NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY
GREATER NOIDA
(An Autonomous Institute)
School of Computer Sciences & Engineering in Emerging Technologies

Lab File

Of

Operating System Lab

(ACSE-0453B)

(4th Semester)

Session (2023 – 2024)

Affiliated to Dr. A.P.J Abdul Kalam Technical University, Uttar Pradesh, Lucknow.

Submitted By:

Priyanshu Singh

2201331550094

Iot-B

Submitted to:

Ms.Savita Yadav

Index

B. TECH. SECOND YEAR (CSE(IoT)) 4th Sem			
Course Code	**ACSE0453B**	**L T P**	**Credit**
Course Title	**Operating System Lab**	**0 0 2**	**1**

List of Experiments:

Sr. No.	Name of Experiment	Date	Sign
1	Lab1: Install an Operating System on the Raspberry Pi.		
2	Lab 2:Execute Various types of Linux Commands (Miscellaneous, File oriented, Directory oriented)		
3	Lab 3: Shell Programming Write a shell program, which accepts the name of a file from standard input and perform the following test on it: i. File readable ii. File writable iii. Both readable and writable		
4	Lab 4: Implement CPU Scheduling Algorithms: 1. FCFS 2. SJF 3. PRIORITY		
5	Lab 5: 4. Round Robin 5. Multi-level Queue Scheduling		
6	Lab 6: Implementation of Banker's algorithm for the purpose of Deadlock Avoidance.		
7	Lab 7: Write a program to simulate the following contiguous memory allocation techniques: a) First fit b) Best fit c) Worst Fit		
8	Lab 8: a) Write a Program for implementation of Contiguous memory fixed partition technique. b) Write a program for implementation of Contiguous memory variable partition technique.		
9	Lab 9: Write a program to simulate page replacement algorithms: a) FIFO b) LRU c) Optimal		

10	Lab 10: Write a program to simulate Disk Scheduling Algorithms: a) FCFS b) SSTF		
11	Lab11: Disk Scheduling Techniques c) SCAN & C-SCAN d) Look & C-LOOK		
12	Lab12: Process Synchronization Write a program to simulate Producer Consumer problem		

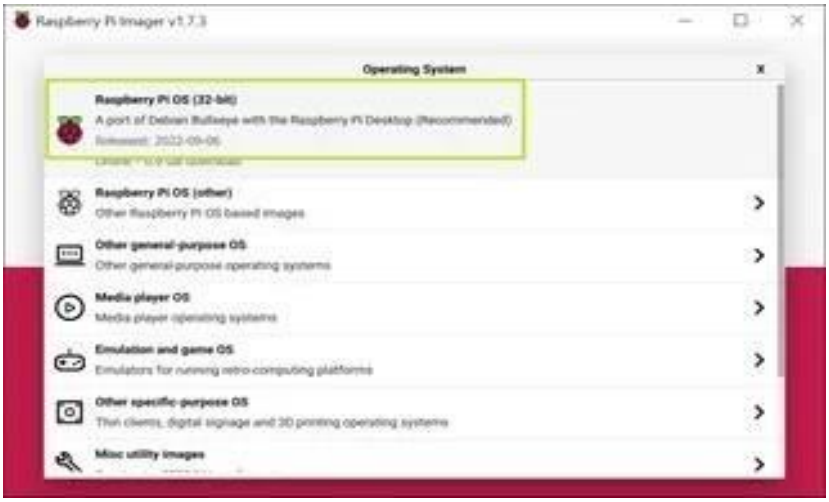
EXPERIMENT NO-1

Aim: : Install an Operating System on the Raspberry Pi.

Steps to install Operating System on a Raspberry Pi are :-

1. **Insert a microSD card / reader** into your computer.
2. **Download and install** the official Raspberry Pi Imager (https://www.raspberrypi.org/downloads/)**.** Available for Windows, macOS or Linux, this app will both download and install the latest Raspberry Pi OS.
3. **Click Choose OS.**

4. **Select Raspberry Pi OS (32-bit)** from the OS menu (there are other choices, but for most uses, 32-bit is the best).



4. **Click Choose storage** and **pick the SD card** you're using.

5. **Click the settings button** or hit CTRL + SHIFT + X to enter settings.

6. **Fill in settings fields** as follows and then **hit Save**. All of these fields are technically optional, but highly recommended so that can get your Raspberry Pi set up and online as soon as you boot it. If you don't set a username and password here, you'll have to go through a setup wizard that asks you to create them on first boot.

- **Set hostname**: the name of your Pi. It could be "raspberrypi" or anything you like.
- **Enable SSH**: Allow SSH connections to the Pi. Recommended.
- **Use password authentication / public key:** method of logging in via SSH
- **Set username and password:** Pick the username and password you'll use for the Pi
- **Configure wireless LAN: s**et the SSID and password of Wi-FI network

- **Wireless LAN country:** If you're setting up Wi-Fi, you must choose this.
- **Set locale settings:** Configure keyboard layout and time zone (probably chosen correctly by default)

7. **Click Write.** The app will now take a few minutes to download the OS and write to your card.

Booting Your Raspberry Pi for the First Time

After you're done writing the Raspberry Pi OS to a microSD card, it's time for the moment of truth.

1. **Insert the microSD card** into the Raspberry Pi.
2. **Connect the Raspberry Pi** to a monitor, keyboard and mouse.
3. **Connect an Ethernet cable** if you plan to use wired Internet.
4. **Plug the Pi in** to power it on.

If you had used the Raspberry Pi Imager settings to create a username and password, you'll be able to go straight into the desktop environment, but if not, you will get a setup wizard.

Using the Raspberry Pi First-Time Setup Wizard

If you chose a username and password in Raspberry Pi Imager settings, before writing the microSD card, you will get the desktop on first boot. But, if you did not, you'll be prompted to create a username and password and enter all the network credentials by a setup wizard on first boot. If that happens, follow these steps to finish setting up your Raspberry Pi.

1. **Click Next** on the dialog box.

2. **Set your country and language** and click Next. The default choices may already be the correct ones.

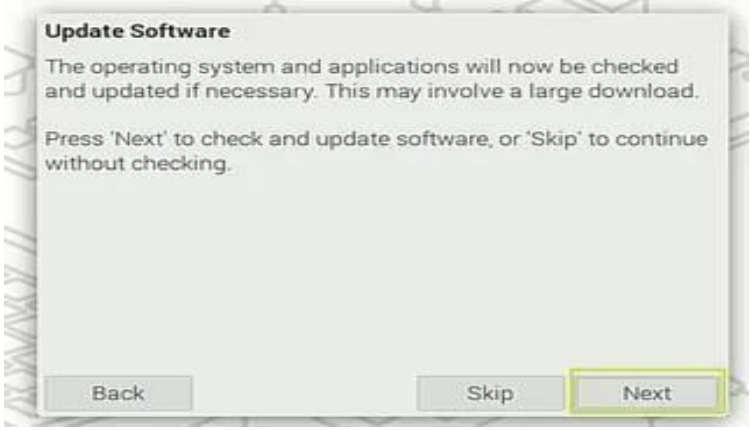
3. **Enter a username and password** you wish to use for your primary login. **Click Next**.

4. **Toggle Reduce the size of the desktop" to on** if the borders of the desktop are cut off. Otherwise, just **click Next**.

5. **Select the appropriate Wi-Fi network** on the screen after, provided that you are connecting via Wi-Fi. If you don't have Wi-Fi or are using Ethernet, you can skip this.

6. **Enter your Wi-Fi password** (unless you were using Ethernet and skipped).

7. **Click Next** when prompted to Update Software. This will only work when you are connected to the Internet, and it can take several minutes. If you are not connected to the Internet, click Skip.



8. **Click Restart.**

If you wish to change these settings later, you can find the region and password settings, along with many other options, by clicking on the Pi icon in the upper left corner of the screen and navigating to Preferences ->

Raspberry Pi Configuration. You can configure Wi-Fi by clicking on the Wi-Fi / network icon on the taskbar.

Fig.1

EXPERIMENT -2

AIM: - Execute Various types of Linux Commands (Miscellaneous, File oriented, Directory oriented)

Command: - (A).File Oriented:--

- PWD –
- The "pwd" command prints the full name (the full path) of current/working directory. By default, right after ssh-ing to a Linux machine you would find yourself in your home directory, usually /home/<username>.

```
theia@theia-priyans6388:/home/project$ pwd
/home/project
theia@theia-priyans6388:/home/project$
```

- MKDIR –
- The mkdir command in Linux/Unix is a command-line utility that allows users to create new directories. mkdir stands for "make directory.

```
theia@theia-priyans6388:/home/project$ mkdir priyanshu
theia@theia-priyans6388:/home/project$ ls
priyanshu
```

- RMDIR-
- The rmdir command removes the directory, specified by the Directory parameter, from the system. The directory must be empty before you can remove it, and you must have write permission in its parent directory.

```
theia@theia-priyans6388:/home/project$ rmdir priyanshu
theia@theia-priyans6388:/home/project$ ls
theia@theia-priyans6388:/home/project$
```

- LS-
- The ls command writes to standard output the contents of each specified Directory or the name of each specified File, along with any other information you ask for with the flags

```
theia@theia-priyans6388:/home/project$ ls Priyanshu
Priyanshu
theia@theia-priyans6388:/home/project$
```

- CD-
- The cd command in Linux stands for change directory. It is used to change the current directory of the terminal. The terminal, by default, opens the home directory.

```
theia@theia-priyans6388:/home/project$ cd ~
theia@theia-priyans6388:~$ cd ..
theia@theia-priyans6388:/home$ cd /
theia@theia-priyans6388:/$ cd bin
theia@theia-priyans6388:/bin$ cd ../home/theia
theia@theia-priyans6388:~$ cd ~
theia@theia-priyans6388:~$ cd ../project
theia@theia-priyans6388:/home/project$
```

File oriented :-

1.TOUCH-

The touch command in Linux is used to create a new empty file and to change the timestamps of existing files. You can use it with the syntax: touch myfile. txt .

```
theia@theia-priyans6388:/home/project$ touch myfile.txt
theia@theia-priyans6388:/home/project$ ls
Priyanshu  myfile.txt
theia@theia-priyans6388:/home/project$
```

2. RM-

rm is a general command in Unix and other Unix-like systems. It is used to delete objects like symbolic links, directories, and computer files from the file systems.

```
theia@theia-priyans6388:/home/project$ touch myfile.txt
theia@theia-priyans6388:/home/project$ ls
Priyanshu  myfile.txt
theia@theia-priyans6388:/home/project$ rm myfile.txt
theia@theia-priyans6388:/home/project$ ls
Priyanshu
theia@theia-priyans6388:/home/project$
```

3. CAT-

The cat command on Linux concatenates files together. It's often used to concatenate one file to nothing to print the single file's contents to the terminal. This is a quick way to preview the contents of a text file without having to open the file in a large application.

```
theia@theia-priyans6388:/home/project$ ls Priyanshu
Priyanshu
theia@theia-priyans6388:/home/project$ cat Priyanshu
I love javatheia@theia-priyans6388:/home/project$
```

4. CP-

Use the cp command to create a copy of the contents of the file or directory specified by the Source File or Source Directory parameters into the file or directory specified by the Target File or Target Directory parameters.

```
theia@theia-priyans6388:/home/project$ ls
Priyanshu
theia@theia-priyans6388:/home/project$ cp Priyanshu Pratyush
theia@theia-priyans6388:/home/project$ ls
Pratyush  Priyanshu
theia@theia-priyans6388:/home/project$
```

5.MV-

The mv command moves files and directories from one directory to another or renames a file or directory. If you move a file or directory to a new directory, it retains the base file name. When you move a file, all links to other files remain intact, except when you move it to a different file system.

```
theia@theia-priyans6388:/home/project$ ls
Pratyush  Priyanshu
theia@theia-priyans6388:/home/project$ mv Pratyush Rajat
theia@theia-priyans6388:/home/project$ ls
Priyanshu  Rajat
theia@theia-priyans6388:/home/project$ s
```

EXPERIMENT NO-3

AIM: Shell Programming

Write a shell program, which accepts the name of a file from standard input and perform the following test on it: i. File readable ii. File writable

iii. Both readable and writable

1. check_readable.sh:

```
#!/bin/bash

# Read the filename from standard input
read -p "Enter filename: " filename

# Check if file exists if [ ! -f
"$filename" ]; then
  echo "Error: File '$filename' does not exist."   exit
1 fi

# Check read permission if [ -r
"$filename" ]; then   echo "$filename
is readable." else
  echo "$filename is not readable." fi
```

Running check_readable.sh:

Enter filename: myfile.txt

Assuming myfile.txt exists and is readable:
myfile.txt is readable.

Assuming myfile.txt doesn't exist:

Enter filename: myfile.txt

Error: File 'myfile.txt' does not exist.

2. check_writable.sh:

```
#!/bin/bash

# Read the filename from standard input
read -p "Enter filename: " filename

# Check if file exists if [ ! -f
"$filename" ]; then
  echo "Error: File '$filename' does not exist."   exit
1 fi

# Check write permission if [ -w
"$filename" ]; then   echo "$filename
is writable." else
  echo "$filename is not writable." fi
```

Running check_writable.sh:

Enter filename: myfile.txt

Assuming myfile.txt exists and is writable:
myfile.txt is writable.

Assuming myfile.txt doesn't exist (same output as check_readable.sh for this case):

Enter filename: myfile.txt
Error: File 'myfile.txt' does not exist.

3. check_read_write.sh:

```
#!/bin/bash

# Read the filename from standard input
read -p "Enter filename: " filename

# Check if file exists if [ ! -f
"$filename" ]; then
```

```
  echo "Error: File '$filename' does not exist."   exit
1 fi


# Check read permission if [ -r
"$filename" ]; then   echo "$filename
is readable." else

  echo "$filename is not readable." fi


# Check write permission if [ -w
"$filename" ]; then   echo "$filename
is writable." else

  echo "$filename is not writable." fi


# Check for both read and write permissions if [ -r
"$filename" -a -w "$filename" ]; then

  echo "$filename is both readable and writable."

fi
```

Running check_read_write.sh:

Enter filename: myfile.txt

Assuming myfile.txt exists, is readable, and writable:

myfile.txt is readable. myfile.txt is writable.
myfile.txt is both readable and writable.

Assuming myfile.txt doesn't exist (same output as check_readable.sh and check_writable.sh for this case):

Enter filename: myfile.txt
Error: File 'myfile.txt' does not exist.

EXPERIMENT NO-4

Aim: Implement CPU Scheduling Algorithms:

1. FCFS
2. SJF
3. PRIORITY