|  |
| --- |
| ***Linux Architecture with diagram:*** |

|  |
| --- |
| ***5 State process model with diagram:*** |

|  |
| --- |
| ***Functions Of OS:***   1. User Interface ⊇This interface can take several forms. 1. DTrace command line interface - which uses text commands and a method for entering them (say, a program to allow entering and editing of commands). 2. Batch interface - in which commands and directives to control those commands are entered into files, and those files are executed. 3. Graphical user interface - the interface with a pointing device to direct I/0, choose from menus, and make selections and a keyboard to enter text. ⊇ Some systems provide two or all three of these variations 2. 2. Program Execution ⊇ The system must be able to load a program into memory and to run that program.The program must be able to end its execution, either normally or abnormally (indicating error). 3. 3. I/O Operations ⊇ A running program may require I/O, which may involve a file or an I/O device. ⊇ For efficiency and protection, users usually cannot control I/O devices directly. ⊇ Therefore, the operating system must provide a means to do I/O 4. 4. File Management ⊇The operating system manages resource allocation and de-allocation. ⊇It specifies which process receives the file and for how long. ⊇It also keeps track of information, location, uses, status, and so on. ⊇These groupings of resources are referred to as file systems. ⊇The files on a system are stored in different directories. ⊇The OS: ♣ Keeps records of the status and locations of files. ♣ Allocates and deallocates resources. ♣ Decides who gets the resources. 5. 5. Resource Allocation ⊇When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them. ⊇The operating system manages many different types of resources, such as CPU cycles, main memory, file storage, I/O devices 6. 6. Communication ⊇In multitasking environment, the processes need to communicate with each other and to exchange their information. ⊇Operating system performs the communication among various types of processes in the form of shared memory or message passing, in which packets of information in predefined formats are moved between processes by the operating system 7. 7. Error detection and response ⊇An error may occur in CPU, in I/O devices or in the memory hardware. ⊇Following are the major activities of an operating system with respect to error handling − ♣ The OS constantly checks for possible errors. ♣ The OS takes an appropriate action to ensure correct and consistent computing. 8. 8. Accounting ⊇We want to keep track of which users use how much and what kinds of computer resources. ⊇This record keeping may be used for accounting (so that users can be billed) or simply for accumulating usage statistics. 9. 9. Protection & Security ⊇Protection involves ensuring that all access to system resources is controlled. ⊇Security of the system from outsiders is also important. ⊇Such security starts with requiring each user to authenticate himself or herself to the system, usually by means of a password, to gain access to system resources. ⊇If a system is to be protected and secure, precautions must be instituted throughout it. 10. 10. Virtual Machine Manager ⊇ The fundamental idea behind a virtual machine is to abstract the hardware of a single computer (the CPU, memory, disk drives, network interface cards, and so forth) into several different execution environments, thereby creating the illusion that each separate execution environment is running its own private computer. |

|  |
| --- |
| ***System Call and its Types:***  A system call is a way for programs to interact with the operating system. } A system call is a mechanism that provides the interface between a process and the operating system. } A computer program makes a system call when it makes a request to the operating system's kernel. } It is a programmatic method in which a computer program requests a service from the kernel of the OS. } System call provides the services of the operating system to the user programs via Application Program Interface(API). } System calls are the only entry points for the kernel system   1. Process Control: This system calls perform the task of process creation, process termination, etc. ⊇ Functions: ♣ End and abort ♣ Load and execute ♣ Create process and terminate process ♣ Wait and signed event ♣ Allocate and free memory } 2. File Management: File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc. ⊇ Functions: ♣ Create a file ♣ Delete file ♣ Open and close file ♣ Read, write and reposition ♣ Get and set file attributes 3. } Device Management: Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc. ⊇ Functions ♣ Request and release device ♣ Logically attach/ detach devices ♣ Get and Set device attributes } 4. Information Maintenance: It handles information and its transfer between the OS and user program. ⊇ Functions: ♣ Get or set time and date ♣ Get process and device attributes 5. }Communication: These types of system calls are specially used for interprocess communications (IPC). ⊇Functions: ♣ Create, delete communications connections ♣ Send, receive message ♣ Help OS to transfer status information ♣ Attach or detach remote devices |

|  |
| --- |
| ***Types of Schedulers with example:*** |
| ***Condition for deadlock and prevention:*** |

|  |
| --- |
| ***Types of Threads:*** |

***Scheduling algorithm/Sums/Bankers Algorithm:***

***Race Around Condition:***

|  |
| --- |
| The situation where several processes access and manipulate shared data concurrently. The final value of the shared data depends upon which process finishes last.  A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time. But, because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly. To prevent race conditions, concurrent processes must be synchronized |
|  |

***Characteristics of Deadlock:***

|  |
| --- |
| ***Mutal Exclusion, Hold and wait, No Pre-emption, Circular Hold*** |

|  |
| --- |
| ***Explain Semaphore operation with codes:*** |