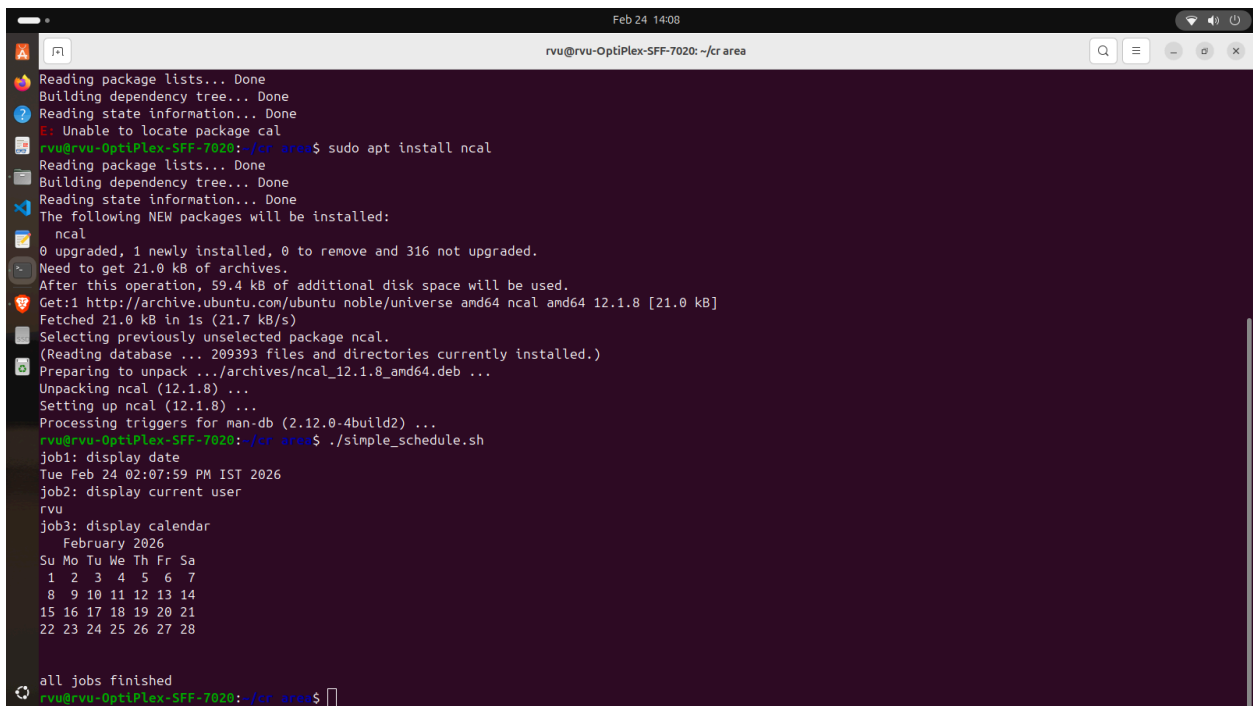


```
echo "job1: display date"
date
sleep 3
echo "job2: display current user"
whoami
sleep 3
echo "job3: display calendar"
cal
echo "all jobs finished"
```

Date cmd show the date.

Whoami shows the current user.

Cal show the calender of current month

A terminal window titled 'rvu@rvu-OptiPlex-SFF-7020: ~/cr area' showing the output of a script. The script first installs 'ncal' using 'sudo apt install ncal'. It then runs three jobs: 'job1: display date' showing 'Tue Feb 24 02:07:59 PM IST 2026', 'job2: display current user' showing 'rvu', and 'job3: display calendar' showing a calendar for February 2026. The script ends with 'all jobs finished'.

```
Feb 24 14:08
rvu@rvu-OptiPlex-SFF-7020: ~/cr area
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package cal
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ sudo apt install ncal
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ncal
0 upgraded, 1 newly installed, 0 to remove and 316 not upgraded.
Need to get 21.0 kB of archives.
After this operation, 59.4 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 ncal amd64 12.1.8 [21.0 kB]
Fetched 21.0 kB in 1s (21.7 kB/s)
Selecting previously unselected package ncal.
(Reading database ... 209393 files and directories currently installed.)
Preparing to unpack .../archives/ncal_12.1.8_amd64.deb ...
Unpacking ncal (12.1.8) ...
Setting up ncal (12.1.8) ...
Processing triggers for man-db (2.12.0-4build2) ...
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./simple_schedule.sh
job1: display date
Tue Feb 24 02:07:59 PM IST 2026
job2: display current user
rvu
job3: display calendar
  February 2026
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28

all jobs finished
rvu@rvu-OptiPlex-SFF-7020:~/cr area$
```

Output of file simple_scheduling.sh

```
#!/bin/bash
read -p "enter filename to create:" file
if [ -f "$file" ]; then
    echo "error: file already exists!"
else
    touch "$file"

    if [ $? -eq 0 ]; then
        echo "file '$file' created successfully"
    else
        echo "error creating file"
    fi
fi
```

In this [creatingfile.sh](#) file we are going to create a file and check if it exists or not.

1. By read cmd we are taking file names.
2. In 2 line we are checking if the file exists or not using -f "\$file" cmd.
3. And else we are giving touch cmd to create file.
4. We created a nested if for printing a success message.

```
rvu@rvu-OptiPlex-SFF-7020: ~/cr area
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./creatingfile.sh
enter filename to create:s
./creatingfile.sh: line 4: syntax error near unexpected token `else'
./creatingfile.sh: line 4: `else'
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./creatingfile.sh
enter filename to create:s
./creatingfile.sh: line 4: syntax error near unexpected token `else'
./creatingfile.sh: line 4: `else'
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./creatingfile.sh
bash: ./creatingfile.sh: cannot execute: required file not found
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ chmod +x creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./creatingfile.sh
bash: ./creatingfile.sh: cannot execute: required file not found
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:sp
./creatingfile.sh: line 3: [-fsp]: command not found
file 'sp' created successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:f
./creatingfile.sh: line 3: [-f: command not found
file 'f' created successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:s
./creatingfile.sh: line 3: [: missing `]'
file 's' created successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:s
error: file already exists!
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$
```

Output of previous one.

```

read -p "enter filename to copy" source
read -p "enter new file (destination): " dest
if [ -f "$source" ]; then
    cp "$source" "$dest"
    if [ $? -eq 0 ]; then
        echo "file copied successfully"
    else
        echo "error copying file"
    fi
fi

```

In [copyingfile.sh](#) we are going to create program for copying file from source to destination.

1. 1 line we are taking filename and storing in source.
2. 2line we are taking filename and storing it in dest.
3. Then we are starting if in which we are checking if the condition of the file exists or not if yes, then copy to dest.
4. And there is a nested if which tells you file is successfully created or not.

```

Feb 24 14:51
rvu@rvu-OptiPlex-SFF-7020: ~/cr area
enter filename to create:s
./creatingfile.sh: line 4: syntax error near unexpected token `else'
./creatingfile.sh: line 4: `else'
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./creatingfile.sh
bash: ./creatingfile.sh: cannot execute: required file not found
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ chmod +x creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./creatingfile.sh
bash: ./creatingfile.sh: cannot execute: required file not found
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:sp
./creatingfile.sh: line 3: [-fsp]: command not found
file 'sp' created successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:f
./creatingfile.sh: line 3: [-f: command not found
file 'f' created successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:s
./creatingfile.sh: line 3: [: missing `]'
file 's' created successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ bash ./creatingfile.sh
enter filename to create:s
error: file already exists!
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano copyingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ chmod +x copyingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./copyingfile.sh
enter filename to copy:
enter new file (destination): b
file copied successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$

```

output of program above.

```

read -p "enter filename to move" movefile
read -p "enter new new/location" newloc
if [ -f "$movefile" ]; then
    mv "$movefile" "$newloc"
    if [ $? -eq 0 ]; then
        echo "file moved successfully"
    else
        echo "error moving file"
    fi
else
    echo "error: file to move does not exist"
fi

```

You can paste the image fro

Program for moving the file from source to destination.

```

enter filename to create.s
error: file already exists!
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano creatingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano copyingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ chmod +x copyingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./copyingfile.sh
enter filename to copys
enter new file (destination): b
file copied successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano copyingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano movingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ chmod +x movingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./movingfile.sh
enter filename to moves
enter new new/location..desktop
file moved successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ 

```

Output

```

GNU nano 2.2.1 deletingfile.sh
read -p "enter filename to delete" delfile
if [ -f "$delfile" ]; then
    rm "$delfile"
    if [ $? -eq 0 ]; then
        echo "file deleted successfully"
    else
        echo "error deleting file"
    fi
else
    echo "error: file does not exist"
fi

```

Program for deleting the file.

```

rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./movingfile.sh
enter filename to moves
enter new new/location..desktop
file moved successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano movingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ nano deletingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ chmod +x deletingfile.sh
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./deletingfile.sh
enter filename to deletes
error: file does not exist
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./deletingfile.sh
enter filename to deleteg
error: file does not exist
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ ./deletingfile.sh
enter filename to deletesp
file deleted successfully
rvu@rvu-OptiPlex-SFF-7020:~/cr area$ 

```

Output

```
Feb 24 15:21
rvu@rvu-OptiPlex-SFF-7020: ~/cr area
GNU nano 7.2 menu.sh
#!/bin/bash
do
echo ""
echo "1. Create File"
echo "2. Copy File"
echo "3. Move File"
echo "4. Delete File"
echo "5. Exit"
read -p "Enter your choice: " choice

case $choice in
1)
read -p "Enter filename to create: " file
if [ -f "$file" ]; then
echo "Error: File already exists!"
else
touch "$file" && echo "File created successfully." || echo "Error creating file."
fi
;;
2)
read -p "Enter filename to copy: " source
read -p "Enter new filename (destination): " dest
if [ -f "$source" ]; then
cp "$source" "$dest" && echo "File copied successfully." || echo "Error copying file."
else
echo "Source file does not exist."
fi
;;
3)
read -p "Enter filename to move: " movefile
read -p "Enter new name/location: " newloc
[ Read 59 lines ]
Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-] To Bracket
Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy M-_ Where Was
```

```
Feb 24 15:23
rvu@rvu-OptiPlex-SFF-7020: ~/cr area
GNU nano 7.2 menu.sh
fi
;;
3)
read -p "Enter filename to move: " movefile
read -p "Enter new name/location: " newloc
if [ -f "$movefile" ]; then
mv "$movefile" "$newloc" && echo "File moved successfully." || echo "Error moving file."
else
echo "File does not exist."
fi
;;
4)
read -p "Enter filename to delete: " delfile
if [ -f "$delfile" ]; then
rm "$delfile" && echo "File deleted successfully." || echo "Error deleting file."
else
echo "File does not exist."
fi
;;
5)
echo "Exiting program..."
break
;;
*)
echo "Invalid choice! Try again."
;;
esac
done
[
Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-] To Bracket
Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy M-_ Where Was
```