Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
Institute of Computer Technology

Name: Tushar Panchal

En.No: 21162101014

Sub: ITIM (IT Infrastructure & Management)

Branch: CBA

Batch:61

----------PRACTICAL 11----------

- **Task 1 : Create a user of your name and schedule a job to create a file which have the current date and time stored in it. The job should be executed on a specific date and time (you can specify the date and time as per your convenience):**

```
[student@workstation ~]$ su root
Password:

(process:2804): dconf-WARNING **: 11:27:10.204: fa
ection is closed
[root@workstation student]# useradd tushar
[root@workstation student]# crontab -u tushar -e
no crontab for tushar - using an empty one
crontab: installing new crontab
[root@workstation student]#
```

Commands :

```
su root
useradd tushar
crontab -u tushar -e
```

Add this in httpd.conf file as shown below :

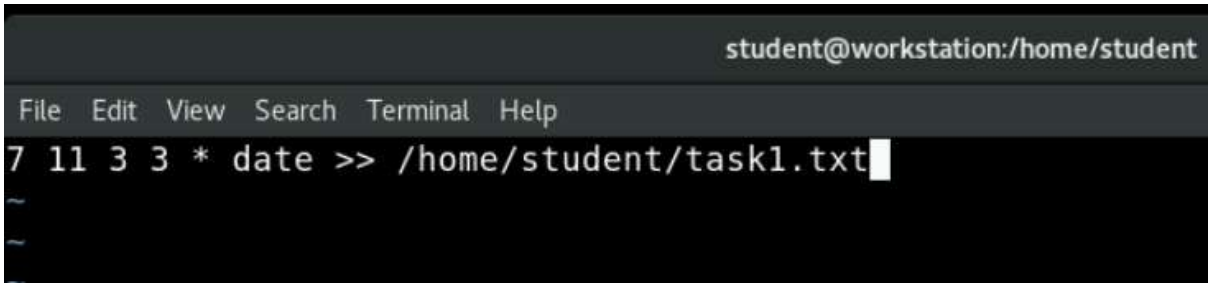
```
7 11 3 3 * date >> /home/student/task1.txt
```

Just to clarify, the cron schedule i've written is in the format:

minute hour day month day_of_week command

Here's a breakdown of cron job:

- **Minute:** 7
- **Hour:** 11
- **Day:** 3
- **Month:** 3 (March)
- **Day of Week:** * (any day of the week)




```
[root@workstation student]# date>test1.txt
[root@workstation student]# date>test1.txt | at 11:40
warning: commands will be executed using /bin/sh
job 1 at Wed Mar  6 11:40:00 2024
[root@workstation student]# atq
1       Wed Mar  6 11:40:00 2024 a root
[root@workstation student]# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  test1.txt  Videos
[root@workstation student]# cat test1.txt
Tue Mar  5 11:44:29 EST 2024
[root@workstation student]# crontab -l
no crontab for root
[root@workstation student]#
```

Commands :

`date>task1.txt | at 11:40`

`atq`

`cat task1.txt`

Here's a breakdown of commands :

1. **date > task1.txt**: This command writes the current date to the file "task1.txt." The ">" operator overwrites the file if it already exists.
2. **at 11:40**: This command schedules a job to be executed at 11:40. The job will involve the execution of the previous "date > task1.txt" command.
3. **atq**: This command displays the current **at** queue, showing scheduled jobs.
4. **cat task1.txt**: This command displays the content of the "task1.txt" file, showing the date that was written earlier.

» **Task 2: Demonstrate how to remove a Scheduled job :**

```
[root@workstation student]# date>test1.txt | at 11:45
warning: commands will be executed using /bin/sh
job 2 at Wed Mar  6 11:45:00 2024
[root@workstation student]# atq
1       Wed Mar  6 11:40:00 2024 a root
2       Wed Mar  6 11:45:00 2024 a root
[root@workstation student]#
[root@workstation student]# atrm 1
[root@workstation student]# atq
2       Wed Mar  6 11:45:00 2024 a root
[root@workstation student]#
```

Commands :

```
date>task1.txt | at 11:45
atq
atrm 1
atq
```

Here's a breakdown of commands :

1. **date > task1.txt | at 11:45**: This schedules a new **at** job to execute the specified command at 11:45.

2. **atq**: This displays the current **at** queue, showing scheduled jobs.
3. **atrm 1**: This removes the job with ID 1 from the **at** queue. Adjust the job ID accordingly based on the output of **atq**.
4. **atq**: This again displays the **at** queue after removing the specified job.

» **Task 3: Do the configuration in such a way that only user1 and user2 is able to schedule the job, while user3 and user4 should not be able to schedule any job.**

```
[root@workstation student]# vim /etc/cron.deny
[root@workstation student]# cat /etc/cron.deny
user3
user4
[root@workstation student]#
```

```
File  Edit  View  Search  Terminal  Help
user3
user4
~
```

For not allowing user3 and user4 we edit cron.deny file in system and add user3 and user4 in that file in one at line and save it so than user3 and user4 will not be able to schedule any jobs

Commands :

```
vim /etc/cron.deny
```

» Task 5: You need to create a script that will store the details about the kernel messages related to drivers in a specific file. And this task should be executed every day at 3 pm.

```
#!/bin/bash
dmesg | grep -i "driver" >> /home/student/kernel_logs.txt
```

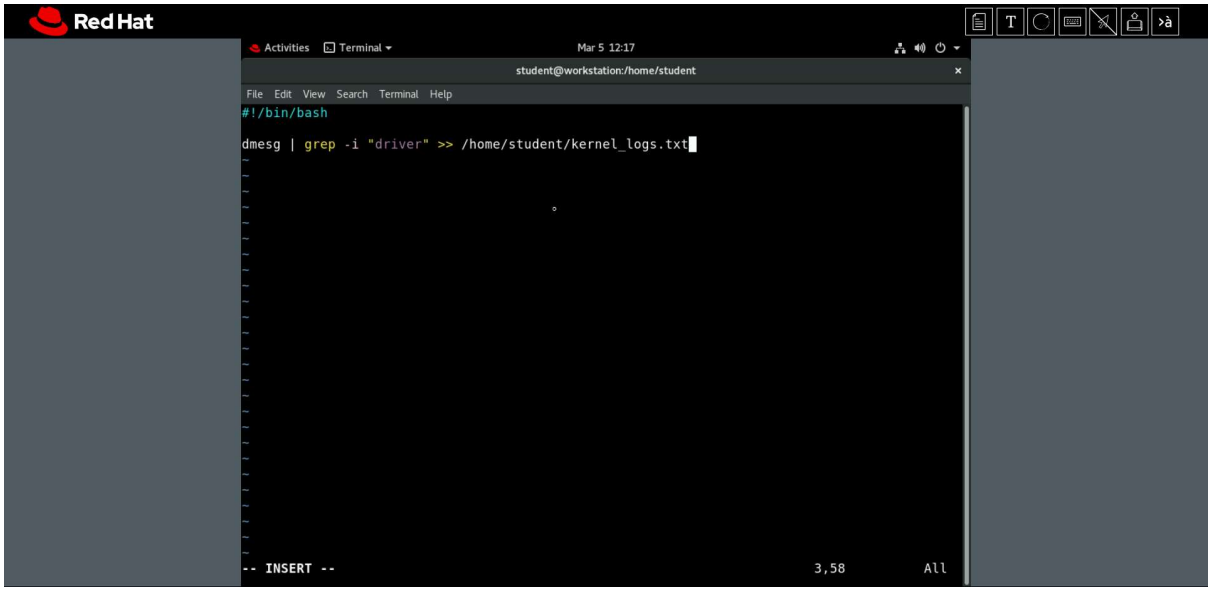
This code will store logs into kernal_logs.txt files Now to make the script executable we'll give permission to that script with

Command :

```
chmod +x task1.sh
```

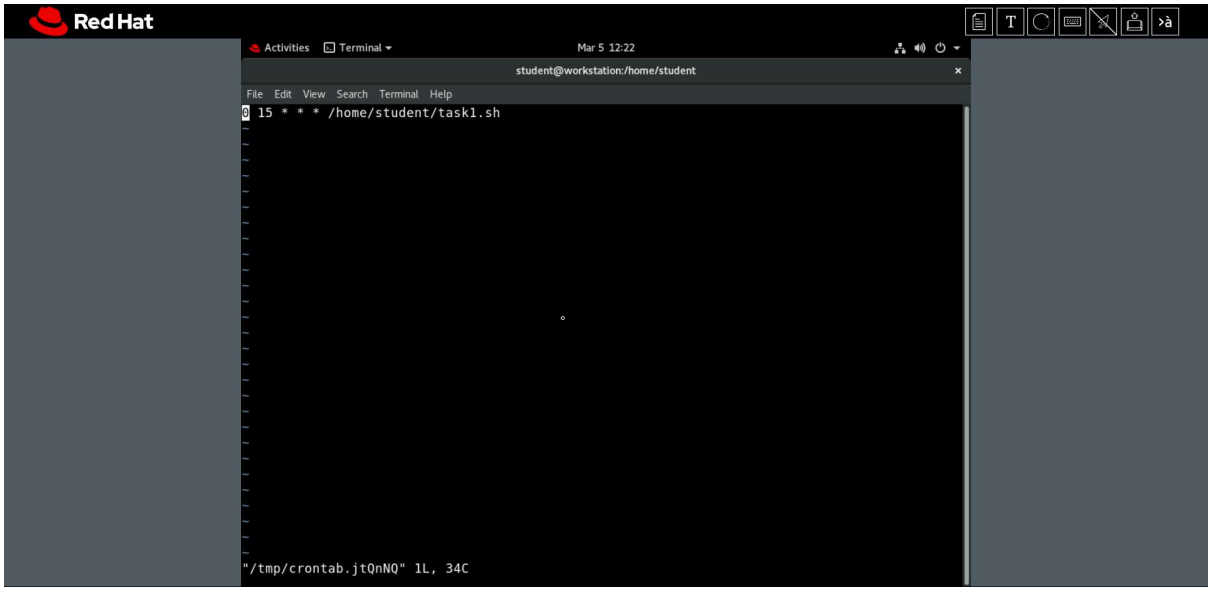
```
[root@workstation student]# vim task1.sh
[root@workstation student]# cat task1.sh
#!/bin/bash

dmesg | grep -i "driver" >> /home/student/kernel_logs.txt
[root@workstation student]#  chmod +x task1.sh
[root@workstation student]#
```



Here 0 means 0 minute an the 15th hour means 3 PM

```
[root@workstation student]# crontab -e
crontab: no changes made to crontab
[root@workstation student]# crontab -l
0 15 * * * /home/student/task1.sh
[root@workstation student]#
```



Commands :

```
crontab -e
crontab -l
```

Edited the user's crontab using the crontab -e command. Checked the user's crontab to ensure the cron job entry was successfully added using the crontab -l command.

- **Task 6: Schedule the backup script to run at 1 pm every Saturday.**

→First I created backup directory.

→The script backup_script.sh contains the command **`rsync -av /home/student /home/student/backup_dir`** to perform the backup operation. This command uses rsync to synchronize the contents of the /home/student directory to the backup_dir directory.

→Made the backup_script.sh script executable using

Command :

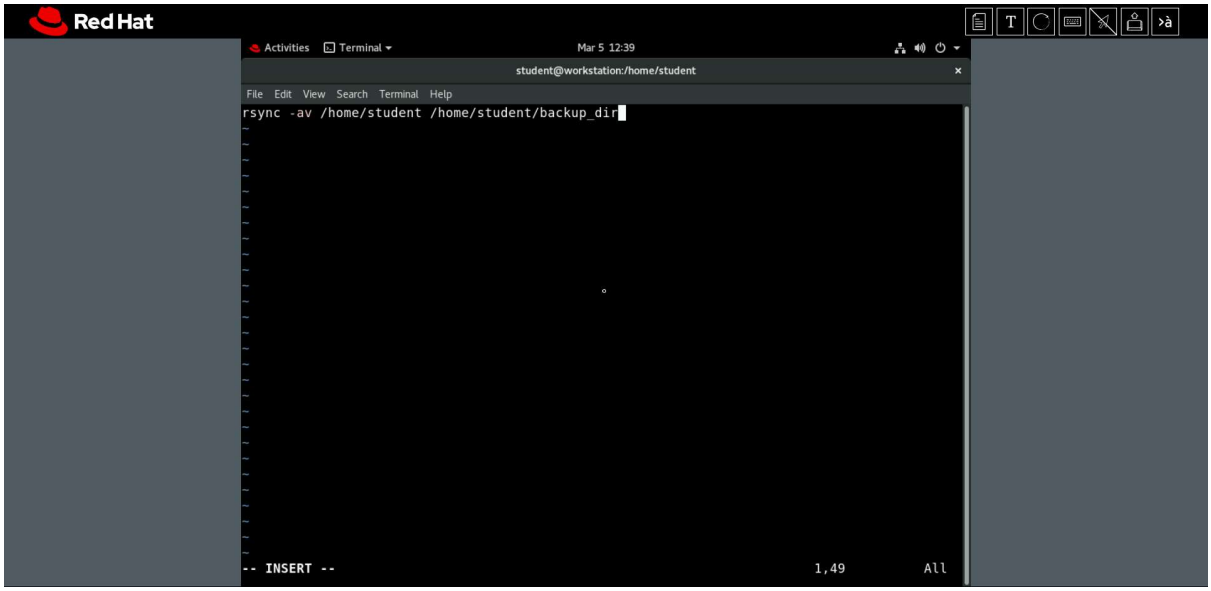
```
chmod +x backup_script.sh
```

→Added a new cron job to the crontab to execute the backup_script.sh script at 1 pm every Saturday.

→Checked the crontab entries using the command crontab -l to ensure that both cron jobs (kernel_logs.txt and backup_script.sh) are correctly scheduled.

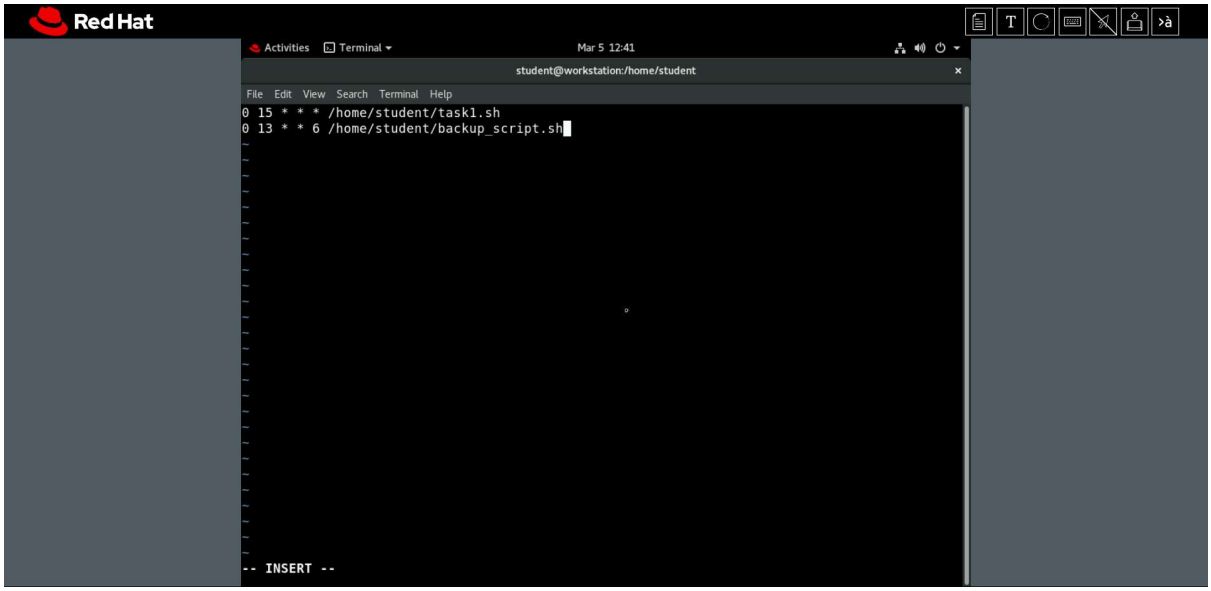
```
[root@workstation student]# mkdir backup_dir
[root@workstation student]# ls
backup_dir  Documents  kernel_logs.txt  Pictures  task1.sh   test1.txt
Desktop     Downloads  Music            Public    Templates  Videos
[root@workstation student]# vim backup_script.sh
[root@workstation student]# cat backup_script.sh
rsync -av /home/student /home/student/backup_dir
[root@workstation student]# chmod +x backup_script.sh
[root@workstation student]# crontab -e
crontab: installing new crontab
[root@workstation student]# crontab -l
0 15 * * * /home/student/task1.sh
0 13 * * 6 /home/student/backup_script.sh
[root@workstation student]#
```

Add this In backup_script.sh :



Command :

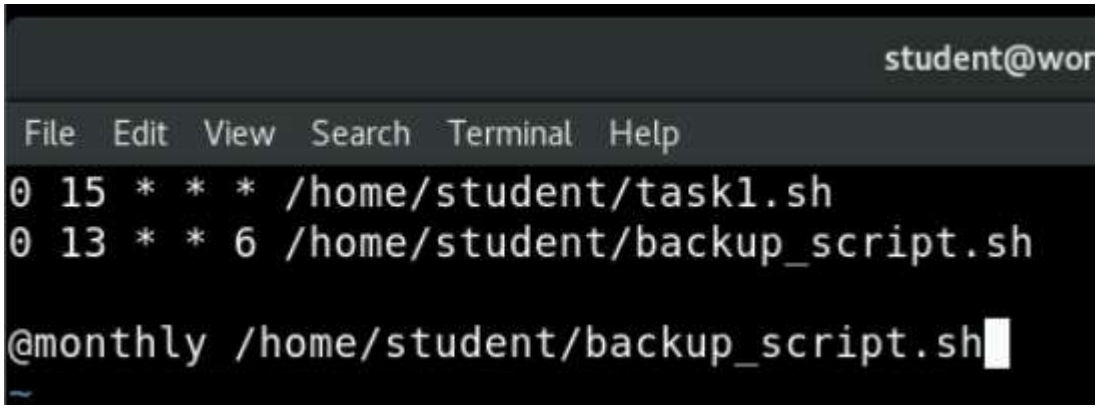
```
crontab -e
```



» Task 7: Schedule a job beginning of every month by using @monthly keyword.

→Added a new cron job to the crontab using the @monthly keyword to execute the backup_script.sh script at the beginning of every month.

→ Checked the crontab entries using the command crontab -l to ensure that the new monthly cron job is correctly scheduled along with the existing cron jobs for the backup and drivers log scripts.



Commands :

```
crontab -e
crontab -l
```

```
[root@workstation student]# crontab -e
crontab: installing new crontab
[root@workstation student]# crontab -l
0 15 * * * /home/student/task1.sh
0 13 * * 6 /home/student/backup_script.sh

@monthly /home/student/backup_script.sh
[root@workstation student]#
```

» Task 8: Script to check the disk space every 10 minutes.

→Created a new script called check_space.sh, specifically designed to keep track of disk space usage by employing the **df -h** command.

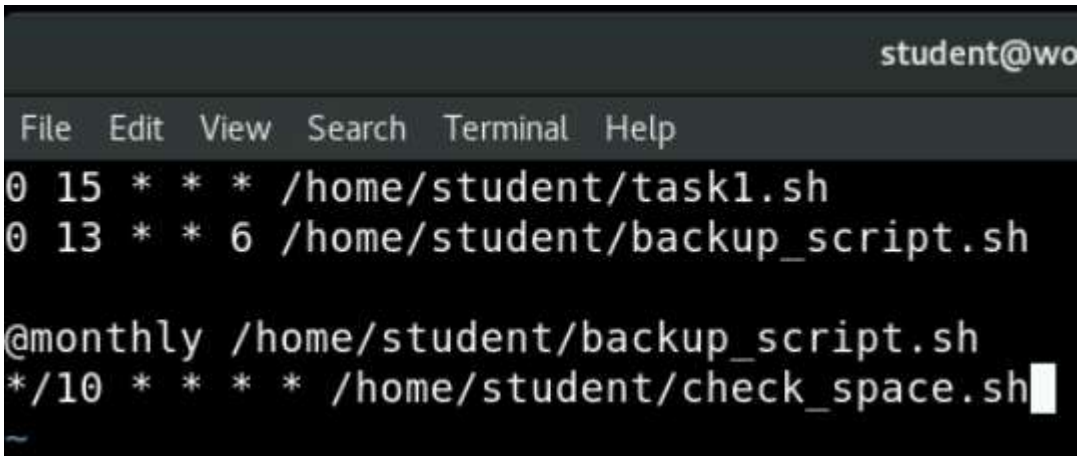
→Made the check_space.sh script executable using the command **chmod +x check_space.sh.**

→Added a new cron job to the crontab to execute the check_space.sh script every 10 minutes. This ensures that disk space is checked regularly.

→Checked the crontab entries using the command **crontab -l** to ensure that the new cron job for check_space.sh is correctly scheduled along with the existing cron jobs.

Command :

```
crontab -e
```



```
[root@workstation student]# vim check_space.sh
[root@workstation student]# cat check_space.sh
df -h
[root@workstation student]# chmod +x check_space.sh
[root@workstation student]# crontab -e
crontab: installing new crontab
[root@workstation student]# crontab -l
0 15 * * * /home/student/task1.sh
0 13 * * 6 /home/student/backup_script.sh

@monthly /home/student/backup_script.sh
*/10 * * * * /home/student/check_space.sh
[root@workstation student]#
```

» Task 9: Schedule a job that will be executed after every reboot.

Added a cron job using @reboot to execute backup_script.sh automatically after every system reboot.

Commands :

`crontab -e` **`crontab -l`**

```
student@v
File  Edit  View  Search  Terminal  Help
0 15 * * * /home/student/task1.sh
0 13 * * 6 /home/student/backup_script.sh
@monthly /home/student/backup_script.sh
*/10 * * * * /home/student/check_space.sh

@reboot /home/student/backup_script.sh
~
```

```
[root@workstation student]# crontab -e
crontab: installing new crontab
[root@workstation student]# crontab -l
0 15 * * * /home/student/task1.sh
0 13 * * 6 /home/student/backup_script.sh
@monthly /home/student/backup_script.sh
*/10 * * * * /home/student/check_space.sh

@reboot /home/student/backup_script.sh
[root@workstation student]#
```

» Task 10: Schedule a job that will display a hello message with the current time after every 5 minutes.

Added a cron job to display a hello message with the current time every 5 minutes by directly echoing the message into the cron configuration. This ensures that the specified message will be shown at regular intervals without the need for a separate script.

Command :

```
crontab -e
```

```
student@workstation:/home/stu
File  Edit  View  Search  Terminal  Help
0 15 * * * /home/student/task1.sh
0 13 * * 6 /home/student/backup_script.sh
@monthly /home/student/backup_script.sh
*/10 * * * * /home/student/check_space.sh
@reboot /home/student/backup_script.sh

*/5 * * * * echo "Hello! the current time is $(date) "
~
```

```
[root@workstation student]# crontab -e
crontab: installing new crontab
[root@workstation student]# crontab -l
0 15 * * * /home/student/task1.sh
0 13 * * 6 /home/student/backup_script.sh
@monthly /home/student/backup_script.sh
*/10 * * * * /home/student/check_space.sh
@reboot /home/student/backup_script.sh

*/5 * * * * echo "Hello! the current time is $(date) "
[root@workstation student]#
```