**Name :- Purval Madhukar Bhude**

**Roll No. :- S20230010193**

**Assignment No. :- 05**

**Date :- 11-10-2023**

**Course Name :- OCW**

# Q1.Explain they different types of operating system?

**Ans. These are the following types of operating system mention below.**

1. **Multiprogramming Operating System:**
   **It increases CPU utilization by organizing jobs so that the CPU always has one job to execute. This type of OS is used to execute more than one jobs simultaneously by single processor. It is use the mechanism of job scheduling and CPU scheduling.**

2. **Batch Operating System:**
   **Batch Operating systems process jobs in batches, meaning that a group of jobs are submitted to the system and then executed one at a time. It is typically used in mainframe computers and for high-volume processing tasks such as scientific computing and payroll processing. The users of this type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches.**

3. **Time-Sharing Operating Systems:**
   **Time-sharing operating systems allow multiple users to access and use the same computer system simultaneously. This is done by dividing the computer's time into slices and allocating each slice to a different user. Time-sharing operating systems are commonly used in personal computers and workstations. Each task is given some time to**

execute so that all the tasks work smoothly. These system is also known as multi-tasking systems. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.

4. Real-Time Operating System:
   Real-time operating systems are designed to respond to events within a certain amount of time. This is important for applications where even a small delay can have serious consequences, such as industrial control systems and medical devices. The time interval required to process and respond to inputs is very small. This time interval is called Response Time. Real-Time operating system are used when there are time requirements that are very strict like missile system, air traffic control system, robots, etc.

5. Embedded Operating System:
   Embedded operating systems are designed for use in embedded devices, such as smartphones, smart TVs, and routers. Embedded operating systems are typically smaller and more efficient than general-purpose operating systems, and they are often optimized for specific tasks. The most popular embedded operating systems for consumer products such as PDAs includes-
   →Windows XP Embedded
   →Windows CE .NET
   →Palm OS

→Symbian

6. Network Operating System:
   These systems runs on a server and provide the capability to manage data, users, security, applications, and other networking functions. This is also known as tightly coupled systems. These types of operating systems allow shared access of files, printers, security over a small private network.

# Q2. Explain the functions of operating system?

**Ans. The functions of operating system can be broadly classified into the following categories:**

- **Process management:**

  **The OS is responsible for managing the execution of multiple processes on the computer. This includes tasks such as scheduling processes, allocating resources to processes, and terminating processes.**

  **The process needs certain resources, including CPU time, Memory, files, and I/O devices to accomplish its task. If process are dependent, than they may tey to share same resources. Thus task of process synchronization comes to the picture. If process are independent, than a due care needs to be taken to avoid their overlapping in memory area. Based on priority, it is important to allow more important processes to execute first than others.**

- **Memory management:**

  **The OS is responsible for managing the computer's memory. This includes tasks such as allocating memory to processes, swapping memory to disk, and freeing up memory when it is no longer needed. It allocate and de-allocate memory space as needed. It decide which processes to load when memory space become available. It keep trach of which parts of memory are currently being used and by whom.**

- **File management:**

**The OS is responsible for managing the computer's files and directories. This includes tasks such as creating, deleting, and renaming files, organizing files into directories, and providing access to files. File systems provide the convention for the encoding, storage and management of data on a storage device such as hard disk.**

- **Device management:**

  **The OS is responsible for managing the computer's hardware devices. This includes tasks such as initializing devices, allocating devices to processes, and handling device interrupts. Device controllers are components on the motherboard that act as an interface between the CPU and the actual device. Device drivers are the operating system software components that interact with the devices controllers.**

- **Security:**

  **The OS is responsible for protecting the computer from unauthorized access and malicious software. This includes tasks such as authenticating users, managing permissions, and detecting and removing malware. It also prevents unauthorized access to programs and user data by assigning access right permission to file and directories. The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other.**

# Q3. Try the following commonds in terminal.

## Ans.

1. **Touch**
   **Create a new file :- $ touch myfile.txt**

```
localhost:~# touch myfile.txt

localhost:~# ls
bench.py    dir1        hello.c     hello.js    myfile.txt  readm
e.txt
```

2. **ls**
   **list all files in the current directory :- $ls**
   **list all files in the current directory, including hidden files :- $ls -a**
   **list all files in the current directory, in long format:- $ls -r**

```
localhost:~# ls
bench.py    dir1        hello.c     hello.js    myfile.txt  readm
e.txt
```

3. **rm**
   **Remove the file :- $rm myfile.txt**

```
localhost:~# rm myfile.txt
localhost:~# ls
bench.py    dir1        hello.c     hello.js    readme.txt
```

4. **cat**
   **display content of the file :- $cat myfile.txt**

```
localhost:~# touch myfile.txt
localhost:~# echo "Purval" > myfile.txt
localhost:~# cat myfile.txt
Purval
```

5. cp
   copy the file to new file :- $cp myfile.txt mycopy.txt

```
localhost:~# cp myfile.txt mycopy.txt
localhost:~# ls
bench.py    hello.c     mycopy.txt  readme.txt
dir1        hello.js    myfile.txt
```

6. mkdir
   create a new directory :- $mkdir mydirectory

```
localhost:~# mkdir dir1
localhost:~# ls
bench.py    dir1        hello.c     hello.js    readme.txt
```

7. pwd
   display the current working directory :- $pwd

```
localhost:~# pwd
/root
```

8. cd
   It is used to change directory :- $cd mydirectory

```
localhost:~# cd dir1
localhost:~/dir1# pwd
/root/dir1
```

9. man
   it is used to display the user manual of any command that we can run on the terminal :- $man -f ls

10. echo
    it is used for displaying lines of text or string which are passed as arguments on the commond line :- $echo “Purval”

11. exit

it used to test for the existence of any record in a subquery :-

```
SELECT lname, fname
FROM Customers
WHERE NOT EXISTS (SELECT *
        FROM Orders
        WHERE Customers.customer_id =
Orders.c_id);
```

12. sort

sorts the contents of a file, in numeric or alphabetic order, and prints the results to standard output :-

```
$sort file.txt
```

13. wc

find out **number of lines, word count, byte and characters count** in the files specified in the file arguments :-

```
$wc file.txt
```

14. head

print the top N number of data of the given input :-

```
$head file.txt
```

15. tail

It is the complementary of head command. print the last N number of data of the given input :-

```
$tail file.txt
```

16. cmp

It is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not :- $cmp file1.txt file2.txt

17. diff

This command is used to display the differences in the files by comparing the files line by line :- $diff file1.txt file2.txt

## Q4

## Ans.

## Below is the answer of the above question

```
localhost:~# pwd
/root
localhost:~# mkdir OCW_Lab5
localhost:~# pwd
/root
localhost:~# cd OCW_Lab5
localhost:~/OCW_Lab5# pwd
/root/OCW_Lab5
localhost:~/OCW_Lab5# touch Countries.txt
localhost:~/OCW_Lab5# echo "india USA Germany Finland Iceland Aus
tria Canada Russia Ukraine taiwan" > Contries.txt
localhost:~/OCW_Lab5# cat Contries.txt
india USA Germany Finland Iceland Austria Canada Russia Ukraine t
aiwan
localhost:~/OCW_Lab5# touch cities.txt
localhost:~/OCW_Lab5# echo "Mumbai Pune Newyork Lasvegas berlin h
amburg helsinki oulu akranes reykjavik vienna innsburck toronto m
ontreal omsk moscow kyiv kharkiv taipei tainan" > cities.txt
localhost:~/OCW_Lab5# cat cities.txt
Mumbai Pune Newyork Lasvegas berlin hamburg helsinki oulu akranes
reykjavik vienna innsburck toronto montreal omsk moscow kyiv khar
kiv taipei tainan
localhost:~/OCW_Lab5# cat cities.txt >> Contries.txt && cat Contr
ies.txt
india USA Germany Finland Iceland Austria Canada Russia Ukraine t
aiwan
Mumbai Pune Newyork Lasvegas berlin hamburg helsinki oulu akranes
reykjavik vienna innsburck toronto montreal omsk moscow kyiv khar
kiv taipei tainan
```

```
localhost:~/OCW_Lab5# mv Countries.txt newdata.txt

localhost:~/OCW_Lab5# ls
Contries.txt  cities.txt    newdata.txt
```