Unix assignment 2

Name : Guthikonda kovida
Section : A
Roll no : 422154

Write shell script for pattern matching using awk, sed utilities, tar , cpio.
Tar
echo "Creating an uncompressed tar Archive"
tar cvf file.tar hello1.txt
ls

echo ""
echo ""

echo " gzip compression on the tar Archive, using option -z"
tar cvzf file1.tar.gz *.txt
ls

echo ""
echo ""

echo " Creating compressed tar archive file using option -j"
tar cvfj file2.tar.tbz hello1.txt
ls

echo ""
echo ""

echo " Check size of existing tar, tar.gz, tar.tbz"
tar czf file2.tar.tbz | wc -c

echo ""
echo ""

echo " Update existing tar file"
tar rvf file.tar hello2.txt
ls

echo ""
echo ""

echo " List the contents and specify the tarfile using option"

```
tar tf file.tar

echo ""
echo ""

echo "Applying pipe to through grep command"
tar tvf file.tar | grep "hello"

echo ""
echo ""

echo "Viewing the Archive"
tar tvf file.tar

echo ""
echo ""

echo "pass a file name as an argument to search a tarfile"
tar tvf file.tar hello1.txt

echo ""
echo ""

echo "Verifies the integrity of an archive"
tar -vcW 100M file.tar hello2.txt

echo ""
echo ""

echo "Updates or adds files or directories to an already existing archive"
tar -Avf file.tar hello3.txt
ls

echo ""
echo ""

echo "Extracting files from Archive using option -xvf"
tar xvf file.tar
ls

echo ""
```

```
echo ""

echo "Untar single tar file "
tar xvfj file2.tar.tbz
ls

echo ""
echo ""

echo " Extracting a gzip tar Archive *.tar.gz using option -xvzf :"
tar xvzf file1.tar.gz
ls
```
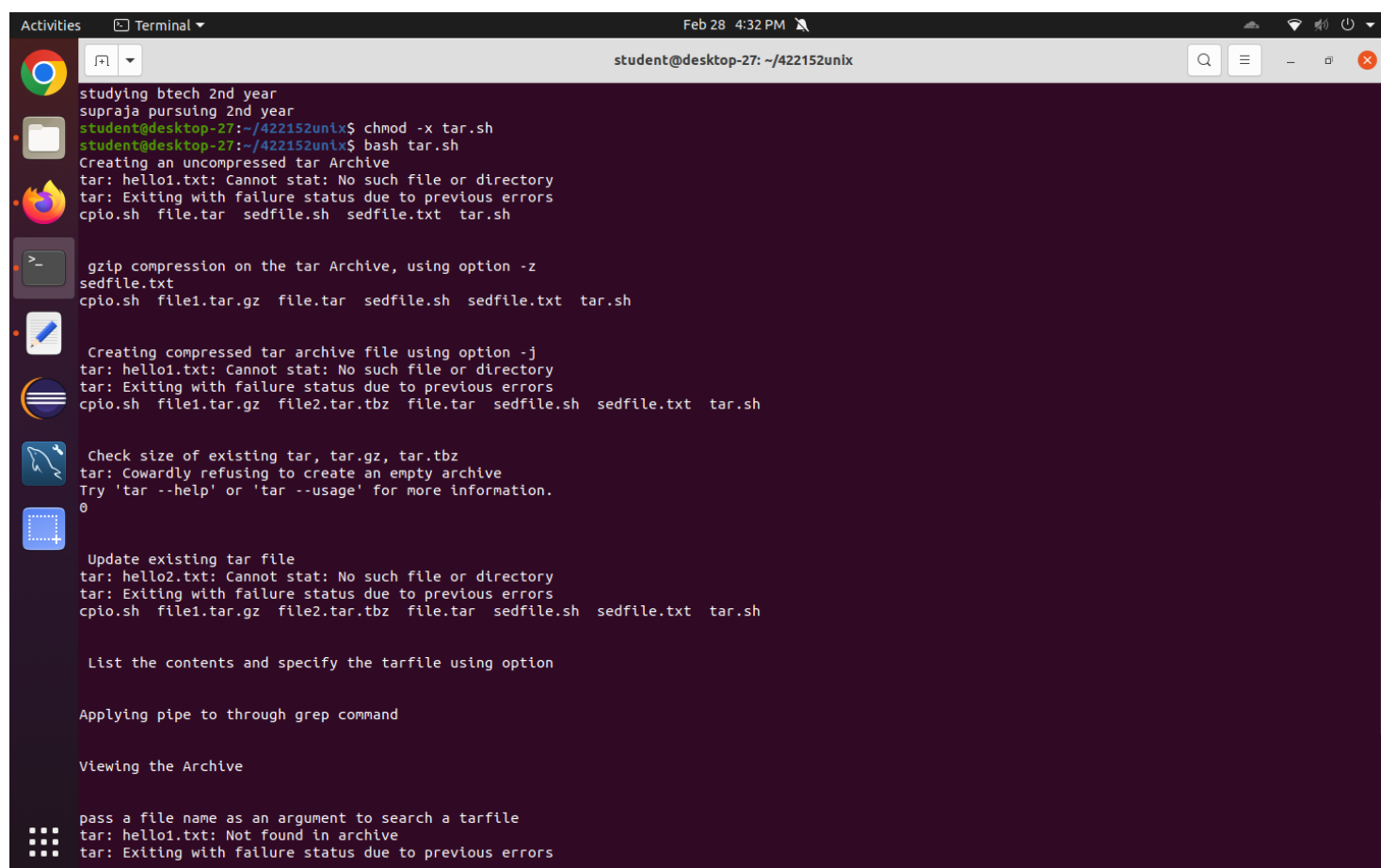
Output:

```
 student@desktop-27: ~/422152unix
```

```
 Update existing tar file
tar: hello2.txt: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
cpio.sh  file1.tar.gz  file2.tar.tbz  file.tar  sedfile.sh  sedfile.txt  tar.sh


 List the contents and specify the tarfile using option

Applying pipe to through grep command

Viewing the Archive

pass a file name as an argument to search a tarfile
tar: hello1.txt: Not found in archive
tar: Exiting with failure status due to previous errors

Verifies the integrity of an archive
tar: Refusing to write archive contents to terminal (missing -f option?)
tar: Error is not recoverable: exiting now

Updates or adds files or directories to an already existing archive
tar: hello3.txt: Cannot open: No such file or directory
tar: Exiting with failure status due to previous errors
cpio.sh  file1.tar.gz  file2.tar.tbz  file.tar  sedfile.sh  sedfile.txt  tar.sh

Extracting files from Archive using option -xvf
cpio.sh  file1.tar.gz  file2.tar.tbz  file.tar  sedfile.sh  sedfile.txt  tar.sh

Untar single tar file
cpio.sh  file1.tar.gz  file2.tar.tbz  file.tar  sedfile.sh  sedfile.txt  tar.sh

 Extracting a gzip tar Archive *.tar.gz using option -xvzf :
sedfile.txt
cpio.sh  file1.tar.gz  file2.tar.tbz  file.tar  sedfile.sh  sedfile.txt  tar.sh
student@desktop-27:~/422152unix$ ▯
```

Sed file

#!/bin/bash

echo "#replaces the word unix with linux in the file line by line"

sed 's/kovida/geetha/' sedfile.txt

echo ""

echo ""

echo "#Replacing the nth occurrence of a pattern in a line"

sed 's/2nd/3rd/2' sedfile.txt

echo ""

echo ""

echo "#Replacing all the occurrence of the pattern in a line "

sed 's/kovida/geetha/g' sedfile.txt

echo ""

echo ""

echo "#Replacing string on a specific line number "

sed '3 s/kovida/geetha/' sedfile.txt

echo ""

echo ""

echo "#Replacing string on a specific line numbers range "

sed '3,5 s/2nd/4th/' sedfile.txt

echo ""

echo ""

```
echo "#Replacing string on a particular line to entire numbers range "
sed '2,$ s/geetha/kovida/' sedfile.txt
echo ""
echo ""
echo "#Duplicating the replaced line with /p flag"
sed 's/kovida/vijaya/p' sedfile.txt
echo ""
echo ""
echo "#printing only the replaced lines"
sed -n 's/kovida/vijya/p' sedfile.txt
echo ""
echo ""
echo "#To Delete a particular line say "
sed '5d' sedfile.txt
echo ""
echo ""
echo "#To Delete a last line"
sed '$d' sedfile.txt
echo ""
echo ""
echo "#To Delete line from range x to y"
sed '3,6d' sedfile.txt
echo ""
echo ""
echo "#To Delete from nth to last line"
sed '12,$d' sedfile.txt
echo ""
echo ""
echo "#To Delete pattern matching line"
sed '/os/d' sedfile.txt
Output:
```

```
student@desktop-27:~$ cd 422152unix
student@desktop-27:~/422152unix$ chmod -x sedfile.sh
student@desktop-27:~/422152unix$ bash sedfile.sh
#replaces the word unix with linux in the file line by line
this is kovidaa
studying btech 2nd year
kovidaa pursuing 2nd year


#Replacing the nth occurrence of a pattern in a line
this is supraja
studying btech 2nd year
supraja pursuing 2nd year


#Replacing all the occurrence of the pattern in a line
this is kovidaa
studying btech 2nd year
kovidaa pursuing 2nd year


#Replacing string on a specific line number
this is supraja
studying btech 2nd year
supraja pursuing 2nd year


#Replacing string on a specific line numbers range
this is supraja
studying btech 2nd year
supraja pursuing 4th year


#Replacing string on a particular line to entire numbers range
this is supraja
studying btech 2nd year
supraja pursuing 2nd year


#Duplicating the replaced line with /p flag
this is ramya
this is ramya
studying btech 2nd year
ramya pursuing 2nd year
ramya pursuing 2nd year
```

```
this is supraja
studying btech 2nd year
supraja pursuing 2nd year

#Duplicating the replaced line with /p flag
this is ramya
this is ramya
studying btech 2nd year
ramya pursuing 2nd year
ramya pursuing 2nd year

#printing only the replaced lines
this is ramya
ramya pursuing 2nd year

#To Delete a particular line say
this is supraja
studying btech 2nd year
supraja pursuing 2nd year

#To Delete a last line
this is supraja
studying btech 2nd year


#To Delete line from range x to y
this is supraja
studying btech 2nd year


#To Delete from nth to last line
this is supraja
studying btech 2nd year
supraja pursuing 2nd year


#To Delete pattern matching line
this is supraja
studying btech 2nd year
supraja pursuing 2nd year
student@desktop-27:~/422152unix$
```
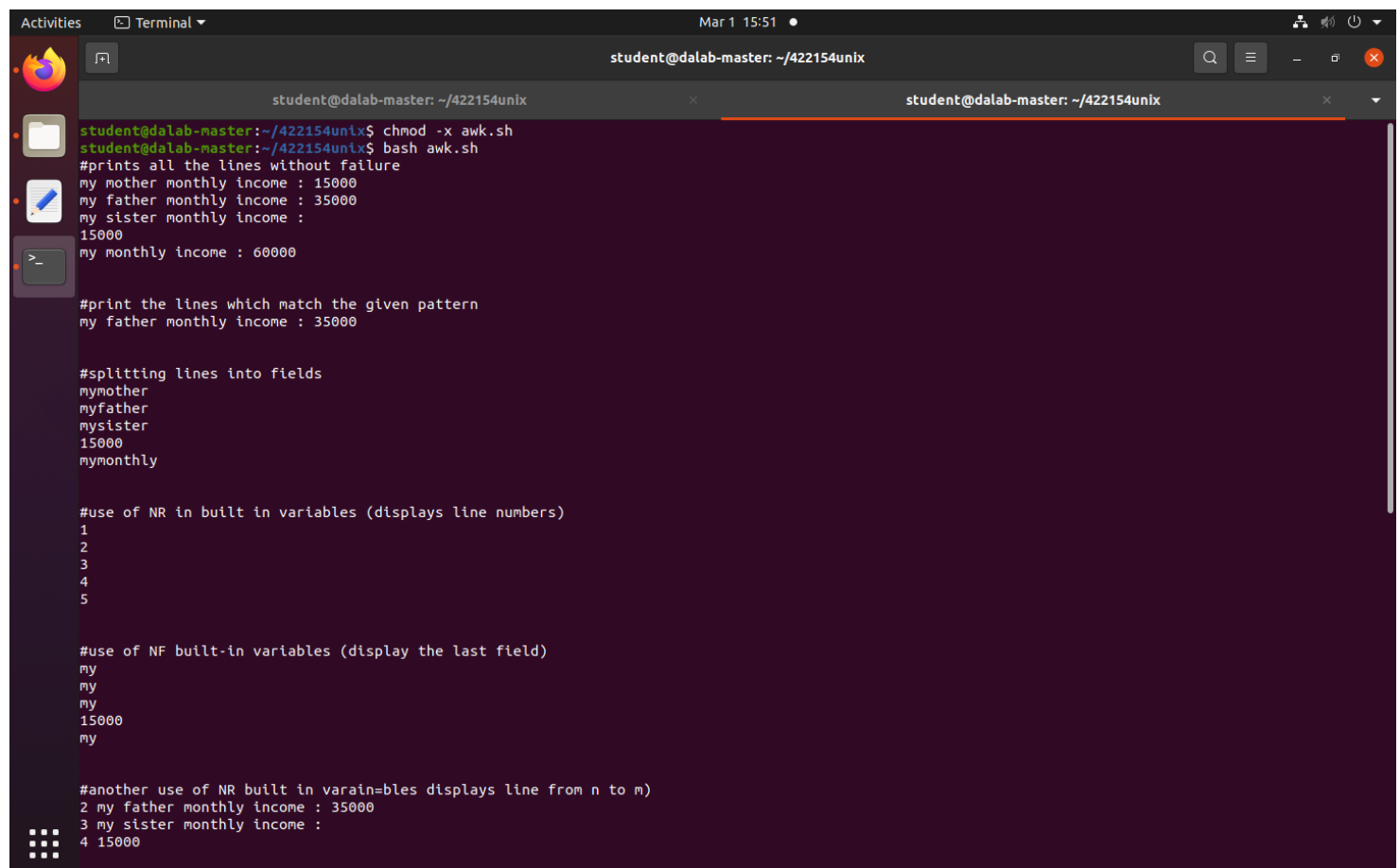
Awk

#!/bin/bash

```
echo "#prints all the lines without failure"
awk '{print}' awk.txt
echo ""
echo ""
echo "#print the lines which match the given pattern"
awk '/father/{print}' awk.txt
echo ""
echo ""
echo "#splitting lines into fields"
awk '{print $1 $2}' awk.txt
echo ""
echo ""
echo "#use of NR in built in variables (displays line numbers)"
awk '{print NR, S0}' awk.txt
echo ""
echo ""
echo "#use of NF built-in variables (display the last field)"
awk '{print $1,SNF}' awk.txt
echo ""
echo ""
echo "#another use of NR built in varain=bles displays line from n to m)"
awk 'NR==2, NR==4 {print NR ,$0}' awk.txt
echo ""
echo ""
echo "# print the first item along with a row number separated with "-" from each line in file"
awk '{print NR "- " $1}' awk2.txt
echo ""
echo ""
echo "#to return the second item from file"
awk '{print $2}' awk2.txt
echo ""
echo ""
echo "#to find the length of the longest line present"
awk '{if (length($0) > max)max = length($0) } END { print max }' awk2.txt
echo ""
echo ""
echo "#to count the lines in file"
awk 'END { print NR }' awk2.txt
echo ""
echo ""
echo "#printing the lines more than 10 characters"
```

```
awk 'length($0) >8' awk2.txt
echo ""
echo ""
echo "#to check for any string in any specific column"
awk '{ if($3 == "Q3") print $0;}' awk2.txt
echo ""
echo ""
echo "#BEGIN is used to execute the block of code before processing any input."
awk 'BEGIN { for (i=1; i<=3; i++) print "cube of ", i , "is" , i*i*i;}'
```

Output:

```
2 my father monthly income : 35000
3 my sister monthly income :
4 15000


# print the first item along with a row number separated with "-" from each line in file
1- C
2- Q3
3- hkc
4- lab


#to return the second item from file
B
4TGT
6tsa
assI-5


#to find the length of the longest line present
31


#to count the lines in file
4


#printing the lines more than 10 characters
A          B              C
H33        4TGT           Q3
UNIX       6tsa           hkc
422154     assI-5         lab


#to check for any string in any specific column
H33        4TGT           Q3


#BEGIN is used to execute the block of code before processing any input.
cube of  1 is 1
cube of  2 is 8
cube of  3 is 27
student@dalab-master:~/422154unix$
```

Cpio

```bash
#!/bin/bash

echo "#create a cpio file"
ls | cpio -ov > one.cpio

echo ""
echo ""

echo "#extract a cpio file"
ls | cpio -iv < one.cpio

echo ""
echo ""

echo "#creating tar using cpio"
ls | cpio -ov -H tar > one.tar

echo ""
echo ""

echo "#extract tar using cpio"
```

```
cpio -iv -F  one.tar

echo ""
echo ""

echo "#To create a *.tar archive with selected files"
find . -name "*.txt" | tar -cvf one.tar -T -

echo ""
echo ""

echo "#To only view *.tar archive file "
cpio -it < one.tar
```
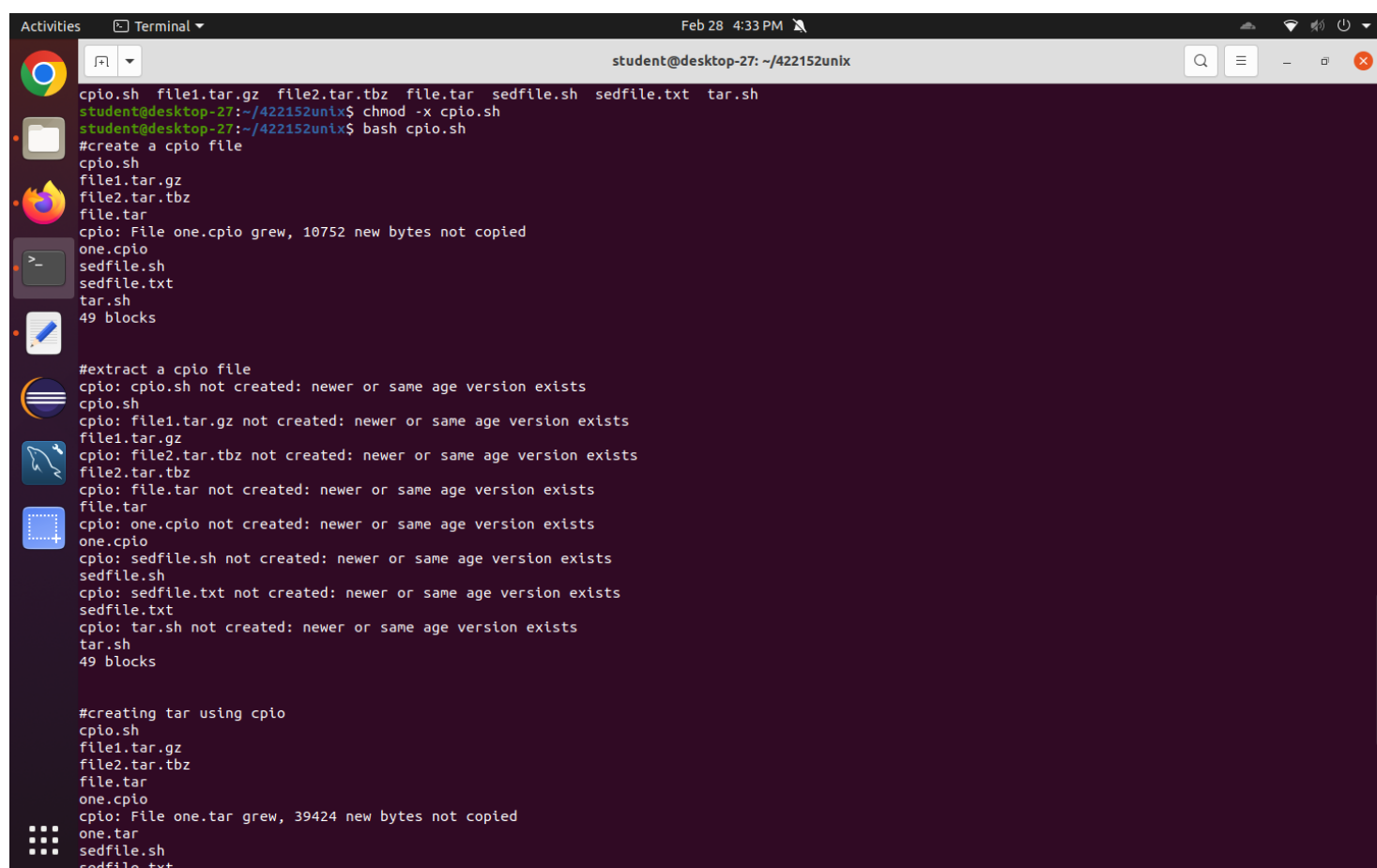
Output:

```
#creating tar using cpio
cpio.sh
file1.tar.gz
file2.tar.tbz
file.tar
one.cpio
cpio: File one.tar grew, 39424 new bytes not copied
one.tar
sedfile.sh
sedfile.txt
tar.sh
166 blocks

#extract tar using cpio
cpio: cpio.sh not created: newer or same age version exists
cpio.sh
cpio: file1.tar.gz not created: newer or same age version exists
file1.tar.gz
cpio: file2.tar.tbz not created: newer or same age version exists
file2.tar.tbz
cpio: file.tar not created: newer or same age version exists
file.tar
cpio: one.cpio not created: newer or same age version exists
one.cpio
cpio: one.tar not created: newer or same age version exists
one.tar
cpio: sedfile.sh not created: newer or same age version exists
sedfile.sh
cpio: sedfile.txt not created: newer or same age version exists
sedfile.txt
cpio: tar.sh not created: newer or same age version exists
tar.sh
165 blocks

#To create a *.tar archive with selected files
./sedfile.txt

#To only view *.tar archive file
./sedfile.txt
3 blocks
student@desktop-27:~/422152unix$ 
```

Text files

Sed file

this is kovida
studying btech 2nd year
kovida pursuing 2nd year

Awk
First file

my mother monthly income : 15000
my father monthly income : 35000
my sister monthly income :
15000
my monthly income : 60000

Second file

A	B	C
H33	4TGT	Q3

UNIX   6tsa   hkc
422154 assI-5   lab