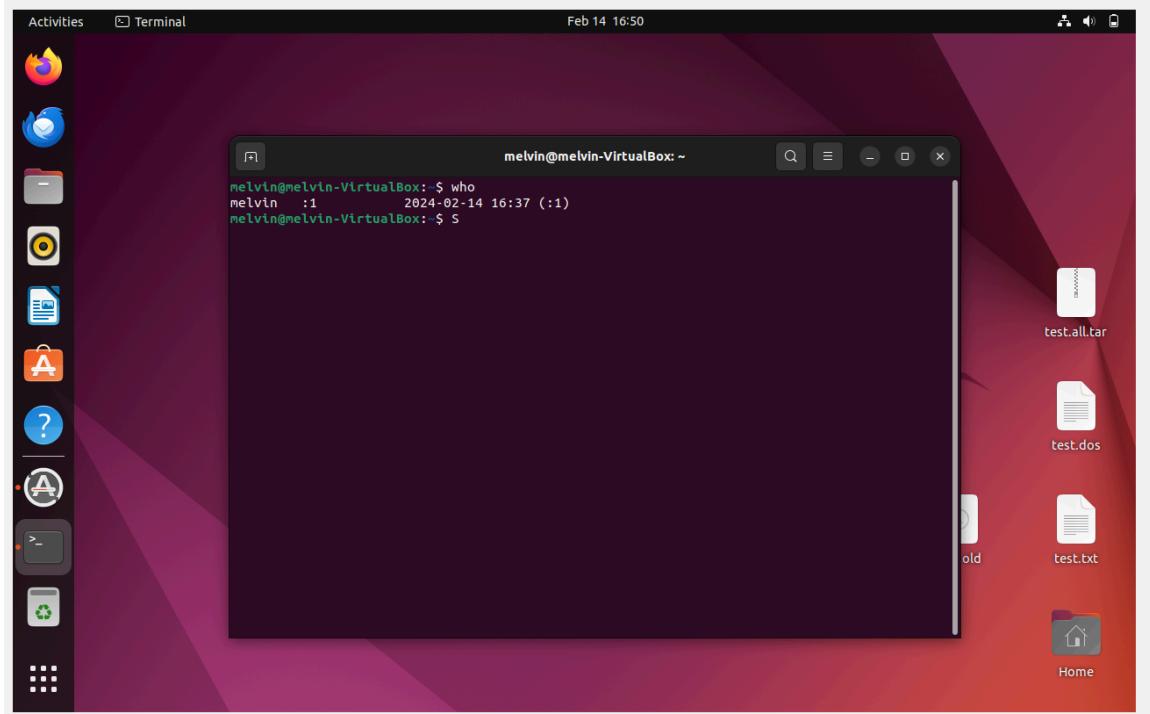


Melvin Musau - 663511
APT 2040: Operation Systems
Hands-on Exercise 2
14/02/2024

-
1. Write a shell script program to display a list of users currently logged in.



Observation: A user called Melvin is logged in. He logged in at 16:47 O'clock

2. Write a (i) shell script program in C to display “HELLO WORLD”.

File for this example is called helloworld.c

Inside helloworld.c

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "melvin@melvin-VirtualBox: ~". The terminal displays the following C code:

```
GNU nano 6.2          helloworld.c
#include <stdio.h>

int main(){
    printf("HELLO WORLD\n");
    return 0;
}
```

The terminal window has a dark background and light-colored text. At the bottom, there is a menu bar with options like Help, Write Out, Read File, Replace, Cut, Paste, Justify, Location, and Go To Line. The desktop background is a purple and red abstract design. On the left, there is a vertical dock with icons for various applications, and on the right, there are several files and folders listed.

Printing "HELLO WORLD"

A screenshot of a Linux desktop environment showing a terminal window titled "melvin@melvin-VirtualBox: ~". The terminal displays the following command-line session:

```
Setting up libubsan1:amd64 (12.3.0-1ubuntu1-22.04) ...
Setting up libnsl-dev:amd64 (1.3.0-2build2) ...
Setting up libcrypt-dev:amd64 (114.4.27-1) ...
Setting up libbinutils:amd64 (2.38-4ubuntu2.5) ...
Setting up libc-dev-bin (2.35-0ubuntu3.6) ...
Setting up libcc1-0:amd64 (12.3.0-1ubuntu1-22.04) ...
Setting up libisano:amd64 (12.3.0-1ubuntu1-22.04) ...
Setting up libbitnri:amd64 (12.3.0-1ubuntu1-22.04) ...
Setting up libc-devtools (2.35-0ubuntu3.6) ...
Setting up libtsan0:amd64 (11.4.0-1ubuntu1-22.04) ...
Setting up libctf0:amd64 (2.38-4ubuntu2.5) ...
Setting up libgcc-11-dev:amd64 (11.4.0-1ubuntu1-22.04) ...
Setting up libc6-dev:amd64 (2.35-0ubuntu3.6) ...
Setting up binutils-x86-64-linux-gnu (2.38-4ubuntu2.5) ...
Setting up binutils (2.38-4ubuntu2.5) ...
Setting up gcc-11 (11.4.0-1ubuntu1-22.04) ...
Setting up gcc (4:11.2.0-1ubuntu1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
melvin@melvin-VirtualBox: $ gcc helloworld.c -o helloworld
melvin@melvin-VirtualBox: $ ./helloworld
HELLO WORLD
melvin@melvin-VirtualBox: $ gcc helloworld.c -o helloworld
melvin@melvin-VirtualBox: $ ./helloworld
HELLO WORLD
melvin@melvin-VirtualBox: $ nano helloworld.c
melvin@melvin-VirtualBox: $ ./helloworld
HELLO WORLD
melvin@melvin-VirtualBox: $
```

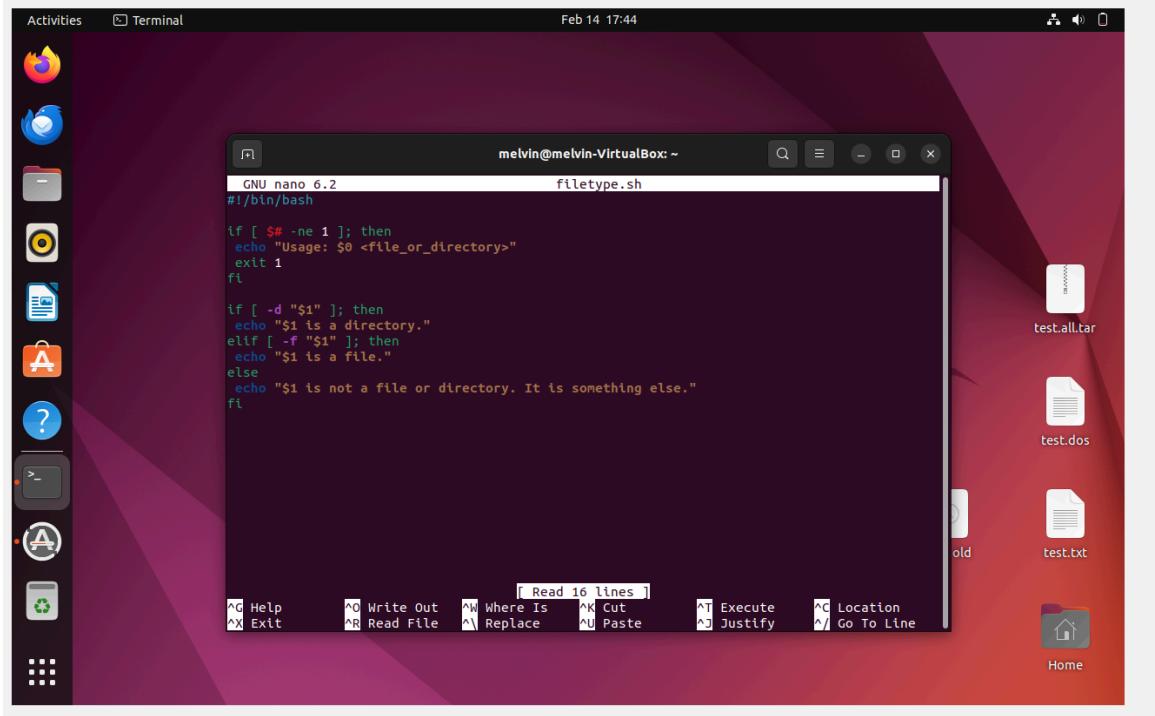
This C shell script program displays a “HELLO WORLD” message when running it.

3. Shell:

- Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.

File for this example is called filetype.sh.

Inside filetype.sh



The screenshot shows a Linux desktop environment with a dark purple background. A terminal window titled "filetype.sh" is open, displaying the following script:

```
GNU nano 6.2               filetype.sh
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 <file_or_directory>"
    exit 1
fi

if [ -d "$1" ]; then
    echo "$1 is a directory."
elif [ -f "$1" ]; then
    echo "$1 is a file."
else
    echo "$1 is not a file or directory. It is something else."
fi
```

The terminal window has a title bar "melvin@melvin-VirtualBox: ~". The status bar at the bottom shows keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^/ Go To Line. The date "Feb 14 17:44" is also visible in the top right corner of the desktop.

Checking if filetype is a file or directory.

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window. The terminal window title is "melvin@melvin-VirtualBox: ~". The terminal content shows a shell script named "filetype.sh" being run. The script uses the "nano" editor to create the file, changes its mode to executable, runs it, and then runs it again with its own path. The output indicates that "/home/melvin/filetype.sh" is a file. The desktop background is dark purple/red, and there are several icons on the right side of the screen, including "test.all.tar", "test.dos", "old", "test.txt", and "Home".

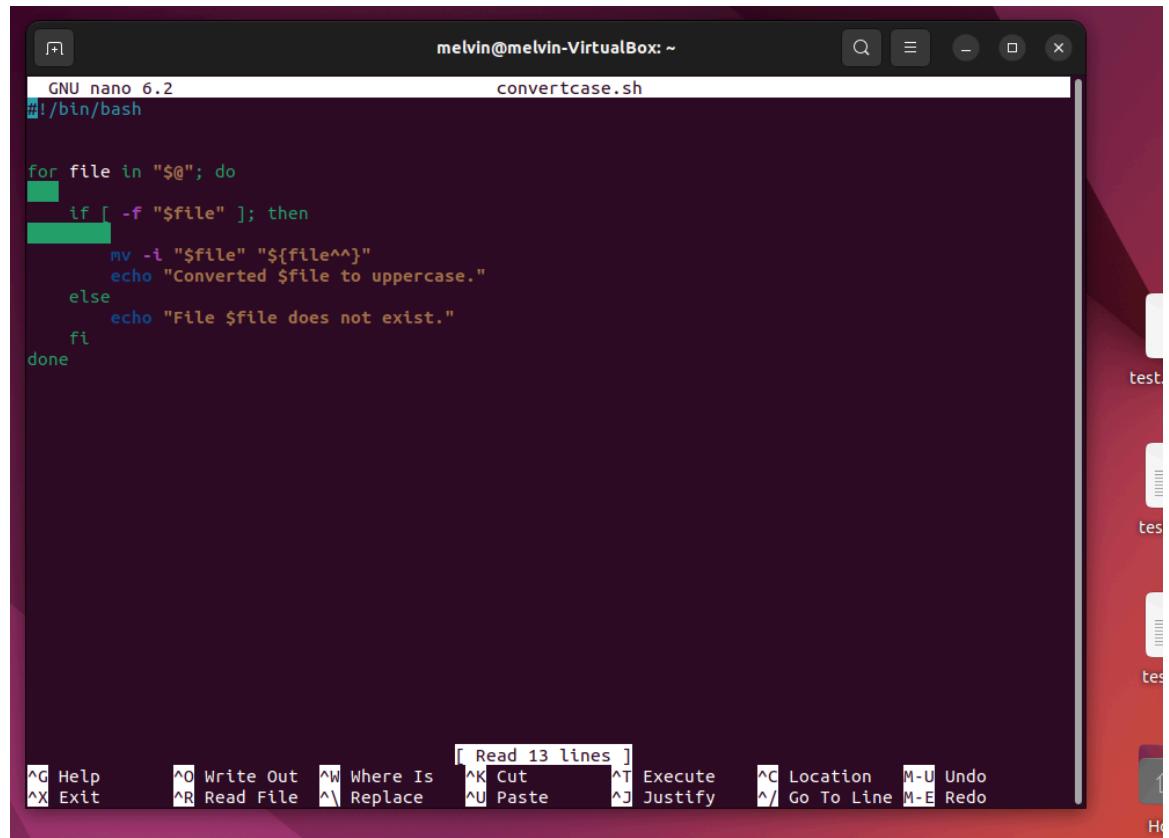
```
melvin@melvin-VirtualBox:~$ nano filetype.sh
melvin@melvin-VirtualBox:~$ chmod +x filetype.sh
melvin@melvin-VirtualBox:~$ ./filetype.sh /home/melvin/filetype.sh
/home/melvin/filetype.sh is a file.
melvin@melvin-VirtualBox:~$ nano filetype.sh
melvin@melvin-VirtualBox:~$ nano filetype.sh
melvin@melvin-VirtualBox:~$ ./filetype.sh /home/melvin/filetype.sh
/home/melvin/filetype.sh is a file.
melvin@melvin-VirtualBox:~$ nano filetype.sh
melvin@melvin-VirtualBox:~$ nano filetype.sh
melvin@melvin-VirtualBox:~$ ./filetype.sh /home/melvin/filetype.sh
/home/melvin/filetype.sh is a file.
melvin@melvin-VirtualBox:~$
```

Above is a shell script that reports on whether a program is a directory, a file, or something else.

- b. Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.

File for this example is called convertcase.sh

Inside convertcase.sh



```
melvin@melvin-VirtualBox: ~
GNU nano 6.2                               convertcase.sh

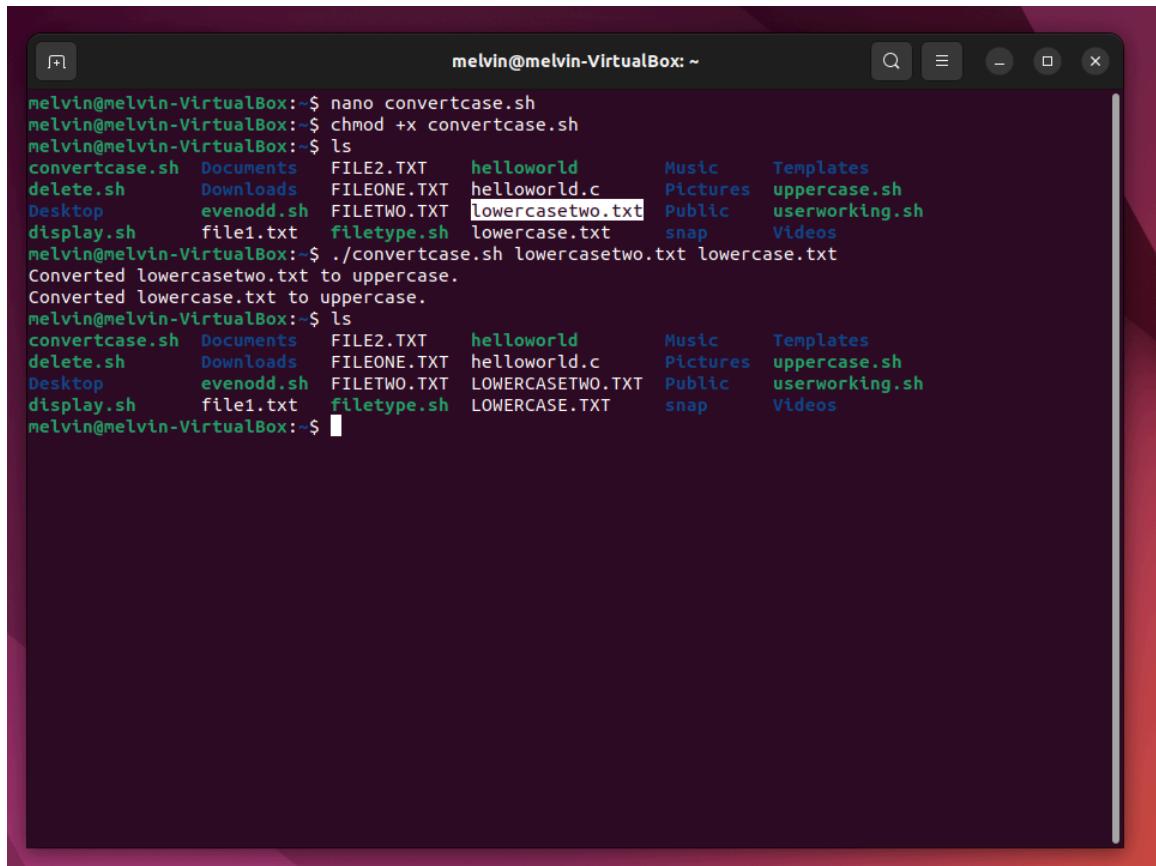
#!/bin/bash

for file in "$@"; do
    if [ -f "$file" ]; then
        mv -i "$file" "${file^^}"
        echo "Converted $file to uppercase."
    else
        echo "File $file does not exist."
    fi
done
```

[Read 13 lines]

^G Help **^O Write Out** **^W Where Is** **^K Cut** **^T Execute** **^C Location** **M-U Undo**
^X Exit **^R Read File** **^A Replace** **^U Paste** **^J Justify** **^/ Go To Line** **M-E Redo**

Changing files lowercase.txt and lowercasetwo.txt to uppercase



```
melvin@melvin-VirtualBox: $ nano convertcase.sh
melvin@melvin-VirtualBox: $ chmod +x convertcase.sh
melvin@melvin-VirtualBox: $ ls
convertcase.sh  Documents  FILE2.TXT  helloworld  Music  Templates
delete.sh       Downloads  FILEONE.TXT  helloworld.c Pictures uppercase.sh
Desktop        evenodd.sh FILETWO.TXT  lowercase.txt  Public  userworking.sh
display.sh      file1.txt  filetype.sh  lowercase.txt  snap   Videos
melvin@melvin-VirtualBox: $ ./convertcase.sh lowercase.txt uppercase.txt
Converted lowercase.txt to uppercase.
Converted lowercase.txt to uppercase.
melvin@melvin-VirtualBox: $ ls
convertcase.sh  Documents  FILE2.TXT  helloworld  Music  Templates
delete.sh       Downloads  FILEONE.TXT  helloworld.c Pictures uppercase.sh
Desktop        evenodd.sh FILETWO.TXT  LOWERCASE.TXT Public  userworking.sh
display.sh      file1.txt  filetype.sh  LOWERCASE.TXT  snap   Videos
```

```
melvin@melvin-VirtualBox: $ nano convertcase.sh
melvin@melvin-VirtualBox: $ chmod +x convertcase.sh
melvin@melvin-VirtualBox: $ ls
convertcase.sh  Documents  FILE2.TXT  helloworld      Music      Templates
delete.sh       Downloads   FILEONE.TXT helloworld.c    Pictures   uppercase.sh
Desktop        evenodd.sh  FILETWO.TXT lowercasetwo.txt Public    userworking.sh
display.sh     file1.txt   filetype.sh  lowercase.txt  snap      Videos
melvin@melvin-VirtualBox: $ ./convertcase.sh lowercase.txt lowercasetwo.txt
Converted lowercase.txt to uppercase.
Converted lowercasetwo.txt to uppercase.
melvin@melvin-VirtualBox: $ ls
convertcase.sh  Documents  FILE2.TXT  helloworld      Music      Templates
delete.sh       Downloads   FILEONE.TXT helloworld.c    Pictures   uppercase.sh
Desktop        evenodd.sh  FILETWO.TXT LOWERCASETWO.TXT Public    userworking.sh
display.sh     file1.txt   filetype.sh  LOWERCASE.TXT  snap      Videos
melvin@melvin-VirtualBox: $
```

The above shell script accepts one or more file names(lowercase.txt and lowercasetwo.text) as arguments and converts all of them to uppercase.

- c. Write a shell script that determines the period for which a specified user is working on the system.

File for this example is called userworking.sh

Inside userworking.sh.

Activities Terminal Feb 14 21:23
melvin@melvin-VirtualBox: ~

```
GNU nano 6.2
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 <username>"
    exit 1
fi

username=$1

login_time=$(last -n 1 "$username" | head -n 1 | awk '{print $5, $6, $7, $8}'| sed 's/still//')
if [ -z "$login_time" ]; then
    echo "User $username is not currently logged in."
    exit 1
fi

echo "Login time: $login_time"
login_timestamp=$(date -d "$login_time" +%s)
current_timestamp=$(date +%s)
current_time=$(date -d "@$current_timestamp" +"%H:%M:%S")
echo "Current time: $current_time"
time_difference=$((current_timestamp - login_timestamp))

hours=$((time_difference / 3600))
minutes=$((($time_difference % 3600) / 60 ))
seconds=$((time_difference % 60))

echo "Period user has been working on the system: $hours hours, $minutes minutes, $seconds seconds"
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^L Replace ^U Paste ^J Justify ^I Go To Line M-E Redo M-C Copy

Ubuntu 22.0 (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Feb 14 21:24
melvin@melvin-VirtualBox: ~

```
GNU nano 6.2
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 <username>"
    exit 1
fi

username=$1

login_time=$(last -n 1 "$username" | head -n 1 | awk '{print $5, $6, $7, $8}'| sed 's/still//')
if [ -z "$login_time" ]; then
    echo "User $username is not currently logged in."
    exit 1
fi

echo "Login time: $login_time"
login_timestamp=$(date -d "$login_time" +%s)
current_timestamp=$(date +%s)
current_time=$(date -d "@$current_timestamp" +"%H:%M:%S")
echo "Current time: $current_time"
time_difference=$((current_timestamp - login_timestamp))

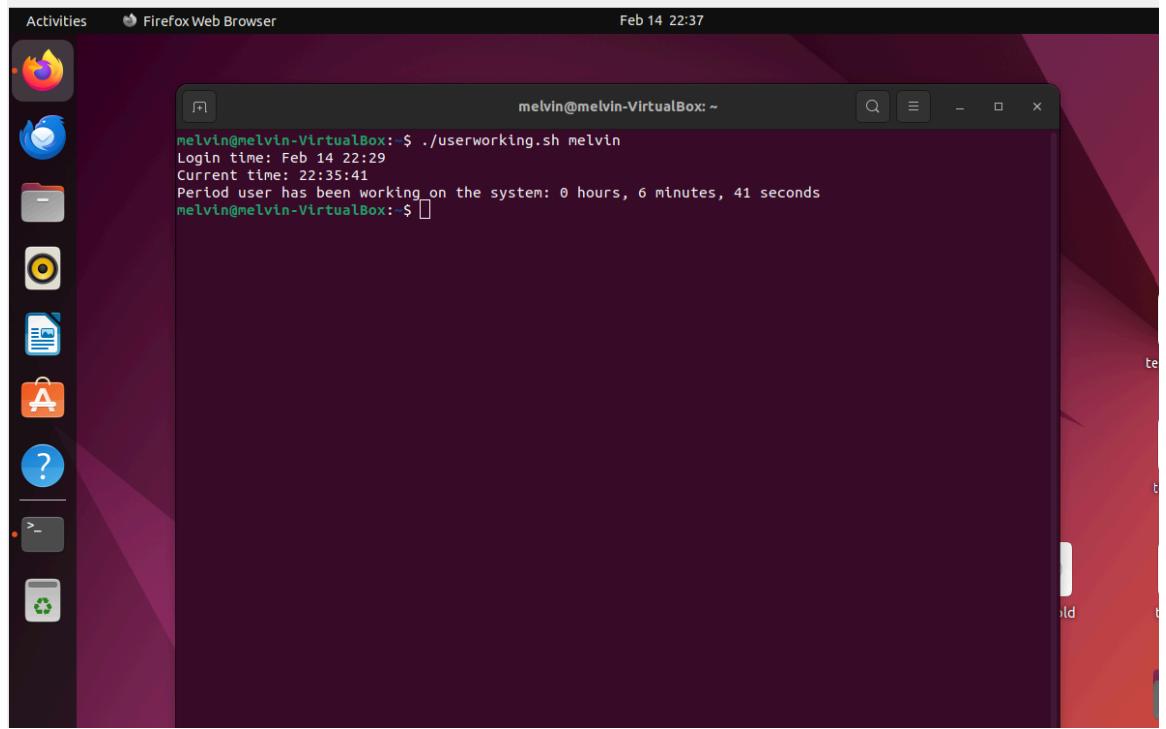
hours=$((time_difference / 3600))
minutes=$((($time_difference % 3600) / 60 ))
seconds=$((time_difference % 60))

echo "Period user has been working on the system: $hours hours, $minutes minutes, $seconds seconds"
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^L Replace ^U Paste ^J Justify ^I Go To Line M-E Redo M-C Copy

Activate Windows
Go to Settings to activate Windows.

Seeing the period that the user has been working on the system.



This is a shell script that determines the period for which a specified user(melvin in this case) has been working on the system. ie 6 minutes and 41 sec.

- d. (a) Write a shell script that accepts a filename, starting and ending line numbers as arguments and displays all the lines between the given line numbers.

File for this example is called display.sh

Inside display.sh

Activities Terminal Feb 14 21:12 melvin@melvin-VirtualBox: ~

```
GNU nano 6.2
#!/bin/bash

if [ $# -ne 3 ]; then
    echo "Usage: $0 <file_name> <start_line_number> <end_line_number>"
    exit 1
fi

file_name=$1
start_line=$2
end_line=$3

sed -n "${start_line},${end_line}p" "$file_name"
```

^G Help ^X Exit ^Q Write Out ^R Read File ^W Where Is ^K Cut ^U Paste ^T Execute ^J Justify ^C Location ^G Go To Line M-U Undo M-E Redo M-A Set Mark M-G Copy

Activate ^

Inside userworking.sh

Activities Terminal Feb 14 21:18 melvin@melvin-VirtualBox: ~

```
userworking.sh

GNU nano 6.2
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 <username>"
    exit 1
fi

username=$1

login_time=$(last -n 1 "$username" | head -n 1 | awk '{print $5, $6, $7, $8}' | sed 's/still//')

if [ -z "$login_time" ]; then
    echo "User $username is not currently logged in."
    exit 1
fi

echo "Login time: $login_time"
login_timestamp=$(date -d "$login_time" +%s)

current_timestamp=$(date +'%s')
current_time=$(date -d "@$current_timestamp" +"%H:%M:%S")
echo "Current time: $current_time"

time_difference=$((current_timestamp - login_timestamp))

hours=$((time_difference / 3600))
minutes=$((($time_difference % 3600) / 60 ))
seconds=$((time_difference % 60))

echo "Period user has been working on the system: $hours hours, $minutes minutes, $seconds seconds"
```

[Read 33 lines] ^G Help ^X Exit ^Q Write Out ^R Read File ^W Where Is ^K Cut ^U Paste ^T Execute ^J Justify ^C Location ^G Go To Line M-U Undo M-E Redo M-A Set Mark M-G Copy

showing userworking.sh's line 3 - 7

A screenshot of a Linux desktop environment. At the top, there's a header bar with "Activities" and "Terminal" tabs, the date "Feb 14 21:22", and system icons. Below the header is a terminal window titled "melvin@melvin-VirtualBox: ~". The terminal shows the following command-line session:

```
melvin@melvin-VirtualBox: $ nano display.sh
melvin@melvin-VirtualBox: $ nano display.sh
melvin@melvin-VirtualBox: $ nano display.sh
melvin@melvin-VirtualBox: $ nano display.sh
melvin@melvin-VirtualBox: $ [A
^CFatal Python error: init_sys_streams: can't initialize sys standard streams
Python runtime state: core initialized
Traceback (most recent call last):
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1006, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 688, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 879, in exec_module
  File "<frozen importlib._bootstrap_external>", line 975, in get_code
  File "<frozen importlib._bootstrap_external>", line 1074, in get_data
KeyboardInterrupt
melvin@melvin-VirtualBox: $ nano display.sh
melvin@melvin-VirtualBox: $ nano userworking.sh
melvin@melvin-VirtualBox: $ nano userworking.sh
melvin@melvin-VirtualBox: $ ./display.sh userworking.sh 3 7
if [ $# -ne 1 ]; then
    echo "Usage: $0 <username>"
    exit 1
fi
melvin@melvin-VirtualBox: $
```

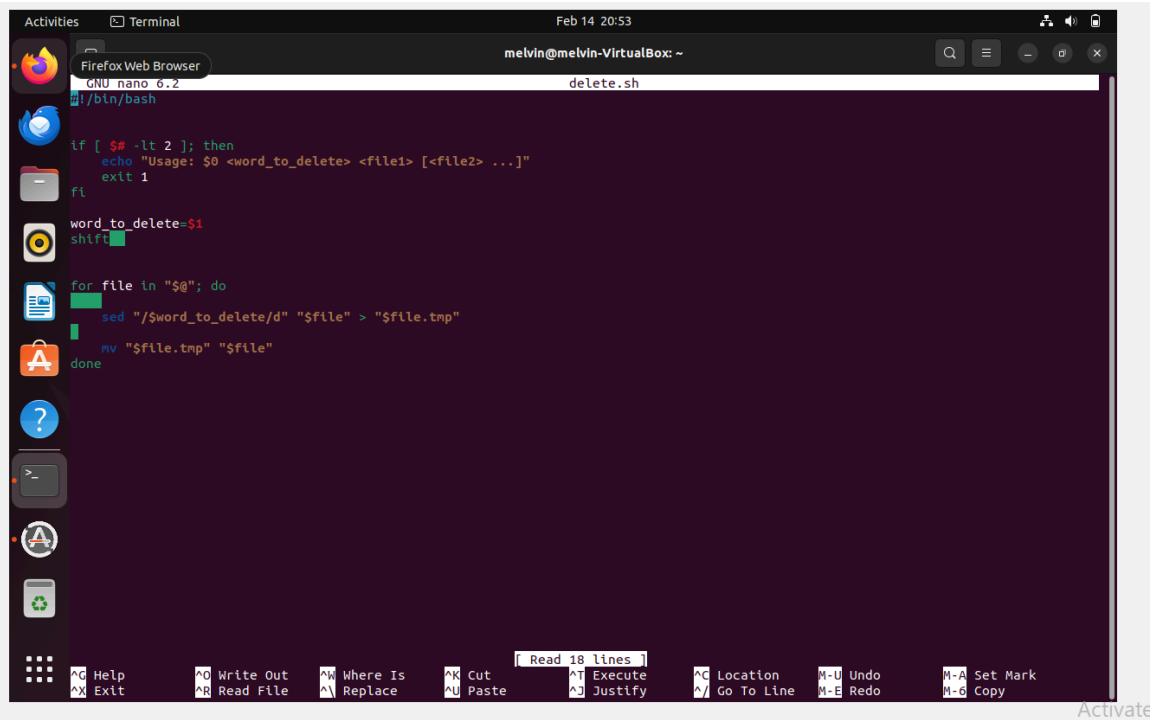
The desktop interface includes a dock with icons for a file manager, terminal, browser, and other applications. A vertical sidebar on the right contains "Activate" and "Go to Settings".

The above shell script accepts a filename(userworking.sh), starting and ending line numbers(3 and 7) and displays all the lines between the given line numbers.

- e. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

File for this example is called delete.sh

Contents of delete.sh file created



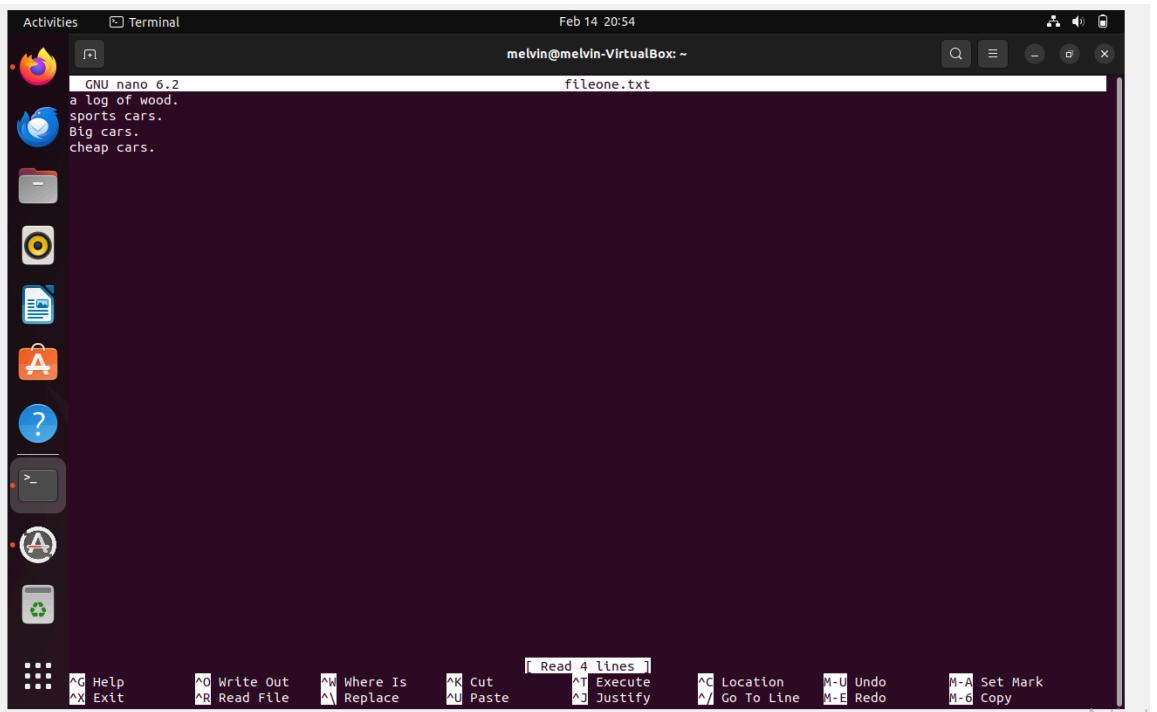
```
Activities Terminal Feb 14 20:53
melvin@melvin-VirtualBox: ~
GNU nano 6.2
#!/bin/bash

if [ $# -lt 2 ]; then
    echo "Usage: $0 <word_to_delete> <file1> [<file2> ...]"
    exit 1
fi

word_to_delete=$1
shift

for file in "$@"; do
    sed "/$word_to_delete/d" "$file" > "$file.tmp"
    mv "$file.tmp" "$file"
done
```

Contents of fileone.txt



```
Activities Terminal Feb 14 20:54
melvin@melvin-VirtualBox: ~
GNU nano 6.2
fileone.txt
a log of wood.
sports cars.
Big cars.
cheap cars.
```

Contents of filetwo.txt

A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for various applications like a browser, file manager, terminal, and system settings. The main window is a terminal titled "Terminal" with the command "GNU nano 6.2". The text in the editor is:

```
GNU nano 6.2
paper is made of wood.
GNU
TXT
```

The status bar at the bottom shows keyboard shortcuts for various functions like Help, Write Out, Cut, Paste, and Undo.

Deleting lines with the word “wood” in fileone.txt an filetwo.txt

A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for various applications like a browser, file manager, terminal, and system settings. The main window is a terminal titled "Terminal" with the command "melvin@melvin-VirtualBox: ~". The text in the terminal is a shell script named "delete.sh" which creates two files ("fileone.txt" and "filetwo.txt"), then attempts to delete them using the "sed" command. The output shows that the script fails because the files do not exist.

```
melvin@melvin-VirtualBox:~$ nano delete.sh
melvin@melvin-VirtualBox:~$ touch fileone.txt
melvin@melvin-VirtualBox:~$ nano fileone.txt
melvin@melvin-VirtualBox:~$ touch filetwo.txt
melvin@melvin-VirtualBox:~$ nano filetwo.txt
melvin@melvin-VirtualBox:~$ nano delete.sh
melvin@melvin-VirtualBox:~$ nano fileone.txt
melvin@melvin-VirtualBox:~$ nano filetwo.txt
melvin@melvin-VirtualBox:~$ chmod +x delete.sh
melvin@melvin-VirtualBox:~$ ./delete.sh wood file1.txt file2.txt
sed: can't read file1.txt: No such file or directory
sed: can't read file2.txt: No such file or directory
melvin@melvin-VirtualBox:~$ ./delete.sh wood fileone.txt filetwo.txt
melvin@melvin-VirtualBox:~$
```

Showing that the lines with “wood” have been deleted in fileone.txt and filetwo.txt

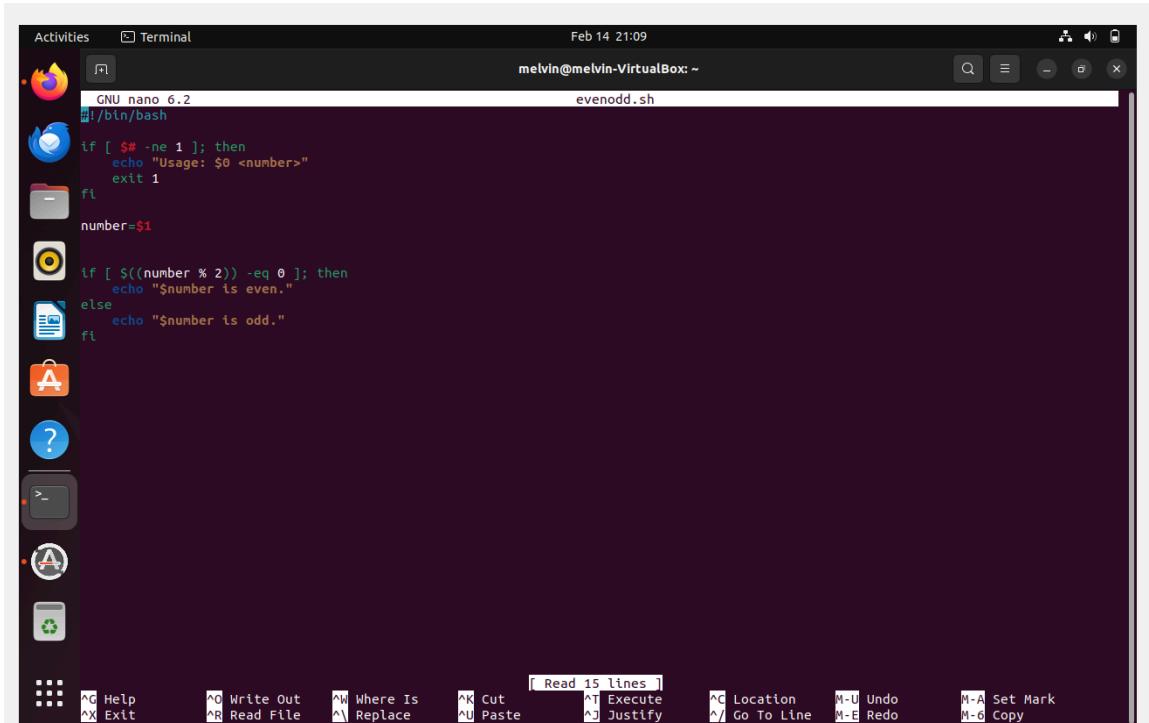
The image shows two terminal windows side-by-side. Both windows are titled "Terminal" and show the command "GNU nano 6.2". The top window is titled "fileone.txt" and contains the text "sports cars." and "Big cars.". The bottom window is titled "filetwo.txt" and contains the text "GNU TXT". Both windows have a dark background and a light-colored text area. The bottom of each window shows a menu bar with various keyboard shortcuts for file operations like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, and Copy.

The above shell script deletes all lines containing a specified word(wood in this case) in one or more files supplied as arguments(fileone.txt and filetwo.txt) to it.

f. Write a shell Script program to check whether the given number is even or odd.

File for this example is called evenodd.sh

Inside evenodd.sh



The screenshot shows a terminal window titled "evenodd.sh" running on a Linux system. The window title bar includes the date and time: "Feb 14 21:09". The terminal content displays the following shell script:

```
GNU nano 6.2
#!/bin/bash

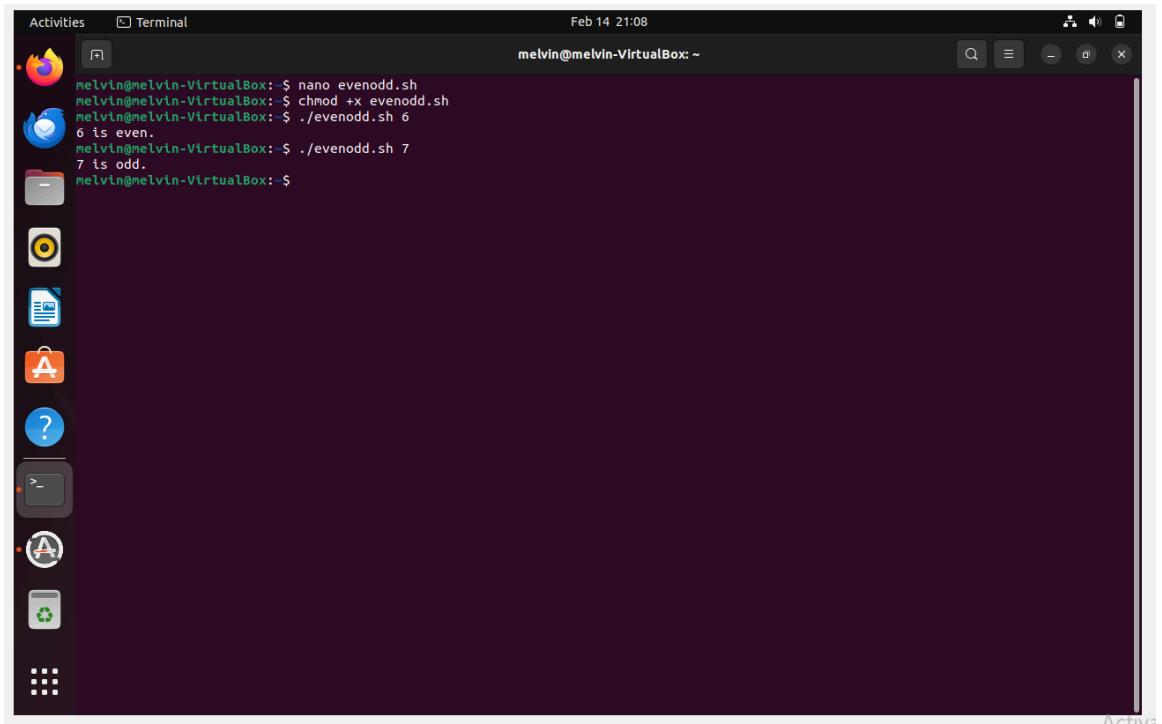
if [ $# -ne 1 ]; then
    echo "Usage: $0 <number>"
    exit 1
fi

number=$1

if [ $((number % 2)) -eq 0 ]; then
    echo "$number is even."
else
    echo "$number is odd."
fi
```

The terminal interface includes a docked application menu on the left and a toolbar at the bottom with various keyboard shortcut icons.

Testing the number 6 and 7.



A screenshot of a Linux desktop environment, likely Elementary OS, showing a terminal window titled "Terminal". The terminal window has a dark background and contains the following text:

```
melvin@melvin-VirtualBox: $ nano evenodd.sh
melvin@melvin-VirtualBox: $ chmod +x evenodd.sh
melvin@melvin-VirtualBox: $ ./evenodd.sh 6
6 is even.
melvin@melvin-VirtualBox: $ ./evenodd.sh 7
7 is odd.
melvin@melvin-VirtualBox: $
```

The terminal window is located in the top right corner of the desktop. To its left is a vertical dock containing icons for various applications: a browser, file manager, terminal, text editor, help, file browser, and system tray.

The above shell Script program checks whether the given number is even or odd.