**Name**= Shubham Abasaheb Sathe

**Email ID**= shubhamsathe66@gmail.com

**Mobile No**= 9405005055

**Batch**= 208

**Que1) Get user info from /etc/passwd and change ownership of userzs home directory (select userid higher than 1000)**

ANS=

Cat /etc/passwd

cat /etc/passed | awk -F ":" '{print $1}' /etc/passwd

Awk -F : '{if($3>=1000) {print "Valid user" $1" "$3}}' /etc/passwd

Awk -F : '{if($3>=1000) {print $6}}' /etc/passwd

Awk -F : '{if($3>=1000) {print $1" "$6}}' /etc/passwd

username=Awk -F : '{if($3>=1000) {print $1}}' /etc/passwd

homedirectory=Awk -F : '{if($3>=1000) {print $6}}' /etc/passwd

Sudo chown -R $username:$homedirectory

1. **Move files from one folder to the respective folders.mkdir folder/**

ANS=

cd folder/

touch abc.txt def.txt ghi.txt

for file in 'ls \*.txt'

> do

> echo $file;

> done

ls abc.txt def.txt ghi.txt

for file in 'ls \*.txt'

> do

> folderName=echo $file | awk -F. '{print $1}'

> echo $folderName

> done

for file in 'ls \*.txt'

> do

> folderName='echo $file | awk -F. '{print $1}'';

> mkdir $folderName;

> done

for file in 'ls \*.txt';

> do

> folderName='echo $file | awk -F. '{print $1}'';

> rm -r $folderName;

> mkdir $folderName;

> cp $file $folderName;

> done

ls abc def ghi

abc:

abc.txt

def:

def.txt

ghi:

Ghi.txt

**3)Append current date to all log files name which has extension .log.1 from a folder**

ANS=

cd newfolder/

touch abc.log.1 def.log.1 ghi.log.1 jkl.log.1 mno.log.1

ls

abc.log.1 def.log.1 ghi.log.1 jkl.log.1 mno.log.1

for file in 'ls \*.log.1'

> do

> filename=echo $file | awk -F. '{print $1}

> echo $filemane

> done

abc

def

ghi

jkl

mno

echo $(date)

Sun Nov 8 13:31:09 IST 2020

for file in 'ls \*.log.1'

>do

>filename=echo $file | awk -F. '{print $1}

>echo $filemane-"'date +%d%m%y'"

>done

abc.11102020

def.11102020

ghi.11102020

jkl.11102020

mno.11102020

**4)Archive the files from /var/log folder which have modified 7 days ago and move it to your backup folder**

ANS=

nano number5

#!/bin/bash

files=()

while IFS= read -r -d $'\0'; do

files+=("$REPLY")

done < <(find /var/log/ -mtime +7 -print0)

tar cvfz backup.tar.gz "${files[@]}"

chmod +x number5

./number5

**5)Check if a folder exists or not. If it’s not present, create it**

**nano number2**

#!/bin/bash

if [ -d ~/myfolder ]

then

echo "The folder exists."

else

echo "The folder does not exists."

mkdir myfolder/

fi

chmod +x number2

./number2

The folder does not exists.

./number2

The folder exists

**6)Execute command "hello" and "Is" and check its execution status and print whether command executed successful or not.**

ANS=

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ hello

-bash: hello: command not found

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ -bash: 127: command not found

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ echo hello

hello

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$-bash: 0: command not found

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ nano hello

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ chmod +x hello

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ ./hello

hello

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ -bash: 0: command not found

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ ls

= Music myfolder

Applications NetBeansProjects number10

Desktop NewFolder number3

Documents Pictures number4

Downloads Public number5

Hello.class VirtualBox VMs problem1

Hello.java backup.tar.gz userids

Library hello

Movies linux-content

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ -bash: 0: command not found

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$

**7)Set environment usersecret="dH34xJaa23” if its already not set**

ANS=

$ printenv usersecret

$ export usersecret

$ export usersecret=$(echo "dH34xJaa23")

$ printenv usersecret

dH34xJaa23

**8)Find a word "system" from all log files in the folder /var/log and print**

**number of occurrence more than 0 against each file.**

ANS=

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208 (master)

$ pwd

/Users/lenovo/Desktop/linux-content

lenovo-MacBook-Air:linux-content lenovo$ cat access.log | grep -c system

6

**9)Create process list table displays process id, parent process id,**

**command name, % of memory consumption, % of cpu utilization.**

ANS=

top | awk '{print "PID="$1 " " "COMMAND="$2 " " "%CPU="$3 " " "%MEM="$8 " " "PPID="$12}' | head -5

PID=Processes:  COMMAND=308    %CPU=total,    %MEM=1126 PPID=

PID=2020/10/19  COMMAND=17:52:04 %CPU=    %MEM=     PPID=

PID=Load    COMMAND=Avg:    %CPU=1.11,    %MEM=     PPID=

PID=CPU    COMMAND=usage:    %CPU=8.88%    %MEM=idle PPID=

PID=SharedLibs: COMMAND=449M    %CPU=resident %MEM=     PPID=

**10)Print last 4 frequently access urls count in sorted order from /var/log/httpd/access.log**

ANS=

cat access.log | awk '{print $11}' | grep -v " "| sort | uniq -c | sort -nr | head -4

1475 "https://fundoopush-dev.bridgelabz.com/login"

1141 "https://fundoopush-dev.bridgelabz.com/dashboard/article"

377 "-"

176 "https://fundoopush-dev.bridgelabz.com/add-post"

**11)Print list of last 4 frequently access unique urls at particular hours from /var/log/httpd/access.log**

ANS=

cat access.log | awk '{print $11}' | grep -v " "| sort | uniq -c | sort -nr | head -4

1475 "https://fundoopush-dev.bridgelabz.com/login"

1141 "https://fundoopush-dev.bridgelabz.com/dashboard/article"

**12)Print list of web response code count in the unique sorted order at specific hours**

ANS=

cat access.log | awk '{print $9}' | grep -v " "| sort | uniq -c | sort -n

8 206

26 304

3176 200

**13)Print list of last 10 unique sorted client IP from /var/log/httpd/access.log**

ANS=

cat access.log | awk '{print $1}' | grep -v " "| sort | uniq | sort -nr | head -10

10.56.9.3

10.56.6.4

10.56.5.2

10.56.46.2

10.56.44.4

10.56.4.2

10.56.34.4

10.56.3.4

10.56.22.3

10.56.21.2

**Q14) Data analysis / manipulation (Awk)**

i) Print Employee Name and TotalPay who has BasePay greater than 10000

a) Read data file data.csv' from command line and extract rows which have BasePay

>10000

b) Print only Employee Name and TotalPay

ANS=

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208/linux-content (master)

$ cat data.csv | awk '{if ($4 > 10000) print ("$2" "$7")}'

EmployeeName TotalPay

NATHANIEL 567595

GARY 538909

ALBERT 335279

CHRISTOPHER 332343

PATRICK 326373

DAVID 316285

ALSON 315981

DAVID 307899

JOANNE 302377

PATRICIA 297608

EDWARD 294580

ii) What is the aggregate Total Pay of employees whose jobtitle is 'CAPTAIN

a) Read data file 'data.csv' from command line and extract rows which have CAPTAIN' in

the column jobtitle

b) Extract TotalPay and calculate sum. Print the result on terminal.

ANS=

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208/linux-content (master)

$ cat data.csv | awk '{if($3 == "CAPTAIN") print($0)}'

2 GARY CAPTAIN 155966 245131 137811 538909 538909

3 ALBERT CAPTAIN 212739 106088 16452 335279 335279

12 PATRICIA CAPTAIN 99722 87082 110804 297608 297608

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208/linux-content (master)

$ cat data.csv | awk '{if($3 == "CAPTAIN") (sum+=$7) } END{print sum}'

1171796

iii) Print JobTitle and Overtimepay who has Overtimepay is between 7000 and 10000

a) Read data file data.csv from command line and extract jobtitle and overtime pay for

column value range

between 700010000

b) Print the result on terminal.

ANS=

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208/linux-content (master)

$ cat data.csv | awk '{if(7000 < $5 && $5 < 10000)print($3" "$5)}'

DEPUTYCHIEF 9737

ASSTDEPUTY 8601

iv) Print average Base Pay

a) Read data file 'data.csv' from command line and extract BasePay values and

calculate its average

b) Print the result on terminal.

ANS=

lenovo@DESKTOP-SCJFA4V MINGW64 ~/batch208/linux-content (master)

$ cat data.csv | awk '{(sum+=$4)} END{print(sum/NR)}'

157972

**15)Find the difference between original file and the updated file. Apply changes to the original file.**

a) Create two directories as "original and "updated"

b) Copy given file 'original-file.sh' to the folder "original" and "updated-file.sh"

to the

folder "updated"

c) Find the difference between these directories using linux command

d) Make copy of folder "original" to some other directory as "original-backup" and

apply changes to 'original-file.sh' file

e) Verify that both folders "updated" and "original-backup" have no difference.

ANS=

mkdir original/ updated/

nano original-file.sh

echo aurangabad

echo pune

echo nanded

echo ratnagiri

echo mumbai

chmod +x original-file.sh

./original-file.sh

aurangabad

pune

nanded

ratnagiri

mumbai

nano updated-file1.sh

echo pune

echo aurangabad

echo ratnagiri

echo nanded

echo mumbai

chmod +x updated-file1.sh

./updated-file1.sh

pune

aurangabad

ratnagiri

nanded

mumbai

cp original-file.sh original/

cp updated-file1.sh updated/

diff original/ updated/

Only in original/: original-file.sh

Only in updated/: updated-file1.sh

mkdir original-backup/

cp original-file.sh  original-backup/

diff original-backup/ updated/

Only in original-backup/: original-file.sh

Only in updated/: updated-file1.sh

diff original-file.sh  updated-file1.sh

1d0

< echo aurangabad

3c2

< echo nanded

---

> echo aurangabad

4a4

> echo nanded

5a6

>

nano original-file.sh

echo pune

echo aurangabad

echo ratnagiri

echo nanded

echo mumbai

chmod +x original-file.sh

./original-file.sh

pune

aurangabad

ratnagiri

nanded

Mumbai

diff original-file.sh  updated-file1.sh

5a6

>