OPERATING SYSTEMS ASSIGNMENT - 2

CREATE A DEVICE DRIVER FOR COMPILED KERNEL MODULE 5.14

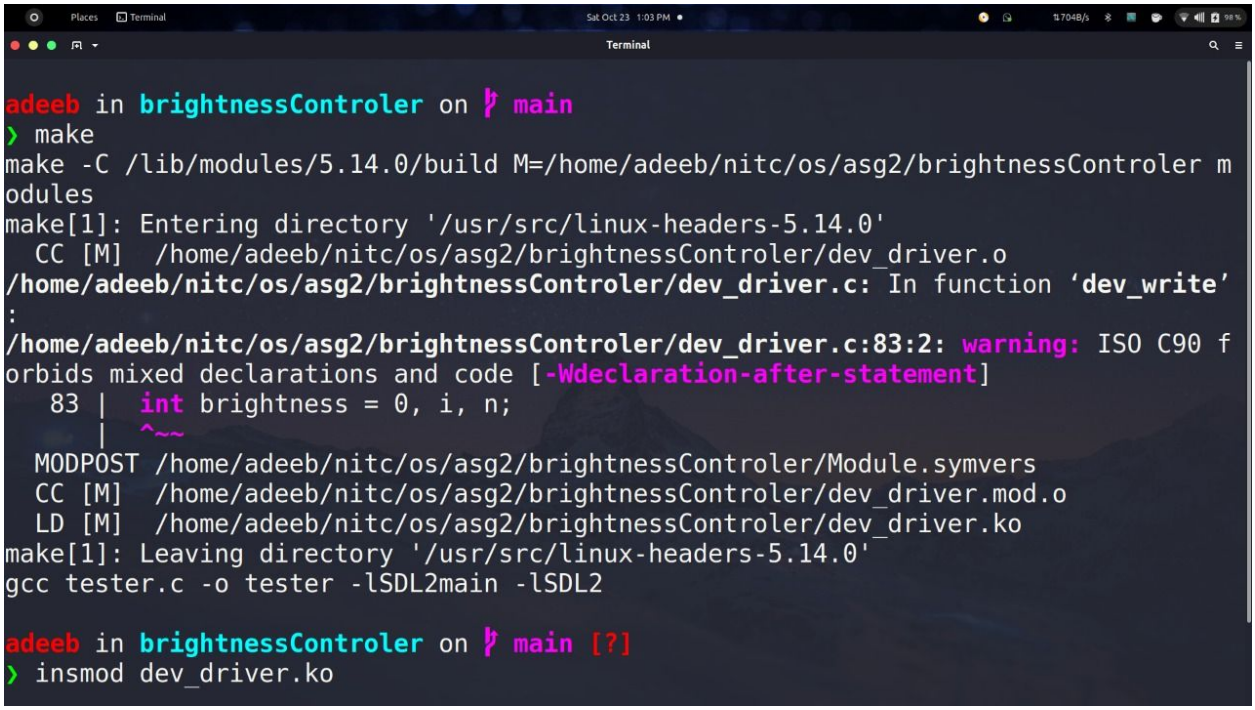
PROBLEM STATEMENT

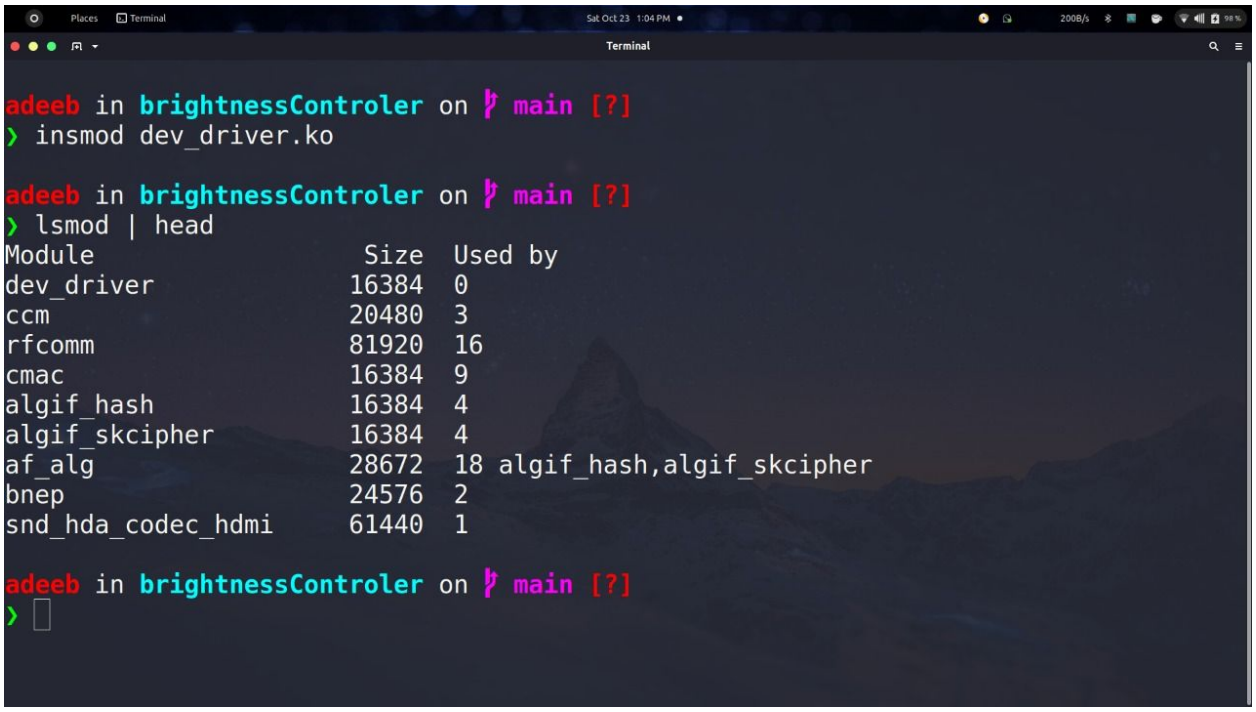
A character device is one of the simplest ways to communicate with a module in the Linux kernel. These devices are presented as special files in a /dev directory and support direct reading and writing of any data, byte by byte, like a stream.The implementation done here is of a custom Linux device driver. The device driver reads the mouse scroll as input and changes the brightness of the screen.

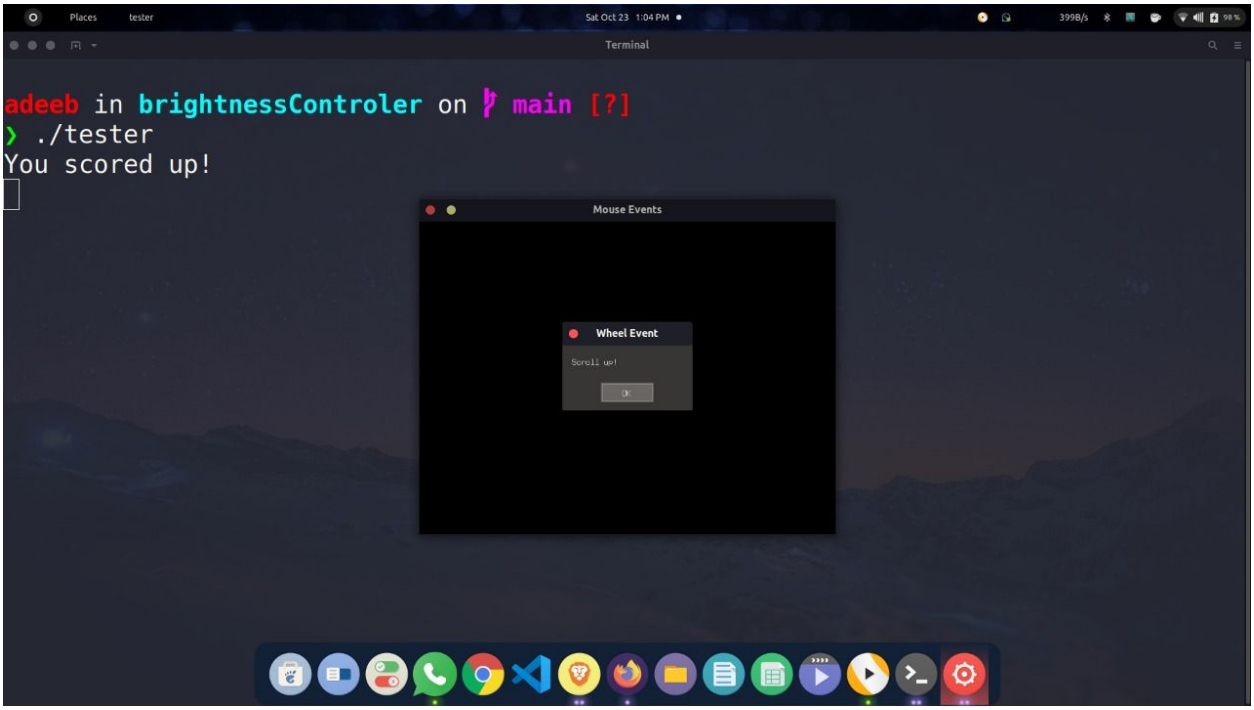
METHODOLOGY

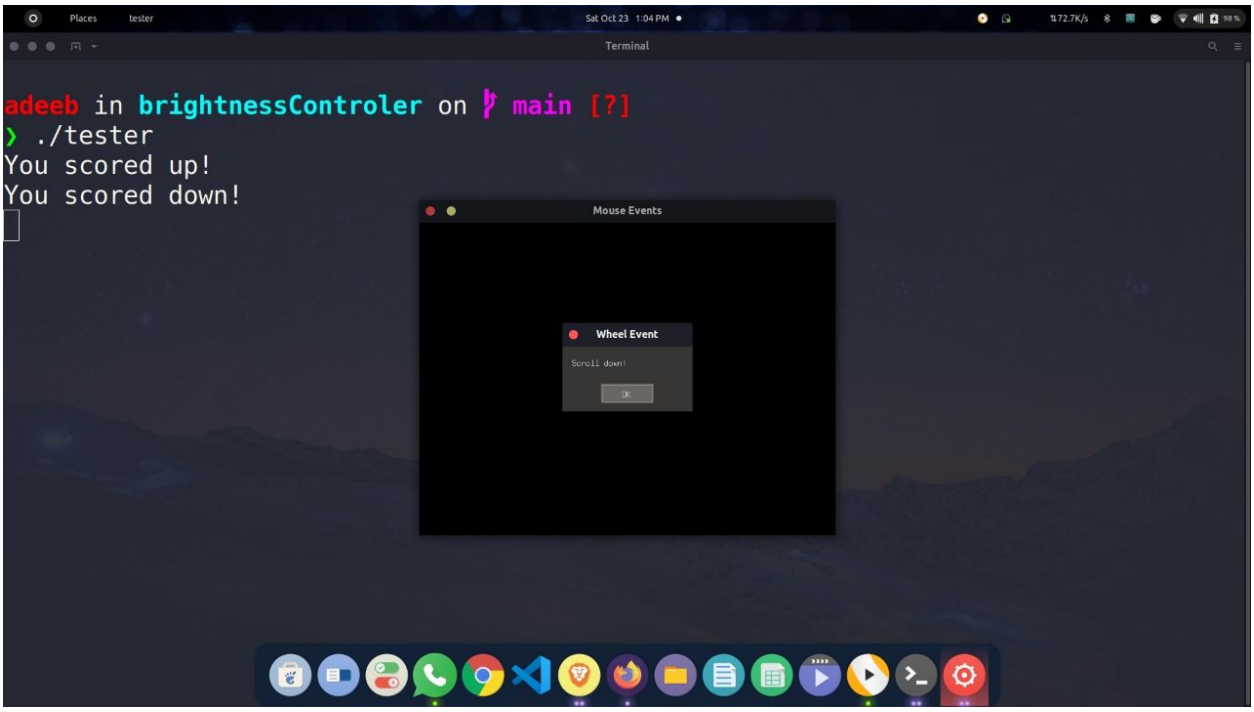
- Execute first:
  - Change to Superuser using su command
  - Use the command ‘ mknod -m 666 /dev/mbdriver c 45 1 ’ to create the character device file in /dev/mbdriver directory
  - Then the command ‘ ls -l /dev/mbdriver ’ to view the driver file in the folder.

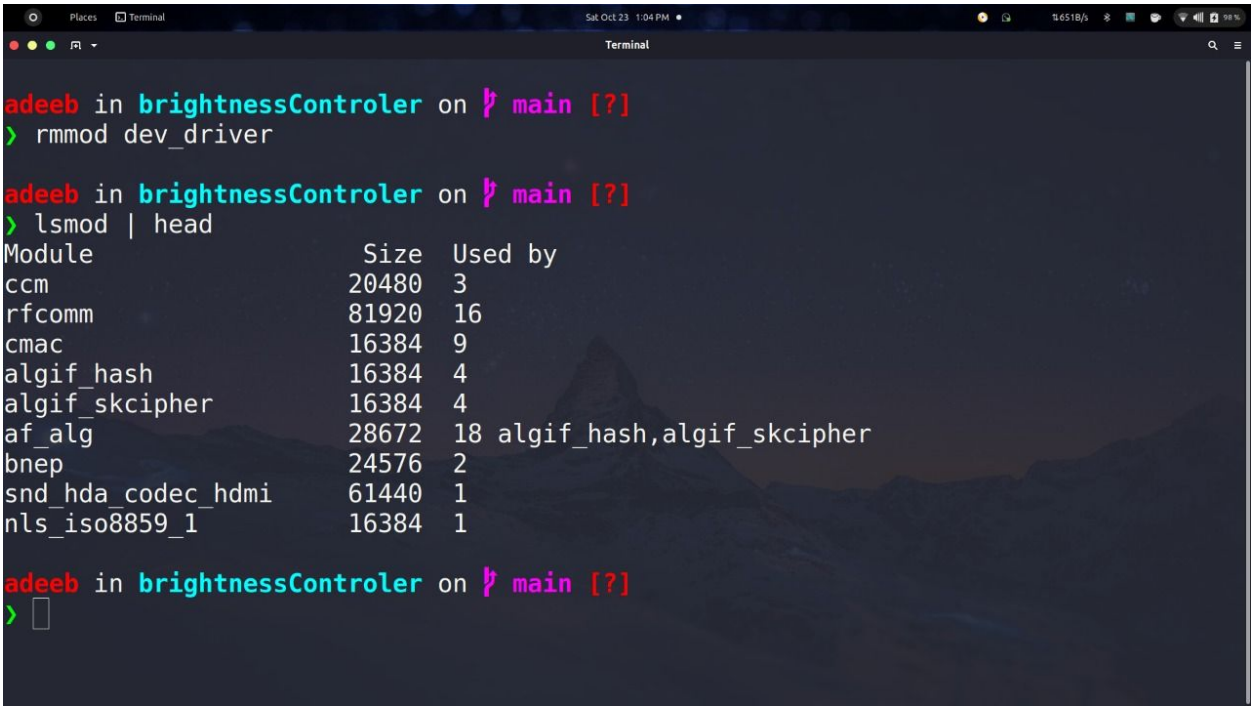
- Repeat to test the driver (dmesg is to debug print and to debug):
    - After that use the ‘ make ’ command to compile the files.
    - Then use the ‘ insmod dev_driver.ko ’ command to load the modules in the dev_driver.ko file.
    - Use the ‘ lsmod | head ’ command to check if the module is loaded.
    - Use ‘ ./tester ’ command to run the device driver and scroll to see the brightness change.
    - Use the ‘ rmmode dev_driver ’ command to unload the module.
    - And check if the module if the module is unloaded using ‘ lsmod | head ’ command.
    - Use the dmesg command to view the kernel messages.



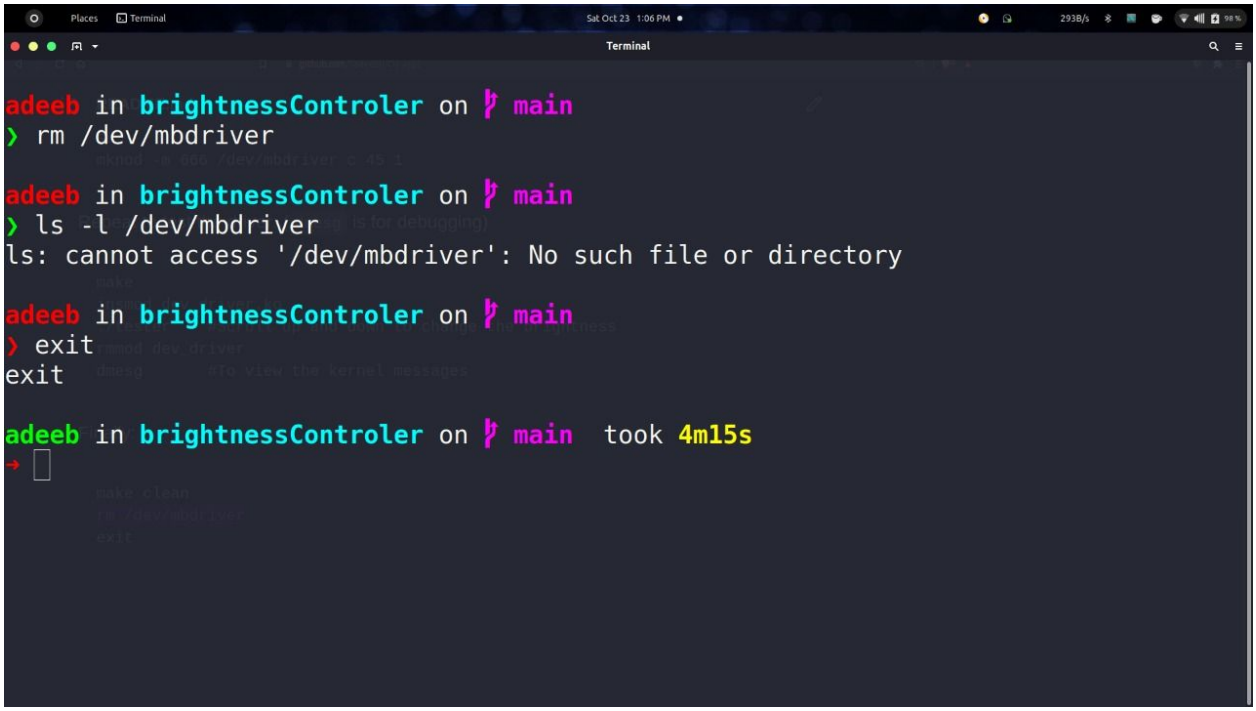

Places
Terminal
Sat Oct 23 1:04 PM
200B/s
Terminal
adeeb in brightnessControler on main [?]
❯ insmod dev_driver.ko
adeeb in brightnessControler on main [?]
❯ lsmod | head
Module Size Used by
dev_driver 16384 0
ccm 20480 3
rfcomm 81920 16
cmac 16384 9
algif_hash 16384 4
algif_skcipher 16384 4
af_alg 28672 18 algif_hash,algif_skcipher
bnep 24576 2
snd_hda_codec_hdmi 61440 1
adeeb in brightnessControler on main [?]
❯


Places
tester
Sat Oct 23 1:04 PM
399B/s
Terminal
adeeb in brightnessControler on main [?]
❯ ./tester
You scored up!
Mouse Events
Wheel Event
Scroll up!
OK


Places
tester
Sat Oct 23 1:04 PM
Terminal
adeeb in brightnessControler on main [?]
./tester
You scored up!
You scored down!
Mouse Events
Wheel Event
Scroll down!
OK


Places
Terminal
Sat Oct 23 1:04 PM
Terminal
adeeb in brightnessControler on main [?]
rmmod dev_driver
adeeb in brightnessControler on main [?]
lsmod | head
Module Size Used by
ccm 20480 3
rfcomm 81920 16
cmac 16384 9
algif_hash 16384 4
algif_skcipher 16384 4
af_alg 28672 18 algif_hash,algif_skcipher
bnep 24576 2
snd_hda_codec_hdmi 61440 1
nls_iso8859_1 16384 1
adeeb in brightnessControler on main [?]

- Finally :
    - Use the ‘ make clean ‘ command to undo all the changes .
    - Use the ‘ rm /dev/mbdriver ’ command to delete the mbdriver directory.
    - Then exit.



EXPLANATION

This is an implementation of a custom Linux device driver. It reads the mouse scroll and changes the brightness of the screen.

Device drivers are able to access privileged functions and have access to stuff userland software is not. So this device driver runs in kernel mode and they are more scrutinized and more difficult to load and run, because of the trouble they could cause.

The Kernel mode is the privileged mode where the process has unrestricted access to system resources and access to system resources like hardware, memory, etc. A process can access I/O Hardware registers to program it, can execute OS kernel code and access I/O Hardware registers to program it, can execute OS kernel code and access kernel data in Kernel mode. Anything related to process management, IO hardware management, and memory management requires processes to execute in Kernel mode. User mode is the normal mode where the process has limited access.

EXPLANATION OF THE WORKING

Devices are generally represented by their respective files in the /dev directory. Device files are created using the mknod system call.

mknod path type major minor

- Path - Path where the file is to be created.
- Type - ‘c’ or ‘b’ to indicate whether the device is a character device or a block device.
- Major, Minor - the major and Minor number of the device. Major number identifies the device driver. Minor number is used to identify the specific instance of the device (if there is more than one). *ls - l* is used to find these numbers.

Device drivers can be built either as part of the kernel or separately as loadable modules. *lsmod* (or, alternatively, *cat /proc/modules*) prints the contents of the */proc/modules* file, which shows the loadable kernel modules that are currently loaded.

Modules can be loaded using the *insmod* command, by giving the name of the object file (.ko) to be loaded.

insmod module_name

Modules can be unloaded using the *rmmod* command.

rmmod module_name

APIs USED

- kernel_read()
- kernel_write()
- copy_from_user()
- register_chrdev()
- unregister_chrdev()

SUBMITTED BY :

- HAMRAZ K V B190407CS
- ADEEB HADI SEEYAD B190737CS
- ROHIT ROBIN MAMPILLY B190345CS
- NIHAL MUHAMMED ASHARAF B190721CS
- NAZIM JABIR B190524CS