2.sh
This actually won't work as file2.sh didn't take any argument from keyboard-it can't take input from output of another command using pipes

74. CREATE A SHELL SCRIPT WHICH TAKES IN 2 WORDS AND counts the number of FILES AND ALL DIRECTORIES IN /var/log directory which begin with 1st word and end with second word
Use f directly rather than -f
~$ echo '#!/bin/bash' > file.sh
~$ echo 'var1=$(find /var/log -type f –name "$1*$2")' >> file.sh
~$ echo 'var2=$(find /var/log -type d –name "$1*$2")' >> file.sh
~$ echo 'echo "$var1"' >> file.sh
~$ echo 'echo "$var2"' >> file.sh
$1*$2 means that name should start with $1 variable with some characters following it and then followed by $2 variable
B*bag means that name should start with B and some characters after it and then end with "bag"

75. *Write a shell script named `latest_warnings.sh` that prints the latest 5 warnings from the /var/log/bootstrap.log file.*

~$ echo '#!/bin/bash' > latest_warnings.sh
~$ echo 'echo $(grep 'warning' /var/log/bootstrap.log | sort | head –n 5)' >> latest_warnings.sh

76. Open default text editor to edit a crontab file
~$ crontab -e

77. Append current date in a text file on every wednesday at 4:25 pm
~$ 25 16 * * 3 date >> file.txt

78. Run a shell script file.sh every morning at 7:30
~$ 30 7 * * * /home/Documents/file.sh

79. Run a file called f.sh every january's all sundays at 10:30 pm
~$ 30 22 * 1 0 /home/Documents/f.sh

80. See a list of all currently runnnng,scheduled,and pending cron jobs for the system
~$ crontab –l

81. echo command should run when the minute is 0 and the hours is 21. It effectively means the job runs at 9.00 p.m every day.
The output of the command should be sent to a file /tmp/echo.txt.

Write in crontab file
0 21 * * * echo "welcome" > /tmp/echo.txt

82. Write a script which prints date and disk usage-run this script every february 27th at 7:46 pm and store the result of this script in another file
~$ echo '#!/bin/bash' > file.sh
~$ echo ' echo "$(date)"' >> file.sh
~$ echo 'echo "$(df –h)"' >> file.sh
Now open crontab and schedule running this script
~$ crontab –e
46 19 27 2 * ./file.sh > file2.txt

83. Remove the current crontab
~$ crontab –r

84. *Create a cron job that runs the task `date >> /tmp/everymin.txt` every minute.*
~$ crontab –e

```
~$ * * * * * date >> /tmp/everymin.txt
```

Practical applications

Sunday, January 8, 2023 10:28 PM

Link to lab 1
Link to pipeline
ADVANCED SCRIPTING LINK
Final lab
Cheat sheet

A shell script is composed of a text file with many commands and in which the first line has the form of an interpreter directive
The interpreter directive is also known as "shebang" directive
! Is also known as "bang" symbol
is called pound symbol
Interpreter is absolute path to an executable program
Optional argument-string representing a single argument
Shell scripts are scripts which invoke a shell program
Shebang directives are related to only shell programs
Filters are standard commands which take input from standard input stream(keyboard) and return output to the standard output stream(console)
Examples of filter shell cmmands- wc,cat,more,head,sort,grep

DEFAULT TEXT EDITOR FOR BASH IS GNU NANO

Cron command is used for scheduling running of shell commands or scripts
Crond is the service which interprets crontab files every minute and submits the corresponding jobs to cron at scheduled times
Crontab(short for cron table) is the file containing scheduled jobs and data
Crontab command allows a text editor to edit a crontab file
Job scheduling syntax-

m h dom mon dow command

- Command is any shell commmand or a call to a shell script
- M stands for minute
- H stands for hour
- Dom stands for day of month(not date of month)
- mon stands for month
- Dow stands for day of week(not date of week)

ALL 5 POSITIONS MUST EITHER HAVE A NUMERIC ENTRY OR * SYMBOL(STANDS FOR "ANY" IN THE CATEGORY)
DAY OF WEEK STARTS AT SUNDAY(0 IS FOR SUNDAY AND 6 IS FOR SATURDAY)
Time is given in 24 hour clock
4pm is given by 16:00

Cron is a system daemon used to execute desired tasks in the background at designated times.

A crontab file is a simple text file containing a list of commands meant to be run at specified times. It is edited using the **crontab** command.

Each line in crontab file has five time-and-date fields, followed by a command, followed by a newline character ('\n'). The fields are separated by spaces.

The five time-and-date fields cannot contain spaces.
The five time-and-date fields and their allowed values are as follows:

minute (0-59)
hour(0-23, 0 = midnight)
day (1-31)
month (1-12)
weekday (0-6, 0 = Sunday)

Command pipeline is a feature of the shell, that helps us to combine different unrelated commands in such a way that one command's output is sent directly as input to the next command. This way, what is not possible with a single command can be made possible by connecting multiple commands.

Only filter commands can be used in this manner.

A **filter command** is a command which can accept input from standard input and send output to standard output.

tr command does not accept file names as arguments. But it accepts standard input.

If any special character has to be treated without their special meaning, we need to quote them.
Backslash removes the meaning of the special character that follows it.

A pair of single quotes removes special meanings of all special characters within them (except another single quote).
A pair of double quotes removes special meanings of all special characters within them *except another double quote, variable substitution and command substitution.*.

To display or interpolate the value of a variable in a command, we use the feature of shell called **variable substitution**.

It is done by preceding the name of the variable with a **$** (dollar) symbol.

Command substitution is a feature of the shell, which helps save the output generated by a command in a variable.

It can also be used to nest multiple commands , so that the innermost command's output can be used by outer commands.
The inner command is enclosed in **$()** and will execute first.

Command line arguments can be accessed inside the script as $1, $2 and so on. $1 is the first arugment, $2 is the second argument.

Linux sends the output of a command to **standard output (display)** and any error generated is sent to **standard error (display)**.

Similarly, the input required by a command is received from **standard input (keyboard)**.

If we need to change these defaults, shell provides a feature called **I/O Redirection**.

This is achieved using the following special characters.

When you redirect using `>` the contents of the target file are overwritten.

Symbol	Meaning
<	Input Redirection
>	Output Redirecton
>>	Append Output
2>	Error Redirection

2 MODES OF OPERATION IN BASH----

- **BATCH MODE**
  In this mode,commands run sequentially(one command only runs after the previous command is successfully run)
  Use ; for batch mode
- **CONCURRENT MODE**
  In this mode,commands run in parallel to each other
  The first command runs in background while the 2nd command runs in foreground
  Then 2nd command also tarts running in background while the 3rd command runs in foreground

df -h command gives disk usage for all individual filesystems including the total usage across the server under the head `overlay`.

QUOTING I USED TO SPECIFY WHETHER THE SHELL SHOULD INTERPRET ANY CHARACTER AS META CHARACTERS OR ESCAPE THEM(INTERPRET THEM AS ESCAPE CHARACTERS)
\ used to escape interpretation of a single character as meta character
Double quotes are used to interpret characters literally except meta characters(if they are written directly or with \)
Single quotes are used to interpret charcters literally(no meaning of any meta characters of escape characters)

METACHARACTERS-SPECIAL CHARACTERS WHICH HAVE A SPECIAL MEANING TO SHELL
COMMENTS
; separate characters typed on same line but are intended to be outputted on different line
Symbol "*" is used to represent any number of consecutive characters in the same line in name of a file only
"*" is also called asterisk or filename expansion wildcard
? Is single character version of "*" character
? Is single character wildcard in filename expansion

FILTERS CAN BE CHAINED TOGETHER
PIPE COMMAND "|"
Pipe command allows us to chaiin filter commands(not all types of commands but only filter commands)
If u want to chain cd .. command with cat command then it won't happen as cd .. is not a filter command
THE OUTPUT OF COMMAND 1 BECOMES THE INPUT OF COMMAND 2
Shell variables are variables which are limited in scope of the shell script/file they are created in
One shell cannot see/use other shell's variables
"Set" comand is used to see definitions and names of all variables defined in a particular shell file/script
Environment variables have more extended scope than shell variables
These variables can be used in any child processes originated by the shell in which they were declared
The backslash \ character used here after the pipe | allows you to write the expression on multiple lines.

Some commands, such as `tr`, *only* accept "standard input" as input (not strings or filenames):

- `tr (translate) - replaces characters in input text.`
  - Syntax: `tr [OPTIONS] [target characters] [replacement characters]`

REPLACE A COMMAND WITH ITS OUTPUT-USE COMMAND SUBSTITUTION
2 NOTATIONS-

- **$(command)**
- | **`command`** | **Used back quotes** |

COMMAND LINE ARGUMENTS-ARGUMENTS SPECIFIED ON THE COMAND LINE
I/O REDIRECTION COMMANDS ARE USED TO REDIRECT STANDARD INPUT(KEYBOARD) OR STANDARD OUTPUT(CONSOLE)
> redirect standard output to a file
If file doesn't exist then first it is created and then the standard output is written on it
If it exists then all of its previous contents are overwritten by the incoming standard output
>> append output to existing content of a file-the new incoming output is just merged after the last line written in the file
2> redirect an error message to a file(overwrite previous contents)
2>> append an error message to a file(merge error message with previous contents)
< pass file's contents as input to the standard input

All the lines starting with # are comments. They are not executed by the shell.

What is a script?

- Script: list of commands interpreted by a scripting language
- Commands can be entered interactively or listed in a text file
- Scripting languages are interpreted at runtime
- Scripting is slower to run, but faster to develop

What is a script used for?

- Widely used to automate processes

- ETL jobs, file backups and archiving, system admin

- Used for application integration, plug-in development, web apps, and many other tasks

```
theia@theia-vardaan12345:/home/project$ which bash
/bin/bash
theia@theia-vardaan12345:/home/project$ which sh
/bin/sh
theia@theia-vardaan12345:/home/project$ #!/bin/bash
theia@theia-vardaan12345:/home/project$ #!/bin/sh
theia@theia-vardaan12345:/home/project$ echo '#!/bin/bash' >> file.sh
theia@theia-vardaan12345:/home/project$ echo '# my first script' >> file.sh
theia@theia-vardaan12345:/home/project$ echo 'give name' >> file.sh
theia@theia-vardaan12345:/home/project$ echo 'read name11' >> file.sh
theia@theia-vardaan12345:/home/project$ echo 'echo 'welcome $name11'' >> file.sh
theia@theia-vardaan12345:/home/project$ ls -l
total 4
-rw-r--r-- 1 theia users 66 Jan  9 03:06 file.sh
theia@theia-vardaan12345:/home/project$ chmod +x file.sh
theia@theia-vardaan12345:/home/project$ cat file.sh
#!/bin/bash
# my first script
give name
read name11
echo welcome
theia@theia-vardaan12345:/home/project$ rm file.sh
theia@theia-vardaan12345:/home/project$ echo '#!/bin/bash' >> f.sh
theia@theia-vardaan12345:/home/project$ echo 'read name' >> f.sh
theia@theia-vardaan12345:/home/project$ echo 'echo "welcome $name"' >> f.sh
theia@theia-vardaan12345:/home/project$ cat f.sh
#!/bin/bash
read name
echo "welcome $name"
theia@theia-vardaan12345:/home/project$ ls -lst f.sh
4 -rw-r--r-- 1 theia users 43 Jan  9 03:10 f.sh
theia@theia-vardaan12345:/home/project$ chmod +x f.sh
theia@theia-vardaan12345:/home/project$ ls -lst f.sh
4 -rwxr-xr-x 1 theia users 43 Jan  9 03:10 f.sh
theia@theia-vardaan12345:/home/project$ bash f.sh
dd
welcome dd
theia@theia-vardaan12345:/home/project$ ./fsh
```

```
theia@theia-vardaan12345:/home/project/new$ env | grep gre
AIRFLOW__CORE__SQL_ALCHEMY_CONN=postgresql+psycopg2://airflow:airflow@localhost/airflow
greeting=17
AIRFLOW__CELERY__RESULT_BACKEND=db+postgresql://airflow:airflow@localhost/airflow
theia@theia-vardaan12345:/home/project/new$ env | gre '*gre'
bash: gre: command not found
theia@theia-vardaan12345:/home/project/new$ env | grep '*gre'
theia@theia-vardaan12345:/home/project/new$ env | grep 'gre*'
AIRFLOW__CORE__SQL_ALCHEMY_CONN=postgresql+psycopg2://airflow:airflow@localhost/airflow
greeting=17
AIRFLOW__CELERY__RESULT_BACKEND=db+postgresql://airflow:airflow@localhost/airflow
theia@theia-vardaan12345:/home/project/new$ nano cat.txt
theia@theia-vardaan12345:/home/project/new$ uniq cat.txt | sort
cat
dog
dog
goldfish
goldfish
parrot
theia@theia-vardaan12345:/home/project/new$ cat cat.txt
goldfish
dog
cat
parrot
dog
goldfish
goldfish
theia@theia-vardaan12345:/home/project/new$ sort -r cat.txt | uniq
parrot
goldfish
dog
cat
theia@theia-vardaan12345:/home/project/new$ cat cat.txt | sort -r | uniq | tr '[A-Z]' '[a-z]'
parrot
goldfish
dog
cat
theia@theia-vardaan12345:/home/project/new$
```

```
theia@theia-vardaan12345:/home/project$ garbage
bash: garbage: command not found
theia@theia-vardaan12345:/home/project$ garbage 2> file.txt
theia@theia-vardaan12345:/home/project$ cat file.txt
bash: garbage: command not found
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ # THIS IS A COMEMT
theia@theia-vardaan12345:/home/project$ pwd;touch file.txt
/home/project
theia@theia-vardaan12345:/home/project$ ls
file.txt
theia@theia-vardaan12345:/home/project$ ls -ltr *
-rw-r--r-- 1 theia users 0 Jan  9 18:52 file.txt
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rw-r--r-- 1 theia users 0 Jan  9 18:52 file.txt
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ ls -ltr /etc/*.conf
-rw-r--r-- 1 root root    34 Jan 27  2016 /etc/ld.so.conf
-rw-r--r-- 1 root root   497 Oct  5  2016 /etc/nsswitch.conf
-rw-r--r-- 1 root root   604 Aug 13  2017 /etc/deluser.conf
-rw-r--r-- 1 root root   703 Aug 21  2017 /etc/logrotate.conf
-rw-r--r-- 1 root root  2683 Jan 17  2018 /etc/sysctl.conf
-rw-r--r-- 1 root root  2584 Feb  1  2018 /etc/gai.conf
-rw-r--r-- 1 root root   191 Feb  7  2018 /etc/libaudit.conf
-rw-r--r-- 1 root root  1260 Feb 25  2018 /etc/ucf.conf
-rw-r--r-- 1 root root  2969 Feb 28  2018 /etc/debconf.conf
-rw-r--r-- 1 root root   812 Mar 24  2018 /etc/mke2fs.conf
-rw-r--r-- 1 root root   552 Apr  4  2018 /etc/pam.conf
-rw-r--r-- 1 root root    92 Apr  9  2018 /etc/host.conf
-rw-r--r-- 1 root root  2154 Mar 22  2019 /etc/mongodb.conf
-rw-r--r-- 1 root root  2517 Aug 17  2020 /etc/ntp.conf
-rw-r--r-- 1 root root 10368 Mar 31  2022 /etc/sensors3.conf
-rw-r--r-- 1 root root  3028 Oct 19 15:28 /etc/adduser.conf
-rw-r--r-- 1 root root  5432 Dec  7 00:06 /etc/ca-certificates.conf
-rw-r--r-- 1 root root   115 Jan  9 18:50 /etc/resolv.conf
theia@theia-vardaan12345:/home/project$ ls -ltr /bin/?grep
-rwxr-xr-x 1 root root   28 Sep 18  2019 /bin/fgrep
-rwxr-xr-x 1 root root   28 Sep 18  2019 /bin/egrep
-rwxr-xr-x 1 root root 6456 Apr  8  2022 /bin/zgrep
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ echo 'Following are some special characters in shell - < > ;
" ( ) \ [ ]  '
Following are some special characters in shell - < > ; " ( ) \ [ ]
theia@theia-vardaan12345:/home/project$ echo ' hi how ' are you ' i am good '
 hi how  are you  i am good
theia@theia-vardaan12345:/home/project$ ' hi " how are" you '
bash:  hi " how are" you : command not found
theia@theia-vardaan12345:/home/project$
theia@theia-vardaan12345:/home/project$ echo ' hi " how " are you '
 hi " how " are you
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ name="hi";
theia@theia-vardaan12345:/home/project$ echo $name
hi
theia@theia-vardaan12345:/home/project$ echo "$name"
hi
theia@theia-vardaan12345:/home/project$ echo '$name'
$name
theia@theia-vardaan12345:/home/project$ echo "Current user name: $USERNAME"
Current user name: vardaan12345
theia@theia-vardaan12345:/home/project$ echo 'Current user name: $USERNAME'
Current user name: $USERNAME
theia@theia-vardaan12345:/home/project$ echo "how "are" you"
how are you
theia@theia-vardaan12345:/home/project$ echo "how 'are' you"
how 'are' you
theia@theia-vardaan12345:/home/project$
```

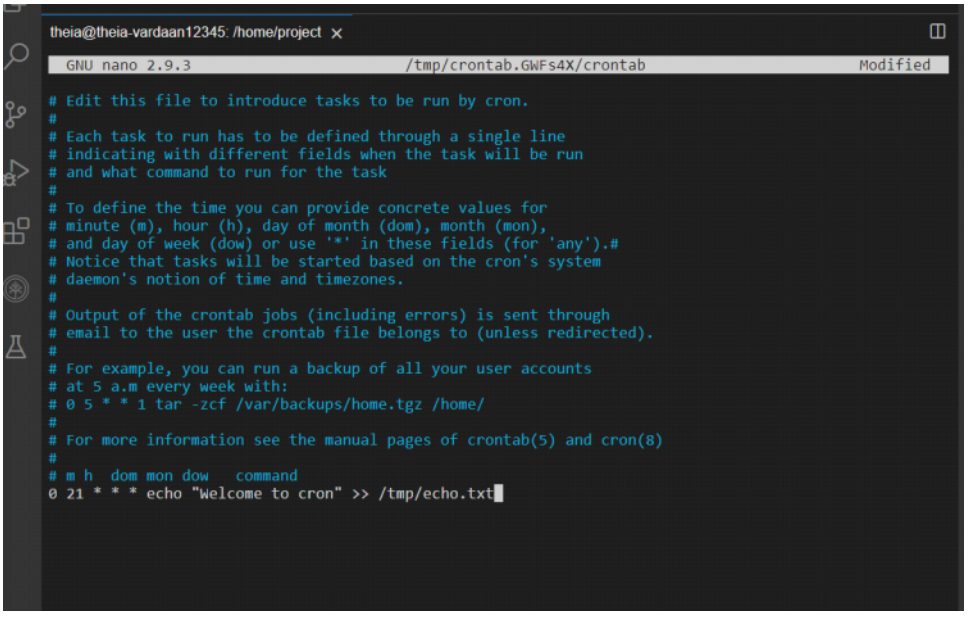
```
theia@theia-vardaan12345:/home/project$ myip=$(hostname -i)
theia@theia-vardaan12345:/home/project$ echo $myip
172.22.160.99
theia@theia-vardaan12345:/home/project$ myip2=`hostname -i`
theia@theia-vardaan12345:/home/project$ echo $myip2
172.22.160.99
theia@theia-vardaan12345:/home/project$
```

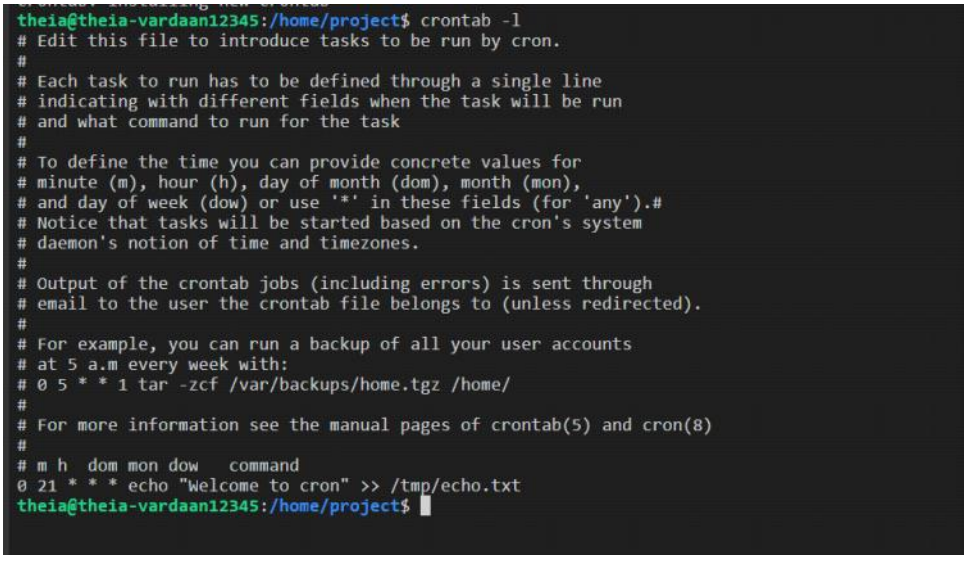
```
theia@theia-vardaan12345:/home/project$ myip=$(hostname -i)
theia@theia-vardaan12345:/home/project$ echo $myip
172.22.160.99
theia@theia-vardaan12345:/home/project$ myip2=`hostname -i`
theia@theia-vardaan12345:/home/project$ echo $myip2
172.22.160.99
theia@theia-vardaan12345:/home/project$ myip3=$(hostname);echo '"Running on host" : $myip3'
"Running on host" : $myip3
theia@theia-vardaan12345:/home/project$ echo "\"Running on host\" : $myip3"
"Running on host" : theia-vardaan12345
theia@theia-vardaan12345:/home/project$ uname
Linux
theia@theia-vardaan12345:/home/project$ uname -a
Linux theia-vardaan12345 4.15.0-200-generic #211-Ubuntu SMP Thu Nov 24 18:16:04 UTC 2022 x86_64 x86_6
4 x86_64 GNU/Linux
theia@theia-vardaan12345:/home/project$ touch fi.txt
theia@theia-vardaan12345:/home/project$ nano fi.txt
theia@theia-vardaan12345:/home/project$ cat fi.txt;tr '[a-z]' '[A-Z]'
Hello hi Bye Tata
ff
FF
^C
theia@theia-vardaan12345:/home/project$ cat fi.txt | tr '[a-z]' '[A-Z]'
HELLO HI BYE TATA
theia@theia-vardaan12345:/home/project$ tr '[a-z]' '[A-Z]' < cat fi.txt
bash: cat: No such file or directory
theia@theia-vardaan12345:/home/project$ tr '[a-z]' '[A-Z]' < newoutput.txt
bash: newoutput.txt: No such file or directory
theia@theia-vardaan12345:/home/project$ tr '[a-z]' '[A-Z]' < fi.txt
HELLO HI BYE TATA
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ ls -ltr | sort | head -n 5
head: cannot open '-n' for reading: No such file or directory
head: cannot open '5' for reading: No such file or directory
ls: cannot access '-ltr': No such file or directory
theia@theia-vardaan12345:/home/project$ ls -ltr
total 8
-rw-r--r-- 1 theia users 18 Jan  9 20:13 fi.txt
-rw-r--r-- 1 theia users 40 Jan  9 20:36 file.txt
theia@theia-vardaan12345:/home/project$ ls | sort
file.txt
fi.txt
theia@theia-vardaan12345:/home/project$ ls | sort | head -n 2
file.txt
fi.txt
theia@theia-vardaan12345:/home/project$ ls | sort | head -n 5
file.txt
fi.txt
theia@theia-vardaan12345:/home/project$ ls -ltr | sort
-rw-r--r-- 1 theia users 18 Jan  9 20:13 fi.txt
-rw-r--r-- 1 theia users 40 Jan  9 20:36 file.txt
total 8
theia@theia-vardaan12345:/home/project$ ls -ltr | sort | head -n 5
-rw-r--r-- 1 theia users 18 Jan  9 20:13 fi.txt
-rw-r--r-- 1 theia users 40 Jan  9 20:36 file.txt
total 8
theia@theia-vardaan12345:/home/project$ ls -s
total 8
4 file.txt  4 fi.txt
theia@theia-vardaan12345:/home/project$ ls -ltr -s
total 8
4 -rw-r--r-- 1 theia users 18 Jan  9 20:13 fi.txt
4 -rw-r--r-- 1 theia users 40 Jan  9 20:36 file.txt
theia@theia-vardaan12345:/home/project$ ls -s -ltr
total 8
4 -rw-r--r-- 1 theia users 18 Jan  9 20:13 fi.txt
4 -rw-r--r-- 1 theia users 40 Jan  9 20:36 file.txt
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ df -h | grep overlay | awk '{print $3}'
40G
theia@theia-vardaan12345:/home/project$ df -h | grep overlay | awk '{print $3 and $2 and $5}'
40G98G43%
```

```
theia@theia-vardaan12345:/home/project$ crontab -l
no crontab for theia
theia@theia-vardaan12345:/home/project$
```


theia@theia-vardaan12345: /home/project
GNU nano 2.9.3 /tmp/crontab.GWFs4X/crontab Modified
Edit this file to introduce tasks to be run by cron.
#
Each task to run has to be defined through a single line
indicating with different fields when the task will be run
and what command to run for the task
#
To define the time you can provide concrete values for
minute (m), hour (h), day of month (dom), month (mon),
and day of week (dow) or use '*' in these fields (for 'any').#
Notice that tasks will be started based on the cron's system
daemon's notion of time and timezones.
#
Output of the crontab jobs (including errors) is sent through
email to the user the crontab file belongs to (unless redirected).
#
For example, you can run a backup of all your user accounts
at 5 a.m every week with:
0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
For more information see the manual pages of crontab(5) and cron(8)
#
m h dom mon dow command
0 21 * * * echo "Welcome to cron" >> /tmp/echo.txt


theia@theia-vardaan12345:/home/project$ crontab -l
Edit this file to introduce tasks to be run by cron.
#
Each task to run has to be defined through a single line
indicating with different fields when the task will be run
and what command to run for the task
#
To define the time you can provide concrete values for
minute (m), hour (h), day of month (dom), month (mon),
and day of week (dow) or use '*' in these fields (for 'any').#
Notice that tasks will be started based on the cron's system
daemon's notion of time and timezones.
#
Output of the crontab jobs (including errors) is sent through
email to the user the crontab file belongs to (unless redirected).
#
For example, you can run a backup of all your user accounts
at 5 a.m every week with:
0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
For more information see the manual pages of crontab(5) and cron(8)
#
m h dom mon dow command
0 21 * * * echo "Welcome to cron" >> /tmp/echo.txt
theia@theia-vardaan12345:/home/project$


theia@theia-vardaan12345:/home/project$ echo $date
theia@theia-vardaan12345:/home/project$ echo $(date)
Tue Jan 10 15:15:13 EST 2023
theia@theia-vardaan12345:/home/project$ theia@theia-vardaan12345:/home/project$ echo $date
theia@theia-vardaan12345:/home/project$ echo $(date)
Tue Jan 10 15:15:13 EST 2023
theia@theia-vardaan12345:/home/project$

3. Filters take input from standard input and then return their output to standard output. What action describes the filter's job? 1 / 1 point

- ○ Provider
- ○ Interpreter
- ○ Analyzer
- ◉ Transformer

✓ Correct
Correct, filters are a program that transforms input data into output data.

4. The cron utility allows you to schedule jobs to run at specific times. What is crond? 1 / 1 point

- ◉ service that interprets crontab files
- ○ crontab
- ○ cron table
- ○ create, read, update, delete

9. Quoting is used to specify whether the shell should interpret special characters as metacharacters, or 'escape' them. When used in conjunction with the echo command, which of the following statements is true? 1 point

- ○ The backslash "\" acts like a new line character.
- ○ You can use the backslash "\" to interpret a single character as a metacharacter.
- ○ Double quotes are used to interpret all contents as literal characters.
- ◉ Single quotes are used to interpret all contents as literal characters.

4. What does chaining filters together produce? 1 point

- ○ Examiner
- ○ Multi-use script
- ◉ Pipeline
- ○ Modifier

Final exam

Wednesday, January 11, 2023 1:44 AM

DOUBTS

1. Which of the following distros is highly configurable but less stable than the others? **1 point**

- Red Hat Enterprise Linux
- Ubuntu
- Arch Linux
- Debian

67:55

Expand

8. Which of the following commands extracts files from an archive named tmp.tar? **1 point**

- tar -tn tmp.tar
- tar -xf tmp.tar
- tar -ca tmp.tar
- tar -vq tmp.tar

66:43

Expand

13. Which of the following operators can you use to make two Bash commands run in parallel? **1 point**

- #
- &
- *
- ;

65:30

Expand

CONFIRM

2. Which layer of the Linux system enables users to interact with the Linux operating system? **1 point**

- Kernel
- User
- Application
- Hardware

67:20

Expand

6. Assume you run the command wc usdoi.txt and receive the following output: 152 1330 8121 usdoi.txt What does the number "1330" indicate about usdoi.txt? 1 point

- Number of lines
- Number of words
- Number of characters
- Number of spaces

67:01

Expand

9. Which of the following commands will print the IP address of the system's host? 1 point

- hostname -d
- hostname -a
- hostname -i
- hostname -y

66:29

Expand

10. Assume you have a file named shoppinglist.txt. In the terminal, you want to print each line of shoppinglist.txt in alphabetical order, but you want to omit consecutive repeated lines from the output. Which of the following Bash inputs should you use? 1 point

- uniq shoppinglist.txt \> sort
- sort shoppinglist.txt \> uniq
- sort shoppinglist.txt | uniq
- uniq shoppinglist.txt | sort

66:03

Expand

15. Which of the following examples of cron syntax will append the current date to the file "mondays.txt" at 8:00 AM every Monday? 1 point

- 1 * 8:00 * 0 date \>\> mondays.txt
- 8 0 0 0 1 date \>\> mondays.txt
- 0 8 * * 1 date \>\> mondays.txt
- 1 8 * * 1 date \>\> mondays.txt

65:13

Expand

Cheat sheet copy

Tuesday, January 10, 2023 12:41 PM

Linux and Bash Command Cheat Sheet: The Basics

Getting information

```
# return your user name
whoami

# return your user and group id
id

# return operating system name, username, and other info
uname -a

# display reference manual for a command
man top

# get help on a command
curl --help

# return the current date and time
date
```

Monitoring performance and status

```
# list selection of or all running processes and their PIDs
ps
ps -e

# display resource usage
top

# list mounted file systems and usage
df
```

Working with files

```
# copy a file
cp file.txt new_path/new_name.txt

# change file name or path
mv this_file.txt that_path/that_file.txt

# remove a file verbosely
rm this_old_file.txt -v

# create an empty file, or update existing file's timestamp
touch a_new_file.txt

# change/modify file permissions to 'execute' for all users
```

chmod +x my_script.sh

get count of lines, words, or characters in file
wc -l table_of_data.csv
wc -w my_essay.txt
wc -m some_document.txt

return lines matching a pattern from files matching a filename pattern - case insensitive and whole words only
grep -iw hello *.txt

return file names with lines matching the pattern 'hello' from files matching a filename pattern
grep -l hello *.txt

Navigating and working with directories

list files and directories by date, newest last
ls -lrt

find files in directory tree with suffix 'sh'
find -name '*.sh'

return present working directory
pwd

make a new directory
mkdir new_folder

change the current directory: up one level, home, or some other path
cd ../
cd ~ or cd
cd another_directory

remove directory, verbosely
rmdir temp_directory -v

Printing file and string contents

print file contents
cat my_shell_script.sh

print file contents page-by-page
more ReadMe.txt

print first N lines of file
head -10 data_table.csv

print last N lines of file
tail -10 data_table.csv

print string or variable value
echo "I am not a robot"
echo "I am $USERNAME"

Compression and archiving

archive a set of files
tar -cvf my_archive.tar.gz file1 file2 file3

compress a set of files
zip my_zipped_files.zip file1 file2
zip my_zipped_folders.zip directory1 directory2

extract files from a compressed zip archive
unzip my_zipped_file.zip
unzip my_zipped_file.zip -d extract_to_this_direcory

Performing network operations

print hostname
hostname

send packets to URL and print response
ping www.google.com

display or configure system network interfaces
ifconfig
ip

display contents of file at a URL
curl <url>

download file from a URL
wget <url>

Bash shebang

```
#!/bin/bash
```

Pipes and Filters

chain filter commands using the pipe operator
ls | sort -r

pipe the output of manual page for ls to head to display the first 20 lines
man ls | head -20

Shell and Environment Variables

list all shell variables
set

# define a shell variable called my_planet and assign value Earth to it
my_planet=Earth

display shell variable

```
echo $my_planet

# list all environment variables
env

# environment vars: define/extend variable scope to child processes
export my_planet
export my_galaxy='Milky Way'
```

Metacharacters

```
# comments
# The shell will not respond to this message

# command separator
echo 'here are some files and folders'; ls

# file name expansion wildcard
ls *.json

# single character wildcard
ls file_2021-06-??.json
```

Quoting

```
# single quotes - interpret literally
echo 'My home directory can be accessed by entering: echo $HOME'

# double quotes - interpret literally, but evaluate metacharacters
echo "My home directory is $HOME"

# backslash - escape metacharacter interpretation
echo "This dollar sign should render: \$"
```

I/O Redirection

```
# redirect output to file
echo 'Write this text to file x' > x

# append output to file
echo 'Add this line to file x' >> x

# redirect standard error to file
bad_command_1 2> error.log

# append standard error to file
bad_command_2 2>> error.log

# redirect file contents to standard input
$ tr “[a-z]” “[A-Z]” < a_text_file.txt

# the input redirection above is equivalent to
$cat a_text_file.txt | tr “[a-z]” “[A-Z]”
```

Command Substitution

# capture output of a command and echo its value
THE_PRESENT=$(date)
echo "There is no time like $THE_PRESENT"

Command line arguments

```
./My_Bash_Script.sh arg1 arg2 arg3
```

Batch vs. concurrent modes

# run commands sequentially
start=$(date); ./MyBigScript.sh ; end=$(date)

run commands in parallel
./ETL_chunk_one_on_these_nodes.sh & ./ETL_chunk_two_on_those_nodes.sh

Scheduling jobs with Cron

open crontab editor
crontab -e

job scheduling syntax
m h dom mon dow command
minute, hour, day of month, month, day of week
* means any

append the date/time to file every Sunday at 6:15 pm
15 18 * * 0 date >> sundays.txt

run a shell script on the first minute of the first day of each month
1 0 1 * * ./My_Shell_Script.sh

back up your home directory every Monday at 3 am
0 3 * * 1 tar -cvf my_backup_path\my_archive.tar.gz $HOME\

deploy your cron job
Close the crontab editor and save the file

list all cron jobs
crontab -l

1. Linux lab week 2 part 1

Thursday, January 5, 2023 5:01 PM

link
Link to code space

PASTE CODE IN UBUNTU-
Use 2 finger click on mousepad

1. **FIND LIST OF ALL USERS LOGGED IN**
   ~$ who
2. **Find name of current user(you)**
   ~$ whoami

3. **Print only kernel name-here linux**
   ~$ uname
4. system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system.
   ~$ uname -a
5. **Display user id and group id for a particular user**
   ~$ id
6. The `ps` command lists each processes that is currently running and its `PID` (process id). However, the output only contains the processes that are owned by you.

   ~$ ps
7. By using the `-e` option, you can display all of the processes running on the system. The includes processes owned by other users.
   ~$ ps –e
8. The `top` command provides a dynamic, real-time view of your system.

   It shows summary information of the system and a table of more detailed information related to the processes or threads which are currently running and managed by the kernel. This includes information related to cpu and memory usage per process.

   ~$ top
   When you start `top`, you'll be presented with the following elements on the main `top` screen.
   Summary area - shows information like system uptime, number of users, load average, and overall memory usage
   Column header - attribute names
   Task area - displays the data for each process, or `PID`

   The output keeps refreshing until you press `q` or `Ctrl+c`.

9. Display some specific no of lines in top command and then stop top
   ~$ top –n 10
   Displays 10 lines of top and then automatically exit

You can press the following keys while `top` is running to sort the table:

Key	Sorts by
M	Memory Usage
P	CPU Usage
N	Process ID (PID)
T	Running Time.

For example, you can find out which process is consuming the most memory by entering `shift + m`.

10. Echo command

These special characters help you better format your output.

Special Character	Effect
\n	start a new line
\t	insert a tab

Use the `-e` option of the echo command when working with special characters.

~$ echo –e "how \n are you"
Output:
How
Are you
Note:always use quotes when u are using –e

11. Display date in mm/dd/yyyy format
`$ date "+%D"
Always use "+%" and then some modifier character

Here are some of the popular format specifiers that you can try out:

Specifier	Explanation
%d	Display the day of the month (01 to 31)
%h	Displays the abbreviated month name (Jan to Dec)
%m	Displays the month of year (01 to 12)
%Y	Displays the four-digit year
%T	Displays the time in 24 hour format as HH:MM:SS
%H	Displays the hour

12. List all binary and executable files which are part of your /bin directory
~$ ls /bin
The `/bin` directory happens to be where Linux commmands such as `ls` and `pwd` are stored.
13. List all files in /bin directory starting with character b
~$ ls ./bin/b*
14. List all files in /bin ending with ar
~$ ls ./bin/*ar

Here are some popular options that you can try with the `ls` command.

Option	Description
`-a`	list all files, including hidden files
`-d`	list directories only, do not include files
`-h`	with `-l` and `-s`, print sizes like 1K, 234M, 2G
`-l`	include attributes like permissions, owner, size, and last-modified date
`-S`	sort by file size, largest first
`-t`	sort by last-modified date, newest first
`-r`	reverse the sort order

15. Get a long list of all files(even hidden files) in /etc directory
    ~$ ls –l –a ./etc
16. Get a simple list of all hidden files in /etc directory
    ~$ ls –a ./etc
17. Find all files in etc directory which start with ffg
    ~$ find ./etc –name 'ffg*'
18. Find all files in etc directory which have a .page extension and have a .txt extension
    ~$ find ./etc -name '*.txt'
    ~$ find ./etc -name '*.page'
19. Ask permission before removing a file
    ~$ rm -i file1.txt
    -i is used to ask permission
20. Copy a file from a below directory into another file which is located in a different below directory
    /home/project$ cp ./my_pr/file.txt ./my_pr2/file2.txt
21. Download a file just with its online url
    Given url: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/usdoi.txt

    ~$ wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/usdoi.txt

Each file and each directory has permissions set for three permission categories: the 'owner', the 'group' and 'all users'.

The following permissions are set for each file and directory:

Permission	symbol
read	`r`
write	`w`
execute	`x`

A sample output looks like:

```
-rw-r--r-- 1 theia theia 8121 May 31 16:45 usdoi.txt
```

The permissions set here are `rw-r--r--`. The `-` preceeding these permissions indicates that `usdoi.txt` is a file. If it were a directory, you would see a `d` instead of the `-`.

The first three entries correspond to the owner, the next three correspond to the group, and the last threee are for all others. You can see the owner of the file has read and write permissions, while the user group only has read permissions, and all other users have read permission. No users have execute permissions, as indicated by the `-` instead of an `x` in the third position for each user category.

The `chmod` (change mode) command lets you change the permissions set for a file.

The change of permissions is specified with the help of a combination of the following characters:

Option	Description
`r`, `w` and `x`	permissions: read, write and execute, respectively
`u`, `g` and `o`	user categories: owner, group and all others, respectively
`+`, `-`	operations: grant and revoke, respectively

22. Remove read permission from user,group,and others for a file
    ~$ chmod –r file1.txt
23. Add permission of read to all 3 categories
    ~$ chmod +r file1.txt
24. Remove write permission for other users
    ~$ chmod o-w file1.txt
25. Remove write and execute permission for user and group
    ~$ chmod ug-wx file1.txt
26. Remove same permission for same category but for many files
    ~$ chmod uo-rx file1.txt file2.txt

27. Display content of /home/ when u are at /home/project
    /home/project$ ls –ltr ./..
28. Find permissions of /home/project directory from /home
    /home$ ls –l –d project or /home$ ls –ltr –d final
    For getting file, use –f
    /home$ ls –ltr –f project
29. Remove a file by specifically giving warning
    ~$ rm –f -i display.sh
    If we write -I before –f then file will be removed but we won't be given warning
30. *List the files in `/etc` directory in the ascending order of their access time.*
    ~$ ls –ltr /etc/
31. C*opy the file `/var/log/bootstrap.log` to your current directory.*

    ~$ cp var/log/bootstrap.log ./ or ~$ var/log/bootstrap.log .

Practical applications of commands part 1

Wednesday, January 4, 2023 8:48 PM

In vim,both :wq and :wq! Work the same way in command mode
TOP command (Table of Processes) ACTS AS A TASK MANAGER AND SHOWS RUNNING PROCESSES WITH THEIR RESOURCE USAGE
Shell is an interface for running linux commands
It is a scripting language
It is an interactive language
Either $ or > can be used for representing command prompt or terminal
Default shell on linux systems is BASH(born again shell)
OTHER SHELLS INCLUDE sh(bourne shell),ksh(korn shell),tcsh(tc shell),zsh(z shell),fish

```
/bin/bash
theia@theia-vardaan12345:/home/project$ fish
bash: fish: command not found
theia@theia-vardaan12345:/home/project$ bash
theia@theia-vardaan12345:/home/project$ sh
$ fsh
```

Running linux on windows machine
Grep command-search for a match
All methods-

- We can install linux on a separate drive
  Then do a reboot
- Install linux on a virtual machine/environment
- Use linux emulator
- Use windows subsystem for linux-which runs linux binary executables natively on windows system

Using wc command
Options-

- **Use wc –m for getting no of characters**
- **Use wc –c for getting no of bytes**
- **Use wc –w to get number of words**

Chmod command

On each line, the first character identifies the type of entry that is being listed. If it is a dash (-) it is a file. If it is the letter d it is a directory.

- The first three characters show the permissions for the user who owns the file (*user permissions*).
- The middle three characters show the permissions for members of the file's group (*group permissions*).
- The last three characters show the permissions for anyone not in the first two categories (*other permissions*).

The letters represent:

- *r*: Read permissions. The file can be opened, and its content viewed.
- *w*: Write permissions. The file can be edited, modified, and deleted.
- *x*: Execute permissions. If the file is a script or a program, it can be run (executed).

- S: The 's' bits are referred to as the "setuid" and "setgid" bits. What it does depends on the file type.
- On a directory, as in your example, these bits set the default user or group for all files created in the directory.
- EG, if you have a directory owned by foo:foo, with the setuid and setgid bits set, then all files created in that directory will be owned by foo:foo, regardless of who creates them.
- In your example, the "setgid" bit is set for each directory. This means that for every file created in these directories, the owner will be the user who created the file, but the group will be set to match the directory's group, rather than the user's main group.
- The character "s" means **"set user or group ID on execution"**.

For example:

- `---` means no permissions have been granted at all.
- `rwx` means full permissions have been granted. The read, write, and execute indicators are all present.
- If we have r-x then we don't have write permission
- If we have –x then we can only execute/run the file/program/script
- We can either use x or s in 3rd place
  If we have r-s then we have r and s permissions but not w and x permissions

Command rm
-r tells to recursively delete/remove the files from the directory
~$ rm –r <directory_name>

VERSIONS OF USING cp command

We can't copy contents of 2 source files into another source file(we can copy though into another directory)

~$ cp src1 dstfile1	**Copy source file into another source file**
~$ cp src1 dstdirectory1	**Copy source file into another directory**
~$ cp src1 src2 destdirectory1	**Copy all source files into another directory**
~$ cp srcdirectory dstdirectory	**Copy all files of one directory into another directory**

Touch command

- **If file doesn't exist then it is created**
- **If file exists,then its timestamp is changed to current time**

Using mv command

If the destination file **doesn't exist**, it will be created. In the above command **mv** simply replaces the source filename in the directory with the destination filename(new name). If the destination file **exist**, then it will be **overwrite** and the source file will be deleted. By

default, **mv** doesn't prompt for overwriting the existing file

~$ mv source destination	Both are files

Options-

- **DESTINATION FILE NON EXISTENT-first it is created and then we copy contents of source file in it and then delete source file**
- **Destination file existent-we overwrite contents of destination file with contents of source file and then delete the source file**
- **If we have to ask user to overwrite destination file then we can put -I before source file's name**

GETTING INFO SHELL COMMANDS

- whoami – username
- id – user ID and group ID
- uname – operating system name
- ps – running processes
- top – resource usage
- df – mounted file systems
- man – reference manual

WORKING WITH FILES COMMANDS

- cp – copy file
- mv – change file name or path
- rm – remove file
- touch – create empty file, update file timestamp
- chmod – change/modify file permissions
- wc – get count of lines, words, characters in file
- grep – return lines in file matching pattern

NAVIGATING AND WORKING WITH DIRECTORIES COMMANDS

- ls – list files and directories
- find – find files in directory tree
- pwd – get present working directory
- mkdir – make directory
- cd – change directory
- rmdir – remove directory

PRINTING CONTENTS OF A FILE OR STRING

- cat – print file contents
- more – print file contents page-by-page
- head – print first N lines of file
- tail – print last N lines of file
- echo – print string or variable value

FILE COMPRESSION AND ARCHIVING APPLICATIONS COMMANDS

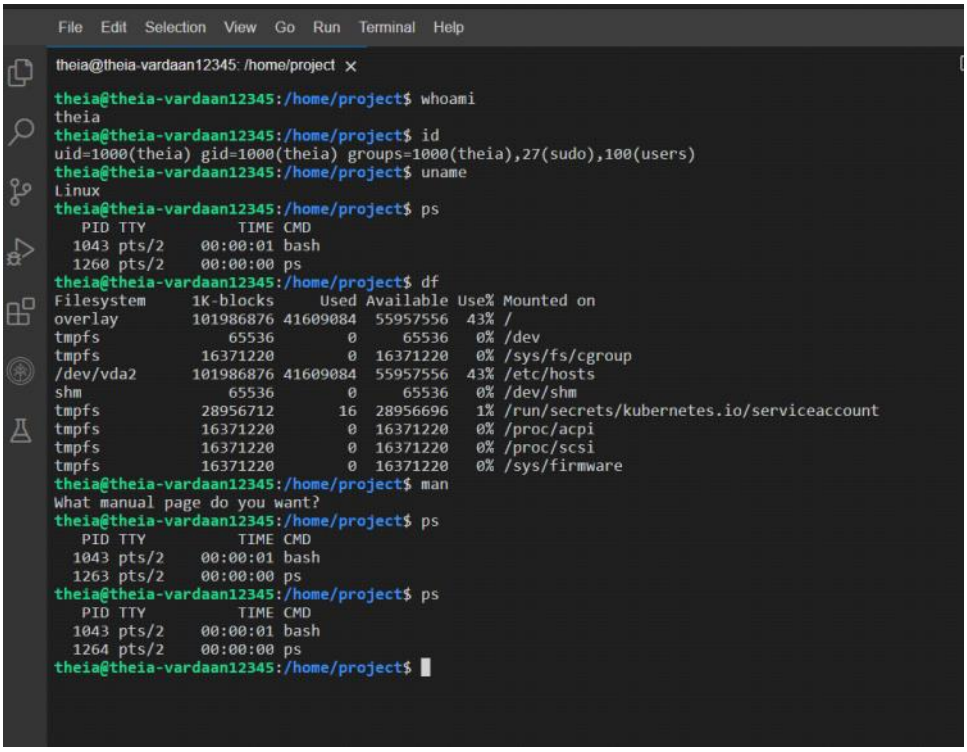
Shell commands related to file compression and archiving applications include:

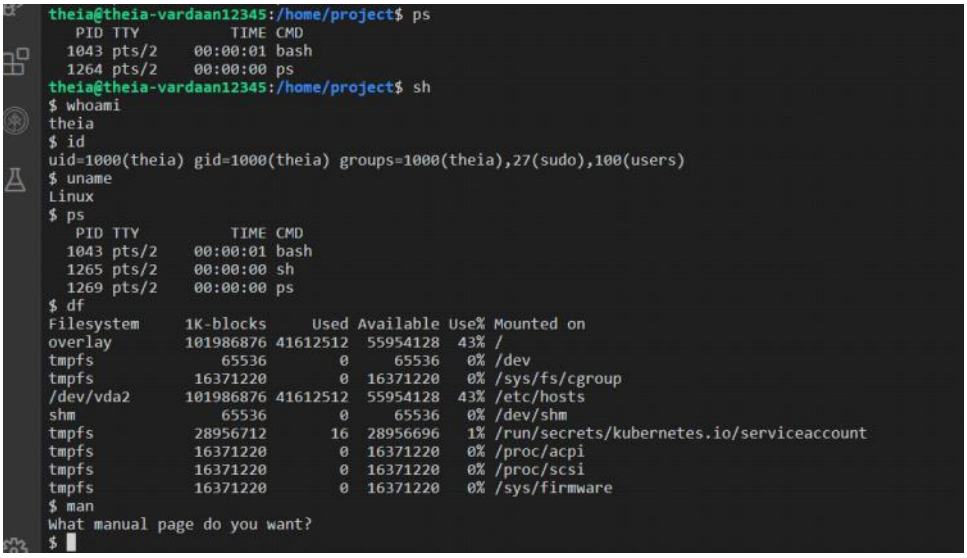
- tar – archive a set of files
- zip – compress a set of files
- unzip – extract files from a compressed zip archive

NETWORKING COMMANDS

Networking applications include the following:

- hostname – print hostname
- ping – send packets to URL and print response
- ifconfig – display or configure system network interfaces
- curl – display contents of file at a URL
- wget – download file from URL





After we do top command

```
theia@theia-vardaan12345:/home/project$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
overlay        101986876 41609084  55957556  43% /
tmpfs              65536        0     65536   0% /dev
tmpfs           16371220        0  16371220   0% /sys/fs/cgroup
/dev/vda2      101986876 41609084  55957556  43% /etc/hosts
shm                65536        0     65536   0% /dev/shm
tmpfs           28956712       16  28956696   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs           16371220        0  16371220   0% /proc/acpi
top - 18:08:16 up 9 days, 12:28,  0 users,  load average: 1.21, 1.43, 1.59
Tasks:  13 total,   1 running,  12 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4.3 us,  3.7 sy,  0.0 ni, 91.4 id,  0.0 wa,  0.0 hi,  0.5 si,  0.1 st
KiB Mem : 32742440 total,  5562896 free, 11381580 used, 15797964 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 21117080 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   59 theia     20   0 1045960  91892  37724 S   1.3  0.3   0:07.77 node
   70 theia     20   0 1095620  77176  32696 S   0.7  0.2   0:16.39 node
    1 theia     20   0    4636    796    724 S   0.0  0.0   0:00.03 sh
    7 theia     20   0   12892   3396   3104 S   0.0  0.0   0:00.01 entrypoint.sh
   35 root      20   0   31324   2912   2632 S   0.0  0.0   0:00.00 cron
   36 theia     20   0  863644  74592  33024 S   0.0  0.2   0:01.22 node
   47 theia     20   0    4644    872    792 S   0.0  0.0   0:00.00 sh
   48 theia     20   0  715812  98284  33040 S   0.0  0.3   0:01.50 node
   85 theia     20   0  868680  67740  34376 S   0.0  0.2   0:01.48 node
  107 theia     20   0  600036  39240  29632 S   0.0  0.1   0:00.29 node
 1043 theia     20   0   25448   7808   3428 S   0.0  0.0   0:01.32 bash
 1265 theia     20   0    4648   1752   1644 S   0.0  0.0   0:00.00 sh
 1302 theia     20   0   41692   3636   3152 R   0.0  0.0   0:00.01 top
```

```
theia@theia-vardaan12345:~$ mkdir my_project
theia@theia-vardaan12345:~$ ls
docker-compose          javasharedresources  package.json  skills-network-extension-v0.1.0.tgz
dsdriver                lib                  plugins       src-gen
entrypoint.sh           my_project           postgres      webpack.config.js
gen-webpack.config.js   node_modules         README.md     yarn.lock
theia@theia-vardaan12345:~$
```

```
theia@theia-vardaan12345:~$ rmdir my_project
theia@theia-vardaan12345:~$ ls
docker-compose          javasharedresources  plugins                          src-gen
dsdriver                lib                  postgres                         webpack.config.js
entrypoint.sh           node_modules         README.md                        yarn.lock
gen-webpack.config.js   package.json         skills-network-extension-v0.1.0.tgz
theia@theia-vardaan12345:~$
```

```
theia@theia-vardaan12345:/$ cd ~/../project
theia@theia-vardaan12345:/home/project$ find prog.py
prog.py
theia@theia-vardaan12345:/home/project$ rmdir prog.py
rmdir: failed to remove 'prog.py': Not a directory
theia@theia-vardaan12345:/home/project$ rmdir pwd
rmdir: failed to remove 'pwd': No such file or directory
theia@theia-vardaan12345:/home/project$ pwd
/home/project
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ nano prog1.py
theia@theia-vardaan12345:/home/project$ python2 prog1.py
h
ht
theia@theia-vardaan12345:/home/project$ cp prog1.py
cp: missing destination file operand after 'prog1.py'
Try 'cp --help' for more information.
theia@theia-vardaan12345:/home/project$ cat prog1.py
print("h")
print("ht")

theia@theia-vardaan12345:/home/project$ more prog1.py
print("h")
print("ht")

theia@theia-vardaan12345:/home/project$ head prog1.py
print("h")
print("ht")

theia@theia-vardaan12345:/home/project$ tail prog1.py
print("h")
print("ht")

theia@theia-vardaan12345:/home/project$ echo

theia@theia-vardaan12345:/home/project$ echo pt
pt
theia@theia-vardaan12345:/home/project$ echo prog1.py
prog1.py
theia@theia-vardaan12345:/home/project$ head prog1.py 1
==> prog1.py <==
print("h")
print("ht")

head: cannot open '1' for reading: No such file or directory
```

```
theia@theia-vardaan12345:/home/project$ head -1 prog1.py
print("h")
theia@theia-vardaan12345:/home/project$ tail -1 prog1.py

theia@theia-vardaan12345:/home/project$ tail -help
tail: invalid option -- 'h'
Try 'tail --help' for more information.
theia@theia-vardaan12345:/home/project$ tail --help
Usage: tail [OPTION]... [FILE]...
Print the last 10 lines of each FILE to standard output.
With more than one FILE, precede each with a header giving the file name.

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.
```

```
theia@theia-vardaan12345:/home/project$ touch hello
theia@theia-vardaan12345:/home/project$ ls
hello  prog1.py  prog.py
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ chmod --version
chmod (GNU coreutils) 8.28
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie and Jim Meyering.
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ wc prog.py
 2  2 27 prog.py
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ grep print prog.py
print("hi")
print("hello")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345: /home/project ×

theia@theia-vardaan12345:/home/project$ touch fle1 file2 file3 file4.py
theia@theia-vardaan12345:/home/project$ ls -ltr
total 4
-rw-r--r-- 1 theia users   0 Jan  5 01:13 file1
-rw-r--r-- 1 theia users 728 Jan  5 01:26 temp.zip
-rw-r--r-- 1 theia users   0 Jan  5 01:49 fle1
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file4.py
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file3
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file2
```

```
theia@theia-vardaan12345:/home/project$ rm file1
theia@theia-vardaan12345:/home/project$ ls -ltr
total 4
-rw-r--r-- 1 theia users 728 Jan  5 01:26 temp.zip
-rw-r--r-- 1 theia users   0 Jan  5 01:49 fle1
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file4.py
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file3
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file2
theia@theia-vardaan12345:/home/project$ rm fle1 file4.py
theia@theia-vardaan12345:/home/project$ ls -ltr
total 4
-rw-r--r-- 1 theia users 728 Jan  5 01:26 temp.zip
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file3
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file2
theia@theia-vardaan12345:/home/project$ r temp.zip file3 file2
bash: r: command not found
theia@theia-vardaan12345:/home/project$ ls -ltr
total 4
-rw-r--r-- 1 theia users 728 Jan  5 01:26 temp.zip
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file3
-rw-r--r-- 1 theia users   0 Jan  5 01:49 file2
theia@theia-vardaan12345:/home/project$ rm file3 file2
theia@theia-vardaan12345:/home/project$ ls -ltr
total 4
-rw-r--r-- 1 theia users 728 Jan  5 01:26 temp.zip
theia@theia-vardaan12345:/home/project$ rm -r temp.zip
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
theia@theia-vardaan12345:/home/project$
```

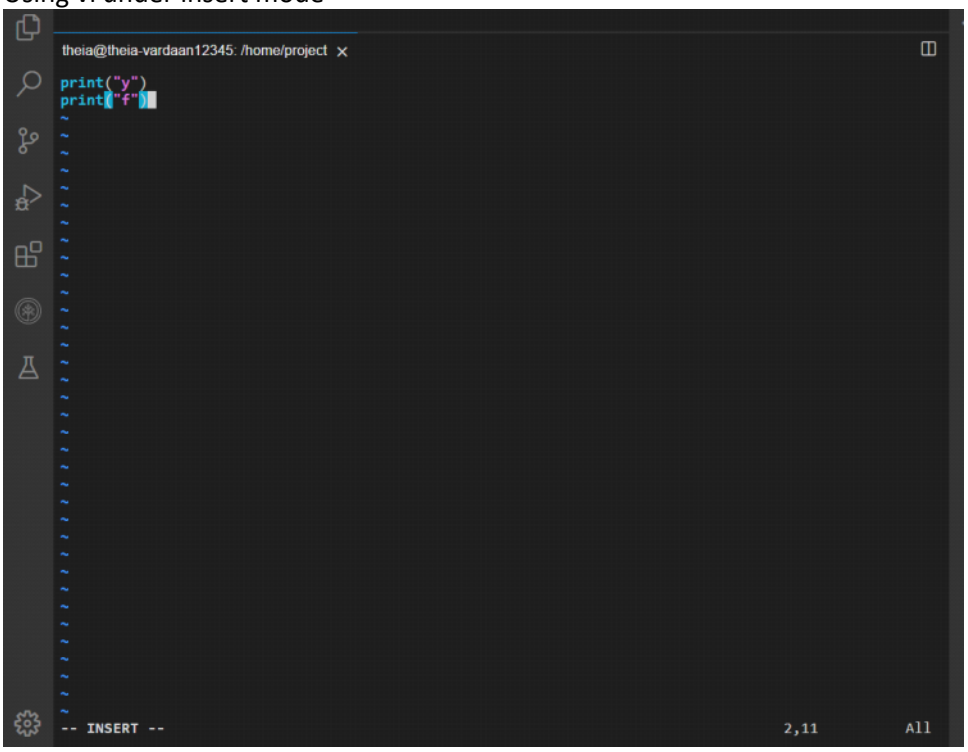
Timestamp of file3.py chnaged

```
theia@theia-vardaan12345:/home/project$ touch file1.py file2 file3.py file4
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file4
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file3.py
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file2
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file1.py
theia@theia-vardaan12345:/home/project$ touch file4.py
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file4
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file3.py
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file2
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file1.py
-rw-r--r-- 1 theia users 0 Jan  5 02:06 file4.py
theia@theia-vardaan12345:/home/project$ touch file3.py
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file4
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file2
-rw-r--r-- 1 theia users 0 Jan  5 01:58 file1.py
-rw-r--r-- 1 theia users 0 Jan  5 02:06 file4.py
-rw-r--r-- 1 theia users 0 Jan  5 02:07 file3.py
theia@theia-vardaan12345:/home/project$
```

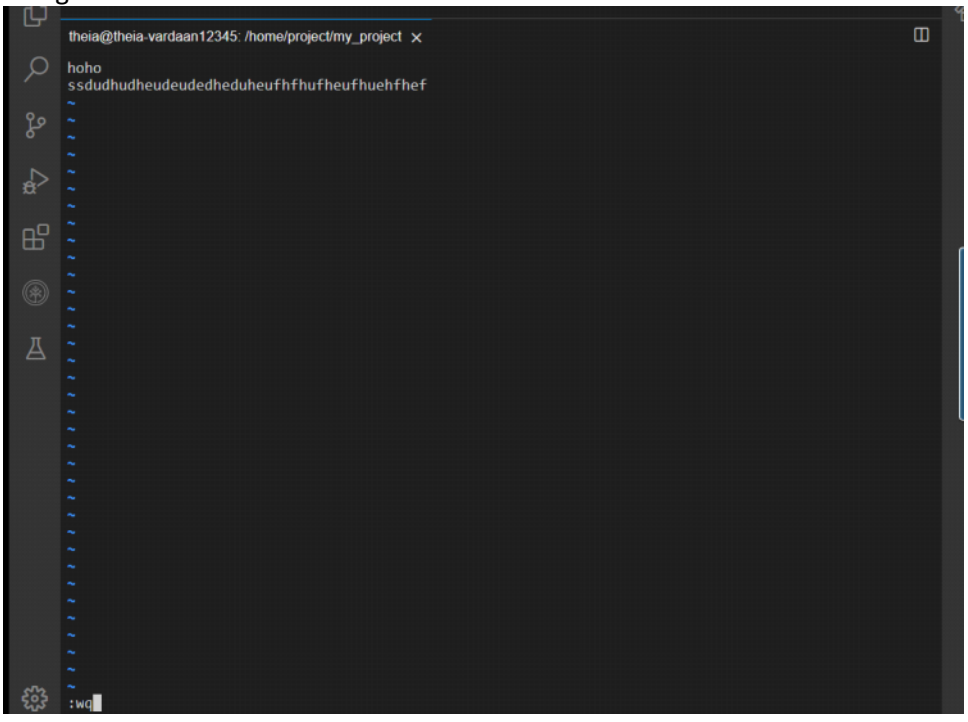
```
theia@theia-vardaan12345:/home/project/my_project$ sudo apt update
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:4 https://packages                                  untu/18.04/prod bionic InRelease
Ign:5 https://storage.  Follow link (ctrl + click)  wnload.dartlang.org/linux/debian stable InRelease
Ign:6 https://storage.googleapis.com/download.dartlang.org/linux/debian testing InRelease
Hit:2 https://apt.llvm.org/bionic llvm-toolchain-bionic InRelease
Ign:7 https://storage.googleapis.com/download.dartlang.org/linux/debian unstable InRelease
Hit:8 https://storage.googleapis.com/download.dartlang.org/linux/debian stable Release
Hit:9 https://storage.googleapis.com/download.dartlang.org/linux/debian testing Release
Hit:10 https://storage.googleapis.com/download.dartlang.org/linux/debian unstable Release
Hit:11 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Hit:14 http://apt.postgresql.org/pub/repos/apt bionic-pgdg InRelease
Hit:15 http://ppa.launchpad.net/ondrej/php/ubuntu bionic InRelease
Hit:1 https://downloads.apache.org/cassandra/debian 311x InRelease
Hit:16 http://ppa.launchpad.net/ubuntu-toolchain-r/test/ubuntu bionic InRelease
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2,344 kB]
Fetched 2,604 kB in 3s (905 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
69 packages can be upgraded. Run 'apt list --upgradable' to see them.
theia@theia-vardaan12345:/home/project/my_project$ sudo apt list --upgradable
Listing... Done
aspnetcore-runtime-3.1/bionic 3.1.32-1 amd64 [upgradable from: 3.1.31-1]
clang-16/unknown 1:16~++20230104064133+352b660c1b05-1~exp1~20230104184235.971 amd64 [upgradable from:
 1:16~++20221206064401+71df24dd3917-1~exp1~20221206184459.913]
clang-tidy-16/unknown 1:16~++20230104064133+352b660c1b05-1~exp1~20230104184235.971 amd64 [upgradable
from: 1:16~++20221206064401+71df24dd3917-1~exp1~20221206184459.913]
clang-tools-16/unknown 1:16~++20230104064133+352b660c1b05-1~exp1~20230104184235.971 amd64 [upgradable
```

```
theia@theia-vardaan12345:/home/project/my_project$ sudo apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
vim is already the newest version (2:8.0.1453-1ubuntu1.9).
0 upgraded, 0 newly installed, 0 to remove and 69 not upgraded.
theia@theia-vardaan12345:/home/project/my_project$ sudo apt install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
nano is already the newest version (2.9.3-2).
0 upgraded, 0 newly installed, 0 to remove and 69 not upgraded.
theia@theia-vardaan12345:/home/project/my_project$
```

Using vi under insert mode



Using vi under command mode



```
theia@theia-vardaan12345:/home/project$ cp file1.py file5
theia@theia-vardaan12345:/home/project$ ls -ltr
total 24
-rw-r--r-- 1 theia users    0 Jan  5 02:06 file4.py
drwxr-sr-x 2 theia users 4096 Jan  5 02:21 my_project
-rw-r--r-- 1 theia users   22 Jan  5 02:22 file1.py
-rw-r--r-- 1 theia users   29 Jan  5 02:22 file2
-rw-r--r-- 1 theia users   20 Jan  5 02:23 file3.py
-rw-r--r-- 1 theia users   17 Jan  5 02:23 file4
-rw-r--r-- 1 theia users   22 Jan  5 02:24 file5
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ ls file5
file5
theia@theia-vardaan12345:/home/project$ cat file5
print("y")
print("f")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ ls file5
file5
theia@theia-vardaan12345:/home/project$ cat file5
print("y")
print("f")
theia@theia-vardaan12345:/home/project$ python3 file5
y
f
theia@theia-vardaan12345:/home/project$ python2 file5
y
f
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat file2
eddededned
edbeudheudheudhed
theia@theia-vardaan12345:/home/project$ cp file5 file2
theia@theia-vardaan12345:/home/project$ ls -ltr
total 24
-rw-r--r-- 1 theia users    0 Jan  5 02:06 file4.py
drwxr-sr-x 2 theia users 4096 Jan  5 02:21 my_project
-rw-r--r-- 1 theia users   22 Jan  5 02:22 file1.py
-rw-r--r-- 1 theia users   20 Jan  5 02:23 file3.py
-rw-r--r-- 1 theia users   17 Jan  5 02:23 file4
-rw-r--r-- 1 theia users   22 Jan  5 02:24 file5
-rw-r--r-- 1 theia users   22 Jan  5 02:32 file2
theia@theia-vardaan12345:/home/project$ cat file2
print("y")
print("f")
theia@theia-vardaan12345:/home/project$ python3 file2
y
f
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat file3.py file2
print("r")
print(5)
print("y")
print("f")
theia@theia-vardaan12345:/home/project$ cat file5
print("y")
print("f")
theia@theia-vardaan12345:/home/project$ cp file2 file3.py file5
cp: target 'file5' is not a directory
theia@theia-vardaan12345:/home/project$ cp file2 ./my_project
theia@theia-vardaan12345:/home/project$ ls my_project
file11.py  file2  file22  file33.py  file44
theia@theia-vardaan12345:/home/project$ ./my_project
bash: ./my_project: Is a directory
theia@theia-vardaan12345:/home/project$ cd ./my_project
theia@theia-vardaan12345:/home/project/my_project$ cat file2
print("y")
print("f")
theia@theia-vardaan12345:/home/project/my_project$
```

Copy 2 files in one directory

```
theia@theia-vardaan12345:/home/project/my_project$ cat file2
print("y")
print("f")
theia@theia-vardaan12345:/home/project/my_project$ cd ..
theia@theia-vardaan12345:/home/project$ cp file1.py file3.py ./my_project
theia@theia-vardaan12345:/home/project$ ls my_project
file11.py  file1.py  file2  file22  file33.py  file3.py  file44
theia@theia-vardaan12345:/home/project$ cd ./my_project
theia@theia-vardaan12345:/home/project/my_project$ cat file3.py
print("r")
print(5)
theia@theia-vardaan12345:/home/project/my_project$
```

Copy one directory into another

```
theia@theia-vardaan12345:/home/project/my_project$ cd ..
theia@theia-vardaan12345:/home/project$ mkdir my_project2
theia@theia-vardaan12345:/home/project$ cp ./my_project ./my_project2
cp: -r not specified; omitting directory './my_project'
theia@theia-vardaan12345:/home/project$ mkdir my_project3
theia@theia-vardaan12345:/home/project$ cp ./my_project2 ./my_project3
cp: -r not specified; omitting directory './my_project2'
theia@theia-vardaan12345:/home/project$ cp -r ./my_project2 ./my_project3
theia@theia-vardaan12345:/home/project$ ls my_project3
my_project2
theia@theia-vardaan12345:/home/project$ cp -r ./my_project1 ./my_project3
cp: cannot stat './my_project1': No such file or directory
theia@theia-vardaan12345:/home/project$ cp -r ./my_project ./my_project3
theia@theia-vardaan12345:/home/project$ ls my_project3
my_project  my_project2
theia@theia-vardaan12345:/home/project$ cd ./my_project3
theia@theia-vardaan12345:/home/project/my_project3$ cat ./my_project/file2
print("y")
print("f")
theia@theia-vardaan12345:/home/project/my_project3$
```

```
theia@theia-vardaan12345:/home/project/my_project3$ cd ./my_project
theia@theia-vardaan12345:/home/project/my_project3/my_project$ ls
file11.py  file1.py  file2  file22  file33.py  file3.py  file44
theia@theia-vardaan12345:/home/project/my_project3/my_project$ cat file3.py
print("r")
print(5)
theia@theia-vardaan12345:/home/project/my_project3/my_project$ cat file33.py

print("hohi")
print("hohihohi")
theia@theia-vardaan12345:/home/project/my_project3/my_project$ mv file33.py filen
theia@theia-vardaan12345:/home/project/my_project3/my_project$ ls
file11.py  file1.py  file2  file22  file3.py  file44  filen
theia@theia-vardaan12345:/home/project/my_project3/my_project$ cat filen

print("hohi")
print("hohihohi")
theia@theia-vardaan12345:/home/project/my_project3/my_project$
```

```
theia@theia-vardaan12345:/home/project/my_project3/my_project$ mv filen file3.py
theia@theia-vardaan12345:/home/project/my_project3/my_project$ ls
file11.py  file1.py  file2  file22  file3.py  file44
theia@theia-vardaan12345:/home/project/my_project3/my_project$ cat file3.py

print("hohi")
print("hohihohi")
theia@theia-vardaan12345:/home/project/my_project3/my_project$
```

```
theia@theia-vardaan12345:/home/project/my_project3/my_project$ python
Python 2.7.17 (default, Jul  1 2022, 15:56:32)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

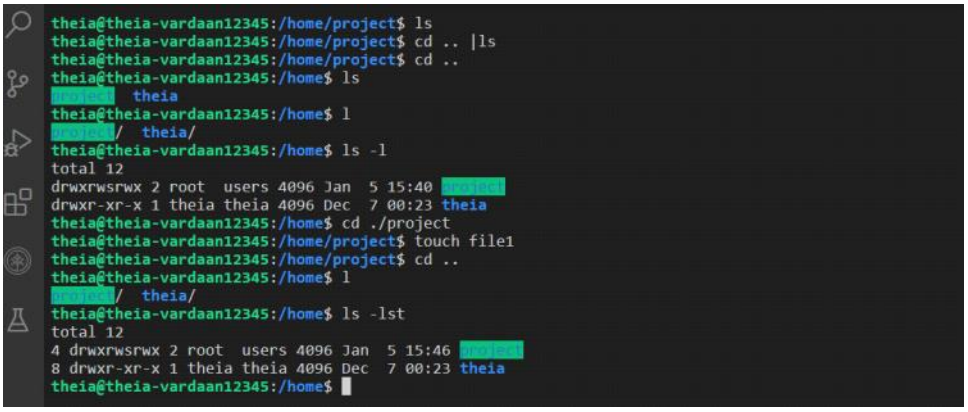
```
theia@theia-vardaan12345:/home/project/my_project3/my_project$ python
Python 2.7.17 (default, Jul  1 2022, 15:56:32)
[GCC 7.5.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
KeyboardInterrupt
>>>
KeyboardInterrupt
>>>
>>> bash
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'bash' is not defined
>>>
KeyboardInterrupt
>>> q
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'q' is not defined
>>> quit()
theia@theia-vardaan12345:/home/project/my_project3/my_project$
```

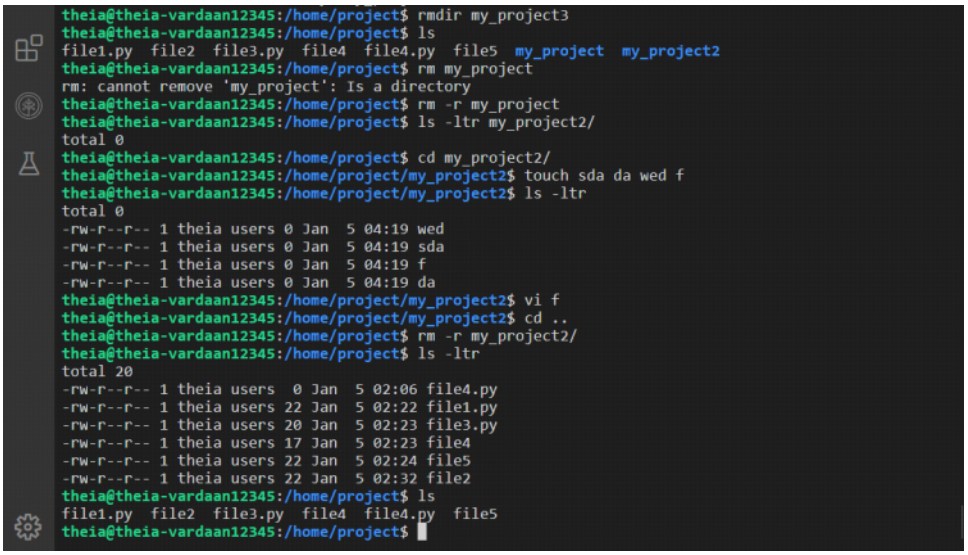
```
theia@theia-vardaan12345:/home/project/my_project3/my_project$ mv -i file3.py file4
theia@theia-vardaan12345:/home/project/my_project3/my_project$ ls
file11.py  file1.py  file2  file22  file4  file44
theia@theia-vardaan12345:/home/project/my_project3/my_project$ cat file44
wdhedhoueduefe
edefbhfbfbfrf
theia@theia-vardaan12345:/home/project/my_project3/my_project$ cat file4

print("hohi")
print("hohihohi")
theia@theia-vardaan12345:/home/project/my_project3/my_project$ python3 file4
hohi
hohihohi
theia@theia-vardaan12345:/home/project/my_project3/my_project$
```

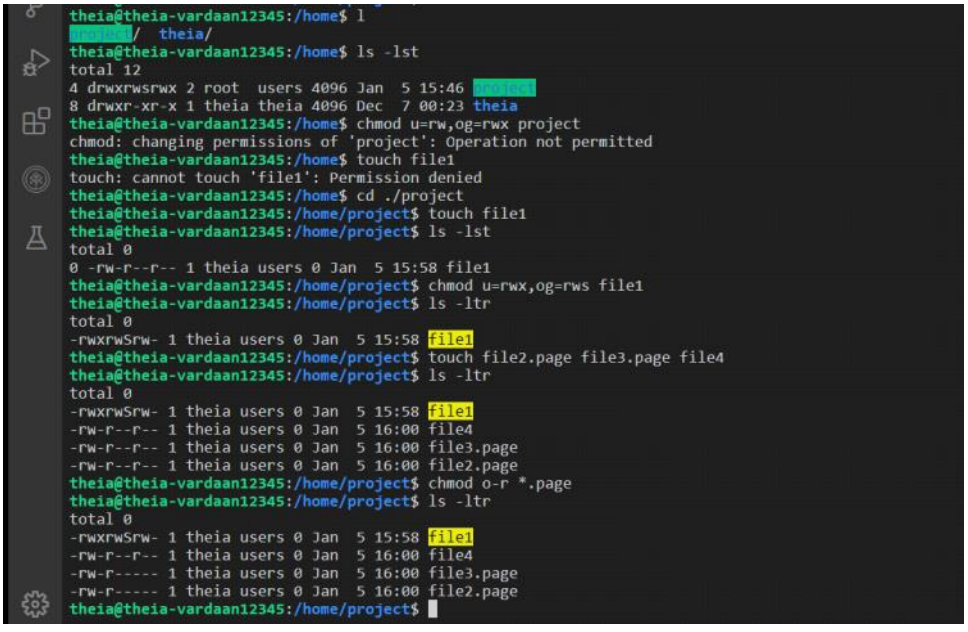
```
theia@theia-vardaan12345:/home/project/my_project3/my_project$ rmdir project1
rmdir: failed to remove 'project1': No such file or directory
theia@theia-vardaan12345:/home/project/my_project3/my_project$ cd ..
theia@theia-vardaan12345:/home/project/my_project3$ rmdir my_project
rmdir: failed to remove 'my_project': Directory not empty
theia@theia-vardaan12345:/home/project/my_project3$ mkdir my_pr
theia@theia-vardaan12345:/home/project/my_project3$ rmdir my_pr
theia@theia-vardaan12345:/home/project/my_project3$
```

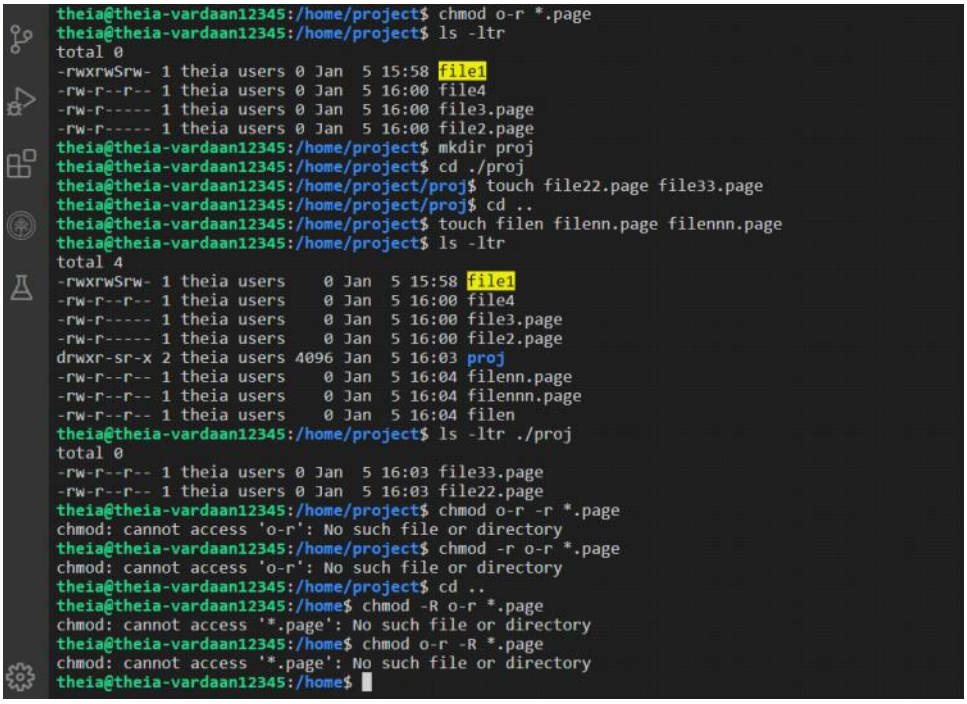
```
theia@theia-vardaan12345:/home/project/my_project3$ rm -r my_project
theia@theia-vardaan12345:/home/project/my_project3$ ls
my_project2
theia@theia-vardaan12345:/home/project/my_project3$
```

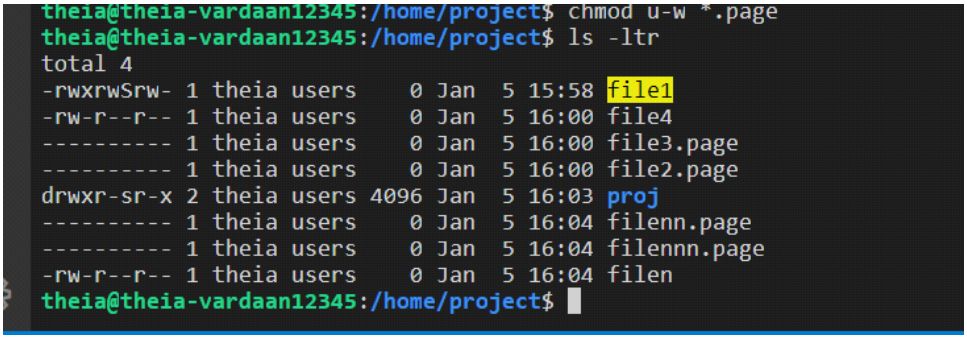
```
theia@theia-vardaan12345:/home/project/my_project3$ cd ./my_project2
theia@theia-vardaan12345:/home/project/my_project3/my_project2$ touch file8.py
theia@theia-vardaan12345:/home/project/my_project3/my_project2$ touch file45.py
theia@theia-vardaan12345:/home/project/my_project3/my_project2$ cd ..
theia@theia-vardaan12345:/home/project/my_project3$ mkdir new_pr
theia@theia-vardaan12345:/home/project/my_project3$ rm -r new_pr my_project
rm: cannot remove 'my_project': No such file or directory
theia@theia-vardaan12345:/home/project/my_project3$ rm -r my_project2 new_pr
rm: cannot remove 'new_pr': No such file or directory
theia@theia-vardaan12345:/home/project/my_project3$ ls
theia@theia-vardaan12345:/home/project/my_project3$ mkdir new
theia@theia-vardaan12345:/home/project/my_project3$ ls
new
theia@theia-vardaan12345:/home/project/my_project3$ mkdir mew
theia@theia-vardaan12345:/home/project/my_project3$ ls
mew  new
theia@theia-vardaan12345:/home/project/my_project3$ rm -r new mew
theia@theia-vardaan12345:/home/project/my_project3$ ls
theia@theia-vardaan12345:/home/project/my_project3$ mkdir new
theia@theia-vardaan12345:/home/project/my_project3$ cd ./new
theia@theia-vardaan12345:/home/project/my_project3/new$ touch file
theia@theia-vardaan12345:/home/project/my_project3/new$ cd ..
theia@theia-vardaan12345:/home/project/my_project3$ mkdir mew
theia@theia-vardaan12345:/home/project/my_project3$ ls
mew  new
theia@theia-vardaan12345:/home/project/my_project3$ rm -r new mew
theia@theia-vardaan12345:/home/project/my_project3$ ls
theia@theia-vardaan12345:/home/project/my_project3$
```

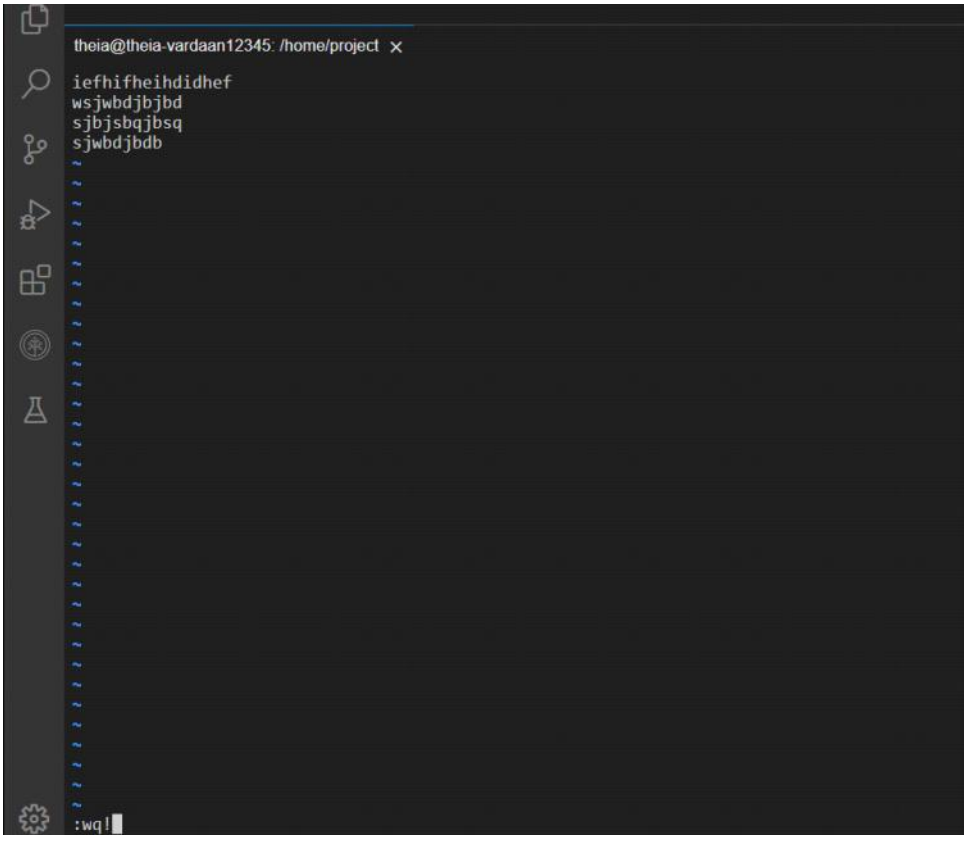

theia@theia-vardaan12345:/home/project$ ls
theia@theia-vardaan12345:/home/project$ cd .. |ls
theia@theia-vardaan12345:/home/project$ cd ..
theia@theia-vardaan12345:/home$ ls
project theia
theia@theia-vardaan12345:/home$ l
project/ theia/
theia@theia-vardaan12345:/home$ ls -l
total 12
drwxrwsrwx 2 root users 4096 Jan 5 15:40 project
drwxr-xr-x 1 theia theia 4096 Dec 7 00:23 theia
theia@theia-vardaan12345:/home$ cd ./project
theia@theia-vardaan12345:/home/project$ touch file1
theia@theia-vardaan12345:/home/project$ cd ..
theia@theia-vardaan12345:/home$ l
project/ theia/
theia@theia-vardaan12345:/home$ ls -lst
total 12
4 drwxrwsrwx 2 root users 4096 Jan 5 15:46 project
8 drwxr-xr-x 1 theia theia 4096 Dec 7 00:23 theia
theia@theia-vardaan12345:/home$


theia@theia-vardaan12345:/home/project$ rmdir my_project3
theia@theia-vardaan12345:/home/project$ ls
file1.py file2 file3.py file4 file4.py file5 my_project my_project2
theia@theia-vardaan12345:/home/project$ rm my_project
rm: cannot remove 'my_project': Is a directory
theia@theia-vardaan12345:/home/project$ rm -r my_project
theia@theia-vardaan12345:/home/project$ ls -ltr my_project2/
total 0
theia@theia-vardaan12345:/home/project$ cd my_project2/
theia@theia-vardaan12345:/home/project/my_project2$ touch sda da wed f
theia@theia-vardaan12345:/home/project/my_project2$ ls -ltr
total 0
-rw-r--r-- 1 theia users 0 Jan 5 04:19 wed
-rw-r--r-- 1 theia users 0 Jan 5 04:19 sda
-rw-r--r-- 1 theia users 0 Jan 5 04:19 f
-rw-r--r-- 1 theia users 0 Jan 5 04:19 da
theia@theia-vardaan12345:/home/project/my_project2$ vi f
theia@theia-vardaan12345:/home/project/my_project2$ cd ..
theia@theia-vardaan12345:/home/project$ rm -r my_project2/
theia@theia-vardaan12345:/home/project$ ls -ltr
total 20
-rw-r--r-- 1 theia users 0 Jan 5 02:06 file4.py
-rw-r--r-- 1 theia users 22 Jan 5 02:22 file1.py
-rw-r--r-- 1 theia users 20 Jan 5 02:23 file3.py
-rw-r--r-- 1 theia users 17 Jan 5 02:23 file4
-rw-r--r-- 1 theia users 22 Jan 5 02:24 file5
-rw-r--r-- 1 theia users 22 Jan 5 02:32 file2
theia@theia-vardaan12345:/home/project$ ls
file1.py file2 file3.py file4 file4.py file5
theia@theia-vardaan12345:/home/project$

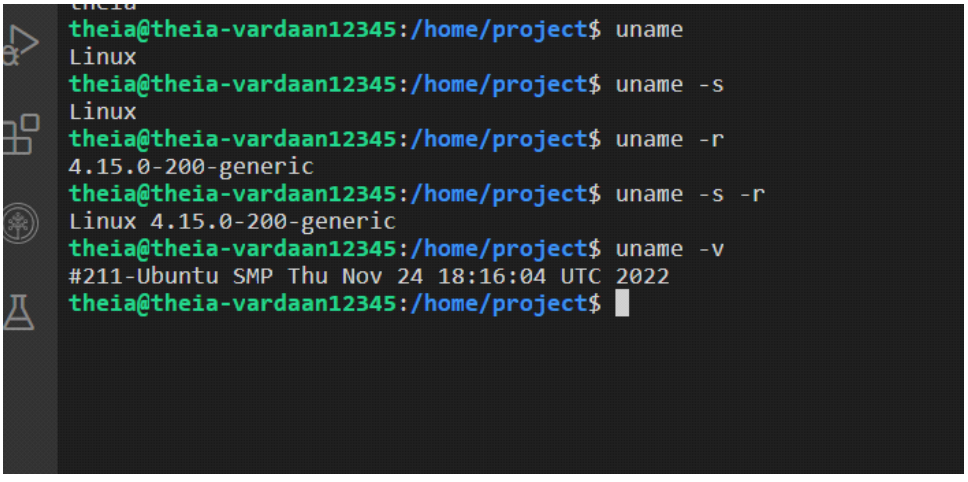

theia@theia-vardaan12345:/home$ l
project/ theia/
theia@theia-vardaan12345:/home$ ls -lst
total 12
4 drwxrwsrwx 2 root users 4096 Jan 5 15:46 project
8 drwxr-xr-x 1 theia theia 4096 Dec 7 00:23 theia
theia@theia-vardaan12345:/home$ chmod u=rw,og=rwx project
chmod: changing permissions of 'project': Operation not permitted
theia@theia-vardaan12345:/home$ touch file1
touch: cannot touch 'file1': Permission denied
theia@theia-vardaan12345:/home$ cd ./project
theia@theia-vardaan12345:/home/project$ touch file1
theia@theia-vardaan12345:/home/project$ ls -lst
total 0
0 -rw-r--r-- 1 theia users 0 Jan 5 15:58 file1
theia@theia-vardaan12345:/home/project$ chmod u=rwx,og=rws file1
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rwxrwSrw- 1 theia users 0 Jan 5 15:58 file1
theia@theia-vardaan12345:/home/project$ touch file2.page file3.page file4
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rwxrwSrw- 1 theia users 0 Jan 5 15:58 file1
-rw-r--r-- 1 theia users 0 Jan 5 16:00 file4
-rw-r--r-- 1 theia users 0 Jan 5 16:00 file3.page
-rw-r--r-- 1 theia users 0 Jan 5 16:00 file2.page
theia@theia-vardaan12345:/home/project$ chmod o-r *.page
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rwxrwSrw- 1 theia users 0 Jan 5 15:58 file1
-rw-r--r-- 1 theia users 0 Jan 5 16:00 file4
-rw-r----- 1 theia users 0 Jan 5 16:00 file3.page
-rw-r----- 1 theia users 0 Jan 5 16:00 file2.page
theia@theia-vardaan12345:/home/project$

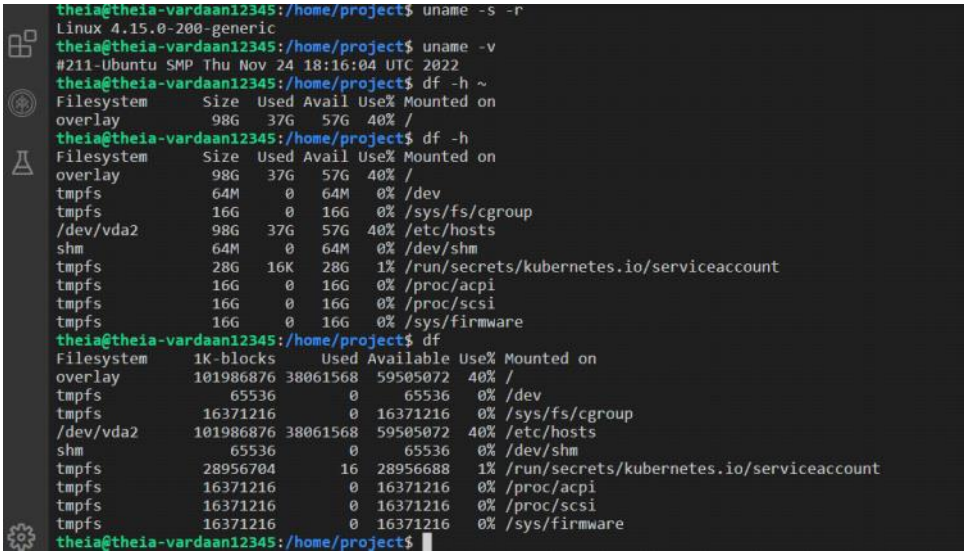

theia@theia-vardaan12345:/home/project$ chmod o-r *.page
theia@theia-vardaan12345:/home/project$ ls -ltr
total 0
-rwxrwSrw- 1 theia users 0 Jan 5 15:58 file1
-rw-r--r-- 1 theia users 0 Jan 5 16:00 file4
-rw-r----- 1 theia users 0 Jan 5 16:00 file3.page
-rw-r----- 1 theia users 0 Jan 5 16:00 file2.page
theia@theia-vardaan12345:/home/project$ mkdir proj
theia@theia-vardaan12345:/home/project$ cd ./proj
theia@theia-vardaan12345:/home/project/proj$ touch file22.page file33.page
theia@theia-vardaan12345:/home/project/proj$ cd ..
theia@theia-vardaan12345:/home/project$ touch filen filenn.page filennn.page
theia@theia-vardaan12345:/home/project$ ls -ltr
total 4
-rwxrwSrw- 1 theia users 0 Jan 5 15:58 file1
-rw-r--r-- 1 theia users 0 Jan 5 16:00 file4
-rw-r----- 1 theia users 0 Jan 5 16:00 file3.page
-rw-r----- 1 theia users 0 Jan 5 16:00 file2.page
drwxr-sr-x 2 theia users 4096 Jan 5 16:03 proj
-rw-r--r-- 1 theia users 0 Jan 5 16:04 filenn.page
-rw-r--r-- 1 theia users 0 Jan 5 16:04 filennn.page
-rw-r--r-- 1 theia users 0 Jan 5 16:04 filen
theia@theia-vardaan12345:/home/project$ ls -ltr ./proj
total 0
-rw-r--r-- 1 theia users 0 Jan 5 16:03 file33.page
-rw-r--r-- 1 theia users 0 Jan 5 16:03 file22.page
theia@theia-vardaan12345:/home/project$ chmod o-r -r *.page
chmod: cannot access 'o-r': No such file or directory
theia@theia-vardaan12345:/home/project$ chmod -r o-r *.page
chmod: cannot access 'o-r': No such file or directory
theia@theia-vardaan12345:/home/project$ cd ..
theia@theia-vardaan12345:/home$ chmod -R o-r *.page
chmod: cannot access '*.page': No such file or directory
theia@theia-vardaan12345:/home$ chmod o-r -R *.page
chmod: cannot access '*.page': No such file or directory
theia@theia-vardaan12345:/home$

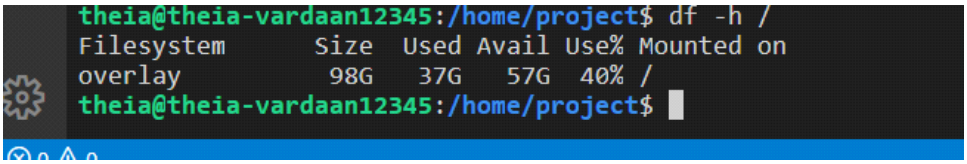

theia@theia-vardaan12345:/home/project$ chmod u-w *.page
theia@theia-vardaan12345:/home/project$ ls -ltr
total 4
-rwxrwSrw- 1 theia users 0 Jan 5 15:58 file1
-rw-r--r-- 1 theia users 0 Jan 5 16:00 file4
---------- 1 theia users 0 Jan 5 16:00 file3.page
---------- 1 theia users 0 Jan 5 16:00 file2.page
drwxr-sr-x 2 theia users 4096 Jan 5 16:03 proj
---------- 1 theia users 0 Jan 5 16:04 filenn.page
---------- 1 theia users 0 Jan 5 16:04 filennn.page
-rw-r--r-- 1 theia users 0 Jan 5 16:04 filen
theia@theia-vardaan12345:/home/project$

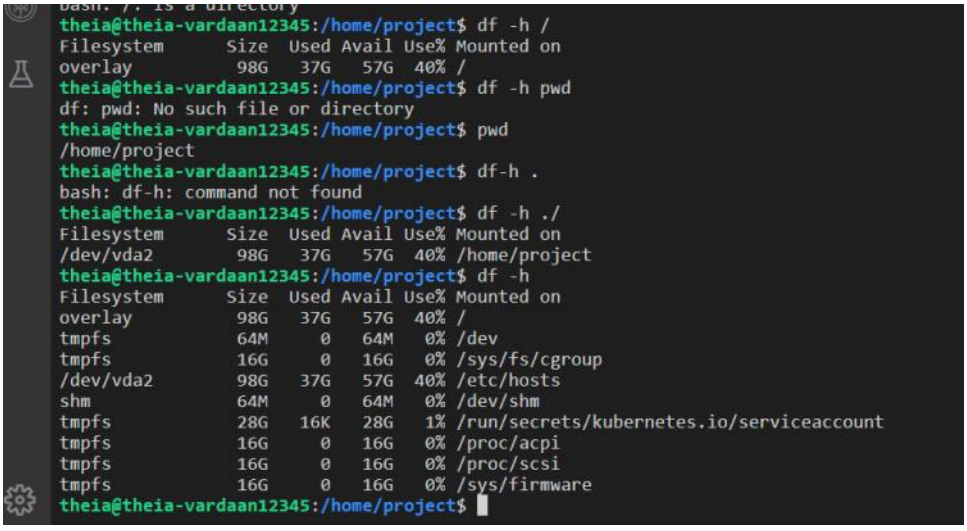

theia@theia-vardaan12345: /home/project
iefhifheihdidhef
wsjwbdjbjbd
sjbjsbqjbsq
sjwbdjbdb
:wq!

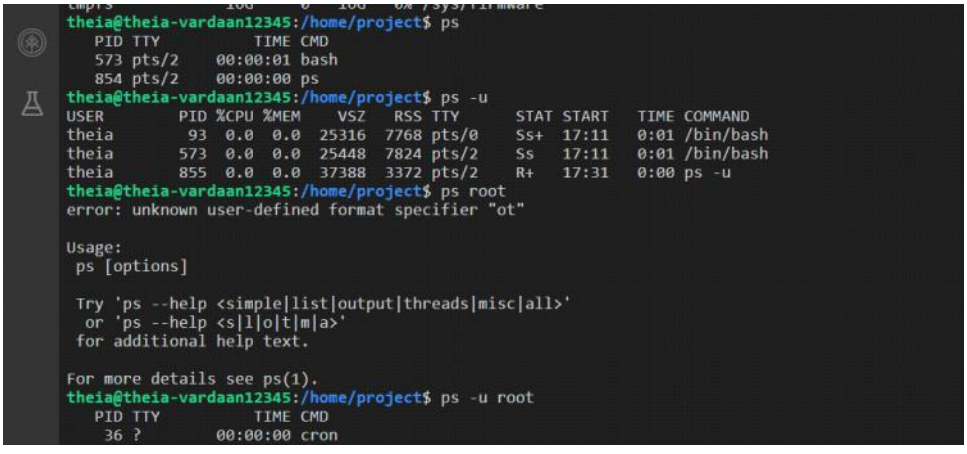

theia@theia-vardaan12345:/home/project$ whoami
theia
theia@theia-vardaan12345:/home/project$ id -u
1000
theia@theia-vardaan12345:/home/project$ id -u -n
theia
theia@theia-vardaan12345:/home/project$


theia@theia-vardaan12345:/home/project$ uname
Linux
theia@theia-vardaan12345:/home/project$ uname -s
Linux
theia@theia-vardaan12345:/home/project$ uname -r
4.15.0-200-generic
theia@theia-vardaan12345:/home/project$ uname -s -r
Linux 4.15.0-200-generic
theia@theia-vardaan12345:/home/project$ uname -v
#211-Ubuntu SMP Thu Nov 24 18:16:04 UTC 2022
theia@theia-vardaan12345:/home/project$









Running top –n 3

```
top - 17:38:51 up 2 days,  6:45,  0 users,  load average: 4.03, 2.29, 1.71
Tasks:  15 total,   1 running,  14 sleeping,   0 stopped,   0 zombie
top - 17:39:09 up 2 days,  6:46,  0 users,  load average: 4.49, 2.51, 1.80
Tasks:  15 total,   1 running,  14 sleeping,   0 stopped,   0 zombie
%Cpu(s): 37.2 us,  7.5 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi,  0.6 si,  0.0 st
KiB Mem : 32742432 total,  5791340 free,  8742392 used, 18208700 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 23794724 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   71 theia     20   0 1089948  71768  32748 S   0.7  0.2   0:08.53 node
   60 theia     20   0 1047852  94684  37720 S   0.3  0.3   0:07.01 node
    1 theia     20   0    4636    844    776 S   0.0  0.0   0:00.02 sh
    8 theia     20   0   12892   3300   3008 S   0.0  0.0   0:00.01 entrypoint.sh
   36 root      20   0   31324   2856   2572 S   0.0  0.0   0:00.00 cron
   37 theia     20   0  863644  74940  32660 S   0.0  0.2   0:00.95 node
   48 theia     20   0    4644    856    780 S   0.0  0.0   0:00.00 sh
   49 theia     20   0  716056  98408  33152 S   0.0  0.3   0:00.98 node
   84 theia     20   0  887132  69636  34488 S   0.0  0.2   0:00.89 node
   93 theia     20   0   25316   7768   3448 S   0.0  0.0   0:01.13 bash
  308 theia     20   0  886828  70184  34356 S   0.0  0.2   0:00.97 node
  528 theia     20   0  600040  38752  29392 S   0.0  0.1   0:00.22 node
  554 theia     20   0  600808  43356  32508 S   0.0  0.1   0:00.24 node
  573 theia     20   0   25448   7824   3440 S   0.0  0.0   0:01.02 bash
  886 theia     20   0   41692   3608   3124 R   0.0  0.0   0:00.00 top
```

```
theia@theia-vardaan12345:/home/project$ echo

theia@theia-vardaan12345:/home/project$ echo hi
hi
theia@theia-vardaan12345:/home/project$ echo "hi hello"
hi hello
theia@theia-vardaan12345:/home/project$ echo $PATH
/home/theia/.local/bin:/home/theia/.asdf/shims:/home/theia/.asdf/bin:/home/theia/.kubectx:/tmp/yarn--
1672956664655-0.19455006404091946:/home/theia/node_modules/.bin:/home/theia/.config/yarn/link/node_mo
dules/.bin:/home/theia/.yarn/bin:/usr/local/nvm/versions/node/v16.13.0/libexec/lib/node_modules/npm/b
in/node-gyp-bin:/usr/local/nvm/versions/node/v16.13.0/lib/node_modules/npm/bin/node-gyp-bin:/usr/loca
l/nvm/versions/node/v16.13.0/bin:/opt/maven/bin:/home/theia/dsdriver/adm:/home/theia/dsdriver/bin:/us
r/lib/dart/bin:/theia/.pub-cache/bin:/usr/local/cargo/bin:/usr/local/go/bin:/usr/local/go-packages/bi
n:/usr/local/nvm/versions/node/v16.13.0/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin:/usr/local/go-packages/bin:/home/theia/.local/bin
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ date
Thu Jan  5 17:48:33 EST 2023
theia@theia-vardaan12345:/home/project$ date %j
date: invalid date ‘%j’
theia@theia-vardaan12345:/home/project$ date "%j"
date: invalid date ‘%j’
theia@theia-vardaan12345:/home/project$ date "%A"
date: invalid date ‘%A’
theia@theia-vardaan12345:/home/project$ date "day of %A"
date: invalid date ‘day of %A’
theia@theia-vardaan12345:/home/project$ date +%j
005
theia@theia-vardaan12345:/home/project$ date +%A
Thursday
theia@theia-vardaan12345:/home/project$ date +%Y
2023
theia@theia-vardaan12345:/home/project$ date "+%Y"
2023
theia@theia-vardaan12345:/home/project$ date "+%j"
005
theia@theia-vardaan12345:/home/project$ date "+%A"
Thursday
theia@theia-vardaan12345:/home/project$ date +date %A
date: extra operand ‘%A’
Try 'date --help' for more information.
theia@theia-vardaan12345:/home/project$ date +hi %A
date: extra operand ‘%A’
Try 'date --help' for more information.
theia@theia-vardaan12345:/home/project$ date "+date %j"
date 005
theia@theia-vardaan12345:/home/project$ date "+date= %j and day= %A and year= %Y"
date= 005 and day= Thursday and year= 2023
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ man id
theia@theia-vardaan12345:/home/project$
theia@theia-vardaan12345:/home/project$ man id
theia@theia-vardaan12345:/home/project$ man ps
No manual entry for ps
See 'man 7 undocumented' for help when manual pages are not available.
theia@theia-vardaan12345:/home/project$ man top
No manual entry for top
See 'man 7 undocumented' for help when manual pages are not available.
theia@theia-vardaan12345:/home/project$ man df
theia@theia-vardaan12345:/home/project$
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ date -r file1.txt
Thu Jan  5 18:53:28 EST 2023
theia@theia-vardaan12345:/home/project$ touch file1.txt
theia@theia-vardaan12345:/home/project$ date -r file1.txt
Thu Jan  5 19:17:39 EST 2023
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ touch file1.txt File1.txt
theia@theia-vardaan12345:/home/project$ find -name file1.txt
./file1.txt
theia@theia-vardaan12345:/home/project$ find -iname file1.txt
./File1.txt
./file1.txt
theia@theia-vardaan12345:/home/project$ cd ..
theia@theia-vardaan12345:/home$ touch file1.txt
touch: cannot touch 'file1.txt': Permission denied
theia@theia-vardaan12345:/home$ cd ./project
theia@theia-vardaan12345:/home/project$ mkdir my_pr
theia@theia-vardaan12345:/home/project$ touch file1.txt
theia@theia-vardaan12345:/home/project$ touch File1.txt
theia@theia-vardaan12345:/home/project$ ls
file1.txt  File1.txt  my_pr
theia@theia-vardaan12345:/home/project$ cd ./my_pr
theia@theia-vardaan12345:/home/project/my_pr$ touch file1.txt File1.txt
theia@theia-vardaan12345:/home/project/my_pr$ find file1.txt
file1.txt
theia@theia-vardaan12345:/home/project/my_pr$ find -name file1.txt
./file1.txt
theia@theia-vardaan12345:/home/project/my_pr$ find -iname file1.txt
./File1.txt
./file1.txt
theia@theia-vardaan12345:/home/project/my_pr$ cd ..
theia@theia-vardaan12345:/home/project$ find -name file1.txt
./file1.txt
./my_pr/file1.txt
theia@theia-vardaan12345:/home/project$ find -iname file1.txt
./File1.txt
./file1.txt
./my_pr/File1.txt
./my_pr/file1.txt
theia@theia-vardaan12345:/home/project$ find . -name file1.txt
./file1.txt
./my_pr/file1.txt
theia@theia-vardaan12345:/home/project$ find -r -name file1.txt
find: unknown predicate `-r'
```

```
theia@theia-vardaan12345:/home/project$ touch b.txt
theia@theia-vardaan12345:/home/project$ find -name file1.txt b.txt
find: paths must precede expression: `b.txt'
find: possible unquoted pattern after predicate `-name'?
theia@theia-vardaan12345:/home/project$ find -name b.txt
./b.txt
theia@theia-vardaan12345:/home/project$ find -name "file1.txt b.txt"
theia@theia-vardaan12345:/home/project$ touch "gil jjj.txt"
theia@theia-vardaan12345:/home/project$ ls
 b.txt   file1.txt   File1.txt  'gil jjj.txt'   my_pr
theia@theia-vardaan12345:/home/project$ find "gil jjj.txt"
gil jjj.txt
theia@theia-vardaan12345:/home/project$ find -name "gil jjj.txt"
./gil jjj.txt
theia@theia-vardaan12345:/home/project$ find k.txt
find: ‘k.txt’: No such file or directory
theia@theia-vardaan12345:/home/project$ ls
 b.txt   file1.txt   File1.txt  'gil jjj.txt'   my_pr
theia@theia-vardaan12345:/home/project$ rm -rf my_pr
theia@theia-vardaan12345:/home/project$ ls
 b.txt   file1.txt   File1.txt  'gil jjj.txt'
theia@theia-vardaan12345:/home/project$ mkdir my_pr
theia@theia-vardaan12345:/home/project$ ls
 b.txt   file1.txt   File1.txt  'gil jjj.txt'   my_pr
theia@theia-vardaan12345:/home/project$ cd ./my_pr
theia@theia-vardaan12345:/home/project/my_pr$ touch file1.txt
theia@theia-vardaan12345:/home/project/my_pr$ ls
file1.txt
theia@theia-vardaan12345:/home/project/my_pr$ cd ..
theia@theia-vardaan12345:/home/project$ rmdir my_pr
rmdir: failed to remove 'my_pr': Directory not empty
theia@theia-vardaan12345:/home/project$ date -r file1.txt
Thu Jan  5 18:53:28 EST 2023
theia@theia-vardaan12345:/home/project$ touch file1.txt
theia@theia-vardaan12345:/home/project$ date -r file1.txt
Thu Jan  5 19:17:39 EST 2023
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ ls
 b.txt   file1.txt   File1.txt  'gil jjj.txt'   mm   my_pr   my_pr2
theia@theia-vardaan12345:/home/project$ cp -r my_pr newnew
theia@theia-vardaan12345:/home/project$ ls
 b.txt   file1.txt   File1.txt  'gil jjj.txt'   mm   my_pr   my_pr2   newnew
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("hi")
print("hello")
theia@theia-vardaan12345:/home/project$ cat file1.txt | 's/hi/hellohi/g'
bash: s/hi/hellohi/g: No such file or directory
theia@theia-vardaan12345:/home/project$ cat file1.txt | sed 's/hello/hihi/g'
print("hi")
print("hihi")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("hi")
print("hello")
theia@theia-vardaan12345:/home/project$ cat file1.txt | 's/hi/hellohi/g'
bash: s/hi/hellohi/g: No such file or directory
theia@theia-vardaan12345:/home/project$ cat file1.txt | sed 's/hello/hihi/g'
print("hi")
print("hihi")
theia@theia-vardaan12345:/home/project$ cat file1.txt | 's/hi/jj/g' > dad
bash: s/hi/jj/g: No such file or directory
theia@theia-vardaan12345:/home/project$ cat file1.txt | sed 's/hi/jj/g' > dad
theia@theia-vardaan12345:/home/project$ ls
 b.txt   dad   file1.txt   File1.txt  'gil jjj.txt'   mm   my_pr   my_pr2   newnew
theia@theia-vardaan12345:/home/project$ ls dad
dad
theia@theia-vardaan12345:/home/project$ cat dad
print("jj")
print("hello")
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("hi")
print("hello")
theia@theia-vardaan12345:/home/project$ cat File1.txt
theia@theia-vardaan12345:/home/project$ nano File1.txt
theia@theia-vardaan12345:/home/project$ cat File1.txt
fijefiefjeifef
eidheifhefh
theia@theia-vardaan12345:/home/project$ cat file1.txt|sed 's/hi/jj/g' > File1.txt
theia@theia-vardaan12345:/home/project$ cat File1.txt
print("jj")
print("hello")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat file1.txt File1.txt
print("hi")
print("hello")
print("jj")
print("hello")
theia@theia-vardaan12345:/home/project$ cat file1.txt | sed 's/hello/ygt/g' >> File1.txt
theia@theia-vardaan12345:/home/project$ cat File1.txt
print("jj")
print("hello")
print("hi")
print("ygt")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ sed 's/he/uhrhgr/g' file1.txt
print("hi")
print("uhrhgrllo")
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("hi")
print("hello")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("hi")
print("hello")
theia@theia-vardaan12345:/home/project$ sed -i 's/hi/ygygj/g' file1.txt
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("ygygj")
print("hello")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ sed 's/hello/yt/g' file1.txt >sdeintern
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("ygygj")
print("hello")
theia@theia-vardaan12345:/home/project$ cat sdeintern
print("ygygj")
print("yt")
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("ygygj")
print("hello")
theia@theia-vardaan12345:/home/project$ cat sdeintern
print("ygygj")
print("yt")
theia@theia-vardaan12345:/home/project$ sed 's/hello/ytyt/g' file1.txt >> sdeintern
theia@theia-vardaan12345:/home/project$ cat sdeintern
print("ygygj")
print("yt")
print("ygygj")
print("ytyt")
theia@theia-vardaan12345:/home/project$ cat file1.txt
print("ygygj")
print("hello")
theia@theia-vardaan12345:/home/project$
```

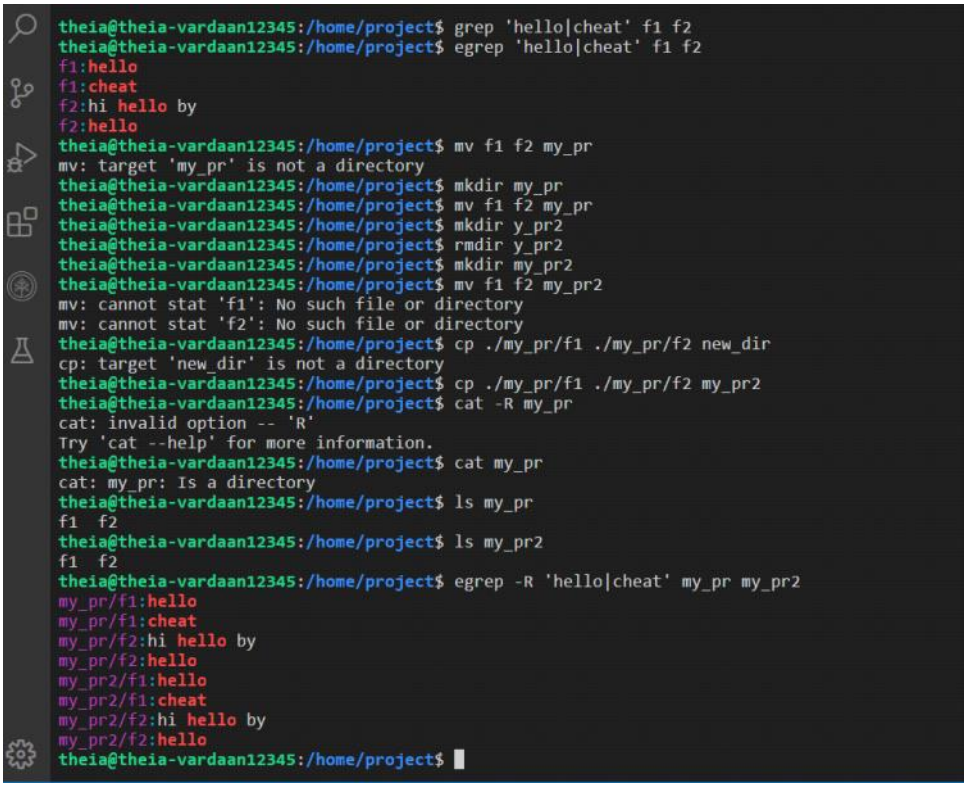
```
theia@theia-vardaan12345:/home/project/my2$ ls -ltr ./..
total 8
drwxr-sr-x 2 theia users 4096 Jan  5 23:37 my1
drwxr-sr-x 2 theia users 4096 Jan  5 23:38 my2
theia@theia-vardaan12345:/home/project/my2$
```

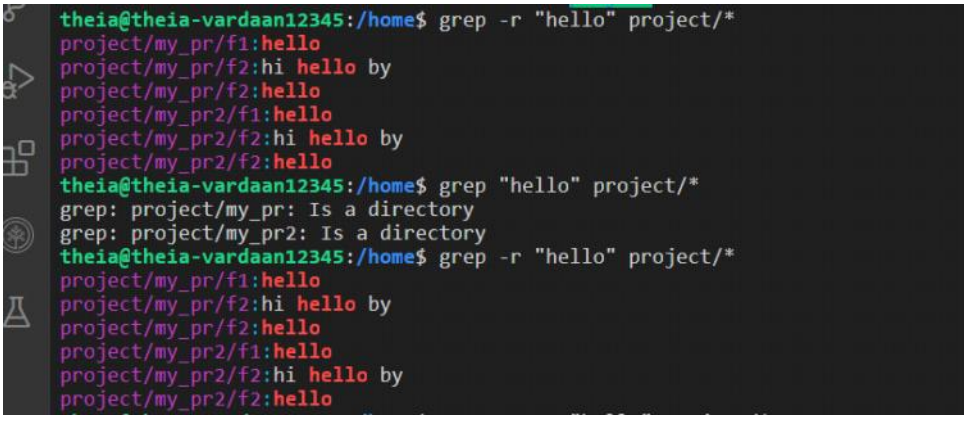
```
theia@theia-vardaan12345:/home/project/my2$ cd ..
theia@theia-vardaan12345:/home/project$ find -type d -empty
./my1
theia@theia-vardaan12345:/home/project$ find -type d
.
./my1
./my2
theia@theia-vardaan12345:/home/project$ find -type f
./my2/f2.txt
./my2/f1.py
theia@theia-vardaan12345:/home/project$ find -type f -empty
theia@theia-vardaan12345:/home/project$ touch new.txt
theia@theia-vardaan12345:/home/project$ find -type f -empty
./new.txt
theia@theia-vardaan12345:/home/project$
```

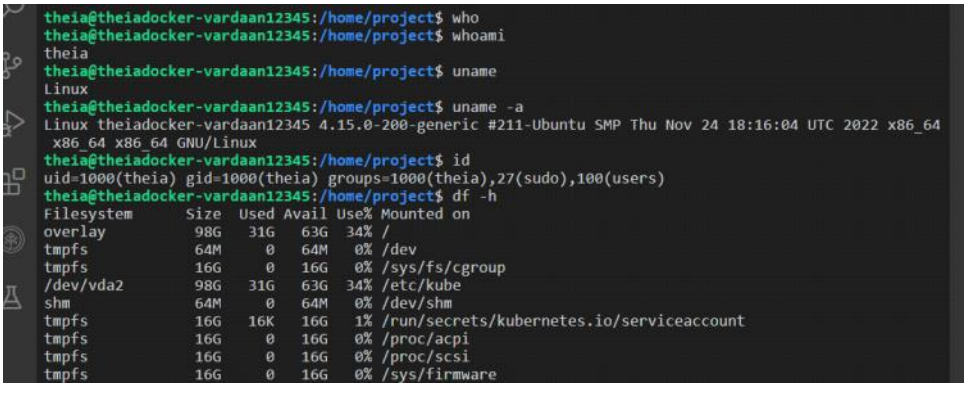
```
theia@theia-vardaan12345:/home/project$ touch f1 f2
theia@theia-vardaan12345:/home/project$ nano f1
theia@theia-vardaan12345:/home/project$ nano f2
theia@theia-vardaan12345:/home/project$ grep -Rw hello *
f1:hello
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$ grep -Rw 'hello' *
f1:hello
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$ grep -Rw 'hello'
f1:hello
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$ grep -Rw hello
f1:hello
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$
```

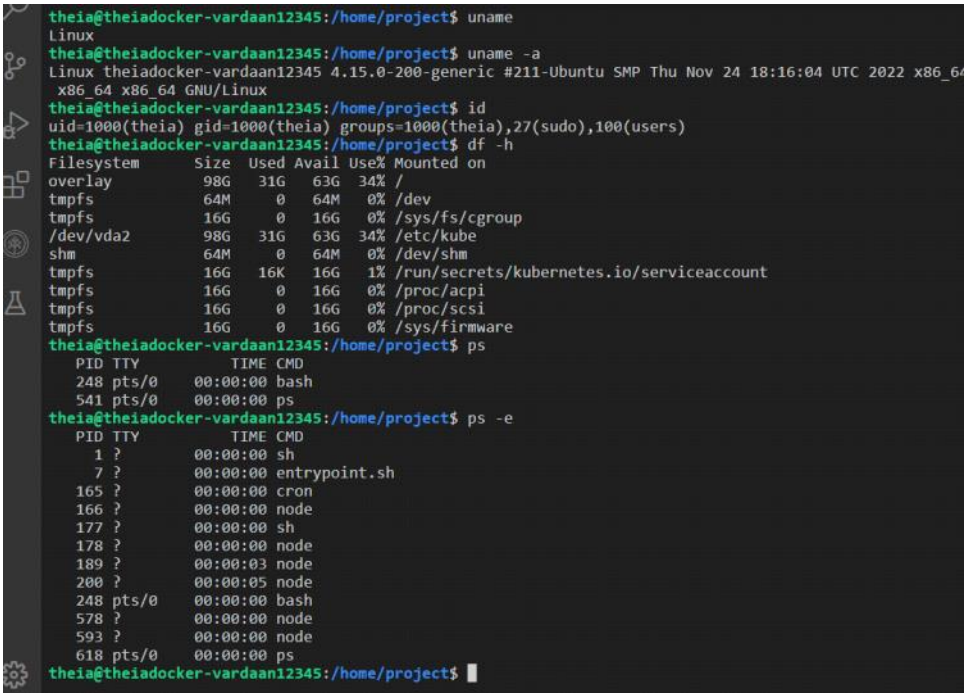
```
theia@theia-vardaan12345:/home/project$ nano f1
theia@theia-vardaan12345:/home/project$ cat f1
hi
by
hello
cheat
theia@theia-vardaan12345:/home/project$ grep -Rw cheat
f1:cheat
theia@theia-vardaan12345:/home/project$ grep -R cheat
f1:cheat
theia@theia-vardaan12345:/home/project$ grep -R by
f1:by
f2:hi hello by
f2:by by
theia@theia-vardaan12345:/home/project$
```

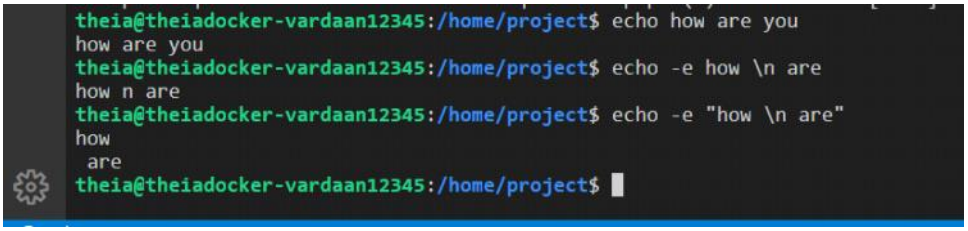
```
theia@theia-vardaan12345:/home/project$ grep -R 'hello'
f1:hello
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$ grep -R 'hello|cheat' *
theia@theia-vardaan12345:/home/project$ egrep -Rw 'hello|cheat' *
f1:hello
f1:cheat
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$ egrep -R 'hello|cheat' *
f1:hello
f1:cheat
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$
```

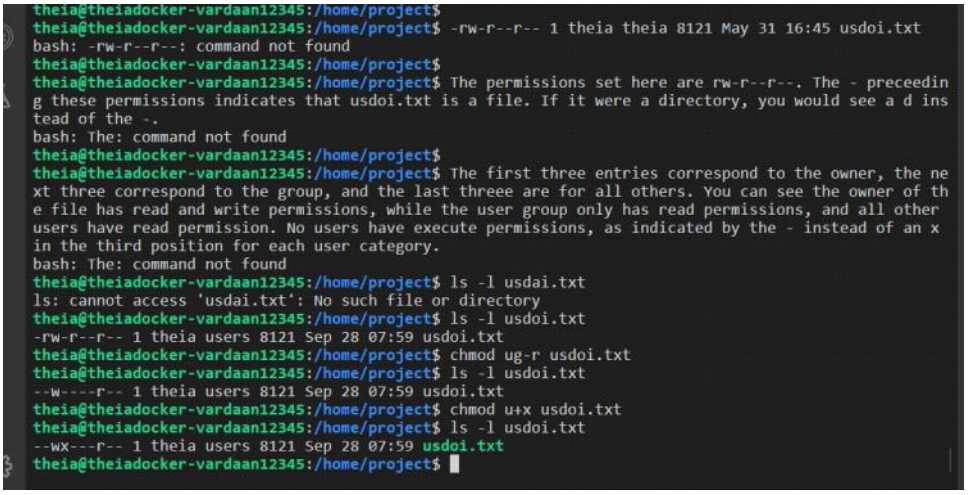

theia@theia-vardaan12345:/home/project$ grep 'hello|cheat' f1 f2
theia@theia-vardaan12345:/home/project$ egrep 'hello|cheat' f1 f2
f1:hello
f1:cheat
f2:hi hello by
f2:hello
theia@theia-vardaan12345:/home/project$ mv f1 f2 my_pr
mv: target 'my_pr' is not a directory
theia@theia-vardaan12345:/home/project$ mkdir my_pr
theia@theia-vardaan12345:/home/project$ mv f1 f2 my_pr
theia@theia-vardaan12345:/home/project$ mkdir y_pr2
theia@theia-vardaan12345:/home/project$ rmdir y_pr2
theia@theia-vardaan12345:/home/project$ mkdir my_pr2
theia@theia-vardaan12345:/home/project$ mv f1 f2 my_pr2
mv: cannot stat 'f1': No such file or directory
mv: cannot stat 'f2': No such file or directory
theia@theia-vardaan12345:/home/project$ cp ./my_pr/f1 ./my_pr/f2 new_dir
cp: target 'new_dir' is not a directory
theia@theia-vardaan12345:/home/project$ cp ./my_pr/f1 ./my_pr/f2 my_pr2
theia@theia-vardaan12345:/home/project$ cat -R my_pr
cat: invalid option -- 'R'
Try 'cat --help' for more information.
theia@theia-vardaan12345:/home/project$ cat my_pr
cat: my_pr: Is a directory
theia@theia-vardaan12345:/home/project$ ls my_pr
f1 f2
theia@theia-vardaan12345:/home/project$ ls my_pr2
f1 f2
theia@theia-vardaan12345:/home/project$ egrep -R 'hello|cheat' my_pr my_pr2
my_pr/f1:hello
my_pr/f1:cheat
my_pr/f2:hi hello by
my_pr/f2:hello
my_pr2/f1:hello
my_pr2/f1:cheat
my_pr2/f2:hi hello by
my_pr2/f2:hello
theia@theia-vardaan12345:/home/project$


theia@theia-vardaan12345:/home$ grep -r "hello" project/*
project/my_pr/f1:hello
project/my_pr/f2:hi hello by
project/my_pr/f2:hello
project/my_pr2/f1:hello
project/my_pr2/f2:hi hello by
project/my_pr2/f2:hello
theia@theia-vardaan12345:/home$ grep "hello" project/*
grep: project/my_pr: Is a directory
grep: project/my_pr2: Is a directory
theia@theia-vardaan12345:/home$ grep -r "hello" project/*
project/my_pr/f1:hello
project/my_pr/f2:hi hello by
project/my_pr/f2:hello
project/my_pr2/f1:hello
project/my_pr2/f2:hi hello by
project/my_pr2/f2:hello


theia@theiadocker-vardaan12345:/home/project$ who
theia@theiadocker-vardaan12345:/home/project$ whoami
theia
theia@theiadocker-vardaan12345:/home/project$ uname
Linux
theia@theiadocker-vardaan12345:/home/project$ uname -a
Linux theiadocker-vardaan12345 4.15.0-200-generic #211-Ubuntu SMP Thu Nov 24 18:16:04 UTC 2022 x86_64
x86_64 x86_64 GNU/Linux
theia@theiadocker-vardaan12345:/home/project$ id
uid=1000(theia) gid=1000(theia) groups=1000(theia),27(sudo),100(users)
theia@theiadocker-vardaan12345:/home/project$ df -h
Filesystem   Size Used Avail Use% Mounted on
overlay   98G   31G   63G 34% /
tmpfs   64M   0   64M   0% /dev
tmpfs   16G   0   16G   0% /sys/fs/cgroup
/dev/vda2   98G   31G   63G 34% /etc/kube
shm   64M   0   64M   0% /dev/shm
tmpfs   16G   16K   16G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs   16G   0   16G   0% /proc/acpi
tmpfs   16G   0   16G   0% /proc/scsi
tmpfs   16G   0   16G   0% /sys/firmware


theia@theiadocker-vardaan12345:/home/project$ uname
Linux
theia@theiadocker-vardaan12345:/home/project$ uname -a
Linux theiadocker-vardaan12345 4.15.0-200-generic #211-Ubuntu SMP Thu Nov 24 18:16:04 UTC 2022 x86_64
x86_64 x86_64 GNU/Linux
theia@theiadocker-vardaan12345:/home/project$ id
uid=1000(theia) gid=1000(theia) groups=1000(theia),27(sudo),100(users)
theia@theiadocker-vardaan12345:/home/project$ df -h
Filesystem Size Used Avail Use% Mounted on
overlay 98G 31G 63G 34% /
tmpfs 64M 0 64M 0% /dev
tmpfs 16G 0 16G 0% /sys/fs/cgroup
/dev/vda2 98G 31G 63G 34% /etc/kube
shm 64M 0 64M 0% /dev/shm
tmpfs 16G 16K 16G 1% /run/secrets/kubernetes.io/serviceaccount
tmpfs 16G 0 16G 0% /proc/acpi
tmpfs 16G 0 16G 0% /proc/scsi
tmpfs 16G 0 16G 0% /sys/firmware
theia@theiadocker-vardaan12345:/home/project$ ps
PID TTY TIME CMD
248 pts/0 00:00:00 bash
541 pts/0 00:00:00 ps
theia@theiadocker-vardaan12345:/home/project$ ps -e
PID TTY TIME CMD
1 ? 00:00:00 sh
7 ? 00:00:00 entrypoint.sh
165 ? 00:00:00 cron
166 ? 00:00:00 node
177 ? 00:00:00 sh
178 ? 00:00:00 node
189 ? 00:00:03 node
200 ? 00:00:05 node
248 pts/0 00:00:00 bash
578 ? 00:00:00 node
593 ? 00:00:00 node
618 pts/0 00:00:00 ps
theia@theiadocker-vardaan12345:/home/project$


theia@theiadocker-vardaan12345:/home/project$ echo how are you
how are you
theia@theiadocker-vardaan12345:/home/project$ echo -e how \n are
how n are
theia@theiadocker-vardaan12345:/home/project$ echo -e "how \n are"
how
are
theia@theiadocker-vardaan12345:/home/project$


theia@theiadocker-vardaan12345:/home/project$
theia@theiadocker-vardaan12345:/home/project$ -rw-r--r-- 1 theia theia 8121 May 31 16:45 usdoi.txt
bash: -rw-r--r--: command not found
theia@theiadocker-vardaan12345:/home/project$
theia@theiadocker-vardaan12345:/home/project$ The permissions set here are rw-r--r--. The - preceedin
g these permissions indicates that usdoi.txt is a file. If it were a directory, you would see a d ins
tead of the -.
bash: The: command not found
theia@theiadocker-vardaan12345:/home/project$
theia@theiadocker-vardaan12345:/home/project$ The first three entries correspond to the owner, the ne
xt three correspond to the group, and the last threee are for all others. You can see the owner of th
e file has read and write permissions, while the user group only has read permissions, and all other
users have read permission. No users have execute permissions, as indicated by the - instead of an x
in the third position for each user category.
bash: The: command not found
theia@theiadocker-vardaan12345:/home/project$ ls -l usdai.txt
ls: cannot access 'usdai.txt': No such file or directory
theia@theiadocker-vardaan12345:/home/project$ ls -l usdoi.txt
-rw-r--r-- 1 theia users 8121 Sep 28 07:59 usdoi.txt
theia@theiadocker-vardaan12345:/home/project$ chmod ug-r usdoi.txt
theia@theiadocker-vardaan12345:/home/project$ ls -l usdoi.txt
--w----r-- 1 theia users 8121 Sep 28 07:59 usdoi.txt
theia@theiadocker-vardaan12345:/home/project$ chmod u+x usdoi.txt
theia@theiadocker-vardaan12345:/home/project$ ls -l usdoi.txt
--wx---r-- 1 theia users 8121 Sep 28 07:59 usdoi.txt
theia@theiadocker-vardaan12345:/home/project$

Practical applications of commands part 2

Saturday, January 7, 2023 12:37 PM

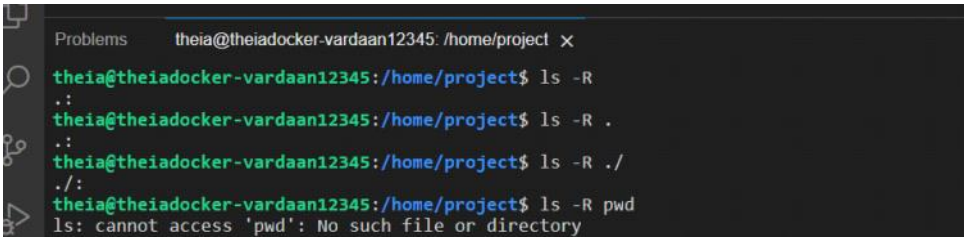
Archiving and compression

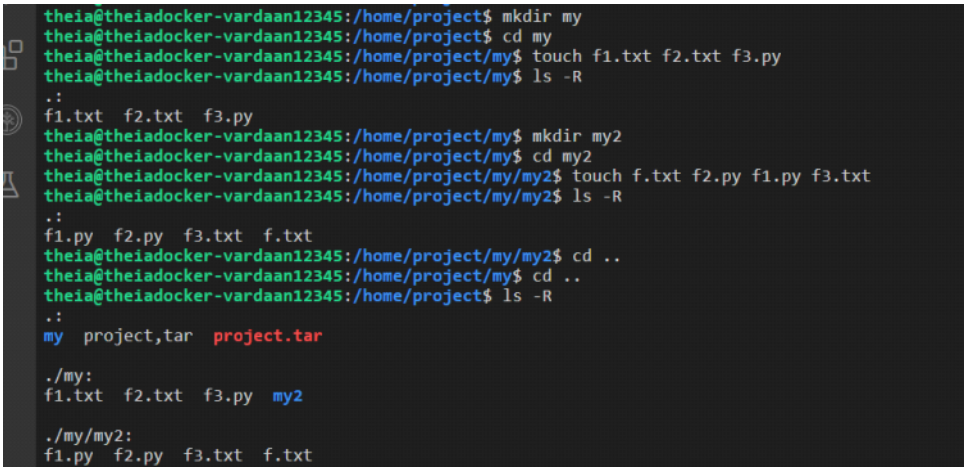
Archives:

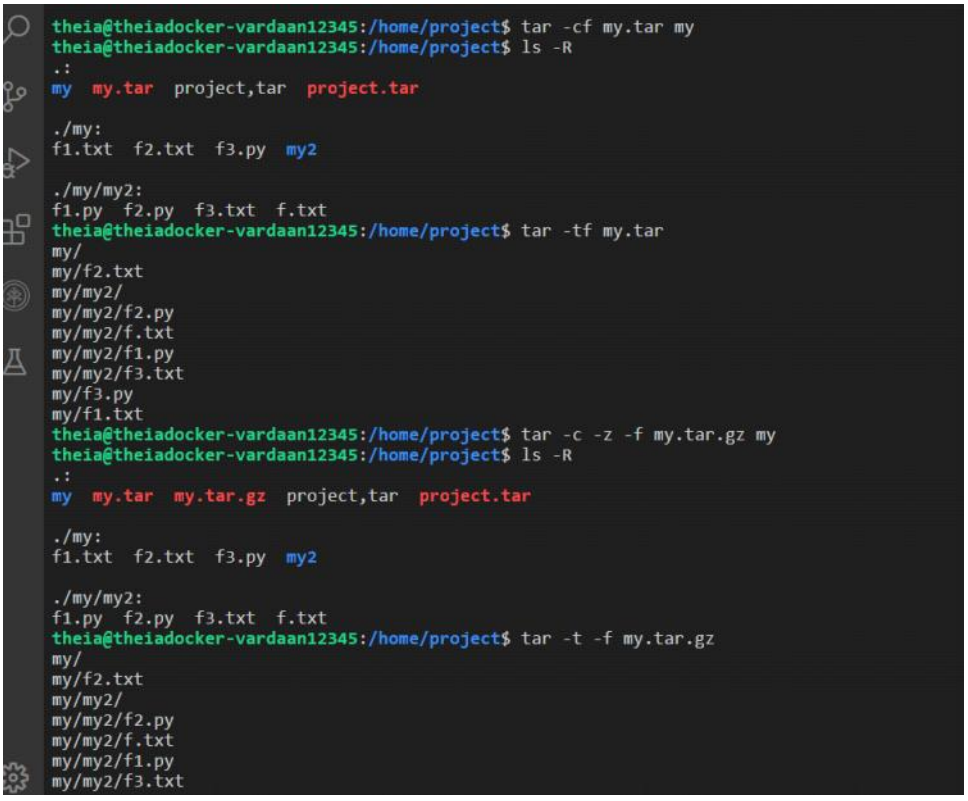
- Store rarely used information and preserve it
- Are a collection of data files and directories stored as a single file
- Make the collection more portable and serve as a backup in case of loss or corruption

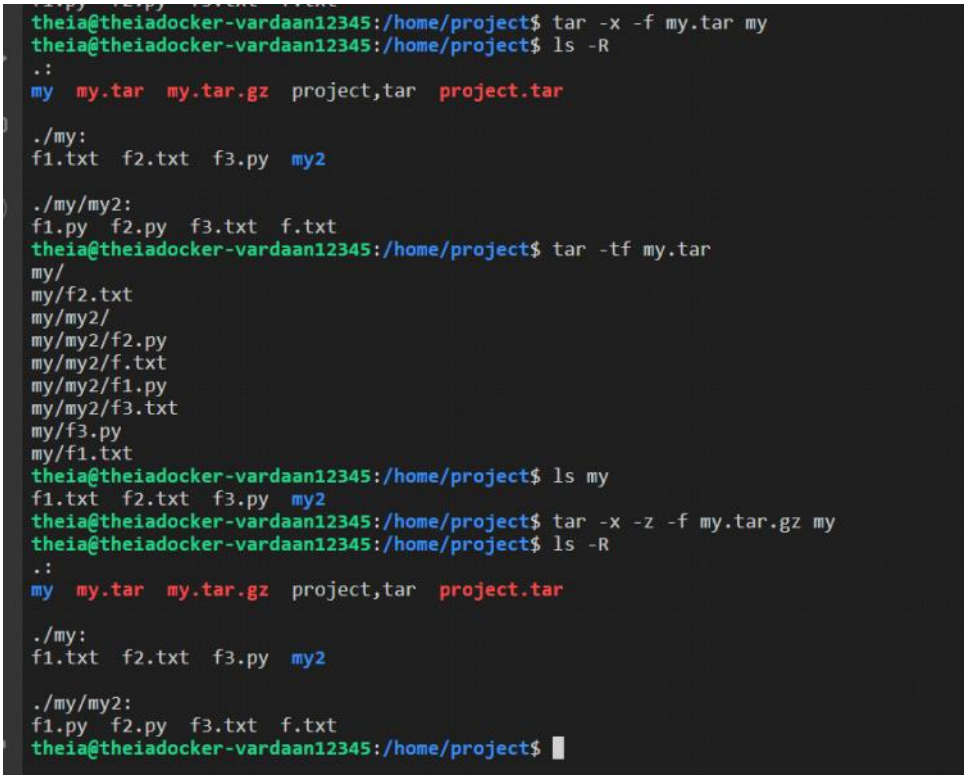
File compression:

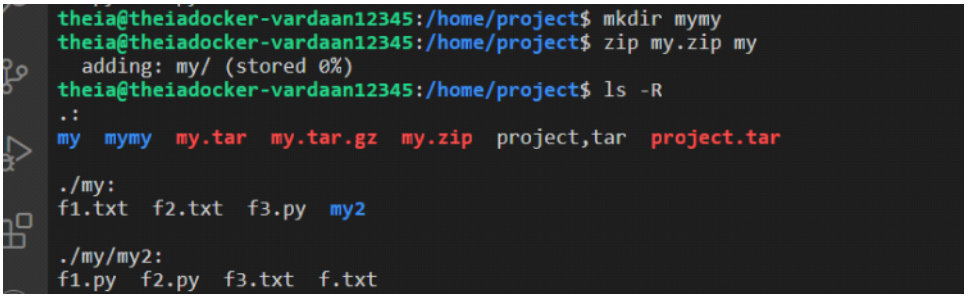
- Reduces file size by reducing information redundancy
- Preserves storage space, speeds up data transfer, and reduces bandwidth load






theia@theiadocker-vardaan12345:/home/project$ tar -cf my.tar my
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my my.tar project,tar project.tar
./my:
f1.txt f2.txt f3.py my2
./my/my2:
f1.py f2.py f3.txt f.txt
theia@theiadocker-vardaan12345:/home/project$ tar -tf my.tar
my/
my/f2.txt
my/my2/
my/my2/f2.py
my/my2/f.txt
my/my2/f1.py
my/my2/f3.txt
my/f3.py
my/f1.txt
theia@theiadocker-vardaan12345:/home/project$ tar -c -z -f my.tar.gz my
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my my.tar my.tar.gz project,tar project.tar
./my:
f1.txt f2.txt f3.py my2
./my/my2:
f1.py f2.py f3.txt f.txt
theia@theiadocker-vardaan12345:/home/project$ tar -t -f my.tar.gz
my/
my/f2.txt
my/my2/
my/my2/f2.py
my/my2/f.txt
my/my2/f1.py
my/my2/f3.txt


theia@theiadocker-vardaan12345:/home/project$ tar -x -f my.tar my
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my my.tar my.tar.gz project,tar project.tar
./my:
f1.txt f2.txt f3.py my2
./my/my2:
f1.py f2.py f3.txt f.txt
theia@theiadocker-vardaan12345:/home/project$ tar -tf my.tar
my/
my/f2.txt
my/my2/
my/my2/f2.py
my/my2/f.txt
my/my2/f1.py
my/my2/f3.txt
my/f3.py
my/f1.txt
theia@theiadocker-vardaan12345:/home/project$ ls my
f1.txt f2.txt f3.py my2
theia@theiadocker-vardaan12345:/home/project$ tar -x -z -f my.tar.gz my
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my my.tar my.tar.gz project,tar project.tar
./my:
f1.txt f2.txt f3.py my2
./my/my2:
f1.py f2.py f3.txt f.txt
theia@theiadocker-vardaan12345:/home/project$


theia@theiadocker-vardaan12345:/home/project$ mkdir mymy
theia@theiadocker-vardaan12345:/home/project$ zip my.zip my
adding: my/ (stored 0%)
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my mymy my.tar my.tar.gz my.zip project,tar project.tar
./my:
f1.txt f2.txt f3.py my2
./my/my2:
f1.py f2.py f3.txt f.txt

```
theia@theiadocker-vardaan12345:/home/project$ mkdir mymy
theia@theiadocker-vardaan12345:/home/project$ zip my.zip my
  adding: my/ (stored 0%)
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my  mymy  my.tar  my.tar.gz  my.zip  project,tar  project.tar

./my:
f1.txt  f2.txt  f3.py  my2

./my/my2:
f1.py  f2.py  f3.txt  f.txt

./mymy:
theia@theiadocker-vardaan12345:/home/project$ unzip my.zip ./mymy
Archive:  my.zip
caution: filename not matched:  ./mymy
theia@theiadocker-vardaan12345:/home/project$ ls my
f1.txt  f2.txt  f3.py  my2
theia@theiadocker-vardaan12345:/home/project$ ls mmy
ls: cannot access 'mmy': No such file or directory
theia@theiadocker-vardaan12345:/home/project$ ls mymy
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my  mymy  my.tar  my.tar.gz  my.zip  project,tar  project.tar

./my:
f1.txt  f2.txt  f3.py  my2

./my/my2:
f1.py  f2.py  f3.txt  f.txt

./mymy:
theia@theiadocker-vardaan12345:/home/project$ unzip my.zip my
Archive:  my.zip
caution: filename not matched:  my
```

```
theia@theiadocker-vardaan12345:/home/project$ unzip my.zip
Archive:  my.zip
theia@theiadocker-vardaan12345:/home/project$ ls -R
.:
my  mymy  my.tar  my.tar.gz  my.zip  project,tar  project.tar

./my:
f1.txt  f2.txt  f3.py  my2

./my/my2:
f1.py  f2.py  f3.txt  f.txt

./mymy:
theia@theiadocker-vardaan12345:/home/project$ ls my.zip
my.zip
theia@theiadocker-vardaan12345:/home/project$ hostname
theiadocker-vardaan12345
theia@theiadocker-vardaan12345:/home/project$
```

Print first 10 lines

```
theia@theia-vardaan12345:/home/project$ head usdoi.txt
The unanimous Declaration of the thirteen united States of America, When in the
Course of human events, it becomes necessary for one people to dissolve the
political bands which have connected them with another, and to assume among the
powers of the earth, the separate and equal station to which the Laws of Nature
and of Nature's God entitle them, a decent respect to the opinions of mankind
requires that they should declare the causes which impel them to the
separation.

We hold these truths to be self-evident, that all men are created equal, that
they are endowed by their Creator with certain unalienable Rights, that among
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ head -n 3 usdoi.txt
The unanimous Declaration of the thirteen united States of America, When in the
Course of human events, it becomes necessary for one people to dissolve the
political bands which have connected them with another, and to assume among the
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ tail usdoi.txt
People of these Colonies, solemnly publish and declare, That these United
Colonies are, and of Right ought to be Free and Independent States; that they
are Absolved from all Allegiance to the British Crown, and that all political
connection between them and the State of Great Britain, is and ought to be
totally dissolved; and that as Free and Independent States, they have full
Power to levy War, conclude Peace, contract Alliances, establish Commerce, and
to do all other Acts and Things which Independent States may of right do. And
for the support of this Declaration, with a firm reliance on the protection of
divine Providence, we mutually pledge to each other our Lives, our Fortunes and
our sacred Honor.
theia@theia-vardaan12345:/home/project$ tail -n 2 usdoi.txt
divine Providence, we mutually pledge to each other our Lives, our Fortunes and
our sacred Honor.
theia@theia-vardaan12345:/home/project$ wc usdoi.txt
 152 1330 8121 usdoi.txt
theia@theia-vardaan12345:/home/project$ wc -l usdoi.txt
152 usdoi.txt
theia@theia-vardaan12345:/home/project$ wc -w usdoi.txt
1330 usdoi.txt
theia@theia-vardaan12345:/home/project$ wc -c usdoi.txt
8121 usdoi.txt
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ wget https://cf-courses-data.s3.us.cloud-object-storage.appdo
main.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/zoo.txt
--2023-01-07 18:13:18--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX011
7EN-SkillsNetwork/labs/module%201/zoo.txt
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-obj
ect-storage.appdomain.cloud)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud
-object-storage.appdomain.cloud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 54 [text/plain]
Saving to: 'zoo.txt'

zoo.txt                   100%[==============================>]      54  --.-KB/s    in 0s

2023-01-07 18:13:18 (7.49 MB/s) - 'zoo.txt' saved [54/54]

theia@theia-vardaan12345:/home/project$ cat zoo.txt
zebra
zebra
lion
lion
anaconda
zebra
zebra
lion
zebra
theia@theia-vardaan12345:/home/project$ uniq zoo.txt
zebra
lion
anaconda
zebra
lion
zebra
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ grep people usdoi.txt
Course of human events, it becomes necessary for one people to dissolve the
people, unless those people would relinquish the right of Representation in the
firmness his invasions on the rights of the people.
to harrass our people, and eat out their substance.
the lives of our people.
Tyrant, is unfit to be the ruler of a free people.
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ grep we usdoi.txt
powers of the earth, the separate and equal station to which the Laws of Nature
they are endowed by their Creator with certain unalienable Rights, that among
rights, Governments are instituted among Men, deriving their just powers from
and organizing its powers in such form, as to them shall seem most likely to
elected; whereby the Legislative powers, incapable of Annihilation, have
for establishing Judiciary powers.
power.
power to legislate for us in all cases whatsoever.
humble terms: Our repeated Petitions have been answered only by repeated
and magnanimity, and we have conjured them by the ties of our common kindred to
our Separation, and hold them, as we hold the rest of mankind, Enemies in War,
connection between them and the State of Great Britain, is and ought to be
Power to levy War, conclude Peace, contract Alliances, establish Commerce, and
divine Providence, we mutually pledge to each other our Lives, our Fortunes and
theia@theia-vardaan12345:/home/project$ grep we h usdoi.txt
grep: h: No such file or directory
usdoi.txt:powers of the earth, the separate and equal station to which the Laws of Nature
usdoi.txt:they are endowed by their Creator with certain unalienable Rights, that among
usdoi.txt:rights, Governments are instituted among Men, deriving their just powers from
usdoi.txt:and organizing its powers in such form, as to them shall seem most likely to
usdoi.txt:elected; whereby the Legislative powers, incapable of Annihilation, have
usdoi.txt:for establishing Judiciary powers.
usdoi.txt:power.
usdoi.txt:power to legislate for us in all cases whatsoever.
usdoi.txt:humble terms: Our repeated Petitions have been answered only by repeated
usdoi.txt:and magnanimity, and we have conjured them by the ties of our common kindred to
usdoi.txt:our Separation, and hold them, as we hold the rest of mankind, Enemies in War,
usdoi.txt:connection between them and the State of Great Britain, is and ought to be
usdoi.txt:Power to levy War, conclude Peace, contract Alliances, establish Commerce, and
usdoi.txt:divine Providence, we mutually pledge to each other our Lives, our Fortunes and
theia@theia-vardaan12345:/home/project$ grep 'we h' usdoi.txt
and magnanimity, and we have conjured them by the ties of our common kindred to
our Separation, and hold them, as we hold the rest of mankind, Enemies in War,
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ grep -v login /etc/passwd
root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
theia:x:1000:1000:,,,:/home/theia:/bin/bash
postgres:x:105:109:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
theia@theia-vardaan12345:/home/project$ cut -c -2 zoo.txt
ze
ze
li
li
an
ze
ze
li
ze
theia@theia-vardaan12345:/home/project$ cut -c 2 zoo.txt
e
e
i
i
n
e
e
i
e
theia@theia-vardaan12345:/home/project$ cut -c 2- zoo.txt
ebra
ebra
ion
ion
naconda
ebra
ebra
ion
ebra
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ cat zoo.txt
zebra
zebra
lion
lion
anaconda
zebra
zebra
lion
zebra
theia@theia-vardaan12345:/home/project$ cat zoo_ages.txt
17
12
7
4
3
4
1
0
1
theia@theia-vardaan12345:/home/project$ paste zoo_ages.txt zoo.txt
17      zebra
12      zebra
7       lion
4       lion
3       anaconda
4       zebra
1       zebra
0       lion
1       zebra
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ paste -d ' ' zoo_ages.txt zoo.txt
17 zebra
12 zebra
7 lion
4 lion
3 anaconda
4 zebra
1 zebra
0 lion
1 zebra
theia@theia-vardaan12345:/home/project$ paste -d ':' zoo_ages.txt zoo.txt
17:zebra
12:zebra
7:lion
4:lion
3:anaconda
4:zebra
1:zebra
0:lion
1:zebra
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ tar bin.tar /bin
tar: /bin: Invalid blocking factor
Try 'tar --help' or 'tar --usage' for more information.
theia@theia-vardaan12345:/home/project$ tar -cf bin.tar /bin
tar: Removing leading `/' from member names
tar: Removing leading `/' from hard link targets
theia@theia-vardaan12345:/home/project$ tar -c -v -f bin.tar /bin
tar: Removing leading `/' from member names
/bin/
/bin/bzcmp
/bin/bzdiff
/bin/dir
/bin/bzegrep
/bin/hostname
/bin/umount
/bin/zegrep
/bin/rm
/bin/chgrp
/bin/dash
/bin/mktemp
/bin/bunzip2
/bin/domainname
/bin/cp
/bin/bzless
/bin/sync
/bin/vdir
/bin/false
/bin/zforce
/bin/which
/bin/true
/bin/mknod
/bin/uncompress
/bin/touch
/bin/dmesg
tar: Removing leading `/' from hard link targets
/bin/gunzip
/bin/bzmore
/bin/sleep
```

```
theia@theia-vardaan12345:/home/project$ tar -t -f bin.tar
bin/
bin/bzcmp
bin/bzdiff
bin/dir
bin/bzegrep
bin/hostname
bin/umount
bin/zegrep
bin/rm
bin/chgrp
bin/dash
bin/mktemp
bin/bunzip2
bin/domainname
bin/cp
bin/bzless
bin/sync
bin/vdir
bin/false
bin/zforce
bin/which
bin/true
bin/mknod
bin/uncompress
bin/touch
bin/dmesg
bin/gunzip
bin/bzmore
bin/sleep
bin/su
```

```
theia@theia-vardaan12345:/home/project$ tar -t -v -f bin.tar
drwxr-xr-x root/root         0 2022-12-07 00:23 bin/
lrwxrwxrwx root/root         0 2019-07-04 08:35 bin/bzcmp -> bzdiff
-rwxr-xr-x root/root      2140 2019-07-04 08:35 bin/bzdiff
-rwxr-xr-x root/root    133792 2018-01-18 04:43 bin/dir
lrwxrwxrwx root/root         0 2019-07-04 08:35 bin/bzegrep -> bzgrep
-rwxr-xr-x root/root     18504 2018-01-31 07:08 bin/hostname
-rwsr-xr-x root/root     26696 2020-09-16 14:43 bin/umount
-rwxr-xr-x root/root       140 2022-04-08 07:12 bin/zegrep
-rwxr-xr-x root/root     63704 2018-01-18 04:43 bin/rm
-rwxr-xr-x root/root     63672 2018-01-18 04:43 bin/chgrp
-rwxr-xr-x root/root    121432 2018-01-25 02:14 bin/dash
-rwxr-xr-x root/root     43192 2018-01-18 04:43 bin/mktemp
-rwxr-xr-x root/root     34888 2019-07-04 08:35 bin/bunzip2
lrwxrwxrwx root/root         0 2018-01-31 07:08 bin/domainname -> hostname
-rwxr-xr-x root/root    141528 2018-01-18 04:43 bin/cp
lrwxrwxrwx root/root         0 2019-07-04 08:35 bin/bzless -> bzmore
-rwxr-xr-x root/root     35000 2018-01-18 04:43 bin/sync
-rwxr-xr-x root/root    133792 2018-01-18 04:43 bin/vdir
-rwxr-xr-x root/root     30904 2018-01-18 04:43 bin/false
-rwxr-xr-x root/root      2131 2022-04-08 07:12 bin/zforce
-rwxr-xr-x root/root       946 2017-12-30 13:15 bin/which
-rwxr-xr-x root/root     30904 2018-01-18 04:43 bin/true
-rwxr-xr-x root/root     67768 2018-01-18 04:43 bin/mknod
-rwxr-xr-x root/root      2301 2022-04-08 07:12 bin/uncompress
-rwxr-xr-x root/root     88280 2018-01-18 04:43 bin/touch
-rwxr-xr-x root/root     72000 2020-09-16 14:43 bin/dmesg
hrwxr-xr-x root/root         0 2022-04-08 07:12 bin/gunzip link to bin/uncompress
-rwxr-xr-x root/root      1297 2019-07-04 08:35 bin/bzmore
-rwxr-xr-x root/root     35000 2018-01-18 04:43 bin/sleep
-rwsr-xr-x root/root     44664 2022-11-29 07:25 bin/su
-rwxr-xr-x root/root      1937 2022-04-08 07:12 bin/zcat
-rwxr-xr-x root/root     67808 2018-01-18 04:43 bin/ln
-rwxr-xr-x root/root     76000 2018-01-18 04:43 bin/dd
-rwxr-xr-x root/root     52664 2022-11-29 07:25 bin/login
-rwxr-xr-x root/root      5782 2022-04-08 07:12 bin/zdiff
-rwxr-xr-x root/root     10104 2017-12-30 13:15 bin/tempfile
-rwxr-xr-x root/root    109000 2018-01-29 21:49 bin/sed
```

```
theia@theia-vardaan12345:/home/project$ tar -x -v -f bin.tar
bin/
bin/bzcmp
bin/bzdiff
bin/dir
bin/bzegrep
bin/hostname
bin/umount
bin/zegrep
bin/rm
bin/chgrp
bin/dash
bin/mktemp
bin/bunzip2
bin/domainname
bin/cp
bin/bzless
bin/sync
bin/vdir
bin/false
bin/zforce
bin/which
bin/true
bin/mknod
bin/uncompress
bin/touch
bin/dmesg
bin/gunzip
bin/bzmore
bin/sleep
bin/su
```

```
theia@theia-vardaan12345:/home/project$ zip config.zip /etc/*.conf
  adding: etc/adduser.conf (deflated 55%)
  adding: etc/ca-certificates.conf (deflated 74%)
  adding: etc/debconf.conf (deflated 56%)
  adding: etc/deluser.conf (deflated 40%)
  adding: etc/gai.conf (deflated 57%)
  adding: etc/host.conf (deflated 13%)
  adding: etc/ld.so.conf (deflated 6%)
  adding: etc/libaudit.conf (deflated 34%)
  adding: etc/logrotate.conf (deflated 50%)
  adding: etc/mke2fs.conf (deflated 58%)
  adding: etc/mongodb.conf (deflated 52%)
  adding: etc/nsswitch.conf (deflated 49%)
  adding: etc/ntp.conf (deflated 52%)
  adding: etc/pam.conf (deflated 62%)
  adding: etc/resolv.conf (deflated 27%)
  adding: etc/sensors3.conf (deflated 82%)
  adding: etc/sysctl.conf (deflated 61%)
  adding: etc/ucf.conf (deflated 61%)
theia@theia-vardaan12345:/home/project$ ls etc/
ls: cannot access 'etc/': No such file or directory
theia@theia-vardaan12345:/home/project$ ls /etc/
adduser.conf            fstab           locale.gen      php               shadow
alternatives            gai.conf        localtime       pm                shadow-
apparmor                gdb             logcheck        postgresql        shells
apparmor.d              groff           login.defs      postgresql-common skel
apt                     groovy          logrotate.conf  profile           ssh
bash.bashrc             group           logrotate.d     profile.d         ssl
bash_completion.d       group-          lsb-release     protocols         subgid
bindresvport.blacklist  gshadow         machine-id      python            subgid-
ca-certificates         gshadow-        magic           python2.7         subuid
ca-certificates.conf    gss             magic.mime      python3           subuid-
calendar                host.conf       mailcap         python3.6         sudoers
cassandra               hostname        mailcap.order   python3.8         sudoers.d
cron.d                  hosts           manpath.config  rc0.d             sysctl.conf
cron.daily              init.d          mime.types      rc1.d             sysctl.d
cron.hourly             inputrc         mke2fs.conf     rc2.d             sysstat
cron.monthly            issue           mongodb.conf    rc3.d             systemd
```

```
theia@theia-vardaan12345:/home/project$ ls /etc/
adduser.conf            fstab           locale.gen      php               shadow
alternatives            gai.conf        localtime       pm                shadow-
apparmor                gdb             logcheck        postgresql        shells
apparmor.d              groff           login.defs      postgresql-common skel
apt                     groovy          logrotate.conf  profile           ssh
bash.bashrc             group           logrotate.d     profile.d         ssl
bash_completion.d       group-          lsb-release     protocols         subgid
bindresvport.blacklist  gshadow         machine-id      python            subgid-
ca-certificates         gshadow-        magic           python2.7         subuid
ca-certificates.conf    gss             magic.mime      python3           subuid-
calendar                host.conf       mailcap         python3.6         sudoers
cassandra               hostname        mailcap.order   python3.8         sudoers.d
cron.d                  hosts           manpath.config  rc0.d             sysctl.conf
cron.daily              init.d          mime.types      rc1.d             sysctl.d
cron.hourly             inputrc         mke2fs.conf     rc2.d             sysstat
cron.monthly            issue           mongodb.conf    rc3.d             systemd
crontab                 issue.net       nanorc          rc4.d             terminfo
cron.weekly             java            network         rc5.d             timezone
dbus-1                  java-11-openjdk NetworkManager  rc6.d             ucf.conf
debconf.conf            java-8-openjdk  networks        rcS.d             ufw
debian_version          kernel          nsswitch.conf   resolv.conf       update-motd.d
default                 ldap            ntp.conf        rmt               wgetrc
deluser.conf            ld.so.cache     opt             rpc               X11
dhcp                    ld.so.conf      os-release      securetty         xdg
docker                  ld.so.conf.d    pam.conf        security          xml
dpkg                    legal           pam.d           selinux
emacs                   libaudit.conf   passwd          sensors3.conf
environment             lighttpd        passwd-         sensors.d
fonts                   locale.alias    perl            services
theia@theia-vardaan12345:/home/project$ zip folder1.zip /etc/cro*
  adding: etc/cron.d/ (stored 0%)
  adding: etc/cron.daily/ (stored 0%)
  adding: etc/cron.hourly/ (stored 0%)
  adding: etc/cron.monthly/ (stored 0%)
  adding: etc/crontab (deflated 55%)
  adding: etc/cron.weekly/ (stored 0%)
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ ls -ltr
total 5596
drwxr-sr-x 2 theia users    4096 Dec  7 00:23 bin
-rw-r--r-- 1 theia users 5703680 Jan  7 19:36 bin.tar
-rw-r--r-- 1 theia users   15486 Jan  7 19:46 config.zip
-rw-r--r-- 1 theia users    1290 Jan  7 19:47 folder1.zip
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ zip folder1.zip /etc/cro*
  adding: etc/cron.d/ (stored 0%)
  adding: etc/cron.daily/ (stored 0%)
  adding: etc/cron.hourly/ (stored 0%)
  adding: etc/cron.monthly/ (stored 0%)
  adding: etc/crontab (deflated 55%)
  adding: etc/cron.weekly/ (stored 0%)
theia@theia-vardaan12345:/home/project$ ls -ltr
total 5596
drwxr-sr-x 2 theia users    4096 Dec  7 00:23 bin
-rw-r--r-- 1 theia users 5703680 Jan  7 19:36 bin.tar
-rw-r--r-- 1 theia users   15486 Jan  7 19:46 config.zip
-rw-r--r-- 1 theia users    1290 Jan  7 19:47 folder1.zip
theia@theia-vardaan12345:/home/project$ mkdir touch11
theia@theia-vardaan12345:/home/project$ zip touch11.zip touch11
  adding: touch11/ (stored 0%)
theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  folder1.zip  touch11  touch11.zip
theia@theia-vardaan12345:/home/project$ cd touch11
theia@theia-vardaan12345:/home/project/touch11$ touch f.txt
theia@theia-vardaan12345:/home/project/touch11$ cd ..
theia@theia-vardaan12345:/home/project$ zip touch.zip touch11
  adding: touch11/ (stored 0%)
theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  folder1.zip  touch11  touch11.zip  touch.zip
theia@theia-vardaan12345:/home/project$ ls touch.zip
touch.zip
theia@theia-vardaan12345:/home/project$ ls -r touch.zip
touch.zip
theia@theia-vardaan12345:/home/project$ zip -r t.zip touch11
  adding: touch11/ (stored 0%)
  adding: touch11/f.txt (stored 0%)
theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  folder1.zip  touch11  touch11.zip  touch.zip  t.zip
theia@theia-vardaan12345:/home/project$ ls t.zip
t.zip
theia@theia-vardaan12345:/home/project$ ls -r t.zip
t.zip
```

```
theia@theia-vardaan12345:/home/project$ unzip -l folder1.zip
Archive:  folder1.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
        0  2022-12-07 00:18   etc/cron.d/
        0  2022-12-07 00:18   etc/cron.daily/
        0  2022-12-07 00:06   etc/cron.hourly/
        0  2022-12-07 00:06   etc/cron.monthly/
      722  2022-05-10 16:59   etc/crontab
        0  2022-12-07 00:16   etc/cron.weekly/
---------                     -------
      722                     6 files
theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  etc  folder1.zip  touch11  touch11.zip  touch.zip  t.zip
theia@theia-vardaan12345:/home/project$ unzip -o folder1.zip
Archive:  folder1.zip
  inflating: etc/crontab
theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  etc  folder1.zip  touch11  touch11.zip  touch.zip  t.zip
theia@theia-vardaan12345:/home/project$ unzip t.zip
Archive:  t.zip
replace touch11/f.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
 extracting: touch11/f.txt
theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  etc  folder1.zip  touch11  touch11.zip  touch.zip  t.zip
```

```
theia@theia-vardaan12345:/home/project$ ping www.yahoo.com
PING new-fp-shed.wg1.b.yahoo.com (74.6.143.25): 56 data bytes
64 bytes from 74.6.143.25: icmp_seq=0 ttl=50 time=16.342 ms
64 bytes from 74.6.143.25: icmp_seq=1 ttl=51 time=16.332 ms
64 bytes from 74.6.143.25: icmp_seq=2 ttl=50 time=16.382 ms
64 bytes from 74.6.143.25: icmp_seq=3 ttl=51 time=16.377 ms
64 bytes from 74.6.143.25: icmp_seq=4 ttl=50 time=16.287 ms
64 bytes from 74.6.143.25: icmp_seq=5 ttl=50 time=16.299 ms
64 bytes from 74.6.143.25: icmp_seq=6 ttl=50 time=16.415 ms
64 bytes from 74.6.143.25: icmp_seq=7 ttl=51 time=16.193 ms
64 bytes from 74.6.143.25: icmp_seq=8 ttl=51 time=16.347 ms
64 bytes from 74.6.143.25: icmp_seq=9 ttl=51 time=16.274 ms
^C--- new-fp-shed.wg1.b.yahoo.com ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 16.193/16.325/16.415/0.061 ms
theia@theia-vardaan12345:/home/project$ ping -c -3 www.google.com
PING www.google.com (172.253.63.99): 56 data bytes
64 bytes from 172.253.63.99: icmp_seq=0 ttl=103 time=1.443 ms
64 bytes from 172.253.63.99: icmp_seq=1 ttl=103 time=1.503 ms
64 bytes from 172.253.63.99: icmp_seq=2 ttl=103 time=1.472 ms
64 bytes from 172.253.63.99: icmp_seq=3 ttl=103 time=1.467 ms
64 bytes from 172.253.63.99: icmp_seq=4 ttl=103 time=1.459 ms
64 bytes from 172.253.63.99: icmp_seq=5 ttl=103 time=1.354 ms
64 bytes from 172.253.63.99: icmp_seq=6 ttl=103 time=1.475 ms
64 bytes from 172.253.63.99: icmp_seq=7 ttl=103 time=1.438 ms
^C--- www.google.com ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.354/1.451/1.503/0.041 ms
theia@theia-vardaan12345:/home/project$ ping -c 3 www.yahoo.com
PING new-fp-shed.wg1.b.yahoo.com (74.6.143.26): 56 data bytes
64 bytes from 74.6.143.26: icmp_seq=0 ttl=51 time=14.874 ms
64 bytes from 74.6.143.26: icmp_seq=1 ttl=50 time=16.387 ms
64 bytes from 74.6.143.26: icmp_seq=2 ttl=51 time=14.977 ms
--- new-fp-shed.wg1.b.yahoo.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 14.874/15.413/16.387/0.690 ms
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1480
        inet 172.22.150.253  netmask 255.255.255.255  broadcast 0.0.0.0
        inet6 fe80::d07a:eff:feb6:c1fc  prefixlen 64  scopeid 0x20<link>
        ether d2:7a:0e:b6:c1:fc  txqueuelen 0  (Ethernet)
        RX packets 15704  bytes 8256285 (8.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15680  bytes 13632282 (13.6 MB)
        TX errors 0  dropped 1 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 48446  bytes 32764661 (32.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 48446  bytes 32764661 (32.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

theia@theia-vardaan12345:/home/project$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1480
        inet 172.22.150.253  netmask 255.255.255.255  broadcast 0.0.0.0
        inet6 fe80::d07a:eff:feb6:c1fc  prefixlen 64  scopeid 0x20<link>
        ether d2:7a:0e:b6:c1:fc  txqueuelen 0  (Ethernet)
        RX packets 15851  bytes 8287995 (8.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15800  bytes 13665754 (13.6 MB)
        TX errors 0  dropped 1 overruns 0  carrier 0  collisions 0

theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:/home/project$ curl -O https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  8121  100  8121    0     0  37082      0 --:--:-- --:--:-- --:--:-- 36913
theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  etc  folder1.zip  touch11  touch11.zip  touch.zip  t.zip  usdoi.txt
theia@theia-vardaan12345:/home/project$ rm usdoi.txt
theia@theia-vardaan12345:/home/project$ wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt
--2023-01-07 20:26:57--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8121 (7.9K) [text/plain]
Saving to: 'usdoi.txt'

usdoi.txt                100%[=================================>]   7.93K  --.-KB/s    in 0s

2023-01-07 20:26:57 (302 MB/s) - 'usdoi.txt' saved [8121/8121]

theia@theia-vardaan12345:/home/project$ ls
bin  bin.tar  config.zip  etc  folder1.zip  touch11  touch11.zip  touch.zip  t.zip  usdoi.txt
theia@theia-vardaan12345:/home/project$
```

```
theia@theia-vardaan12345:~$ uname
Linux
theia@theia-vardaan12345:~$ uname -i
x86_64
theia@theia-vardaan12345:~$ id
uid=1000(theia) gid=1000(theia) groups=1000(theia),27(sudo),100(users)
theia@theia-vardaan12345:~$
```

```
theia@theia-vardaan12345:/home/project$ ~
bash: /home/theia: Is a directory
theia@theia-vardaan12345:/home/project$ cd `
> euhefuef^C
theia@theia-vardaan12345:/home/project$ cd ~
theia@theia-vardaan12345:~$ pwd
/home/theia
theia@theia-vardaan12345:~$ uname
Linux
theia@theia-vardaan12345:~$ uname -i
x86_64
theia@theia-vardaan12345:~$ id
uid=1000(theia) gid=1000(theia) groups=1000(theia),27(sudo),100(users)
theia@theia-vardaan12345:~$ whoami
theia
theia@theia-vardaan12345:~$ hostname
theia-vardaan12345
theia@theia-vardaan12345:~$ whoami -i
whoami: invalid option -- 'i'
Try 'whoami --help' for more information.
theia@theia-vardaan12345:~$ whoami -v
whoami: invalid option -- 'v'
Try 'whoami --help' for more information.
theia@theia-vardaan12345:~$ uname -r
4.15.0-200-generic
theia@theia-vardaan12345:~$ wc -l /etc/passwd file
  25 /etc/passwd
wc: file: No such file or directory
  25 total
```

```
theia@theia-vardaan12345:~$ grep 'not installed' /var/log/bootstrap.log
  Package libc6 is not installed.
  Package libdebconfclient0 is not installed.
  awk is not installed.
  Package awk is not installed.
  libbz2-1.0 is not installed.
  libc6 is not installed.
  liblzma5 is not installed.
  libselinux1 is not installed.
  libzstd1 is not installed.
  zlib1g is not installed.
  Package libbz2-1.0 is not installed.
  Package libc6 is not installed.
  Package liblzma5 is not installed.
  Package libselinux1 is not installed.
  Package libzstd1 is not installed.
  Package zlib1g is not installed.
  Package tar is not installed.
  Package libgcc1 is not installed.
  libtinfo5 is not installed.
  libsystemd0 is not installed.
  libacl1 is not installed.
  libattr1 is not installed.
  libselinux1 is not installed.
  libbz2-1.0 is not installed.
  liblzma5 is not installed.
  libselinux1 is not installed.
  libzstd1 is not installed.
  zlib1g is not installed.
  libblkid1 is not installed.
  libcom-err2 is not installed.
  libext2fs2 is not installed.
  libss2 is not installed.
  libuuid1 is not installed.
  libselinux1 is not installed.
  libpcre3 is not installed.
  libpam0g is not installed.
```

```
theia@theia-vardaan12345:~$ wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/I
BM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt
--2023-01-07 20:38:30--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB025
0EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-obj
ect-storage.appdomain.cloud)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud
-object-storage.appdomain.cloud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1311 (1.3K) [text/plain]
Saving to: 'top-sites.txt'

top-sites.txt             100%[==============================>]   1.28K  --.-KB/s    in 0s

2023-01-07 20:38:30 (126 MB/s) - 'top-sites.txt' saved [1311/1311]

theia@theia-vardaan12345:~$ grep org top-sites.txt
en.wikipedia.org
wordpress.org
mozilla.org
pt.wikipedia.org
es.wikipedia.org
w3.org
wikimedia.org
creativecommons.org
fr.wikipedia.org
apache.org
id.wikipedia.org
de.wikipedia.org
theia@theia-vardaan12345:~$ curl https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/I
BM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt | grep org top-sites.txt
en.wikipedia.org
wordpress.org
mozilla.org
pt.wikipedia.org
es.wikipedia.org
```

```
theia@theia-vardaan12345:~$ curl https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/I
BM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt | grep org top-sites.txt
en.wikipedia.org
wordpress.org
mozilla.org
pt.wikipedia.org
es.wikipedia.org
w3.org
wikimedia.org
creativecommons.org
fr.wikipedia.org
apache.org
id.wikipedia.org
de.wikipedia.org
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1311  100  1311    0     0  14566      0 --:--:-- --:--:-- --:--:-- 14406
(23) Failed writing body
theia@theia-vardaan12345:~$ curl -o https://cf-courses-data.s3.us.cloud-object-storage.appdomain.clou
d/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt | grep org top-sites.txt
en.wikipedia.org
wordpress.org
mozilla.org
pt.wikipedia.org
es.wikipedia.org
w3.org
wikimedia.org
creativecommons.org
fr.wikipedia.org
apache.org
id.wikipedia.org
de.wikipedia.org
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information
theia@theia-vardaan12345:~$
```

```
theia@theia-vardaan12345:~$ cut -c -3 top-sites.txt
you
www
app
mic
clo
pla
sup
www
en.
doc
wor
moz
lin
you
map
ado
dri
eur
goo
plu
vim
sit
acc
pt.
git
fac
uol
vk.
ist
es.
ama
bp.
bbc
cnn
fee
```

1. Which of the following is NOT one of the Linux shell definitions? **1 / 1 point**

- ○ Interactive language
- ○ Scripting language
- ◉ Enable access to applications
- ○ User interface for running commands

✓ **Correct**
Correct, this is one of the applications of a shell, rather than a definition.

9. Similar to curl – the wget command is used to _____. **1 / 1 point**

- ○ write a file to a URL, including the HTML code to build a web page.
- ○ wrangle a file to get user data and other useful information.
- ◉ retrieve the HTML code for a web page.
- ○ upload GNU files to a web server for subsequent hash code analysis.

✓ **Correct**
Correct. wget can retrieve a file located at a URL or it can retrieve the HTML code for a web page and has recursive file downloading capabilities.

2. Which of the following statements would print the paths stored in your system's PATH variable? 1 point

- ○ ps PATH
- ○ man $PATH
- ○ echo PATH
- ◉ echo $PATH

8. Which one of the following four statements regarding file archiving and compression is false? 1 point

- ○ You would archive your file if you want it to fit on your hard drive.
- ○ Archiving and compression are distinct processes which are usually combined.
- ◉ An archive file is a collection of data files and directories that are stored as a single file.
- ○ File compression involves reducing the size of a file by taking advantage of redundancy in its information content.

Shell commands part 2

Saturday, January 7, 2023 12:46 PM

link

1. Display a file's contents page by page
   Here page means the current height of terminal window which displays the contents and not the actual page in the file
   ~$ more file.txt
   When we press space bar, we go to the next contents in the file which are displayed in the terminal window
   Press q or ctrl+C to quit the more command and go to command prompt to enter the next command
2. Print first 10(only 10) lines of a file
   ~$ head file1.txt
3. Print first 15 lines of a file
   ~$ head –n 15 file1.txt
4. Print last 10(only 10) lines of a file
   ~$ tail file.txt
5. Print last 6 lines of a file
   ~$ tail –n 6 file.txt
6. Print number of lines,words,characters in a file in this order only
   It also counts newline characters even if we don't type them and we just type enter to go to next line
   The last newline character represents "end of file" character-which is not "enter" we pressed but it represents automatically that there are no characters in the file beyond this point
   If there are 7 newline characters,it means there are 7 lines and we pressed enter 6 times
   ~$ wc file.txt
7. View only word count,only character count,only lines count
   ~$ wc –w f1.txt
   ~$ wc –c f1.txt
   ~$ wc –l f1.txt
8. Sort lines alphanumerically
   ~$ sort file.txt

9. Sort in reverse order
   ~$ sort –r file.txt
10. Remove consecutive duplicate lines(not all duplicate lines but consecutive duplicates)
    ~$ uniq file.txt
11. Return lines in a file matching a particular pattern of characters(case sensitive search)
    ~$ grep ch file.txt
    All lines containing ch are returned
    Grep stands for global regular expression print
12. Return lines in a file matching a regular pattern of characters but case insensitive
    ~$ grep -i ch file.txt
13. Extract a section of characters from each line
    ~$ cut –c 2-9 file.txt
    This cuts and prints 2^{nd} to 9^{th} characters from each line
14. Extract 3^{rd} to 8^{th} words from every line
    ~$ cut –w 3-8 file.txt
15. Extract 2^{nd} column from a file where by default every column is separated by a space
    ~$ cut -f2 file.txt

16. Use tab as a delimiter and then find the 3rd column of every line
    ~$ cut –d '\t' -f3 file.txt
17. Use colon as a delimiter and find 5th column
    ~4 cut –d ':' -f5 file.txt
18. See contents of many files with tab as a default delimiter b/w corresponding lies of every file
    ~$ paste file.txt file2.txt file3.txt
19. Use a different delimiter
    ~$ paste –d ":" file.txt file2.txt file3.txt
20. List all files in a directory tree from a particular directory
    ~$ ls –R ./project
    Or ~$ ls –R ./../my_project
21. List all files in pwd
    ~$ ls –R
    Or ~$ ls –R .
    Or ~$ ls –R ./
22. Archieve a directory
    ~$ tar –c –f notes.tar notes
    Notes.tar is new archieved directory which contains the notes directory which we archieved
    Tar stands for tape archiever
    -c means "create a new archive diretcory
    -f means "interpret content from files in the notes directory rather than from default,which is standard input"
23. Compress archive directory
    We will use –z which will process newly created archived directory through a zip function
    Now we add notes.tar.gz to the archived file as notes is not only archived but also compressed
    ~$ tar –c –z –f notes.tar.gz notes

24. List all files in an archieved directory
    ~$ tar –t –f notes.tar
25. Dearchive a directory
    ~$ tar –x –f notes.tar notes
    DE archive notes.tar and store its contents in notes directory
    -x options is used to dearchive all files and directories inside notes.tar
26. De archive and de compress an archived directory
    ~$ tar –x –z –f notes.tar.gz notes
27. Compress files and directories to an archive
    Use zip command
    difference b/w zip and tar –z commands
    Zip first compresses everything and then bundles them to form an archive
    While tar –z first bundles every file to an archive and then compresses the full archive
    ~$ zip notes.zip notes
    Form a notes.zip directory by compressing notes directory
28. Unzip a file
    ~$ unzip file.zip
29. Get host's name(our computer's name)
    ~$ hostname
    If our machine has local domains,then ".local" be will attatched at end of host's name
30. Display hostname without domain name
    ~$ hostname –s
31. Get ip address of the host
    ~$ hostname -i
32. Get ip address,mac address,and hardware specificationsof host computer

~$ ifconfig
Ifconfig=interface configuration
Displays information about our device's communication devices
It displays or configures system network interfaces

33. Get info of a particular device we want ifconfig to inspect
    We can get ethernet adapter information
    ~$ ifconfig eth0
    It is not O but is 0
34. Test connectivity to a host or another ip address
    Use ping-send ICMP packets to URL and print response
    ICMP=internet control message protocol requests
    Ping sends a response to each echo request sent by it to the server or url
    To abort.press q or ctrl+C
    Then ping shows a summary of responses
    ~$ ping www.google.com
    ~$ ping www.yahoo.com

35. Ask ping to send a specific number of requests
    After sending that number of requests,ping is automatically aborted and shows the summary
    ~$ ping –c 10 www.yahoo.com
    Sends 10 requests
36. Send data to/from many different URLs
    Get entire content of landing page of a URL
    ~$ curl www.google.com
    Curl=client url
37. Write contents of a URL to a local file
    ~$ curl www.google.com -o file.txt
    Save contents of google website to file.txt
    We used small "o"
38. Retrieve files located at URLs
    Use wget-it has recursive downloading capabilities
    ~$ wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/usdoi.txt
39. Search for lines with only one word to find-quotes not necessary
    ~$ grep we usdoi.txt
40. Search for lines with 2 words seperated by space
    Quotes necessary otherwise it only looks for the first word and not anything after space
    ~$ grep 'we have' usdoi.txt -------correct
41. ~$ grep we have usdoi.txt ---------incorrect
    It only searches for we and not for "we have"

    Some of the frequently used options for `grep` are:

Option	Description
`-n`	Along with the matching lines, also print the line numbers
`-c`	Get the count of matching lines
`-i`	Ignore the case of the text while matching
`-v`	Print all lines which do not contain the pattern
`-w`	Match only if the pattern matches whole words

42. Display 2nd character of every line of a file
    ~$ cut –c 2 file.txt

43. Display first 2 characters of every line
    ~$ cut –c –2 file.txt
44. Display all characters of every line starting from 2n character
    ~$ cut –c 2- file.txt
45. Paste corresponding lines of 2 files or more in separate columns in consecutive rows
    ~$ paste zoo.txt zoo_ages.txt
46. creates an archive of the entire `/bin` directory into a file named `bin.tar`.

    The options used are as follows:

47.

Option	Description
-c	Create new archive file
-v	Verbosely list files processed
-f	Archive file name

    ~$ -c –v –f bin.tar /bin
    It shows all the files inside bin directory that linux is archiving
    If we don't want to see all files inside bin which are being archived
    ~$ tar –c –f bin.tar /bin

48. See only simple details of every file inside an archived directory
    ~$ tar –t –f bin.tar
49. See all details of every file in archived directory
    ~$ tar –t –v –f bin.tar
50. Untar an archived directory without seeing all files in it which are getting unarchived
    ~$ tar –x –f bin.tar
51. See a list of all files which are getting unarchived
    ~$ tar –x –v –f bin.tar
52. Make a compressed directory which contains all files with .conf extension in /etc/ directory
    ~$ zip hello.zip /etc/*.conf
53. Make a compressed directory which contains all files from /bin/ directory which start with "fine"
    ~$ zip folder1.zip /bin/fine*
54. Compress a directory(with some files) but not recursively(files in directpry are not compressed-only the whole big directory is compressed)
    ~$ zip folder1.zip my_pr
    Compress my_pr and store in folder1.zip
55. Compress a directory and all its files recursively
    ~$ zip –r folder1.zip my_pr
56. Unzip a directory
    ~$ unzip folder.zip
57. Unzip a folder by first listing all files we are unzipping
    ~$ unzip –l folder.zip
    We use small "l" and not small "i"
58. Unzip a folder/directory and all files in it by force overwriting-in case we use the command of unzip many times
    ~$ unzip –o folder.zip
59. **Test if a host is reachable**

    `ping`

    Check if **www.google.com is reachable. The command keeps sending data packets to the www.google.com server and prints the response it gets back. (Press `Ctrl+C` to stop pinging)**

~$ ping www.google.com

`eth0` is usually the primary network interface that connects your server to the network.

You can see your server's IP address in line number 2 after the word `inet`

60. Only see contents of a file (we can't see contents of a full firectory) which is located at a particular URL
    ~$ curl https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt
61. Download full contents of a directory(all files in it) or just a single file which is at a URL
    Wget can work recursively to download all files in a directory
    `$ wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt

62. Download only a single file from a URL
    ~$ wget <link> or ~$ curl –O <link>

    Curl –O can't download a full directory with many files
63. Display system name
    ~$ uname
64. Display system number of system name
    ~$ uname -i
65. Display username
    ~$ whoami
66. Get id
    ~$ id
67. Get hostname
    ~$ hostname
68. Get kernel version
    ~$ uname –r
69. *https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt contains most popular websites. Find out all the websites that have the word `org` in them.*

    Method 1:
    ~$ wget <link> | grep org top-sies.txt
    Method 2:
    ~$ curl <link> | grep org top-sites.txt
    Method 3:
    ~$ curl –o <link> | grep org top-sites.txt

70. Print first 3 charaters of every line in a file
    ~$ cut –c –3 top.txt

71. Print path
    ~$ echo $PATH

Shell commands

Wednesday, January 4, 2023 4:15 PM

1. Find out the default shell used
   ~$ printenv SHELL
   It returns path of the default shell program-if it is bash then this is returned
   /bin/bash
2. If shell is not bash and we want to switch shell to it
   ~$ bash
3. Convert to sh shell
   ~$ sh
4. Getting help for a command
   ~$ head –help or ~$ cat –help or ~$ <command_name> --help
5. Echo prints what we give it a argument-I we give it a file name then also it prints that file name again rather than printing the contents of that file
   ~$ echo prog.py
   Output: prog.py
   ~$ echo hihello
   OUTPUT: hihello
6. Find version of chmod command
   ~$ chmod ---version
7. Getting info of all files in a directory(not just name of existing files)
   ~$ ls –ltr or ~$ ls –l
8. Getting info of a particular file
   ~$ ls <filename>
9. Make 4 files in home directory when files previously were non existent
   ~$ touch file1 file2 file3 file4.py
   Default extension of files is txt if no extension is specified
10. Touch command when file is existent-timestamp changed
    ~$ touch file3
    By doing this and writing ~$ ls –ltr
    We can see that timestamp of file3 changed to current time's value
11. Remove a file(not a directory)
    ~$ rm <filename1> <filename2>
12. Remove a zip file which contains many files-we can't use standard rm command
    ~$ rm –r temp.zip
    We used –r to state that we don't just want to remove temp.zip folder-but we want to recursively go into the depths of this folder and delete every file in it

    LETS SAY WE HAVE file1.py file2 file3.py file4 and my_project directory in /home/project directory
    And we have file11.py file22 file33.py file44 in my_project directory

13. Copy contents of one file into a non existent file-new file file5 is made first
    /home/project$ file1.py file5
    File5 is text file but contents of file1.py are copied into this text file
    But we can run this file by python3
14. See contents of file5
    /home/project$ cat file5
15. Copy file5's contents into file2-file2's contents are overwritten by file5's contents
    /home/project$ cp file5 file2
    Now we can run the python script of contents of file2
    /home/project$ python3 file2
    /home/project$ cat file2
16. Copy file2 into my_project directory
    /home/project$ cp file2 ./my_project
    Now see contents of my_project
    /home/project$ ls my_project
    See contents of file2 now inside my_project
    /home/project$ cd ./my_project
    /home/project/my_project$ cat file2
17. Copy 2 files into one directory
    /home/project$ cp file1.py file3.py ./my_project
    /home/project$ ls my_project

/home/project$ cd ./my_project
/home/project/my_project$ cat file3.py

18. Make 2 empty directories in /home/project directory
    /home/project$ mkdir my_project2
    /home/project$ mkdir my_project3
19. Copy my_project2 and my_project into my_project3-we have to use –r for recursively copying all files of the directories into another directory
    /home/project$ cp –r ./my_project2 ./my_project3
    /home/project$ cp –r ./my_project ./my_project3
20. Go into file2 inside my_project3
    /home/project$ cd ./my_project3
    /home/project/my_project3$ cat ./my_project/file2
21. moving one file into another where destination file is non existent
    Move file33.py into new file called filen
    /home/project/my_project3/my_project$ mv file33.py filen
    /home/project/my_project3/my_project$ ls
    /home/project/my_project3/my_project$ cat filen
    We will see that all contents of file33.py are now in filen
22. Move one file into an existing file

    /home/project/my_project3/my_project$ mv filen file3.py
    /home/project/my_project3/my_project$ ls
    /home/project/my_project3/my_project$ cat file3.py
23. Run python script by opening python console directly in terminal
    /home/project/my_project3/my_project$ python
24. Quit python
    >>> quit()
25. Asking user to manually say yes before we overwrite contents of destination file
    ~$ mv -i source destination
    User has to press y for confirmation

    /home/project/my_project3/my_project$ mv -i file3.py file44
    /home/project/my_project3/my_project$ python2 file44
    Terminal first copies text in file44 text file and then runs it in a python terminal and then outputs the result
    If we write this then we get error-------no file exists like file44.py
    /home/project/my_project3/my_project$ python2 file44,py
26. Remove a directory(empty)
    Use rmdir-can't be used to remove non empty directories

    /home/project/my_project3$ mkdir my_pr
    /home/project/my_project3$ rmdir my_pr
    If we write this we get error
    /home/project/my_project3$ rmdir my_project
    As my_project is not empty
27. Delete a non empty directory
    Use –r with rm command
    Rm COMMAND ONLY REMOVS FILES AND NOT DIRECTORIES BUT IF WE USE –r WITH IT THEN IT RECURSIVLY REMOVES ALL FILES IN A PARTICULAR DIRECTORY
    WE CAN EVEN WRITE –rf TO SIGNIFY "remove recursively and forcefully" THE DIRECTORY
    If we just write –f with rm then a file(not a directory) is removed forcefully
    /home/project/my_project3$ rm –r my_project
    Or /home/project/my_project3$ rm –rf my_project

    /home/project/my_project3$ ls
    We won't see my_project directory now
28. Remove 2 or more directories at one time in which anyone of them can be either empty or non empty-it doesn't matter as we are using rm command and not rmdir command
    /home/project/my_project3$ rm –r new mew
    Or /home/project/my_project3$ rm –rf new mew

    New and mew directories can be either empty or non empty-doesn’t matter

29. Remove files with only rm
    /home/project/my_project$ rm file1 file2 file3.py
    3 files removed(doesn't matter if they are empty files or not)
30. Remove files with –f
    /home/project/my_project$ rm –f file1 file2 file3.py
31. See permissions of files in a directory
    ~$ ls –l or ~$ ls -lst
32. See permissions of a particular file
    ~$ ls –l file1 or ~$ ls –lst file1
33. See paths of directories or files in a directory
    ~$ l
34. See only files or directories without their path in a directory
    ~$ ls
35. Change permissions of a file or directory using chmod

36. We want the user to have read and write permissions and the group and other users to have read permissions only
    ~$ chmod u=rw,og=r file1
37. we want to remove the read permissions for the “other” users from files that have a “.page” extension
    ~$ chmod o-r *.page
38. If we had wanted to include files in subdirectories, we could have used the `-R` (recursive) option.

~$ chmod –R o-r *.page

39. **Removing write permission from user in files with .page extension**
    ~$ chmod u-w *.page
40. **Make console a text file editor where we can type anything**
    ~$ wc –m
41. **Echo something which we type**
    First we type characters and when we press enter we see them printed on next line
    ~$ echo <statement>
42. **Get no of characters in a file**
    ~$ wc –m <filename>
43. **Get no of bytes in a file**
    ~$ wc –m <filename>
44. **Get no of words in a file**
    ~$ wc –w <filename>
45. **Get no of lines in a file**
    ~$ wc –l <filename>
46. Get username(not user id)
    ~$ whoami
47. Get user id by id command
    ~$ id –u
48. Get usernamme by id command
    ~$ id –u -n
49. Get unix name,kernel name,versions
    ~$ uname
50. Get os name and version
    ~$ uname –s –r
51. View all versions' information
    ~$ uname –v
52. Get only os name
    ~$ uname -s
53. Get only os version

    ~$ uname –r
54. Get system's disk usage of only home directory
    ~$ df –h ~
55. Get whole system's full disk usage
    ~$ df –h or ~$ df
    Writing –h makes the output of df command human readable like using T and G for showing disk space(T=terabytes,G=gigabytes)

56. Get root directory's disk usage in human readable form
    `$ df –h /
57. **Get current working directory's disk usage**
    ~$ df –h ./

58. See all currently running processes
    ~$ ps
59. See all currently running processes with user privileges
    ~$ ps -u
60. See all running processes with user privileges in root directory
    ~$ ps –u root

61. **SEE ALL running tasks**
    ~$ top
62. **See top 3 running tasks**
    ~$ top –n 3
63. **Print nothing**
    ~$ echo
64. **Print a string without spaces**
    ~$ echo hi
65. **Print a string with spaces-use quotes**
    ~$ echo "hi hello"
66. **Print value of a variable-use $ before variable name**
    ~$ echo $PATH
    PATH is also a variable which stores the absolute path of the present working directory
67. **Display system's current date and time**
    ~$ date
68. **Extract parts of date-**
    Display day number out of 365- %j
    ~$ date +%j or ~$ date "+%j"
    Display year in 4 digits- %Y
    ~$ date +%Y or ~$ date "+%Y"
    Display day of the week- %A
    ~$ date +%A or ~$ date "+%A"
69. **Print date in a format**
    ~$ date "+date= %j and day= %A and year= %Y"
    Output:date= 005 and day= thursday and year= 2023

70. **Get manual for id command and ps command**

~$ man id
~$ man ps
For ps and top,there is no manual

71. Get manual for df
~$ man df

72. Using find command which finds paths to all files in a particular directory which specify a particular criteria—case sensitive search
If we are in ~ directory and want to find all files with name A.txt in this same directory
~$ find . -name "A.txt"
"." specifies to search within the current working directory(pwd)
If we have both A.txt and a.txt files then only A.txt file's path will be returned

73. Case insensitive search-use iname
~$ find . -iname "A.txt" or ~$ find –iname "A.txt" pr ~$ find –iname A.txt
We can even write file name without quotes

74. If there is another directory in home directory called my_pr and it also contains A.path and a.path, then if we write . -name a.txt in home directory to find the file,we see paths of both the a.txt file in home directory and in all the directories inside it
~$ find –name a.txt
Finds all files with this name in every subdirectory of pwd
We can even remove "." before –name as by default we are searching in pwd

75. Make a new file with 2 words as name-use quotes
If file has only one word as name,we can remove quotes
~$ touch "hi hello.txt"

76. Find a file with 2 words-use quotes
~$ find –name "hi hello.txt"
If we just write ~$ find "hi hello.txt" without using –name preceding it, then we are just told by bash whether the file exists or not but we are not given the relative path to the file

77. Check whether a file exists or not-if it exists then its name is returned otherwise an error message is returned
~$ find "hi hello.txt"

78. Remove a file or empty folder
~$ rm <file/folder name>

79. Remove a non empty folder-removes all child elements inside this folder
~$ rm –r <foldername>

80. Remove an empty directory
~$ rmdir <directory name>

81. Remove a non empty directory
~$ rm –r <directoryname> or ~$ rm –rf <directoryname>

82. Find the date when a file was last modified
~$ date –r <filename>

83. Copy all files of one directory into another
~$ cp –r my_pr my_pr2
Recursively copy all files from m_pr directory into my_pr2 directory
Even if my_pr is empty or non empty,we have to use –r

84. Create a non existent directory ot folder and copy another directory/folder in it
~$ cp –r Documents new_pr
New_pr was first created and then Documents was copied in it

85. Move a file into a directory with that file nonexistent in it
~$ cp –r Documents <directory_name>

86. Move 2 or more files into a directory
~$ cp –r <file1> <file2> <directoryname>

87. If a file has rw permissions and we want to add execute permission to it
~$ chmod +x <filename>

88. If we want to add "s" permission
~$ chmod +s <filename>

89. Change contents of a file(replace method) by "sed" command
First read contents of file by cat command and then use sed command -separate both commands by |
~$ cat <filename> | sed 's/<original_text>/<new_text_to_replace_with>/g'

90. Read contents of one file, make a copy of them, overwrite them with given arguments and save them in a new file
If new file is already there, it is not created otherwise it is created
If sad.txt exixts from before,all its contents are overwriten with newly edited contents of file1.txt
But note:the original file is not changed
~$ cat file1.txt | sed 's/ff/gtgrf/g' > sad.txt
If sad.txt is not there,it is created and gets the newly edited contents of file1.txt but file1.txt is not changed
Every instance of ff is replaced with gtgrf

91. Read more than 2 files
~$ cat <file1> <file2>

92. If we don't want to overwrite contents of destination file but want to merge previous contents of destination file with incoming contents of source file
~$ cat file1.txt | sed 's/hi/ygeyfgfe/g' >> sad.txt
Now in sad.txt file,first we will see its contents and then below them the contents of file1.txt(after editing)
But still:contents of file1.txt re not changed

93. If we want to see a "preview" of what our file's contents will look like after editing, then we can do this by using sed command followed by file name and not using cat command
~$ sed 's/hi/yfgryfgfr/g' <filename>
This will print the newly edited contents of file but no changes were made and they were not stored in any other destination file

94. If we want to write in a file but don't want to read it fully as it might contain millions of lines
Use -I
~$ sed -i 's/hi/uhrff/g' <filename>
Directly the file is edited and we didn't have to read it before making changes

95. Make changes to file's contents and store in some other file without actually reading the original source file first
Note:chnages are not made in original file

~$ sed 's/hello/yt/g' file1.txt > sad.txt
Sad.txt is changed but file1.txt is not changed

96. Merge contents in destination file without reading original file first
~$ sed 's/hello/yt/g' file1.txt >> sad.txt

97. Find directories by a particular type(whether empty or non empty)
~$ find –type d -empty or ~$ find . -type d -empty
"." is used to search in pwd
But by default also we search in the pwd

98. If we want to search all empty directories in a directory inside the pwd
"d" stands for directory
~$ find ./Documents -type d –empty
Find all empty directories inside documents directory which is inside pwd

99. Find a particular word or words in a file
~$ egrep 'vivek|vardaan' file1.txt
Search all instances of both vardaan and vivek in file file1.txt
~$ grep vivek file1.txt

If we are searching for 2 words or more,use egrep instead of grep

100. Find all instances of a particular word/words in a pwd directory
~$ egrep –Rw vivek *
~$ egrep –Rw 'vivek|vardaan' *

101. Find all words' instances in a particular directory at same level
/home/Documents$ egrep –Rw 'hello|cheat' ./../pr_new
Pr_new is at same level as Documents directory

102. Find all words' instances in a particular file in a below directory(s)
/home/Documents$ egrep –Rw 'hello|cheat' ./my_pr ./my_pr2
Or /home/Documents$ egrep –Rw 'hello|cheat' my_pr my_pr2
My_pr ad my_pr2 are inside Documents directory

103. Find all files in a directory with a particular word(s)
~$ egrep –R vivek *
~$ egrep –R 'vivek|vardaan' *

104. Find all files with particular words in a below directory(s)
/home/Documents$ egrep –R 'hello|cheat' ./my_pr ./my_pr2
Or /home/Documents$ egrep –R 'hello|cheat' my_pr my_pr2

105. Find contents of a directory above pwd
We can't use cd .. along with ls -ltr
We have to use relative path with ls –ltr as we can't go to above directory and then see its contents
For seeing contents,we have to tell the location to ls –ltr
~$ ls –ltr ./..
It means from pwd,go one level up and then show its contents

106. Find all directories below pwd including pwd
~$ find –type d
Search in particular directory
~$ find ./Documents -type d
~$ find ./Documents -type d -empty

107. Find all empty files in a directory(including subdirectories)
~$ find –type f –empty
Find in a particular directory-search in documents directory which is one level below from pwd
$ find ./Documents -type f –empty
A file is empty if it has nothing written in it

108. Find all files in a directory(including subdirectories)
~$ find –type f
Find in a particular directory-search in documents directory which is one level below from pwd
$ find ./Documents -type f

109. Highlight word when searching in directory with color(default color)-use 2 dashes before color
~$ egrep --color -R 'hi fi' *

110. Find a word(s) in more than one files, directories
~$ egrep 'hello|cheat' f1.txt f2.txt
~$ egrep –R 'hello|cheat' my_pr my_pr2
Here my_pr and my_pr2 are both directories

111. Copy files from one directory below into another directory below pwd
Lets say we are at /home/Documents and it contains my_pr directory with f1.txt and f2.txt files
Lets say it also has an empty directory my_pr2
Copy files form my_pr into m_pr2
/home/Documents$ cp ./my_pr/f1.txt ./my_pr/f2.txt my_pr2

112. Move files from one directory below into another directory below pwd
/home/Documents$ mv ./my_pr/f1.txt ./my_pr/f2.txt my_pr2
Find all files without a particular word(s) inside a particular directory(s)
/home/Documents$ egrep –v –R 'hello|cheat' ./my_pr ./my_pr2
It finds all files in my_pr and my_pr2 directories without the words hello and cheat

113. Find all word instances without a particular word(s)
/home/Documents$ egrep –v –Rw 'hello|cheat' ./my_pr ./my_pr2

114. Find in a non case sensitive manner-use I
/home/Documents$ egrep –v -R -i 'hello|cheat' ./my_pr ./my_pr2
/home/Documents$ egrep –v –Rw -i 'hello|cheat' ./my_pr ./my_pr2

115. Sort ith column of a file in descending order (only sort by numeric values)
For numeric values use –n
For descending order use –r
For ascending write nothing

For ith column write ki where I is a natural number
~$ sort –k3 –n –r <filename>
Didn't read file before
Or ~$ cat <filename> | sort –k3 –n -r
Read file before sorting
Sort 3rd column of file by numeric values in descending order