

LINUX AND SHELL PROGRAMMING - LAB

(Course Code: 22UPCSC1C04)

A laboratory record submitted to Periyar University, Salem

In partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER APPLICATIONS

By

ELANCHEZHIAN M

[Reg. No: U22PG507CAP006]



DEPARTMENT OF COMPUTER SCIENCE

PERIYAR UNIVERSITY

(NACC `A++` Grade with CGPA 3.61) – NIRF RANK 63 – ARIIA RANK 10

PERIYAR PALKALAI NAGAR,

SALEM – 636 011.

(NOVEMBER - 2022)

CERTIFICATE

This is to certify that the Programming Laboratory entitled “**LINUX AND SHELL PROGRAMMING - LAB (22UPCSC1C04)**” is a bonafide record work done by Mr. / Ms. _____

Register No: _____ in partial fulfillment of the requirements for the degree of Master of Computer Applications, in the Department of Computer Science, Periyar University, Salem, during the Academic Year 2022-2023.

Staff In-charge

Head of the Department

Submitted for the practical examination held on.....

Internal Examiner

External Examiner

CONTENTS

S.NO	DATE	TITLE OF THE PROGRAM	PAGE NO	SIGNATURE
1.		CALCULATE THE NUMBER OF DAYS BETWEEN TWO DATES		
2.		CHECK SYSTEM ON LOCAL NETWORK USING CONTROLS STRUCTURE WITH USER INPUT		
3.		CHECK SYSTEM ON LOCAL NETWORK USING CONTROLS STRUCTURE WITH FILE INPUT		
4.		THE SCRIPT CONTROL COMMAND		
5.		SHELL SCRIPT FUNCTION		
6.		REGULAR EXPRESSIONS		
7.		SED AND GAWK COMMANDS		
8.		DEMONSTRATE FILE BACKUP PROCESS THROUGH CREATING A DAILY ARCHIVE LOCATIONS		
9.		CREATE A FOLLOWING GUI TOOLS (A) CREATING TEXT MENUS. (B) BUILDING TEXT WINDOW WIDGETS		
10.		CONNECT A POSTGRESQL DATABASE AND PERFORMING CRUD OPERATIONS		

SOURCE CODE:

```
echo "Enter the date 1: "  
read d1  
echo "Enter the date 2: "  
read d2  
days=$(( ($(date -d $d2 +%s) - $(date -d $d1 +%s)) / 86400))  
echo "The different between $d1 and $d2 is $days day"
```

OUTPUT:

```
Enter the date 1:  
2015-03-05  
Enter the date 2:  
2015-03-11  
The different between 2015-03-05 and 2015-03-11 is 6 day
```

SOURCE CODE:

```
while getopts t: opt
do
case "$opt" in t)
if [ $OPTARG = "IPv4" ]
then
pingcommand=$(which ping)
elif [ $OPTARG = "IPv6" ]
then
pingcommand=$(which ping6)
fi;;
*) echo "Usage: -t IPv4 or -t IPv6"
echo "Exiting script..."
exit;;
esac
shift $(( $OPTIND - 1 ))
if [ $# -eq 0 ]
then
echo "\nIP Address(es) parameters are missing."
echo "\nExiting script..."
exit
fi
for ipaddress in "$@"
do
echo "\nChecking system at $ipaddress..."
echo
$pingcommand -q -c 3 $ipaddress
echo
done
exit
done
```

OUTPUT:

```
elanchezhian@elanchezhian-virtual-machine:~/Desktop/Linux$ sh po2.sh -t IPv4 192.168.240.1
```

```
Checking system at 192.168.240.1...
```

```
PING 192.168.240.1 (192.168.240.1) 56(84) bytes of data.
```

```
--- 192.168.240.1 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
```

```
rtt min/avg/max/mdev = 1.744/2.015/2.421/0.292 ms
```

SOURCE CODE:

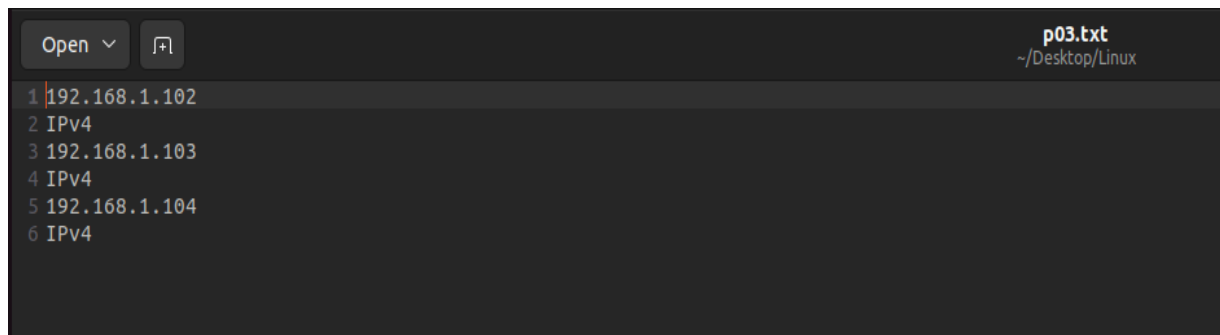
```
echo "\nPlease enter the file name with an absolute directory
reference...\n"
choice=0
while [ $choice -eq 0 ]
do
    read -p "Enter name of file:" filename
    if [ -z $filename ]
    then
        quitanswer=""
    else
        choice=1
    fi
done
if [ -s $filename ] && [ -r $filename ]
then
    echo "$filename is a file, is readable, and is not empty."
    echo
    cat $filename | while read line
    do
        ipaddress=$line
        read line
        iptype=$line
        if [ $iptype = "IPv4" ]
        then
            pingcommand=$(which ping)
        else
            pingcommand=$(which ping6)
        fi
    done
    echo "Checking system at $ipaddress..."
    $pingcommand -q -c 3 $ipaddress
done
```



```
echo "\nFinished processing the file. All systems checked."
else
echo "\n$filename is either not a file, is empty, or is not readable by
you. Exiting script..."
fi
exit
```

OUTPUT:

➤ P03.txt file

A screenshot of a text editor window titled 'p03.txt' with a subtitle '~/.Desktop/Linux'. The editor shows a list of six lines: 1 192.168.1.102, 2 IPv4, 3 192.168.1.103, 4 IPv4, 5 192.168.1.104, and 6 IPv4. The first line is highlighted with a mouse cursor at the end of the IP address.

```
Open  [icon] p03.txt  
~/.Desktop/Linux  
1 192.168.1.102  
2 IPv4  
3 192.168.1.103  
4 IPv4  
5 192.168.1.104  
6 IPv4
```

```
Please enter the file name with an absolute directory reference...
```

```
Enter name of file:p03.txt
```

```
p03.txt is a file, is readable, and is not empty.
```

```
Checking system at 192.168.1.102...
```

```
PING 192.168.1.102 (192.168.1.102) 56(84) bytes of data.
```

```
--- 192.168.1.102 ping statistics ---
```

```
3 packets transmitted, 0 received, 100% packet loss, time 2035ms
```

```
Checking system at 192.168.1.103...
```

```
PING 192.168.1.103 (192.168.1.103) 56(84) bytes of data.
```

```
--- 192.168.1.103 ping statistics ---
```

```
3 packets transmitted, 0 received, 100% packet loss, time 2055ms
```

```
Checking system at 192.168.1.104...
```

```
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
```

```
--- 192.168.1.104 ping statistics ---
```

```
3 packets transmitted, 0 received, 100% packet loss, time 2054ms
```

```
Finished processing the file. All systems checked.
```

SOURCE CODE:

```
#SIGINT

trap "echo 'sorry! I have trapped ctrl+c'" INT

echo "This is a test script"

count=1

while [ $count -le 5 ]
do
    echo "Loop #$count"
    sleep 2
    count=$(( $count+1 ))
done

echo "This is the end of the test script"


#SIGQUIT

trap "echo 'sorry! I have trapped ctrl+\'" QUIT

echo "This is Quit process"

count=1

while [ $count -le 5 ]
do
    echo "Loop #$count"
    sleep 2
    count=$(( $count+1 ))
done

echo "Quit the Process"
```

```
#SIGSTOP
trap STOP
echo "This is Stop process"
count=1
while [ $count -le 5 ]
do
    echo "Loop #$count"
    sleep 2
    count=$(( $count+1 ))
done
echo "Stop the Process"
```

OUTPUT:

```
This is a test script
Loop #1
Loop #2
Loop #3
^C'sorry! I have trapped ctrl+c'
Loop #4
Loop #5
This is the end of the test script
This is Quit process
Loop #1
Loop #2
^\\Quit (core dumped)
'sorry! I have trapped ctrl+'
Loop #3
Loop #4
Loop #5
Quit the Process
This is Stop process
Loop #1
Loop #2
^Z
[1]+  Stopped                               sh p06.sh
```

SOURCE CODE:

```
read -p "Enter the Fibonacci number: " n
fib(){
    i=0
    f1=0
    f2=1
    echo "The Fibonacci Series for $n is:"
    while [ $i -le $n ]
    do
        echo "$f1"
        temp=$((f1+f2))
        f1=f2
        f2=temp
        i=$((i+1))
    done
}
fib
```

OUTPUT:

```
Enter the Fibonacci number: 5
The Fibonacci Series for 5 is:
0
1
1
2
3
5
```

SOURCE CODE:

```
fruits_file=$(cat fruit.txt | grep App.e)
echo "\n1. Using '.' to find out all the original word wheres given
word is 'App.e'"
echo "Output:\n$fruits_file"
fruits_file=$(cat fruit.txt | grep Ap*le)
echo "\n2. Using '*' to find out all the fruits name of 'Ap' one after
another in it"
echo "Output:\n$fruits_file"
fruits_file=$(cat fruit.txt | grep ^B)
echo "\n3. Using '^' to find out all the words that start with the letter
'B'"
echo "output:\n$fruits_file"
fruits_file=$(cat fruit.txt | grep "\ ")
echo "\n4. Using '\' to find out all the fruits name that has single space
in their full name"
echo "Output:\n$fruits_file"
fruits_file=$(cat fruit.txt | grep -E Ch?)
echo "\n5. Using '?' to find out all the fruits name that has 'Ch' in it"
echo "Output:\n$fruits_file"
fruits_file=$(cat fruit.txt | grep -E "(fruit)")
echo "\n6. Using '()' to find out all the fruits name that has word
'fruit' in it"
echo "Output:\n$fruits_file"
```


OUTPUT:

➤ fruit.txt file

```
fruit.txt
~/Desktop/Linux

1 Apple
2 Banana
3 Bil Berry
4 Black Berry
5 custard Apple
6 Currant
7 Cherimoya
8 Chico Fruit
9 Drangonfruit
10 Goji Berry
11 Juniper Berry
12 Passuib Fruit
13 Star Fruit
14 Salal Berry
15 Ugli Fruit
```

```
1. Using '.' to find out all the original word wheres given word is 'App.e'
Output:
Apple
custard Apple

2. Using '*' to find out all the fruits name of 'Ap' one after another in it
Output:
Apple
custard Apple

3. Using '^' to find out all the words that start with the letter 'B'
output:
Banana
Bil Berry
Black Berry

4. Using '\' to find out all the fruits name that has single space in their full name
Output:
Bil Berry
Black Berry
custard Apple
Chico Fruit
Goji Berry
Juniper Berry
Passuib Fruit
Star Fruit
Salal Berry
Ugli Fruit

5. Using '?' to find out all the fruits name that has 'Ch' in it
Output:
Currant
Cherimoya
Chico Fruit

6. Using '()' to find out all the fruits name that has word 'fruit' in it
Output:
Drangonfruit
```

SOURCE CODE (sed command) :

```
echo "1.Replacing or substituting string:"
echo "-----"
sed 's/unix/linux/' sed.txt
echo
echo "2.Replacing the nth occurrence of a pattern in a line:"
echo "-----"
sed 's/unix/linux/2' sed.txt
echo
echo "3.Replacing all the occurrence of the pattern in a line:"
echo "-----"
sed 's/unix/linux/g' sed.txt
echo
echo "4.Replacing from nth occurrence to all occurrences in a line:"
echo "-----"
sed 's/unix/linux/3g' sed.txt
echo
echo "5.Replacing string on a specific line number:"
echo "-----"
sed '3 s/unix/linux/' sed.txt
echo
echo "6.Duplicating the replaced line with /p flag:"
echo "-----"
sed 's/unix/linux/p' sed.txt
echo
echo "7.Printing only the replaced lines:"
echo "-----"
sed -n 's/unix/linux/p' sed.txt
echo
echo "8.Replacing string on a range of lines:"
echo "-----"
sed '2,$ s/unix/linux/' sed.txt
echo
```

```
echo "9.Deleting lines from a particular file:"  
echo "-----"  
sed '2,4d' sed.txt
```

SOURCE CODE (gawk command) :

```
gawk 'BEGIN { print "Enter the mark:"  
getline mark < "-"  
if (mark >= 90) print "A+"  
else if( mark >= 80) print "A"  
else if( mark >= 70) print "B+"  
else if( mark >= 60) print "B"  
else if( mark >= 50) print "C+"  
else print "Fail" }'
```

OUTPUT (sed command):

➤ Sed.txt file:

```
Open  sed.txt
~/Desktop
1 unix is great os. unix is opensource. unix is free os.
2 learn operating system.
3 unix linux which one you choose.
4 unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

```
1.Replacing or substituting string:
-----
linux is great os. unix is opensource. unix is free os.
learn operating system.
linux linux which one you choose.
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

2.Replacing the nth occurrence of a pattern in a line:
-----
unix is great os. linux is opensource. unix is free os.
learn operating system.
unix linux which one you choose.
unix is easy to learn.linux is a multiuser os.Learn unix .unix is a powerful.

3.Replacing all the occurrence of the pattern in a line:
-----
linux is great os. linux is opensource. linux is free os.
learn operating system.
linux linux which one you choose.
linux is easy to learn.linux is a multiuser os.Learn linux .linux is a powerful.

4.Replacing from nth occurrence to all occurrences in a line:
-----
unix is great os. unix is opensource. linux is free os.
learn operating system.
unix linux which one you choose.
unix is easy to learn.unix is a multiuser os.Learn linux .linux is a powerful.

5.Replacing string on a specific line number:
-----
unix is great os. unix is opensource. unix is free os.
learn operating system.
linux linux which one you choose.
unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

6.Duplicating the replaced line with /p flag:
-----
linux is great os. unix is opensource. unix is free os.
linux is great os. unix is opensource. unix is free os.
learn operating system.
linux linux which one you choose.
linux linux which one you choose.
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

7.Printing only the replaced lines:
-----
linux is great os. unix is opensource. unix is free os.
linux linux which one you choose.
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

8.Replacing string on a range of lines:
-----
unix is great os. unix is opensource. unix is free os.
learn operating system.
linux linux which one you choose.
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

9.Deleting lines from a particular file:
-----
unix is great os. unix is opensource. unix is free os.
```

OUTPUT (gawk command) :

```
Enter the mark:  
90  
A+
```

SOURCE CODE:

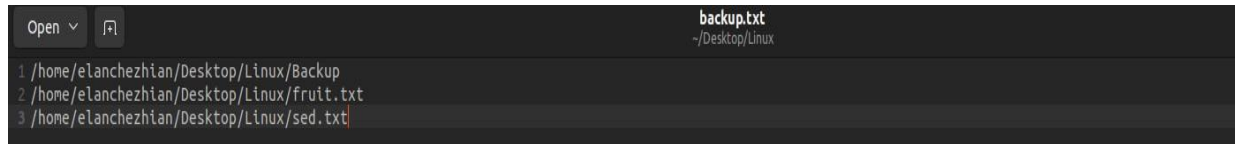
```
DATE=$(date +%y%m%d)
read -p "Give name to the archive file:" file
FILE=$file$DATE.tgz
read -p "Enter the Filename: " SOURCE
read -p "Enter the Destination path: " des
DESTINATION=$des/$FILE
if [ -f $SOURCE ]
then
    echo
else
    echo "$SOURCE doesn't exist, BACKUP INCOMPLETE"
    exit
fi
FILE_NO=1
exec < $SOURCE
read FILE_NAME
while [ $? -eq 0 ]
do
    if [ -f $FILE_NAME -o -d $FILE_NAME ]
    then
        FILE_LIST="$FILE_LIST $FILE_NAME"
    else
        echo "$FILE_NAME doesn't exist, thus it is not included"
        echo "BACKUP is still on process"
        echo
    fi
    FILE_NO=$((FILE_NO+1))
    read FILE_NAME
done
echo "Starting Archive..."
tar -czf $DESTINATION $FILE_LIST 2>/dev/null
echo "Archive COMPLETED at $DESTINATION"
exit
```

OUTPUT:

- Creating the file to Store the backup file:

```
elanchezhian@elanchezhian-virtual-machine:~/Desktop/Linux$ gedit backup.txt
```

- Backup Files:

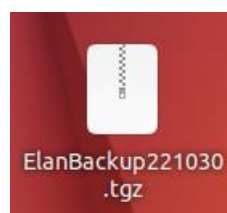


```
Open ▾ [icon] backup.txt  
~/Desktop/Linux  
1 /home/elanchezhian/Desktop/Linux/Backup  
2 /home/elanchezhian/Desktop/Linux/fruit.txt  
3 /home/elanchezhian/Desktop/Linux/sed.txt
```

- Running the Script to Backup the Files

```
Give name to the archive file:ElanBackup  
Enter the Filename : backup.txt  
Enter the Destination path: /home/elanchezhian/Desktop  
  
Starting Archive...  
Archive COMPLETED at /home/elanchezhian/Desktop/ElanBackup221030.tgz
```

- Archive file (ElanBakup221030.tgz):



- Listing of the Archive contents from a Terminal Prompt Type:

```
elanchezhian@elanchezhian-virtual-machine:~/Desktop/Linux$ tar -tzvf /home/elanchezhian/Desktop/ElanBackup221030.tgz  
drwxrwxr-x elanchezhian/elanchezhian 0 2022-10-30 09:25 home/elanchezhian/Desktop/Linux/Backup/  
-rw-rw-r-- elanchezhian/elanchezhian 239 2022-10-29 12:06 home/elanchezhian/Desktop/Linux/Backup/sample1.sh  
-rw-rw-r-- elanchezhian/elanchezhian 166 2022-10-28 22:06 home/elanchezhian/Desktop/Linux/fruit.txt  
-rw-rw-r-- elanchezhian/elanchezhian 133 2022-10-28 22:21 home/elanchezhian/Desktop/Linux/sed.txt
```

SOURCE CODE:

A) Creating Text Menus

```
diskspace() {
    clear
    df -k
}
diskspace
whoseon() {
    clear
    who
}
whoseon
memusage() {
    clear
    cat /proc/meminfo
}
memusage
menu(){
    clear
    echo
    echo "\t\t\tSys Admin Menu\n"
    echo "\t\t1. Display disk space"
    echo "\t\t2. Display logged on users"
    echo "\t\t3. Display memory usage"
    echo "\t\t0. Exit program\n\n"
    echo
    echo "\t\tEnter option: "
    read option
    echo
}
menu
```



```

while [ True ]
do
    menu
    case $option in
    0)
        break ;;
    1)
        diskpace ;;
    2)
        whoseon ;;
    3)
        memusage ;;
    *)
        clear
        echo "Sorry, wrong selection";;
    esac
echo "\n\n\t\tHit any key to continue"
read line
done
clear

```

B) Text window widgets

```

temp=$(mktemp -t test.XXXXXXX)
temp2=$(mktemp -t test2.XXXXXXX)
function diskpace {
    clear
    df -k> $temp
    dialog --textbox $temp 20 50
}
function whoseon {
    clear
    who> $temp
    dialog --textbox $temp 20 50
}

```

```

}
function memusage {
    clear
    cat /proc/meminfo> $temp
    dialog --textbox $temp 20 50
}
while [ 1 ]
do
    clear
    dialog --menu "Sys Admin Menu" 20 30 10 1 "Display
diskspace" 2 "Display users" 3 "Display memory usage"
2> $temp2
    if [ $? -eq 1 ]
    then
        break
    fi
selection=$(cat $temp2)
case $selection in
    1) diskspace ;;
    2) whoseon ;;
    3) memusage ;;
    *) dialog --msgbox "Sorry, invalid selection" 10 30
esac
done
clear
rm -f $temp 2> /dev/null
rm -f $temp2 2> /dev/null

```

OUTPUT:

A) Text Menus

```

                                     Sys Admin Menu

1. Display disk space
2. Display logged on users
3. Display memory usage
0. Exit program

Enter option:
```

1. Displaying Disk Space

```

                                     Enter option:
1

Filesystem      1K-blocks    Used Available Use% Mounted on
tmpfs            198824      1820   197004   1% /run
/dev/sda3       91789000 14707824  72372620 17% /
tmpfs           994120        0   994120   0% /dev/shm
tmpfs           5120         4    5116    1% /run/lock
/dev/sda2       524252     5364   518888   2% /boot/efi
tmpfs           198824     4728   194096   3% /run/user/1000
/dev/sr0        129778    129778        0 100% /media/elanchezhian/CDROM
/dev/sr1        3737140  3737140        0 100% /media/elanchezhian/Ubuntu 22.04.1 LTS amd64
/dev/fd0         1424         9    1415    1% /media/floppy0

Hit any key to continue
```

2. Displaying Logged Users

```
elanchezhian tty2      2022-10-29 09:05 (tty2)

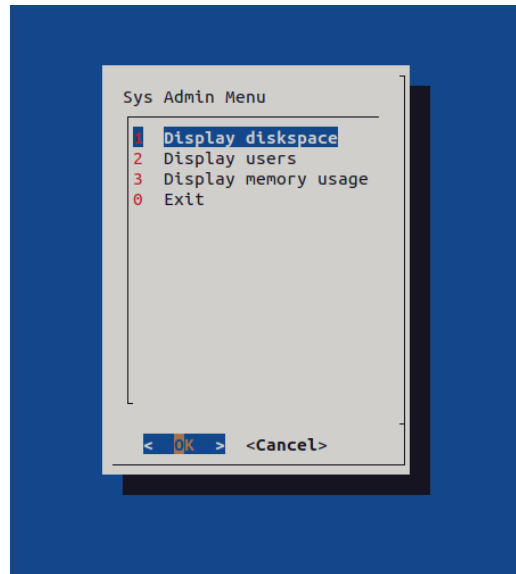
Hit any key to continue
```

3. Displaying Memory Usage

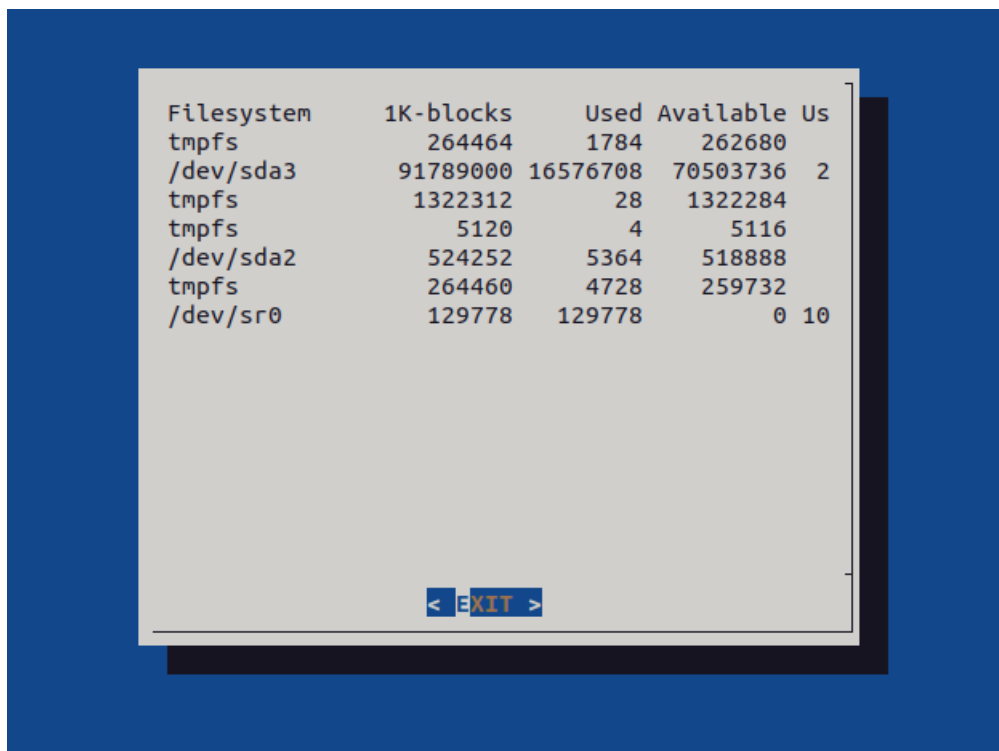
```
MemTotal:      1988240 kB
MemFree:       129276 kB
MemAvailable:  706512 kB
Buffers:       33924 kB
Cached:        642028 kB
SwapCached:    19632 kB
Active:        506536 kB
Inactive:      778100 kB
Active(anon):  111568 kB
Inactive(anon): 509876 kB
Active(file):  394968 kB
Inactive(file): 268224 kB
Unevictable:   16 kB
Mlocked:       16 kB
SwapTotal:     6191100 kB
SwapFree:      5980936 kB
Dirty:         0 kB
Writeback:     0 kB
AnonPages:     599248 kB
Mapped:        175428 kB
Shmem:         17116 kB
KReclaimable:  90708 kB
Slab:          180548 kB
SReclaimable:  90708 kB
SUnreclaim:    89840 kB
KernelStack:   10984 kB
PageTables:    16364 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:   7185220 kB
Committed_AS:  4025280 kB
VmallocTotal:  34359738367 kB
VmallocUsed:    62544 kB
VmallocChunk:   0 kB
Percpu:        112128 kB
HardwareCorrupted: 0 kB
AnonHugePages: 0 kB
ShmemHugePages: 0 kB
ShmemPmdMapped: 0 kB
FileHugePages: 0 kB
FilePmdMapped: 0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize:  2048 kB
Hugetlb:        0 kB
DirectMap4k:   280448 kB
DirectMap2M:   1816576 kB
DirectMap1G:   0 kB

Hit any key to continue
```

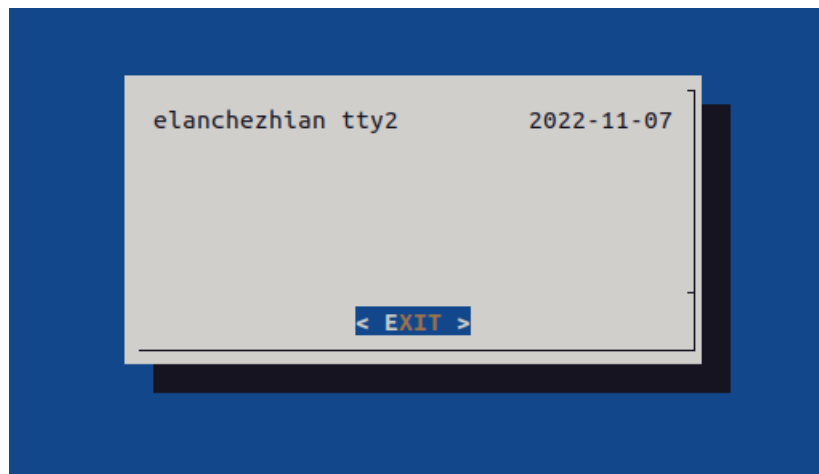
B) Text Window Widgets



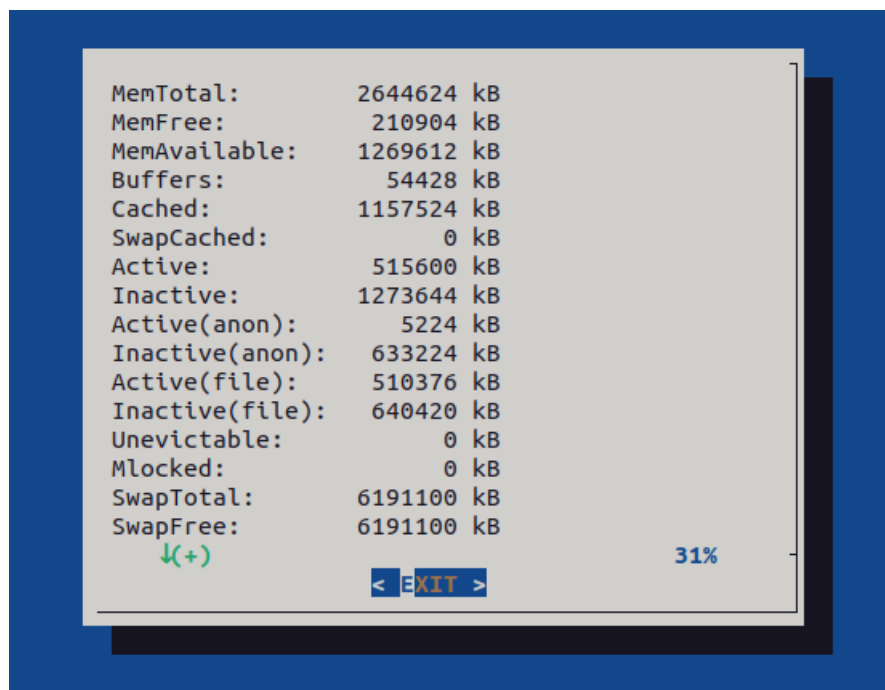
1. Displaying Disk Space:



2. Displaying Logged Users:



3. Displaying Memory Usage



SOURCE CODE AND OUTPUTS:

- To view the list of databases by using \l command:

```
elanchezhian@elanchezhian-virtual-machine:~$ sudo -i -u postgres
[sudo] password for elanchezhian:
postgres@elanchezhian-virtual-machine:~$ psql
psql (14.5 (Ubuntu 14.5-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# \l
```

List of databases						
Name	Owner	Encoding	Collate	Ctype	Access privileges	
postgres	postgres	UTF8	en_IN	en_IN		
template0	postgres	UTF8	en_IN	en_IN	=c/postgres	+
					postgres=CTc/postgres	
template1	postgres	UTF8	en_IN	en_IN	=c/postgres	+
					postgres=CTc/postgres	

(3 rows)

- Creating Database:

```
postgres=# CREATE DATABASE bank_details;
CREATE DATABASE
```

- Listing the Database and Checking Database Which Created by User:

```
postgres=# \l
```

List of databases						
Name	Owner	Encoding	Collate	Ctype	Access privileges	
bank_details	postgres	UTF8	en_IN	en_IN		
postgres	postgres	UTF8	en_IN	en_IN		
template0	postgres	UTF8	en_IN	en_IN	=c/postgres	+
					postgres=CTc/postgres	
template1	postgres	UTF8	en_IN	en_IN	=c/postgres	+
					postgres=CTc/postgres	

(4 rows)

- Changing Path to the Created Database(bank_details):

```
postgres=# \c bank_details;
You are now connected to database "bank_details" as user "postgres".
```

➤ Creating Table:

```
bank_details=# CREATE TABLE BankDetails(acc_no integer, name text, balance numeric, acc_type text);
CREATE TABLE
```

➤ Inserting Values to Table:

```
bank_details=# INSERT INTO BankDetails VALUES(50706,'Elanchezhian',1000.00,'Savings');
INSERT 0 1
bank_details=# SELECT * FROM BankDetails;
 acc_no |      name      | balance | acc_type
-----+-----+-----+-----
  50706 | Elanchezhian   | 1000.00 | Savings
(1 row)
```

➤ Inserting Multiple Values to Table:

```
bank_details=# INSERT INTO BankDetails VALUES(50707,'Hariharan',500.00,'Savings'),(50708,'Lachu',5000.00,'Current'),(50709,'Sanjai',800
0.00,'Current'),(50710,'Mahadevan',7000.00,'Savings');
INSERT 0 4
bank_details=# SELECT * FROM BankDetails;
 acc_no |      name      | balance | acc_type
-----+-----+-----+-----
  50706 | Elanchezhian   | 1000.00 | Savings
  50707 | Hariharan      |  500.00 | Savings
  50708 | Lachu          | 5000.00 | Current
  50709 | Sanjai         | 8000.00 | Current
  50710 | Mahadevan      | 7000.00 | Savings
(5 rows)
```

➤ Updating the Colum in Table:

```
bank_details=# UPDATE BankDetails SET balance=3000.00 WHERE balance=500.00;
UPDATE 1
bank_details=# SELECT * FROM BankDetails;
 acc_no |      name      | balance | acc_type
-----+-----+-----+-----
  50706 | Elanchezhian   | 1000.00 | Savings
  50708 | Lachu          | 5000.00 | Current
  50709 | Sanjai         | 8000.00 | Current
  50710 | Mahadevan      | 7000.00 | Savings
  50707 | Hariharan      | 3000.00 | Savings
(5 rows)
```


➤ Deleting the Column in Table:

```
bank_details=# DELETE FROM BankDetails WHERE acc_no=50710;
DELETE 1
bank_details=# SELECT * FROM BankDetails;
 acc_no |      name      | balance | acc_type
-----+-----+-----+-----
  50706 | Elanchezhian   | 1000.00 | Savings
  50708 | Lachu          | 5000.00 | Current
  50709 | Sanjai         | 8000.00 | Current
  50707 | Hariharan      | 3000.00 | Savings
(4 rows)
```

➤ Deleting the Table:

```
bank_details=# DROP TABLE BankDetails;
DROP TABLE
```

➤ Checking the Table if Exist or Not:

```
bank_details=# SELECT * FROM BankDetails;
ERROR:  relation "bankdetails" does not exist
LINE 1: SELECT * FROM BankDetails;
                        ^
```

➤ Deleting the Database and Listing of Databases:

```
bank_details=# \c postgres;
You are now connected to database "postgres" as user "postgres".
postgres=# DROP DATABASE bank_details;
DROP DATABASE
postgres=# \l

              List of databases
  Name      | Owner   | Encoding | Collate | Ctype  | Access privileges
-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8     | en_IN   | en_IN  |
 template0  | postgres | UTF8     | en_IN   | en_IN  | =c/postgres      +
            |          |          |          |          | postgres=CTc/postgres
 template1  | postgres | UTF8     | en_IN   | en_IN  | =c/postgres      +
            |          |          |          |          | postgres=CTc/postgres
(3 rows)
```

- Quit from Database:

```
postgres=# \q
```

- Logout from psql:

```
postgres@elanchezhian-virtual-machine:~$  
logout  
elanchezhian@elanchezhian-virtual-machine:~$
```