ITIM

PRACTICAL - 11

Name : Yagna Patel
Enrollment No. : 211621020
Batch : 61(CBA)

Tasks :

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)

Now we'll create a user named yagna with useradd yagna

Now for a set date and time, we'll enter into the crontab job schedule file with

Command: crontab -u yagna -e

Here -u yagna This means that you're editing the crontab file for the

specified user.

And in the crontab file, we'll just put the bellowed line

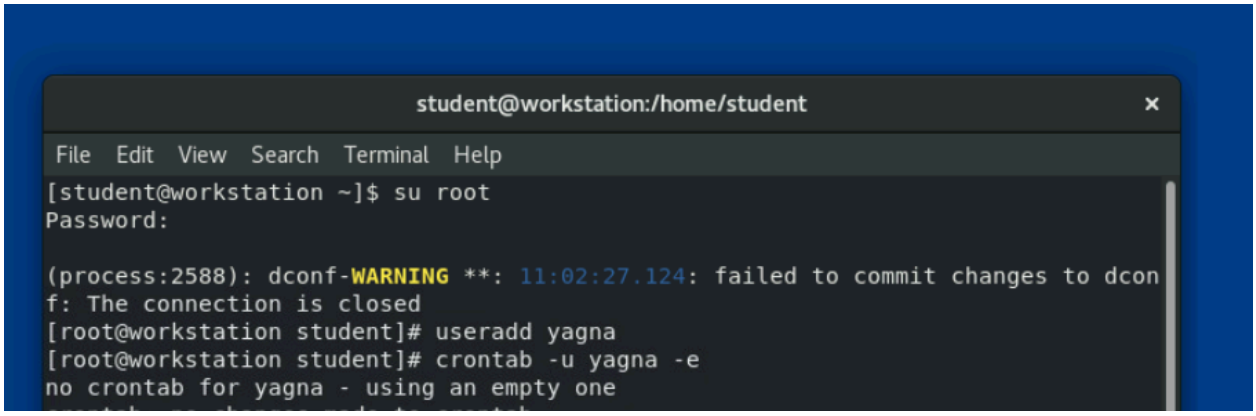
7 11 3 3 * date >> /home/student/task1.txt

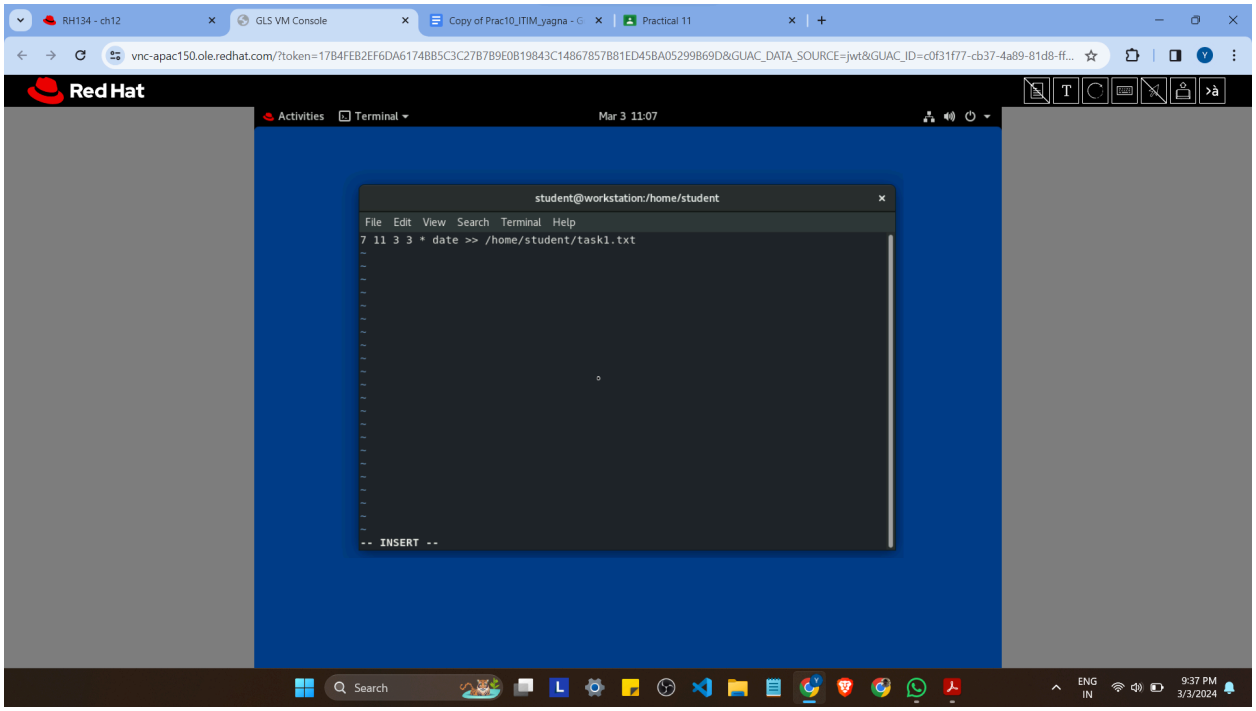
In this command

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

Date is a function that return current date and time

/home/student/task1.txt will save to this path





Task 2: Demonstrate how to remove a Scheduled job

If we have scheduled a job via crontab, we can remove it directly through the crontab file.

But if our job was created via at command than we've to remove via atrm command

Like let suppose

I've some job in queue i can see it with command atq

```
bash: 11:20: command not found...
[root@workstation student]# date>test1.txt | at 11:20
warning: commands will be executed using /bin/sh
job 1 at Sun Mar  3 11:20:00 2024
[root@workstation student]# atq
1       Sun Mar  3 11:20:00 2024 a root
[root@workstation student]# date>test1.txt | at 11:22
warning: commands will be executed using /bin/sh
job 2 at Sun Mar  3 11:22:00 2024
[root@workstation student]# atq
1       Sun Mar  3 11:20:00 2024 a root
2       Sun Mar  3 11:22:00 2024 a root
[root@workstation student]#
```

I can simply remove job with it's job id with atrm command

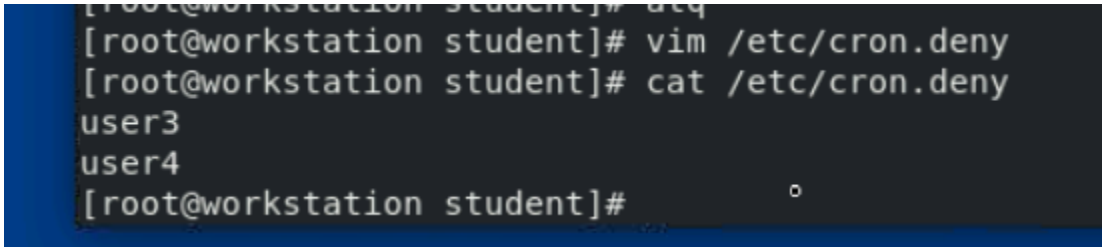
```
[root@workstation student]# atq
3       Mon Mar  4 11:22:00 2024 a root
[root@workstation student]# atrm 3
[root@workstation student]# atrm 3
Cannot find jobid 3
[root@workstation student]# atq
[root@workstation student]#
```

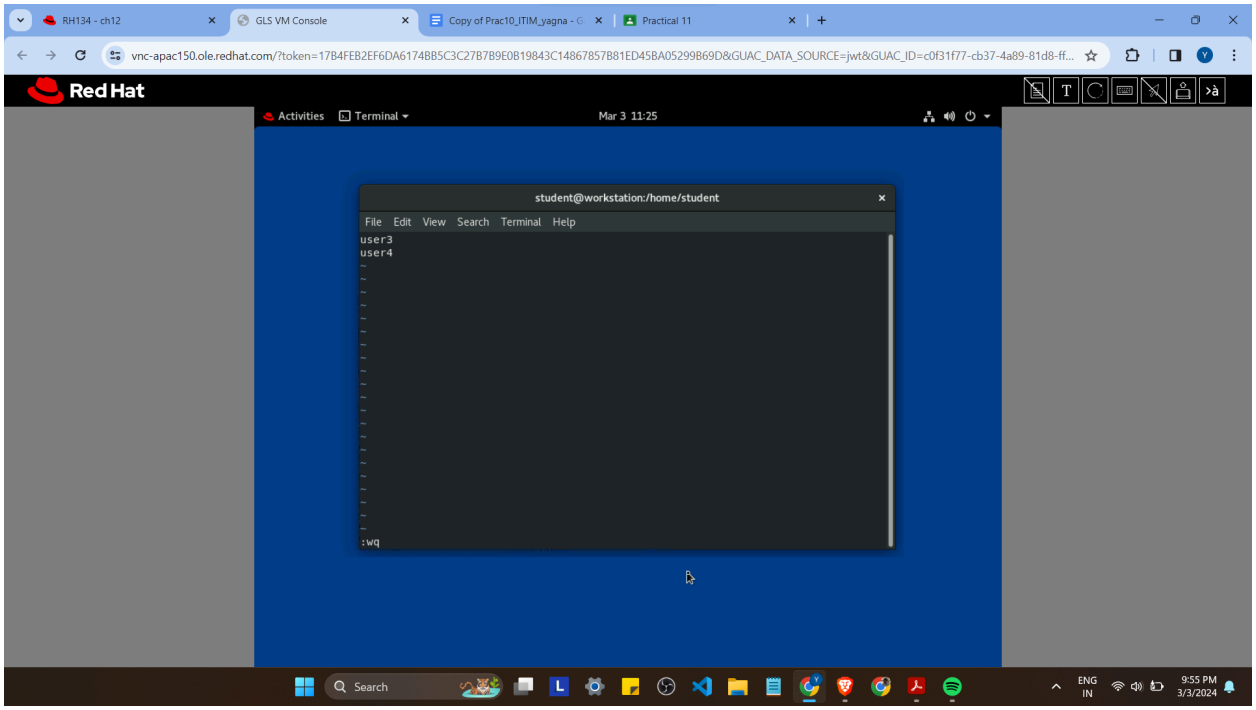
Task 3: Do the configuration in such a way that only user1 and user2 is able to schedule the job, while user3 and user4 should not be able to schedule any job.

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.





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

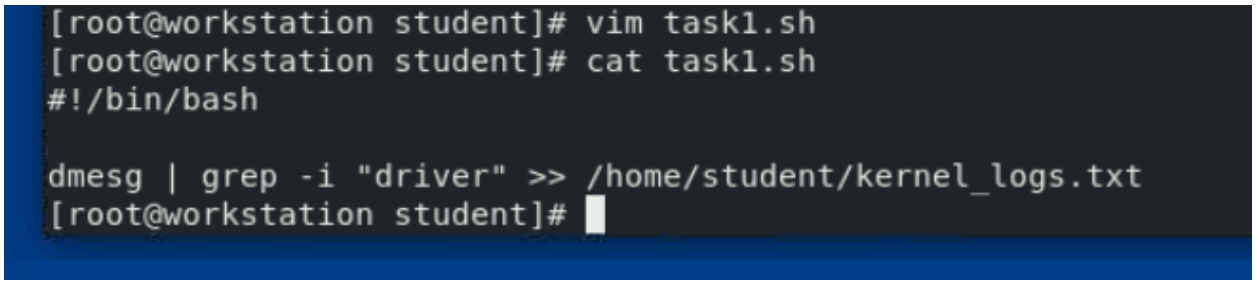
#!/bin/bash

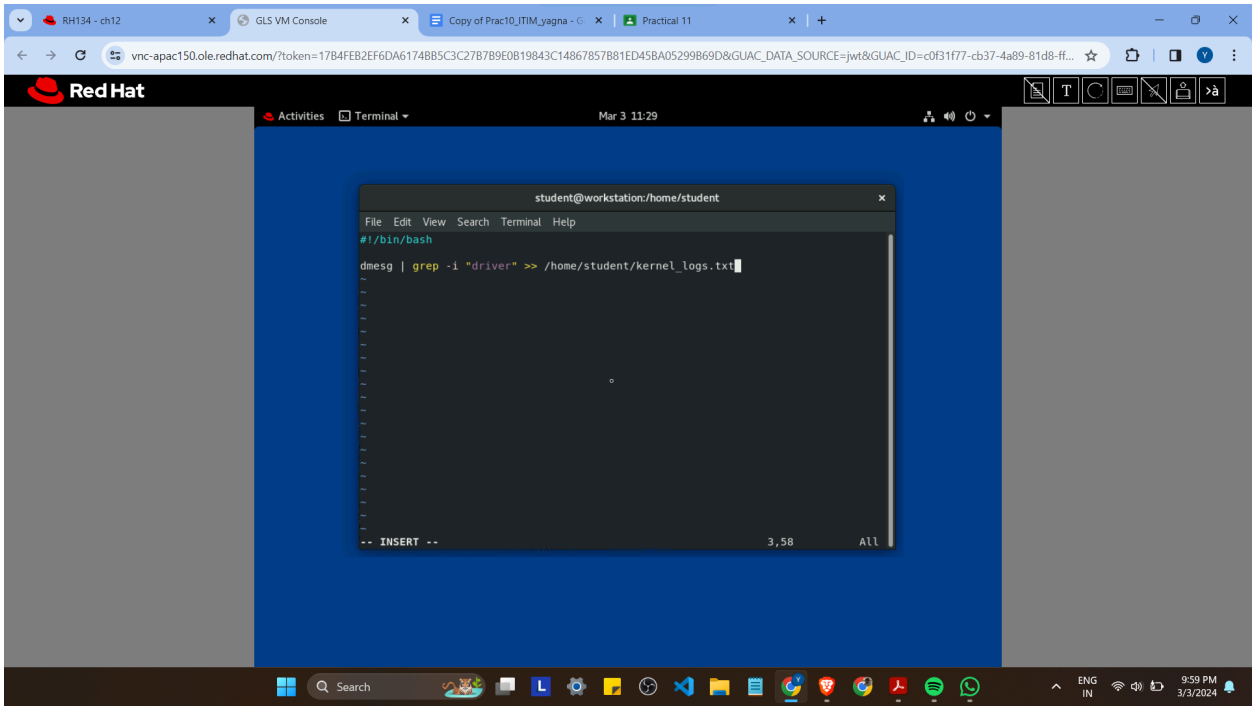
dmesg | grep -i "driver" >> /home/student/kernel_logs.txt

This code will store logs into kernal_logs.txt files

Now to make the script executable we'll give permission to that script with

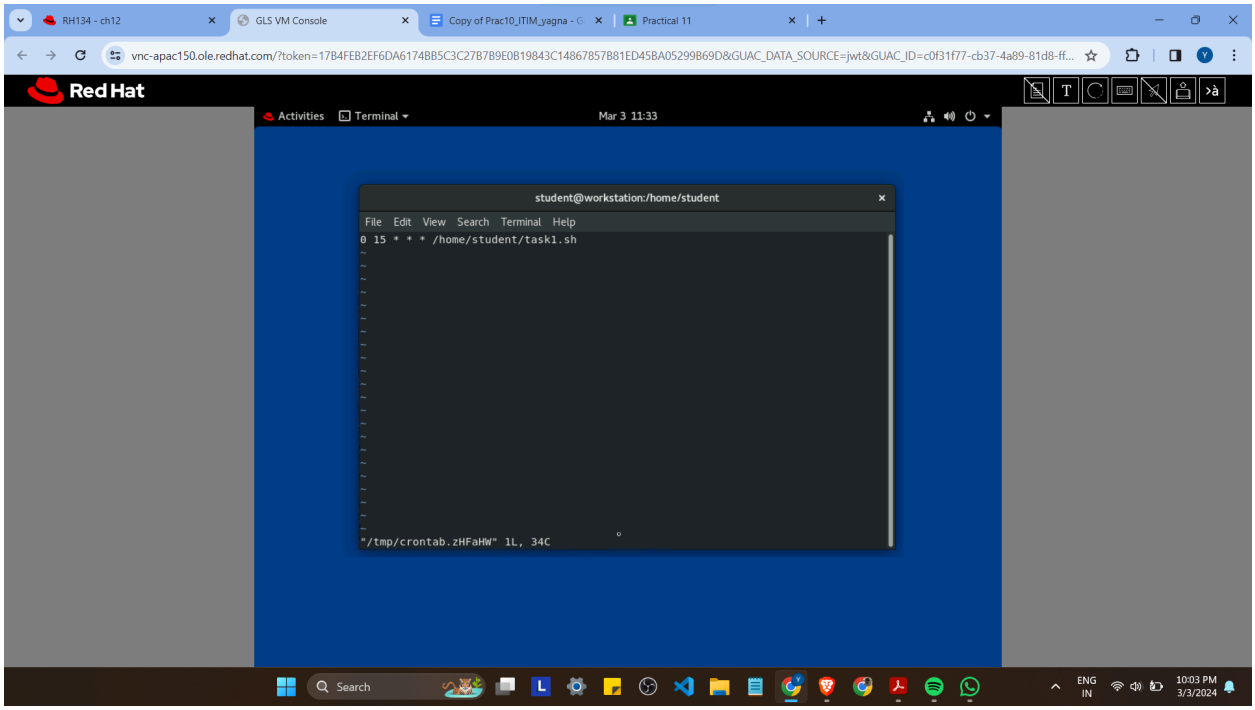
command: chmod +x task1.sh





```
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



should be executed every day at 3 pm.

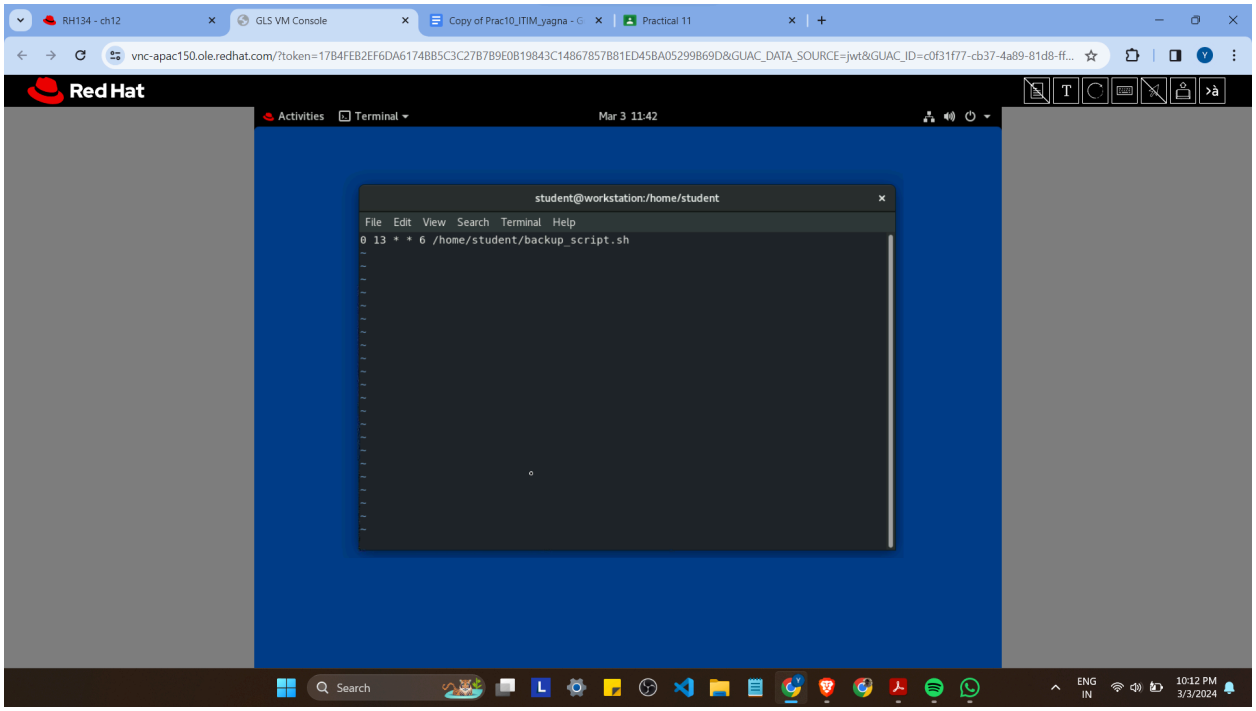
```
[root@workstation student]# cat kernel_logs.txt
[    0.023010] Performance Events: Skylake events, full-width counters, Intel PM
U driver.
[    0.083006] acpiphp: ACPI Hot Plug PCI Controller Driver version: 0.5
[    0.170870] usbcore: registered new interface driver usbfs
[    0.171029] usbcore: registered new interface driver hub
[    0.171536] usbcore: registered new device driver usb
[    1.401819] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 2
46)
[    1.404769] shpchp: Standard Hot Plug PCI Controller Driver version: 0.4
[    1.482135] Serial: 8250/16550 driver, 4 ports, IRQ sharing enabled
[    1.507405] Non-volatile memory driver v1.3
[    1.511618] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    1.512239] ehci-pci: EHCI PCI platform driver
[    1.512691] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
[    1.513269] ohci-pci: OHCI PCI platform driver
[    1.513707] uhci_hcd: USB Universal Host Controller Interface driver
[    1.538629] usbcore: registered new interface driver usbserial_generic
[    1.549348] hidraw: raw HID events driver (C) Jiri Kosina
[    1.549961] usbcore: registered new interface driver usbhid
[    1.550535] usbhid: USB HID core driver
[    1.625155] Loaded X.509 cert 'Red Hat Enterprise Linux Driver Update Program
```

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

make backup sccript and make that script executable and make entry into crontab like :

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

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



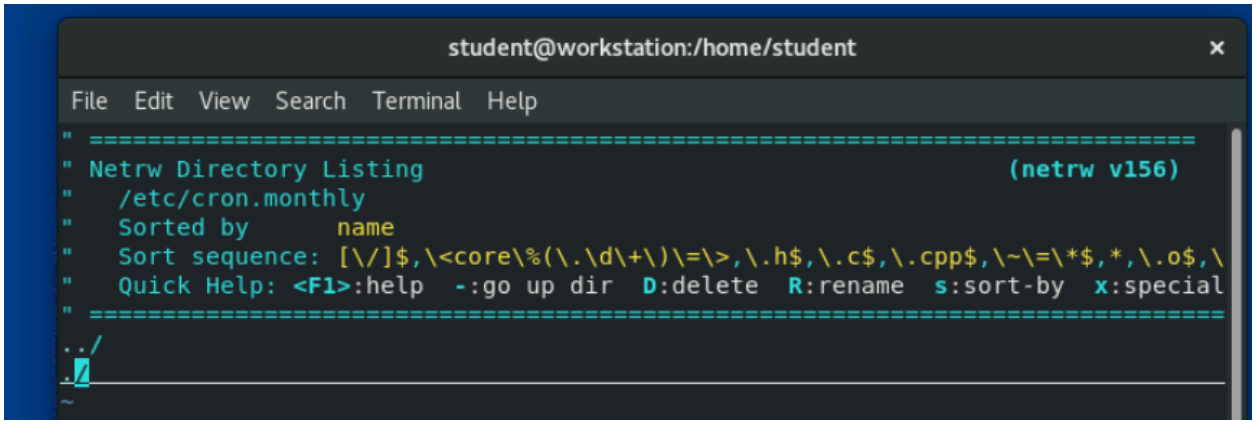

```
[root@workstation student]# cat kernel_logs.txt
[    0.023010] Performance Events: Skylake events, full-width counters, Intel PM
U driver.
[    0.083006] acpiphp: ACPI Hot Plug PCI Controller Driver version: 0.5
[    0.170870] usbcore: registered new interface driver usbfs
[    0.171029] usbcore: registered new interface driver hub
[    0.171536] usbcore: registered new device driver usb
[    1.401819] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 2
46)
[    1.404769] shpchp: Standard Hot Plug PCI Controller Driver version: 0.4
[    1.482135] Serial: 8250/16550 driver, 4 ports, IRQ sharing enabled
[    1.507405] Non-volatile memory driver v1.3
[    1.511618] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    1.512239] ehci-pci: EHCI PCI platform driver
[    1.512691] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
[    1.513269] ohci-pci: OHCI PCI platform driver
[    1.513707] uhci_hcd: USB Universal Host Controller Interface driver
[    1.538629] usbcore: registered new interface driver usbserial_generic
[    1.549348] hidraw: raw HID events driver (C) Jiri Kosina
[    1.549961] usbcore: registered new interface driver usbhid
[    1.550535] usbhid: USB HID core driver
[    1.625155] Loaded X.509 cert 'Red Hat Enterprise Linux Driver Update Program
```

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

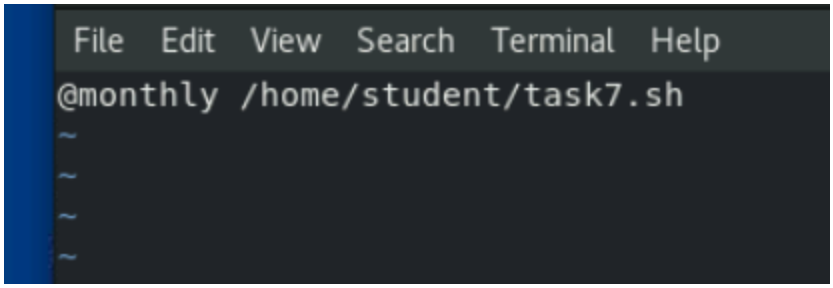
make entry into the crontab.monthly file with which is

/etc/cron.monthly

```
[root@workstation student]# vim /etc/cron.monthly
[root@workstation student]#
```



Or we can just make entry into crontab with just @monthly keyword

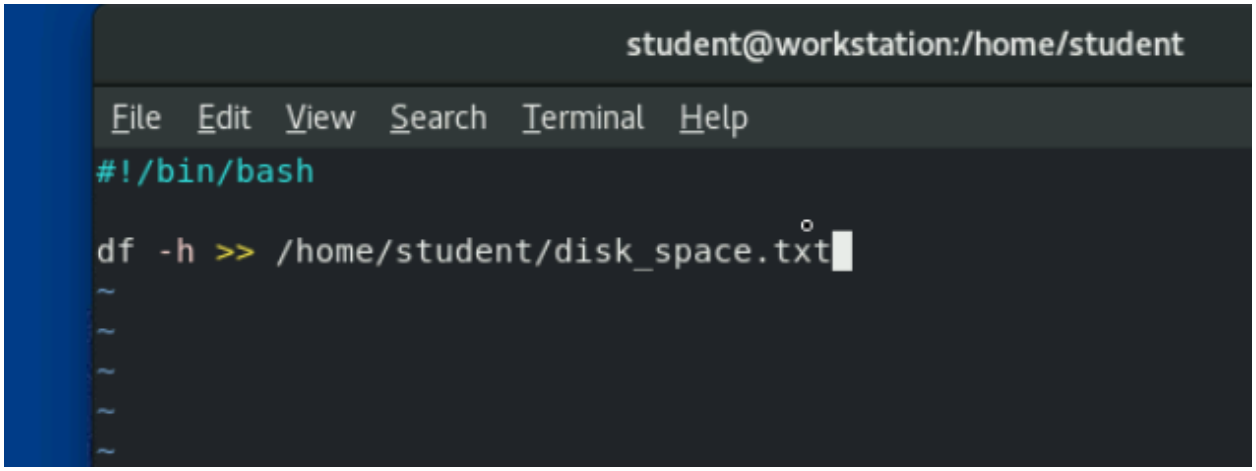


Task 8: Script to check the disk space every 10 minutes.

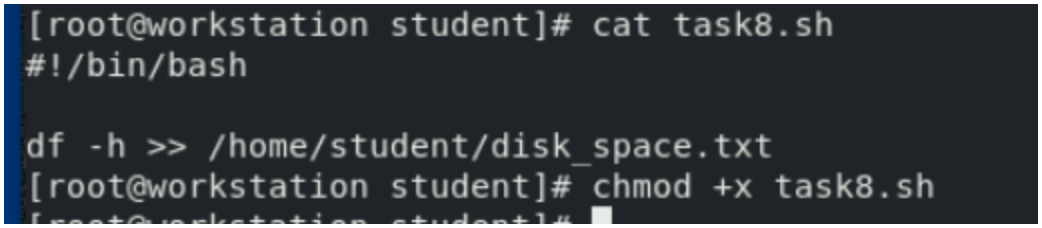
For this we create **task8.sh** file and in that we put code:

#!/bin/bash

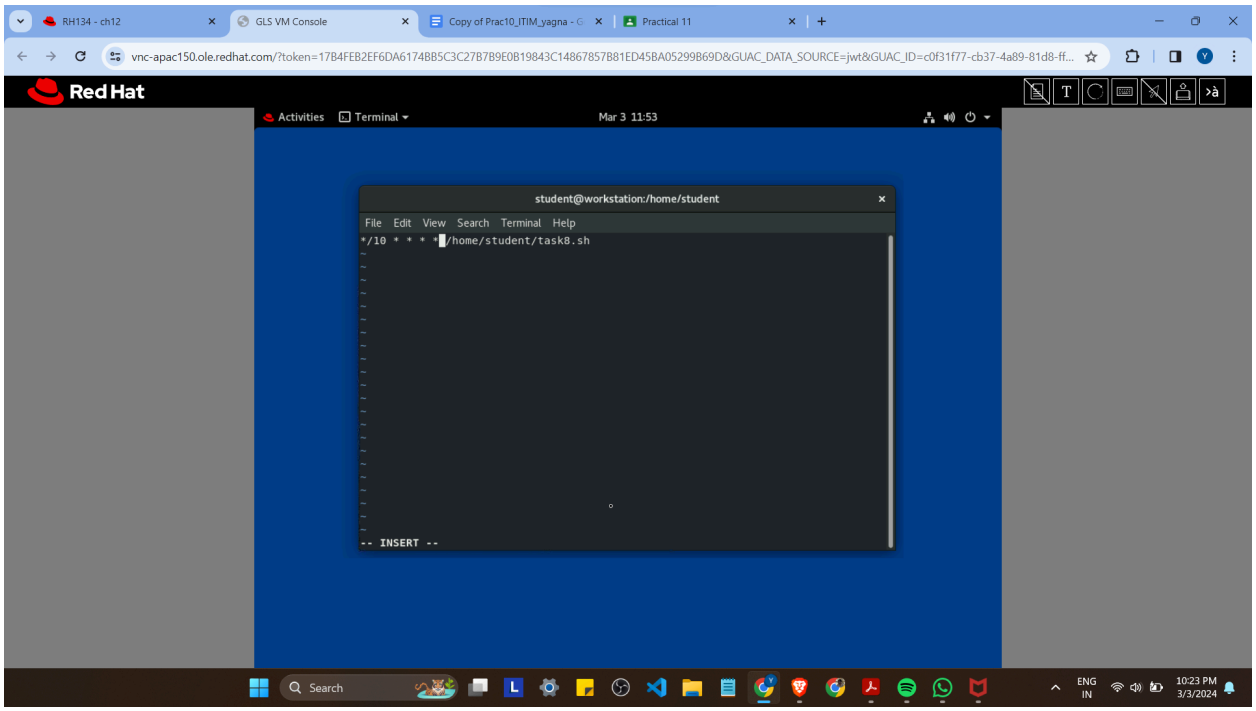
df -h >> /home/student/disk_space.txt



Now make this file executable with chmod +x task8.sh



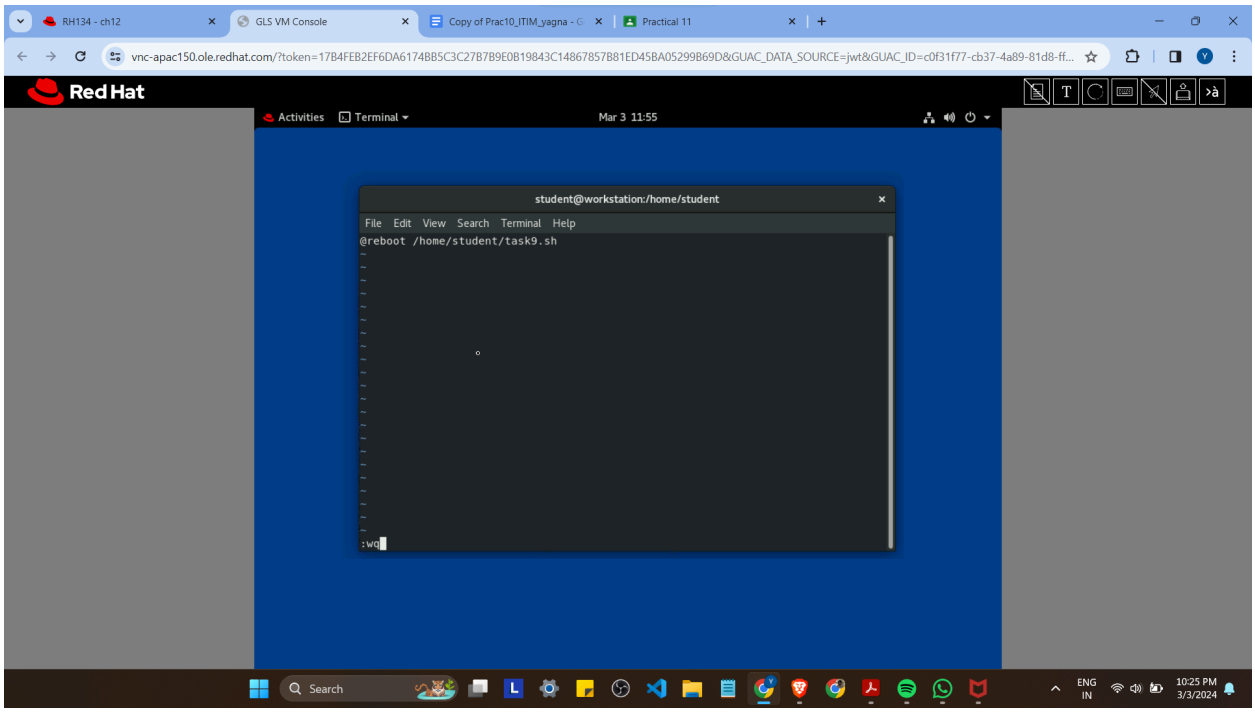
And make entry into crontab file



```
student@workstation:/home/student
File  Edit  View  Search  Terminal  Help
[root@workstation student]# cat disk_space.txt
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        2.8G     0  2.8G   0% /dev
tmpfs           2.9G     0  2.9G   0% /dev/shm
tmpfs           2.9G   18M  2.9G   1% /run
tmpfs           2.9G     0  2.9G   0% /sys/fs/cgroup
/dev/vda3       9.9G  5.2G  4.7G  53% /
/dev/vda2       100M  6.8M   94M   7% /boot/efi
tmpfs           580M   20K  580M   1% /run/user/42
tmpfs           580M   28K  580M   1% /run/user/1000
[root@workstation student]#
```

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

We can do it via simply adding @reboot in crontab entry



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

make entry in crontab and in that we'll */5 in first * means every five minutes