2.4 implementation of ioctl() system call on Ubuntu 18

Before looking the implementation of the systemcall ioctl() first lets see what dose sysem call and ioctl mean;

A system call is a controlled entry point into the kernel of an operating system, enabling a program to request a service from the kernel. This includes operations such as interacting with hardware devices, managing processes, and manipulating files. System calls provide an essential interface between user-space applications and the underlying services offered by the operating system.

examples of system calls include:

- File Operations: open() , read() , write() , close()

- Process Control: fork(), exec(), wait(), exit()

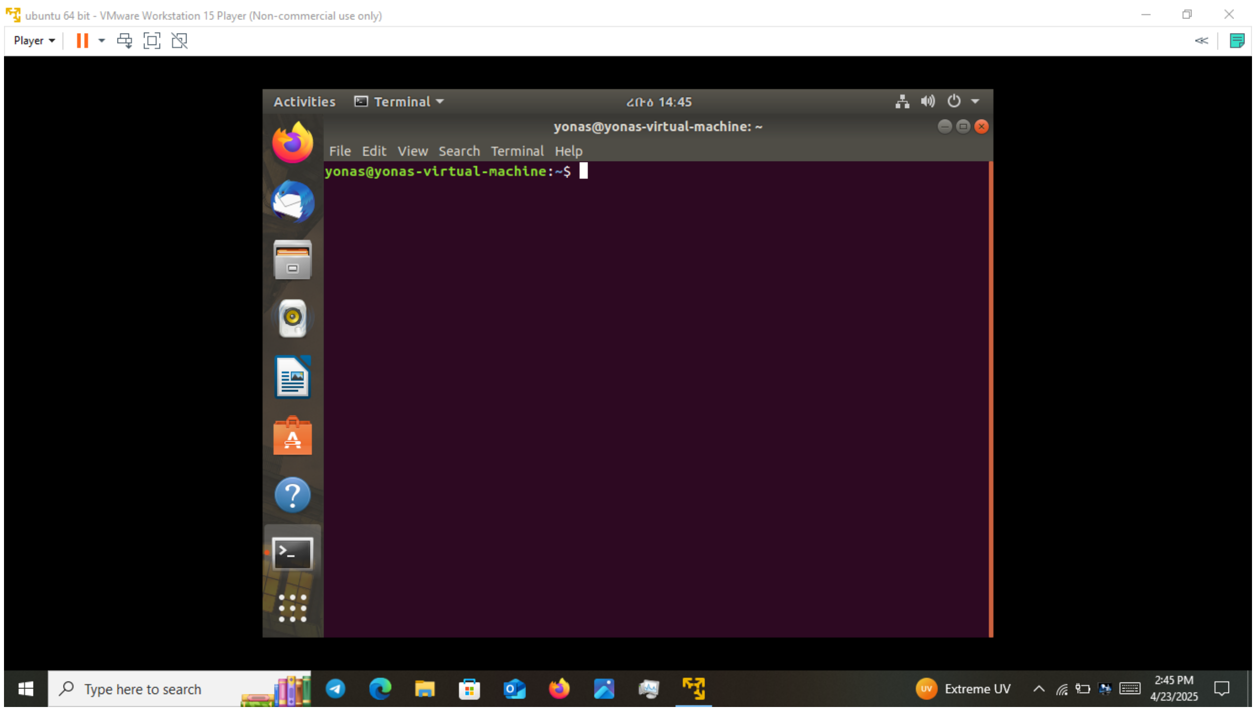
When a system call is made, the program execution is temporarily transferred to the kernel, which performs the requested operation and then returns control to the program. System calls are typically invoked through libraries that wrap around these kernel services, providing a more user-friendly

The **ioctl (input/output control)** is a system call in Unix and Unix-like operating systems used to configure device parameters of special files. It can be used for a variety of operations, such as configuring terminal I/O characteristics, manipulating network interfaces, etc. Here are some key points about `ioctlterface.

To implement ioctl() system call on Ubuntu 18 we can follow the following steps:

1. Open a Terminal:

Openig the terminal on Ubuntu 18

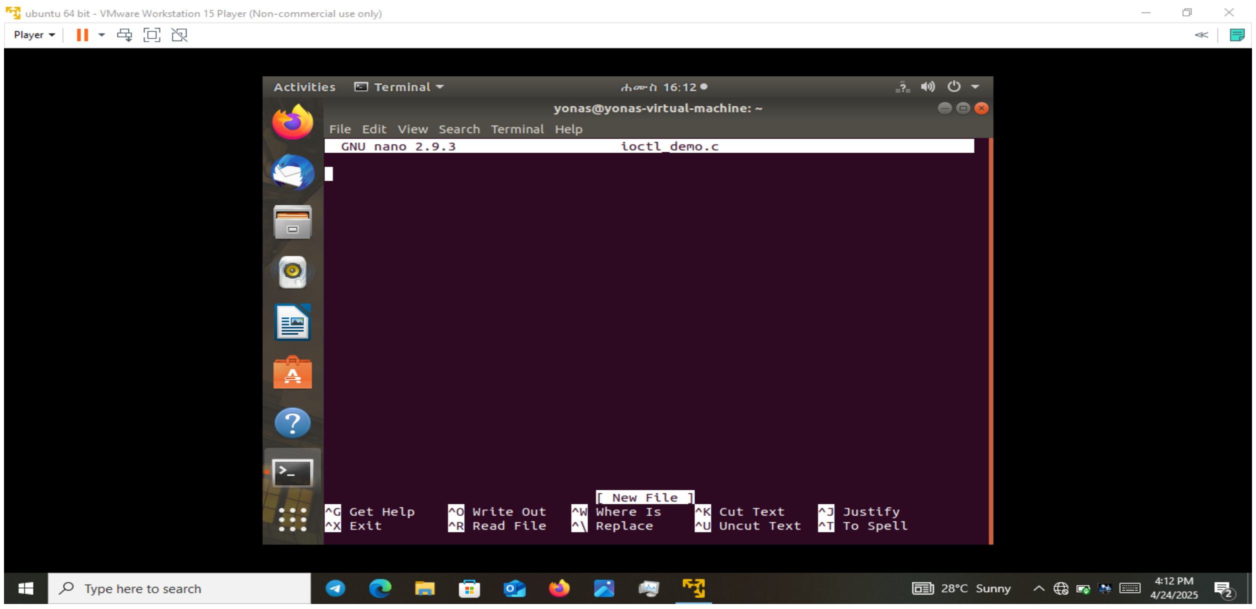

ubuntu 64 bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
ረቡዕ 14:45
yonas@yonas-virtual-machine: ~
File Edit View Search Terminal Help
Type here to search
Extreme UV
2:45 PM
4/23/2025

```
yonas@yonas-virtual-machine:~$
```

Image 3.4 opening a terminal.

2. Create and edit the file in nano:

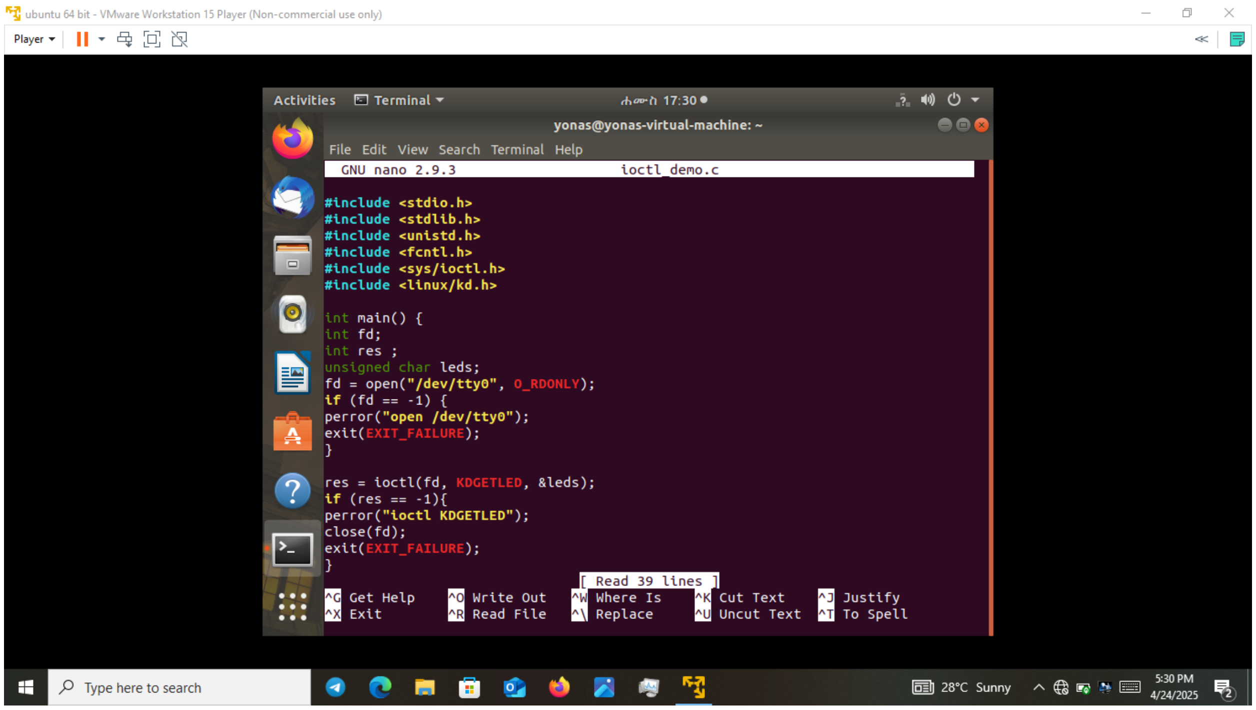
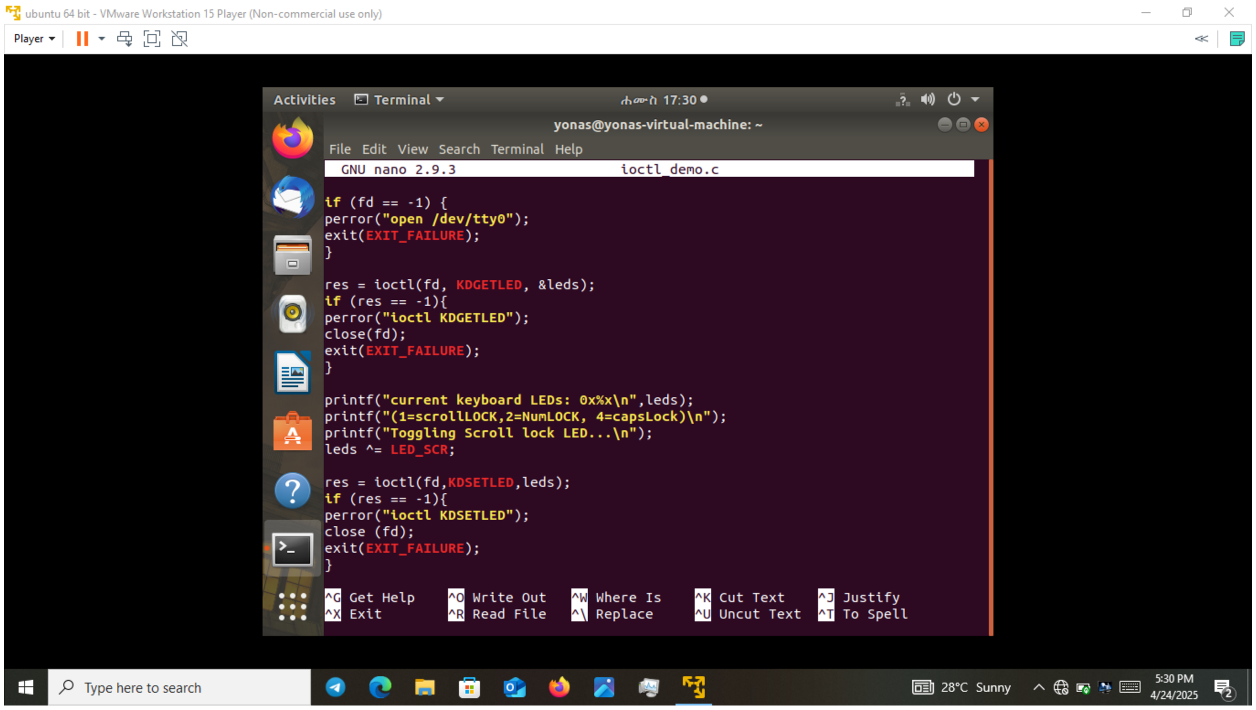
Using the command sudo nano ioctl_demo.c.

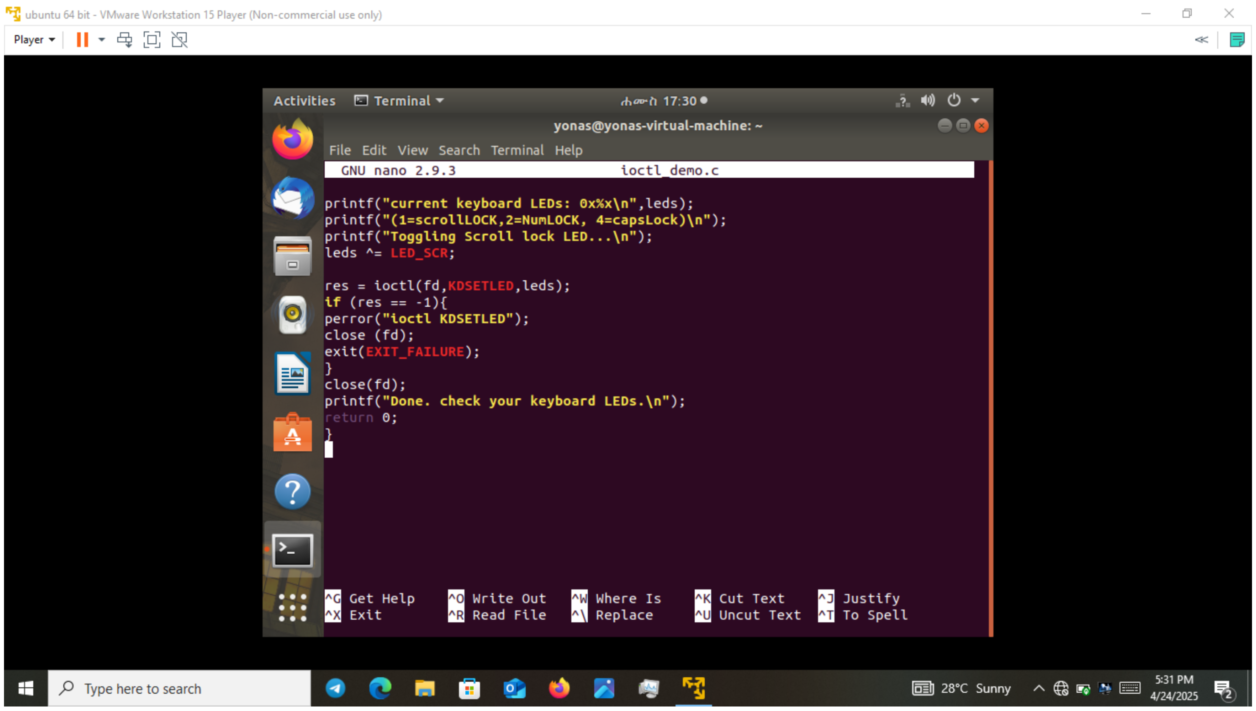

ubuntu 64 bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
ሐሙስ 16:12
yonas@yonas-virtual-machine: ~
File Edit View Search Terminal Help
Type here to search
28°C Sunny
4:12 PM
4/24/2025

```
  GNU nano 2.9.3                   ioctl_demo.c

                                  [ New File ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

Image 3.5 creating nano editor

3 type the code on the nano editor


```
GNU nano 2.9.3                    ioctl_demo.c

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/kd.h>

int main() {
int fd;
int res ;
unsigned char leds;
fd = open("/dev/tty0", O_RDONLY);
if (fd == -1) {
perror("open /dev/tty0");
exit(EXIT_FAILURE);
}

res = ioctl(fd, KDGETLED, &leds);
if (res == -1){
perror("ioctl KDGETLED");
close(fd);
exit(EXIT_FAILURE);
}
                               [ Read 39 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

```
GNU nano 2.9.3                    ioctl_demo.c

if (fd == -1) {
perror("open /dev/tty0");
exit(EXIT_FAILURE);
}

res = ioctl(fd, KDGETLED, &leds);
if (res == -1){
perror("ioctl KDGETLED");
close(fd);
exit(EXIT_FAILURE);
}

printf("current keyboard LEDs: 0x%x\n",leds);
printf("(1=scrollLOCK,2=NumLOCK, 4=capsLock)\n");
printf("Toggling Scroll lock LED...\n");
leds ^= LED_SCR;

res = ioctl(fd,KDSETLED,leds);
if (res == -1){
perror("ioctl KDSETLED");
close (fd);
exit(EXIT_FAILURE);
}

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

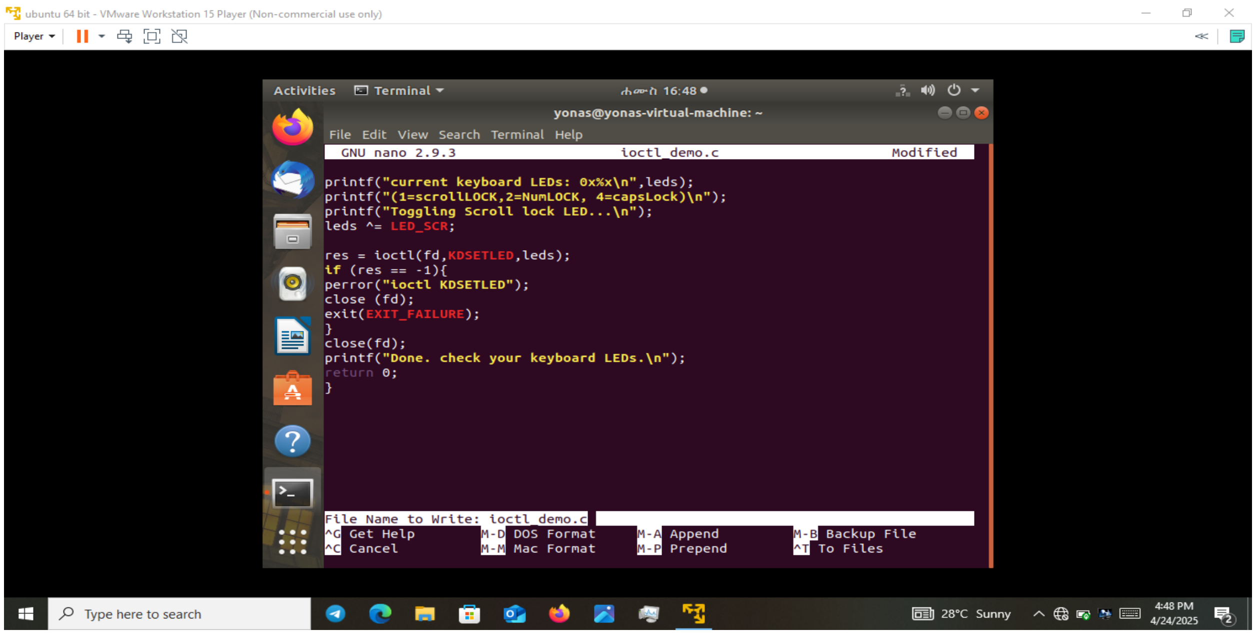

ubuntu 64 bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
17:30
yonas@yonas-virtual-machine: ~
File Edit View Search Terminal Help
Type here to search
28°C Sunny
5:31 PM
4/24/2025

```
  GNU nano 2.9.3                    ioctl_demo.c

printf("current keyboard LEDs: 0x%x\n",leds);
printf("(1=scrollLOCK,2=NumLOCK, 4=capsLock)\n");
printf("Toggling Scroll lock LED...\n");
leds ^= LED_SCR;

res = ioctl(fd,KDSETLED,leds);
if (res == -1){
perror("ioctl KDSETLED");
close (fd);
exit(EXIT_FAILURE);
}
close(fd);
printf("Done. check your keyboard LEDs.\n");
return 0;
}


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

In

Image 3.6 writeing the code on the nano editor

4 Save the file: By Press Ctrl + O (to write out), then Enter to confirm the filename and Press Ctrl + X to exit nano.


ubuntu 64 bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
16:48
yonas@yonas-virtual-machine: ~
File Edit View Search Terminal Help
Type here to search
28°C Sunny
4:48 PM
4/24/2025

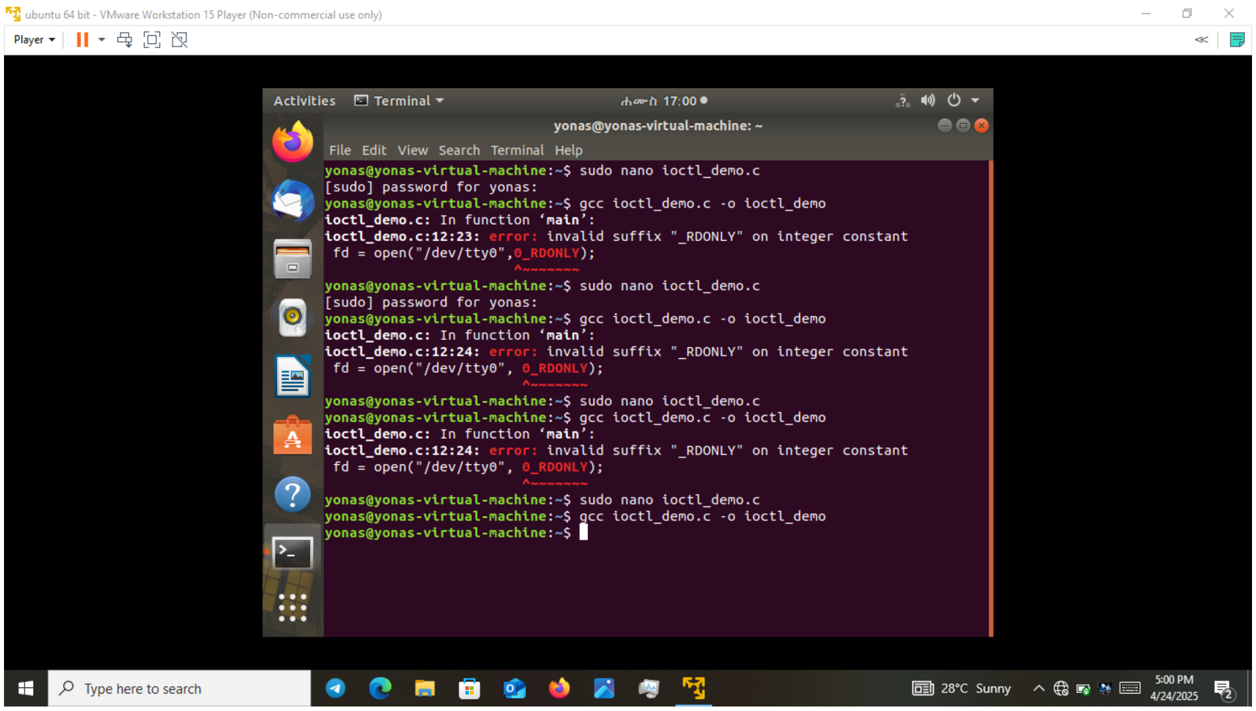
```
  GNU nano 2.9.3                    ioctl_demo.c                        Modified

printf("current keyboard LEDs: 0x%x\n",leds);
printf("(1=scrollLOCK,2=NumLOCK, 4=capsLock)\n");
printf("Toggling Scroll lock LED...\n");
leds ^= LED_SCR;

res = ioctl(fd,KDSETLED,leds);
if (res == -1){
perror("ioctl KDSETLED");
close (fd);
exit(EXIT_FAILURE);
}
close(fd);
printf("Done. check your keyboard LEDs.\n");
return 0;
}

File Name to Write: ioctl_demo.c
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```

Image 3.7 saving the code

Step 5: Compile the Code

1. Compile with gcc.

Compile the code using the comand gcc ioctl_demo.c


ubuntu 64 bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
17:00
yonas@yonas-virtual-machine: ~
File Edit View Search Terminal Help
Type here to search
28°C Sunny
5:00 PM
4/24/2025

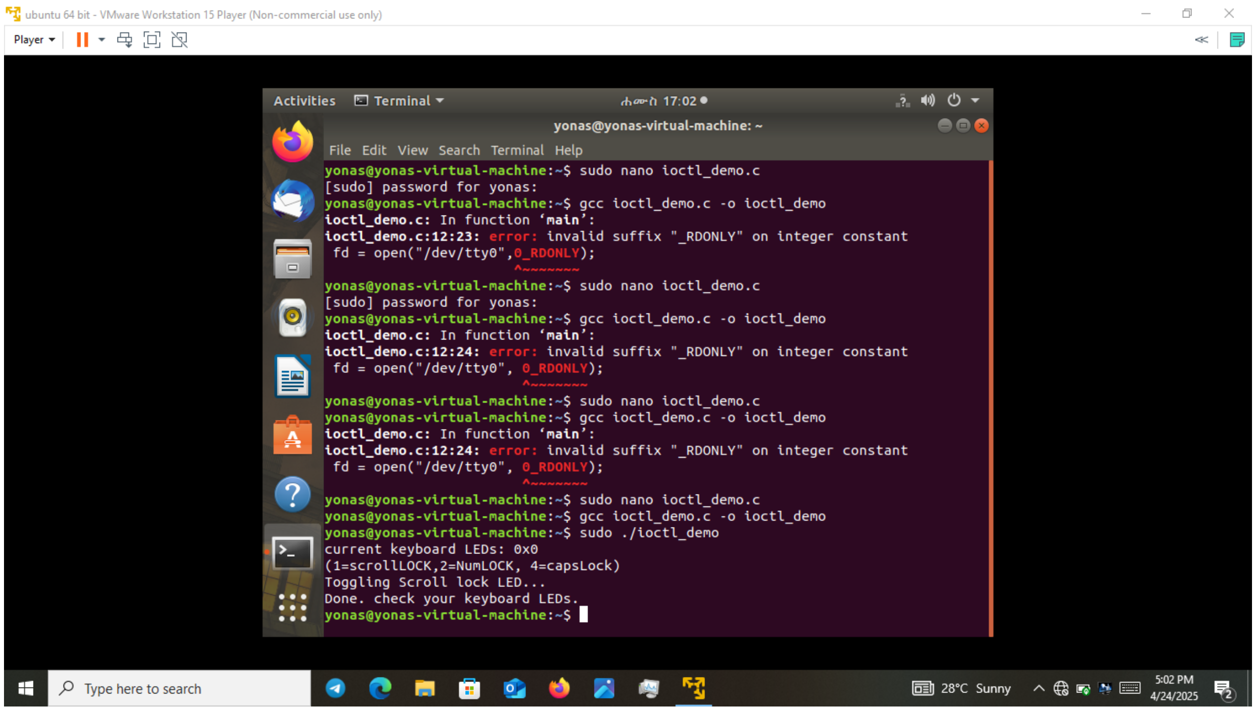
```
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
[sudo] password for yonas:
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
ioctl_demo.c: In function ‘main’:
ioctl_demo.c:12:23: error: invalid suffix "_RDONLY" on integer constant
 fd = open("/dev/tty0",0_RDONLY);
                       ^~~~~~~~
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
[sudo] password for yonas:
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
ioctl_demo.c: In function ‘main’:
ioctl_demo.c:12:24: error: invalid suffix "_RDONLY" on integer constant
 fd = open("/dev/tty0", 0_RDONLY);
                        ^~~~~~~~
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
ioctl_demo.c: In function ‘main’:
ioctl_demo.c:12:24: error: invalid suffix "_RDONLY" on integer constant
 fd = open("/dev/tty0", 0_RDONLY);
                        ^~~~~~~~
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
yonas@yonas-virtual-machine:~$ 
```

Image 3.8 compilling the code

Step 6: Run the Program

1. Execute with sudo :

Excute the code using the command **sudo ./ioctl_demo**


ubuntu 64 bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
yonas@yonas-virtual-machine: ~
File Edit View Search Terminal Help
Type here to search
28°C Sunny
5:02 PM
4/24/2025

```
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
[sudo] password for yonas:
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
ioctl_demo.c: In function ‘main’:
ioctl_demo.c:12:23: error: invalid suffix "_RDONLY" on integer constant
 fd = open("/dev/tty0",0_RDONLY);
                       ^~~~~~~~
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
[sudo] password for yonas:
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
ioctl_demo.c: In function ‘main’:
ioctl_demo.c:12:24: error: invalid suffix "_RDONLY" on integer constant
 fd = open("/dev/tty0", 0_RDONLY);
                        ^~~~~~~~
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
ioctl_demo.c: In function ‘main’:
ioctl_demo.c:12:24: error: invalid suffix "_RDONLY" on integer constant
 fd = open("/dev/tty0", 0_RDONLY);
                        ^~~~~~~~
yonas@yonas-virtual-machine:~$ sudo nano ioctl_demo.c
yonas@yonas-virtual-machine:~$ gcc ioctl_demo.c -o ioctl_demo
yonas@yonas-virtual-machine:~$ sudo ./ioctl_demo
current keyboard LEDs: 0x0
(1=scrollLOCK,2=NumLOCK, 4=capsLock)
Toggling Scroll lock LED...
Done. check your keyboard LEDs.
yonas@yonas-virtual-machine:~$
```

Image 3.9 runing the code and seeing the final output

Finally the final output will be the massages like this:

- 0x0 = All LEDs off.
- 0x2 = NumLock on.
- 0x5 = ScrollLock + CapsLock on.

in summary we can implement and use the system call ioctl() and

other system call using the above procedures with much effort to stregule errors.

// //