

**AROR UNIVERSITY OF ART, ARCHITECTURE,
DESIGN & HERITAGE SUKKUR**

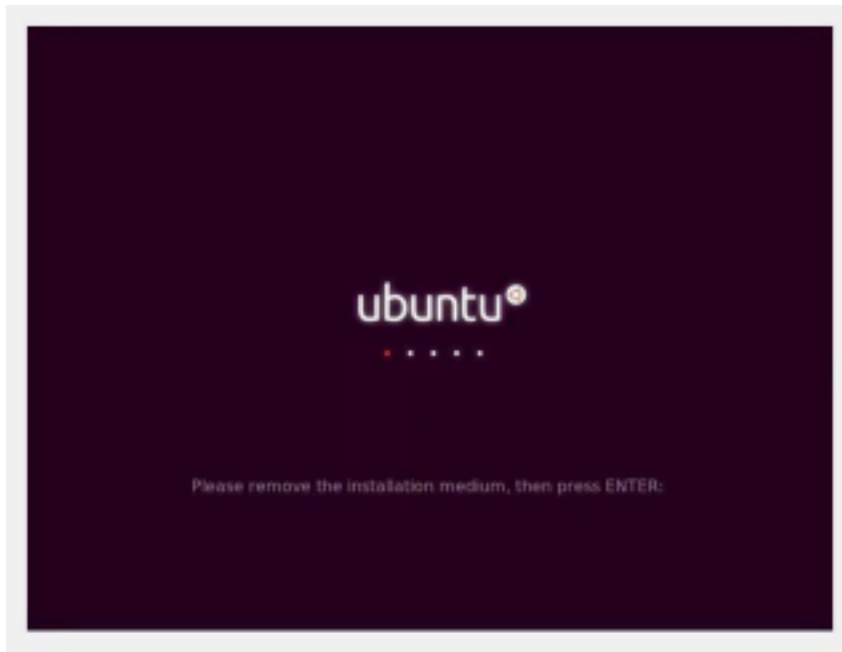


**Operating System
Lab-03**



Compiled by: Aurangzeb Magsi

AROR UNIVERSITY OF ART, ARCHITECTURE, DESIGN & HERITAGE SUKKUR



From now on, just click on the installed Linux virtual machine. You'll be able to use it directly. The installation is a one time only process. You can even delete the Linux ISO that you downloaded earlier.

Linux Shell or “Terminal”

So, basically, a shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output.

To open the terminal, press Ctrl+Alt+T in Ubuntu, or press Alt+F2, type in `gnome-terminal`, and press enter.

Basic Commands

1. **pwd** — When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “pwd” command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash (/). The user directory is usually something like “/home/username”.

```
nayso@Alok-Aspire:~$ pwd
/home/nayso
```

2. **ls** — Use the "ls" command to know what files are in the directory you are in. You can see all the hidden files by using the command "ls -a".

```
nayso@Alok-Aspire:~$ ls
Desktop      itsuserguide.desktop  reset-settings  VCD_Copy
Documents    Music                  School_Resources  Videos
Downloads    Pictures               Students_Works_10
examples.desktop  Public                 Templates
GplatesProject  Qgis Projects         TuxPaint-Pictures
```

3. **Cd** — Use the "cd" command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in "cd Downloads". Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is.

```
nayso@Alok-Aspire:~$ cd Downloads
nayso@Alok-Aspire:~/Downloads$ cd
nayso@Alok-Aspire:~$ cd Raspberry\ Pi
nayso@Alok-Aspire:~/Raspberry Pi$ cd ..
nayso@Alok-Aspire:~$
```

4. **mkdir & rmdir** — Use the mkdir command when you need to create a folder or a directory. For example, if you want to make a directory called "DIY", then you can type "mkdir DIY". Remember, as told before, if you want to create a directory named "DIY Hacking", then you can type "mkdir DIY\ Hacking". Use rmdir to delete a directory. But rmdir can only be used to delete an empty directory. To delete a directory containing files, use rm.

```
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ mkdir DIY
nayso@Alok-Aspire:~/Desktop$ ls
DIY
nayso@Alok-Aspire:~/Desktop$ rmdir DIY
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$
```

5. **rm** - Use the rm command to delete files and directories. Use "rm -r" to delete just the directory. It deletes both the folder and the files it contains when using only the rm command.

```
nayso@Alok-Aspire:~/Desktop$ ls
newer.py  New Folder
nayso@Alok-Aspire:~/Desktop$ rm newer.py
nayso@Alok-Aspire:~/Desktop$ ls
New Folder
nayso@Alok-Aspire:~/Desktop$ rm -r New\ Folder
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$
```

6. **touch** — The touch command is used to create a file. It can be anything, from an empty txt file to an empty zip file. For example, "touch new.txt".

```
nayso@Alok-Aspire:~/Desktop$ ls
nayso@Alok-Aspire:~/Desktop$ touch new.txt
nayso@Alok-Aspire:~/Desktop$ ls
new.txt
```

7. **man & --help** — To know more about a command and how to use it, use the man command. It shows the manual pages of the command. For example, “man cd” shows the manual pages of the cd command. Typing in the command name and the argument helps it show which way the command can be used (e.g., cd –help).

```
TOUCH(1)                                User Commands                                TOUCH(1)

NAME
    touch - change file timestamps

SYNOPSIS
    touch [OPTION]... FILE...

DESCRIPTION
    Update the access and modification times of each FILE to the current
    time.

    A FILE argument that does not exist is created empty, unless -c or -h
    is supplied.

    A FILE argument string of - is handled specially and causes touch to
    change the times of the file associated with standard output.

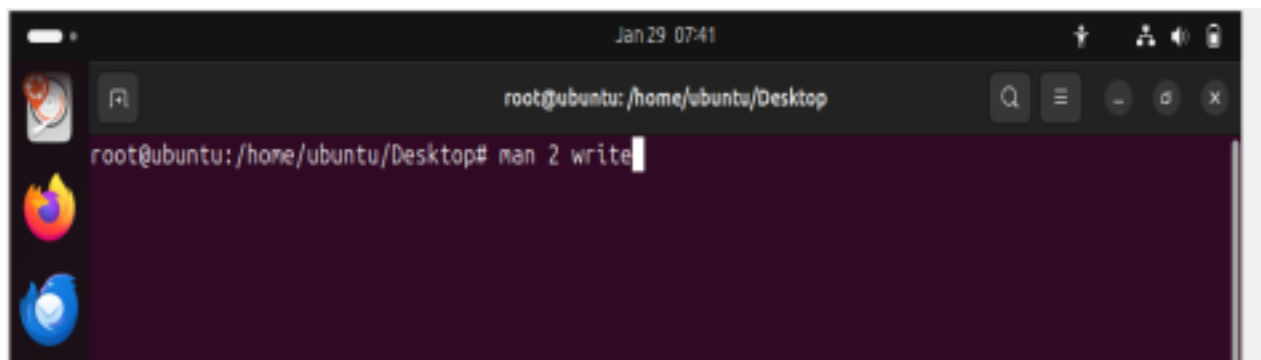
    Mandatory arguments to long options are mandatory for short options
    too.

    -a      change only the access time

Manual page touch(1) line 1 (press h for help or q to quit)
```

Read-Write System Calls – O.S

1. Manual for System Calls:

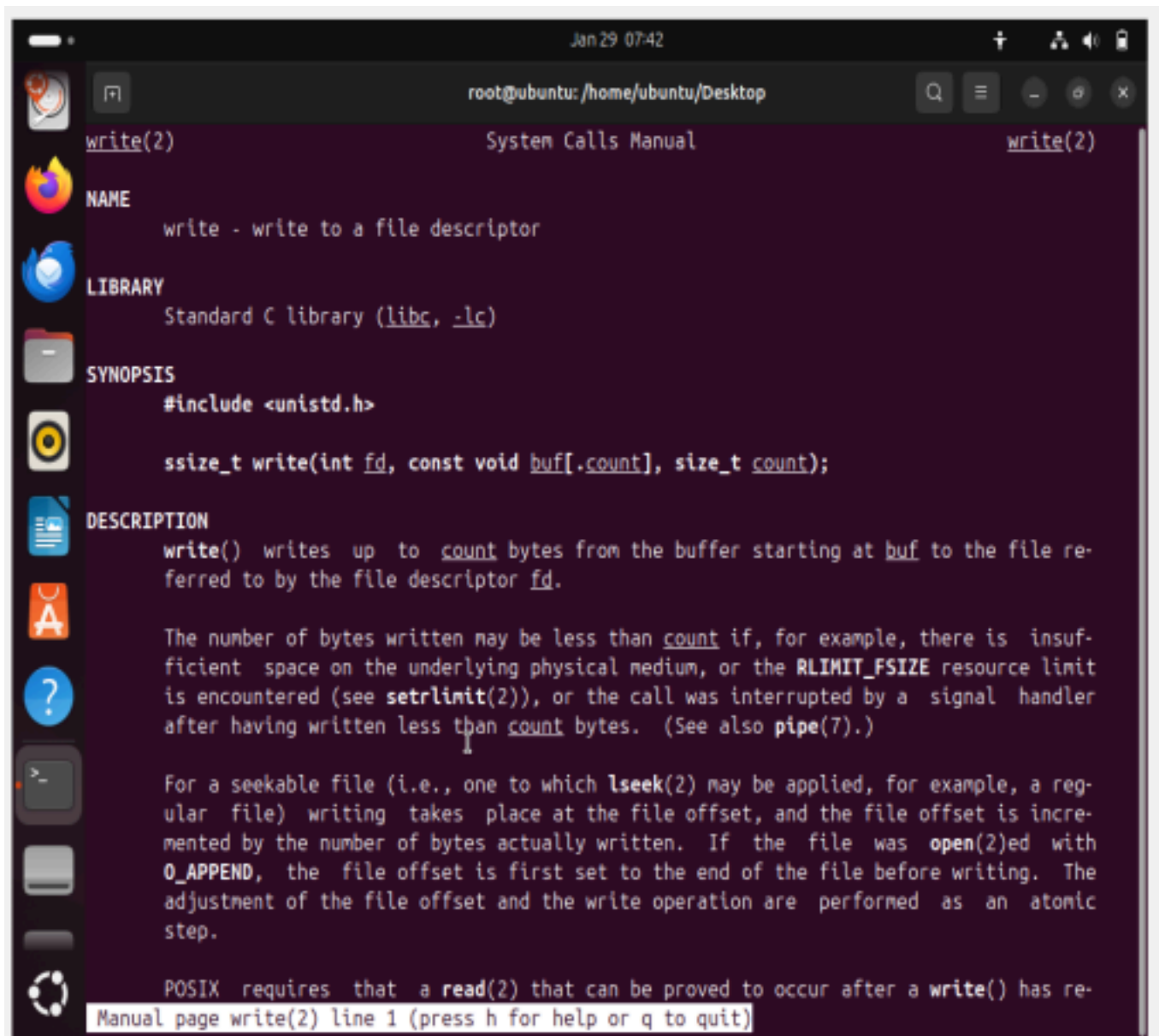


Course Title:

Lab-3

OperatingSystem

Virtual Box Setup Guide & Linux Basic Commands



Jan 29 07:42

root@ubuntu: /home/ubuntu/Desktop

write(2) System Calls Manual write(2)

NAME

write - write to a file descriptor

LIBRARY

Standard C library (`libc`, `-lc`)

SYNOPSIS

```
#include <unistd.h>

ssize_t write(int fd, const void buf[.count], size_t count);
```

DESCRIPTION

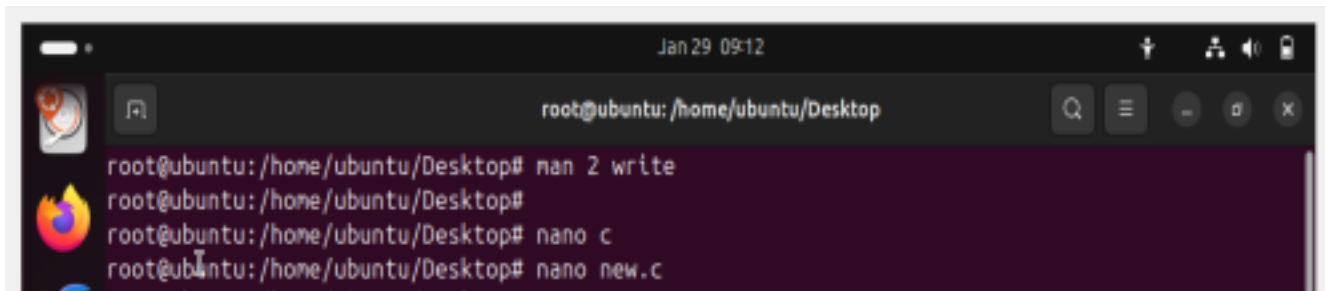
`write()` writes up to `count` bytes from the buffer starting at `buf` to the file referred to by the file descriptor `fd`.

The number of bytes written may be less than `count` if, for example, there is insufficient space on the underlying physical medium, or the `RLIMIT_FSIZE` resource limit is encountered (see `setrlimit(2)`), or the call was interrupted by a signal handler after having written less than `count` bytes. (See also `pipe(7)`.)

For a seekable file (i.e., one to which `lseek(2)` may be applied, for example, a regular file) writing takes place at the file offset, and the file offset is incremented by the number of bytes actually written. If the file was `open(2)`ed with `O_APPEND`, the file offset is first set to the end of the file before writing. The adjustment of the file offset and the write operation are performed as an atomic step.

POSIX requires that a `read(2)` that can be proved to occur after a `write()` has re-

Manual page write(2) line 1 (press h for help or q to quit)



Jan 29 09:12

root@ubuntu: /home/ubuntu/Desktop

```
root@ubuntu:/home/ubuntu/Desktop# nan 2 write
root@ubuntu:/home/ubuntu/Desktop#
root@ubuntu:/home/ubuntu/Desktop# nano c
root@ubuntu:/home/ubuntu/Desktop# nano new.c
```

Lab-1 Exercises

Course Title:

Lab-3

OperatingSystem

Virtual Box Setup Guide & Linux Basic Commands

Jan 29 09:13

root@ubuntu: /home/ubuntu/Desktop

GNU nano 7.2

new.c

```
#include<unistd.h>
int main()
{
write(1, "Hello", 5);
}
```