

2a. Consider the below scenarios and execute the given shell scripts.

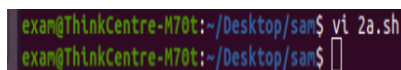
*“Ramaiah College has 10 departments (Say, CS, IS, AI, ML, Cyber Security, EC, Mechanical, EEE, DS, Civil) with UG and PG programs, and in each of the programs student details, course details are maintained in 10 different files (such as Student Details, Course details, Curriculum, Exam, Marks, Research Activity, NBA, Placement Activities, Library Details, Extra Activities....).”*

Develop a shell script for the above scenario to create 10 levels of folders for the departments and inside each level(department) of the folder, create 10 files in each department for maintaining student details. Display the entire hierarchy on the standard output by using the tree command.

Note:- All these Shell script create in your own directory. Create a directory (ex- yourUSN) and save all these scripts inside it.

```
mkdir 1MS23CY099
cd 1MS23CY099
```

then follow the screenshots.



```
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 2a.sh
exam@ThinkCentre-M70t:~/Desktop/sam$
```



```
#!/bin/sh
echo "10 levels of folders are created for the departments and 10 levels of files created for student details"
i=1
while [ $i -le 10 ]
do
    mkdir MSRITDept$i
    cd MSRITDept$i
    j=1
    while [ $j -le 10 ]
    do
        touch MSRITStudentDetails$j
        j=$((j+1))
    done
    cd ..
    i=$((i+1))
done
```

Once the shell script is typed in the vi editor, Let us execute the script.

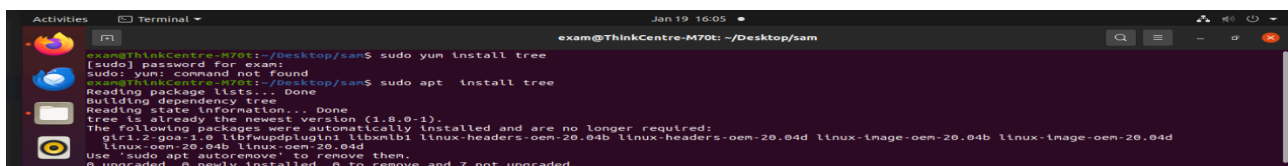
**Output:-**

```

exam@ThinkCentre-M70t:~/Desktop/sam$ vi 2a.sh
exam@ThinkCentre-M70t:~/Desktop/sam$ sh 2a.sh
10 levels of folders are created for the departments and 10 levels of files created for student details
exam@ThinkCentre-M70t:~/Desktop/sam$ ls
2a.sh 8.txt MSRITDept1 MSRITDept2 MSRITDept4 MSRITDept6 MSRITDept8 t1
2b.sh 9.txt MSRITDept10 MSRITDept3 MSRITDept5 MSRITDept7 MSRITDept9
exam@ThinkCentre-M70t:~/Desktop/sam$ cd MSRITDept1
exam@ThinkCentre-M70t:~/Desktop/sam/MSRITDept1$ ls
MSRITStudentDetails1 MSRITStudentDetails2 MSRITStudentDetails4 MSRITStudentDetails6 MSRITStudentDetails8
MSRITStudentDetails10 MSRITStudentDetails3 MSRITStudentDetails5 MSRITStudentDetails7 MSRITStudentDetails9
exam@ThinkCentre-M70t:~/Desktop/sam/MSRITDept1$ cd ..
exam@ThinkCentre-M70t:~/Desktop/sam$ cd MSRITDept2
exam@ThinkCentre-M70t:~/Desktop/sam/MSRITDept2$ ls
MSRITStudentDetails1 MSRITStudentDetails2 MSRITStudentDetails4 MSRITStudentDetails6 MSRITStudentDetails8
MSRITStudentDetails10 MSRITStudentDetails3 MSRITStudentDetails5 MSRITStudentDetails7 MSRITStudentDetails9
exam@ThinkCentre-M70t:~/Desktop/sam/MSRITDept2$ 

```

To install tree command:-



```

exam@ThinkCentre-M70t:~/Desktop/sam$ sudo yum install tree
[sudo] password for exam:
sudo: yum: command not found
exam@ThinkCentre-M70t:~/Desktop/sam$ sudo apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
tree is already the newest version (1.8.0-1).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 libfwupdplugins1 libnbt1 linux-headers-oem-20.04b linux-image-oem-20.04d linux-image-oem-20.04d
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.

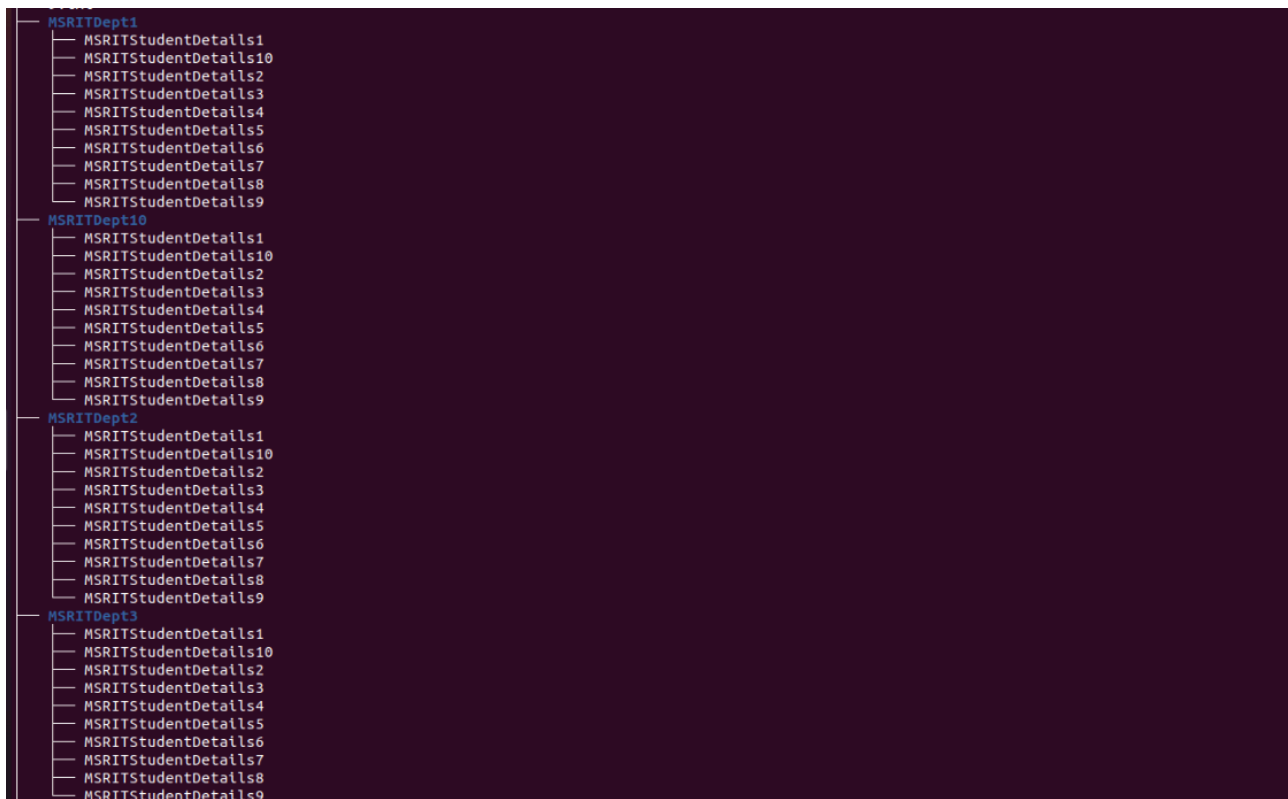
```

In this system, tree command is already installed. Now check the tree output

```

-rw-rw-r-- 1 exam exam 44 Jan 19 15:18 t1
exam@ThinkCentre-M70t:~/Desktop/sam$ tree

```



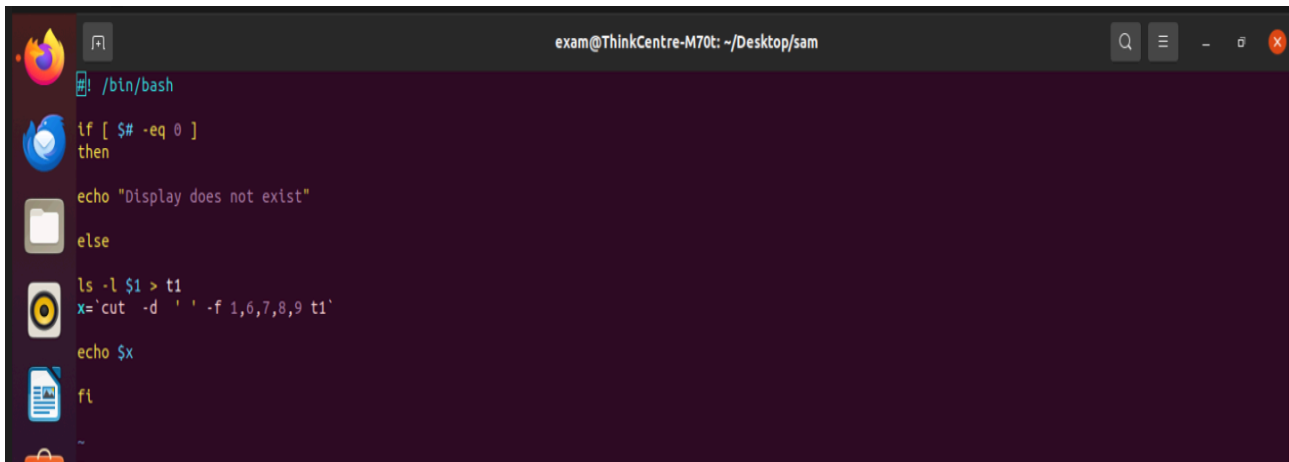
```

MSRITDept1
├── MSRITStudentDetails1
├── MSRITStudentDetails10
├── MSRITStudentDetails2
├── MSRITStudentDetails3
├── MSRITStudentDetails4
├── MSRITStudentDetails5
├── MSRITStudentDetails6
├── MSRITStudentDetails7
├── MSRITStudentDetails8
└── MSRITStudentDetails9
MSRITDept10
├── MSRITStudentDetails1
├── MSRITStudentDetails10
├── MSRITStudentDetails2
├── MSRITStudentDetails3
├── MSRITStudentDetails4
├── MSRITStudentDetails5
├── MSRITStudentDetails6
├── MSRITStudentDetails7
├── MSRITStudentDetails8
└── MSRITStudentDetails9
MSRITDept2
├── MSRITStudentDetails1
├── MSRITStudentDetails10
├── MSRITStudentDetails2
├── MSRITStudentDetails3
├── MSRITStudentDetails4
├── MSRITStudentDetails5
├── MSRITStudentDetails6
├── MSRITStudentDetails7
├── MSRITStudentDetails8
└── MSRITStudentDetails9
MSRITDept3
├── MSRITStudentDetails1
├── MSRITStudentDetails10
├── MSRITStudentDetails2
├── MSRITStudentDetails3
├── MSRITStudentDetails4
├── MSRITStudentDetails5
├── MSRITStudentDetails6
├── MSRITStudentDetails7
├── MSRITStudentDetails8
└── MSRITStudentDetails9

```

**2b.** Develop a shell script that accepts above created filename as argument & display its creation time and permissions of the file, on the standard output

vi 2b.sh



```
exam@ThinkCentre-M70t: ~/Desktop/sam
/bin/bash
if [ $# -eq 0 ]
then
echo "Display does not exist"
else
ls -l $1 > t1
x=`cut -d ' ' -f 1,6,7,8,9 t1`
echo $x
fi
```

Now create files using below screenshot

```
exam@ThinkCentre-M70t:~/Desktop/sam$ echo "hai">8.txt
exam@ThinkCentre-M70t:~/Desktop/sam$ echo "hello" >9.txt
```

**Output:-**

```
exam@ThinkCentre-M70t:~/Desktop/sam$ sh 2b.sh 8.txt
-rw-rw-r-- Jan 19 15:16 8.txt
exam@ThinkCentre-M70t:~/Desktop/sam$
```

The above output displays field 1 (permissions), field 6,7,8(Modification time) and field 9(filename) of ls -l command.

We are executing 2b.sh with an argument as 8.txt(file), to display the 1,6,7,8,9 field of the file 8.txt.

**3a.** Develop a shell script that takes a valid directory name as an argument and recursively descends all the sub-directories, finds the maximum length of any file in that hierarchy, and stores the output in a file.

```
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3a.c
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3a.c
exam@ThinkCentre-M70t:~/Desktop/sam$ mkdir sd
```

```
Activities Terminal Jan 19 15:34 exam@ThinkCentre-M70t: ~/Desktop/sam
#!/bin/bash
for i in $*
do
if [ -d $i ]
then
echo "Large filename size is"
echo `ls -lR $1 | grep "^." | tr -s ' ' | cut -d ' ' -f 5 | sort -n | tail -l`
else
echo "Not a directory"
fi
done
```

Output:-

Here the directory sd is created, 2 files 1.txt and 2.txt are created. While executing 3a.sh, directory created sd is given as a argument. The output of the 3a.sh is larger filesize in a directory will be displayed.

```
exam@ThinkCentre-M70t:~/Desktop/sam$ mkdir sd
exam@ThinkCentre-M70t:~/Desktop/sam$ cd sd
exam@ThinkCentre-M70t:~/Desktop/sam/sd$ echo "haai" >1.txt
exam@ThinkCentre-M70t:~/Desktop/sam/sd$ echo "Hellooo" >2.txt
exam@ThinkCentre-M70t:~/Desktop/sam/sd$ ls
1.txt 2.txt
exam@ThinkCentre-M70t:~/Desktop/sam/sd$ cd ..
exam@ThinkCentre-M70t:~/Desktop/sam$ sh 3a.sh sd
Large filename size is
5 8
```

3b. Create a shell script to find a file with particular name, (show separate outputs for both the conditions)

i) if that file exists then rename the existing file and create an empty file with that name.

```
/hone/exam/Desktop/sam
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3b.sh
```

```
exam@ThinkCentre-M70t: ~/Desktop/sam
#!/bin/bash
cd /home/exam/

filename="rit.txt"

if [ -e $filename ]
then
    echo "Moving the contents of rit.txt to rit.txt_old"
    mv -f $filename $filename"_old"
    touch $filename
fi
```

## Output:-

Before executing 3b.sh, create a file called rit.txt with some contents in it, then execute the shell script.

```
exam@ThinkCentre-M70t: ~/Desktop/sam
exam@ThinkCentre-M70t:~/Desktop/sam$ pwd
/home/exam/Desktop/sam
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3b.sh
exam@ThinkCentre-M70t:~/Desktop/sam$ cd /home/exam
exam@ThinkCentre-M70t:~/Desktop/sam$ echo "hai">rit.txt
exam@ThinkCentre-M70t:~/Desktop/sam$ ls
Desktop Documents Downloads eclipse-workspace Music Pictures Public rit.txt sa sample snap Templates tyu Videos
exam@ThinkCentre-M70t:~/Desktop/sam$ pwd
/home/exam/Desktop/sam
exam@ThinkCentre-M70t:~/Desktop/sam$ cd /home/exam/Desktop/sam
exam@ThinkCentre-M70t:~/Desktop/sam$ sh 3b.sh
Moving the contents of rit.txt to rit.txt_old
exam@ThinkCentre-M70t:~/Desktop/sam$ cd ../../
exam@ThinkCentre-M70t:~/Desktop/sam$ ls
Desktop Documents Downloads eclipse-workspace Music Pictures Public rit.txt rit.txt_old sa sample snap Templates tyu Videos
exam@ThinkCentre-M70t:~/Desktop/sam$ cat rit.txt_old
hai
exam@ThinkCentre-M70t:~/Desktop/sam$ cat rit.txt
exam@ThinkCentre-M70t:~/Desktop/sam$
```

ii) if that file does not exist then create a new empty file.

```
exam@ThinkCentre-M70t:~/Desktop/sam$ cat rit.txt
exam@ThinkCentre-M70t:~/Desktop/sam$ cd /home/exam/Desktop/sam
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3bb.sh
exam@ThinkCentre-M70t:~/Desktop/sam$ sh 3bb.sh
```

```
exam@ThinkCentre-M70t: ~/Desktop/sam
#!/bin/bash
cd /home/exam/
filename="rit.txt"

if [ ! -e $filename ]
then echo "File doesnt exist.. so creatinga empty file"
    touch $filename
fi
```

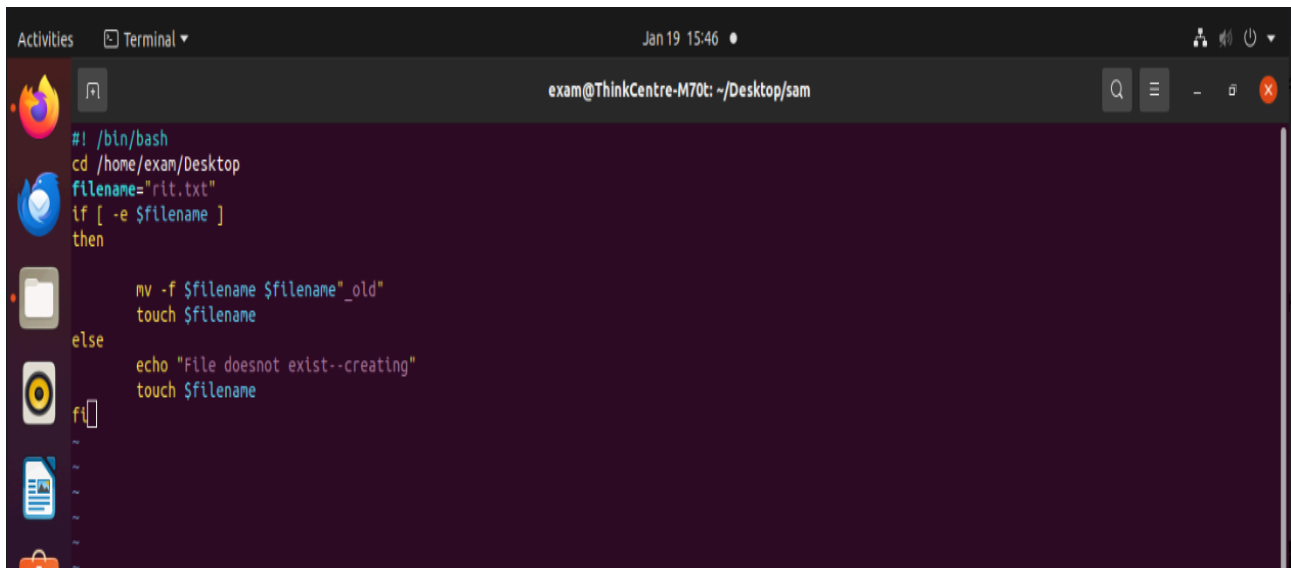
Output:-

Here if the file rit.txt doesnot exist, it will create an empty rit.txt file.

```
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3bb.sh
exam@ThinkCentre-M70t:~/Desktop/sam$ sh 3bb.sh
File doesnot exist.. so creating a empty file
exam@ThinkCentre-M70t:~/Desktop/sam$ cd ..
exam@ThinkCentre-M70t:~/Desktop$ ls
9a.c  rit.txt  sam
exam@ThinkCentre-M70t:~/Desktop$ cat rit.txt
```

iii) Both conditions together(if and else conditions together)

```
exam@ThinkCentre-M70t:~/Desktop$ cd sam/
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3bbb.sh
```



The screenshot shows a terminal window titled "exam@ThinkCentre-M70t: ~/Desktop/sam". The script content is as follows:

```
#!/bin/bash
cd /home/exam/Desktop
filename="rit.txt"
if [ -e $filename ]
then
    mv -f $filename $filename_old
    touch $filename
else
    echo "File doesnot exist--creating"
    touch $filename
fi
```

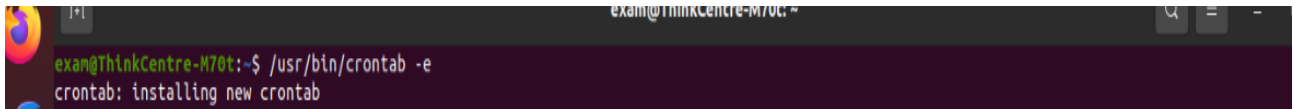
Output:

```
exam@ThinkCentre-M70t:~/Desktop/sam$ vi 3bbb.sh
exam@ThinkCentre-M70t:~/Desktop/sam$ sh 3bbb.sh
exam@ThinkCentre-M70t:~/Desktop/sam$ ls
2a.sh  3a.c  3bbb.sh  3b.sh  9.txt      MSRITDept10  MSRITDept3  MSRITDept5  MSRITDept7  MSRITDept9  t1
2b.sh  3a.sh  3bb.sh  8.txt  MSRITDept1 MSRITDept2   MSRITDept4  MSRITDept6  MSRITDept8  sd
exam@ThinkCentre-M70t:~/Desktop/sam$ cd ..
exam@ThinkCentre-M70t:~/Desktop$ ls
9a.c  rit.txt  rit.txt_old  sam
exam@ThinkCentre-M70t:~/Desktop$ cat rit.txt_old
exam@ThinkCentre-M70t:~/Desktop$ cat rit.txt
exam@ThinkCentre-M70t:~/Desktop$
```

**3.c.** Set up a cron job for the above developed scripts, , that will be execute after every 30 minutes

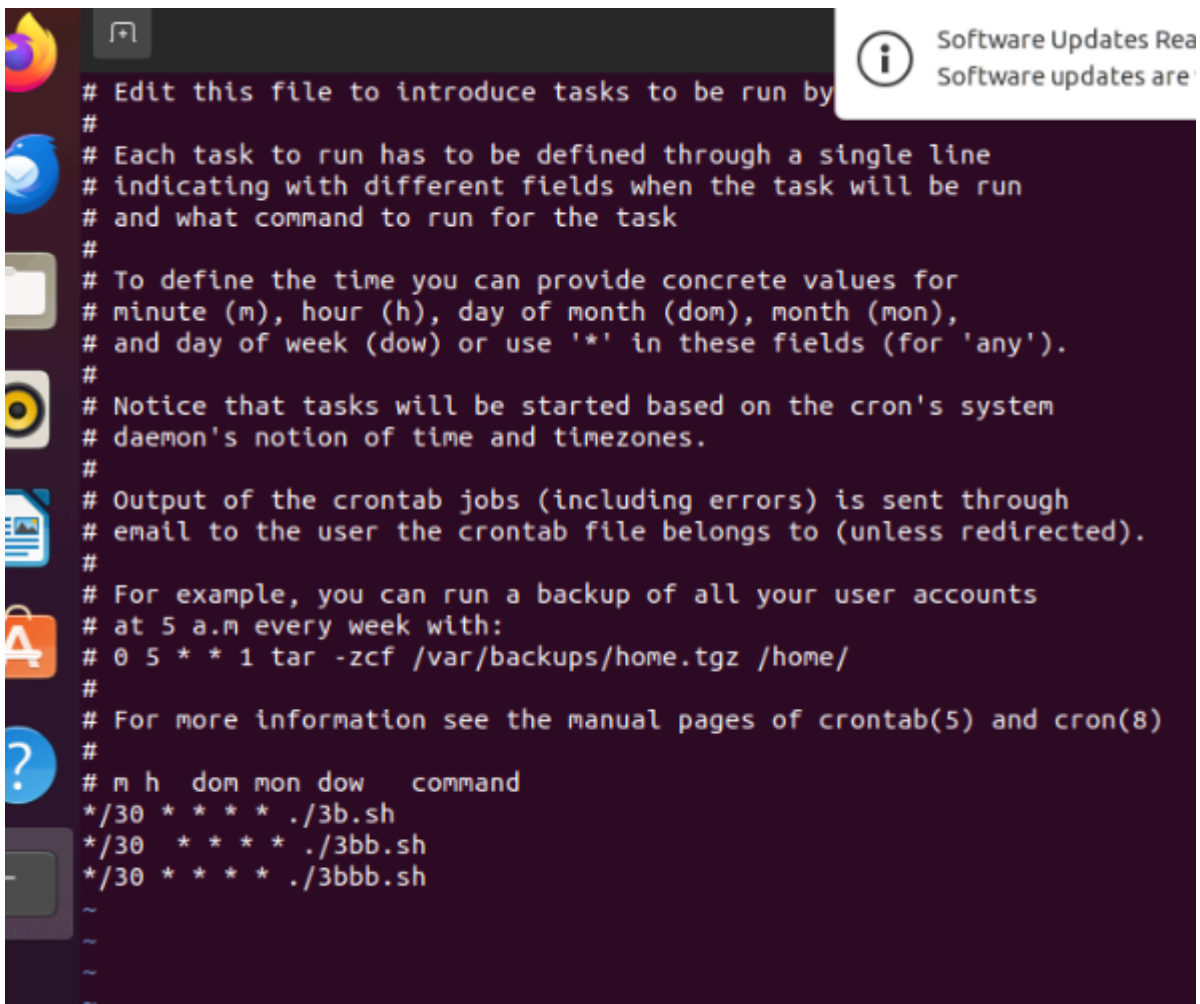
The Cron software utility is a time-based job scheduler in Unix-like operating systems. Cron allows Linux and Unix users to run commands or scripts at a given time and date.

### Setting a cronjob



```
exam@ThinkCentre-M70t:~$ /usr/bin/crontab -e
crontab: installing new crontab
```

Once we type the above command `/usr/bin/crontab -e`, one more window will open, there schedule your jobs. Which is shown in below screenshot (in last 3 lines). We are trying to schedule 3 jobs (executing `3b.sh`, `3bb.sh` and `3bbb.sh` script every 30 minutes)



```
# Edit this file to introduce tasks to be run by
#
# 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
*/30 * * * * ./3b.sh
*/30 * * * * ./3bb.sh
*/30 * * * * ./3bbb.sh
~
~
~
~
```

### Listing out cronjobs:-

```
exam@ThinkCentre-M70t:~$ /usr/bin/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
*/30 * * * * ./3b.sh
*/30 * * * * ./3bb.sh
*/30 * * * * ./3bbb.sh
exam@ThinkCentre-M70t:~$
```

You can observe last 3 lines 3 jobs are set for 3b.sh, 3bb.sh and 3bbb.sh

### 3d. Illustrating shell variables in a shell script

```
exam@ThinkCentre-M70t:~$ vi 3d.sh
```

```
#!/bin/bash
echo " '$*' output is $*"
echo " '$#' output is $# "
echo " '$1 and $2' output is $1 and $2"
echo " '$0' output is $0"
echo " '$?' output is $?"
echo " '$$' output is $$"
echo " '$!' output is $!"
echo " '$0' your current program name is $0
```

Output:-

The script 3d.sh is executed with some arguments.

```
exam@ThinkCentre-M70t:~$ vi 3d.sh
exam@ThinkCentre-M70t:~$ sh 3d.sh hai students welcome to unix lab
'hai students welcome to unix lab' output is hai students welcome to unix lab
'6' output is 6
'hai and students' output is hai and students
'hai students welcome to unix lab' output is hai students welcome to unix lab
'0' output is 0
'4418' output is 4418
'' output is 
'3d.sh' your current program name is 3d.sh
exam@ThinkCentre-M70t:~$
```



