PES UNIVERSITY

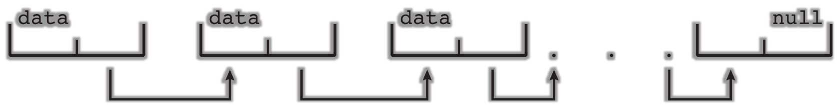
(Established under Karnataka Act No.16 of 2013)
100-ft Ring Road, BSK III Stage, Bangalore – 560 085

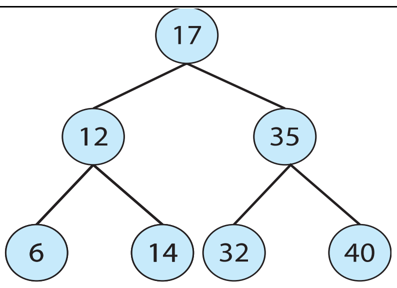
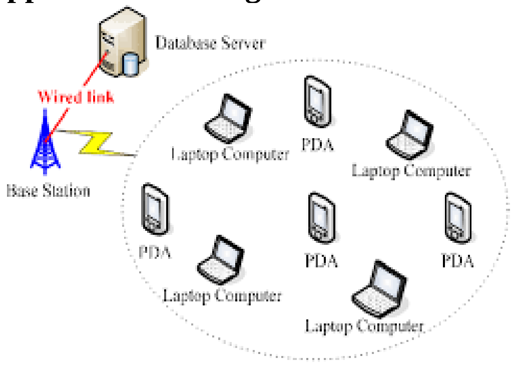
Department of Computer Science & Engg

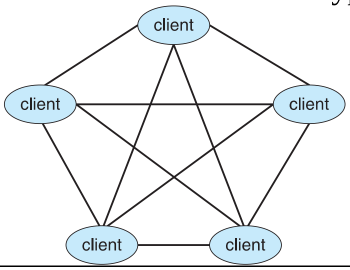
Session: Jan-May 2021

UE19CS254: Operating Systems

Unit 1 Question Bank(Solutions to selected questions)

#	
Chapter 1	
1.	Explain the data structures used in implementing OS. Lists • Standard programming data structures are used extensively in OS **Singly linked list** • The items in a list must be accessed in a particular order. • common method for implementing this structure is a linked list  • In a **singly linked list,** each item points to its successor. • In a **doubly linked list**, a given item can refer either to its predecessor or to its successor. • In a **circularly linked list**, the last element in the list refers to the first element, rather than to null. Lists Adavantages: • Linked lists accommodate items of varying sizes. • Allow easy insertion and deletion of items. Lists disadavantages: • performance for retrieving a specified item in a list of size n is linear — O(n) , worst case. • Usage • List are used by the some of the kernel algorithms, • Constructing more powerful data structures such as stacks and queues Trees • **Data structure used to represent data hierarchically** • **Binary search tree** ordering between 2 children: left <= right • Search performance is *O(n)* • **Balanced binary search tree** is *O(lg n)* • *Used by Linux as part its CPU-Scheduling algorithm for selecting which task to run next*

	 Hash functions • Hash functions can result in the same output value for 2 inputs • **Hash function** can be used to implement a **hash map** ○ Maps or associates key:value pairs using a hash function **Bitmaps** • **Bitmap** - string of *n* binary digits representing the status of *n* items • Availability of each resource is indicated by the value of a binary digit • 0 – resource is available • 1 – resource is unavailable • Value of the **i**$^{\text{th}}$ position in the bitmap is associated with the **i**$^{\text{th}}$ resource • Ex: bitmap 001011101 shows resources 2, 4, 5, 6, and 8 are unavailable; resources 0, 1, 3, and 7 are available
2.	Explain any two computing environments Ans: Mobile computing • Handheld smartphones, tablets, etc • What is the functional difference between them and a “traditional” laptop? • Extra feature – more OS features (GPS, gyroscope) • Allows new types of apps like ***augmented reality*** • Use IEEE 802.11 wireless, or cellular data networks for connectivity • Leaders are **Apple iOS** and **Google Android**  Database Server Wired link Base Station Laptop Computer PDA Laptop Computer PDA PDA PDA Laptop Computer Laptop Computer **Peer-to-Peer computing** • Another model of distributed system • P2P does not distinguish clients and servers ○ Instead all nodes are considered peers ○ May each act as client, server or both ○ Node must join P2P network ▪ Registers its service with central lookup service on network, or ▪ Broadcast request for service and respond to requests for service via ***discovery protocol***

	○ Examples include Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype 
3.	Name the applications of peer-to-peer computing environment Applications of peer-to peer computing are Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype
Chapter 2	
1.	What are the services provided by the OS? Ans: Services provided by the operating system are, • **User interface** - Almost all operating systems have a user interface (**UI**). ○ Varies between **Command-Line (CLI)**, **Graphics User Interface (GUI)**, **Batch** • **Program execution** - The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error) • **I/O operations** - A running program may require I/O, which may involve a file or an I/O device • **File-system manipulation** - The file system is of particular interest. Programs need to read and write files and directories, create and delete them, search them, list file Information, permission management. • **Communications** – Processes may exchange information, on the same computer or between computers over a network. Communications may be via shared memory or through message passing (packets moved by the OS) • **Error detection** – OS needs to be constantly aware of possible errors. May occur in the CPU and memory hardware, in I/O devices, in user program. For each type of error, OS should take the appropriate action to ensure correct and consistent computing. Debugging facilities can greatly enhance the user's and programmer's abilities to efficiently use the system • **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them. Many types of resources - CPU cycles, main memory, file storage, I/O devices. • **Accounting -** To keep track of which users use how much and what kinds of computer resources • **Protection and security -** 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
2.	Why policies are to be separated from mechanisms? Ans: The separation of policy from mechanism is a very important principle, it allows maximum flexibility if policy decisions are to be changed later (example – timer to prevent a user program from running too long

3. Name the language used to build the UNIX and windows operating system

- Much variation
    - Early OSes in assembly language
    - Then system programming languages like Algol, PL/1
    - Now C, C++
- Actually usually a mix of languages
    - Lowest levels in assembly
    - Main body in C
    - Systems programs in C, C++, scripting languages like PERL, Python, shell scripts
- More high-level language easier to **port** to other hardware
    - But slower

Chapter 3

1. Describe the differences among short-term, medium-term, and long term scheduling.

Short term scheduler

It is also called as CPU scheduler. Its main objective is to increase system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects a process among the processes that are ready to execute and allocates CPU to one of them. Short-term schedulers, also known as dispatchers, make the decision of which process to execute next. Short-term schedulers are faster than long-term schedulers.

Medium Term Scheduler

Medium-term scheduling is a part of swapping. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-processes.

Long Term Scheduler

It is also called a job scheduler. A long-term scheduler determines which programs are admitted to the system for processing. It selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling.

The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.

2. Describe the actions taken by a kernel to context-switch between processes.

Ans: OS save the context of the current process and restore the context of the new process to be executed.

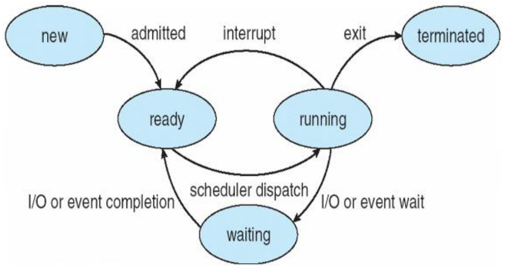
3. What is context switching

Ans:

Context Switching involves storing the context or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier.

4. What are the operations that can be performed on the processes?

- Process creation: process is created in UNIX OS using fork() system call
- Process termination: normal or abnormal termination

	• Process stopping: Temporary suspension of the process.
5.	Differentiate between process & program Program is ***passive*** entity stored on disk (**executable file**), process is ***active*** • Program becomes process when executable file loaded into memory • Execution of program started via GUI mouse clicks, command line entry of its name, etc • One program can be several processes • Consider multiple users executing the same program **Process** – a program in execution; process execution must progress in sequential fashion • Multiple parts • The program code, also called **text section** • Current activity including **program counter**, processor registers • **Stack** containing temporary data, Function parameters, return addresses, local variables • **Data section** containing global variables • **Heap** containing memory dynamically allocated during run time
6.	Identify the section of a process's memory for the following statements. i. The section that contains temporary data ii. The section that contains global variable iii.The section that contains Program code iv. The section that contains Function parameters Ans stack data text stack
7.	What sections of process? text section program counter, processor registers Stack Data section Heap
8.	What are the possible conditions for the process to move from running state to ready state? A diagram depicting the same is required.  new admitted interrupt exit terminated ready running scheduler dispatch I/O or event completion I/O or event wait waiting New born to ready: when the process if selected by the long-term scheduler Ready- running: When the process selected by the dispatcher and the CPU is free. Running- waiting: when the process goes for an I/O operation, waiting for an event to occur. Waiting-ready- After the completion of I/O or an event has occurred Running-termination: after the call to exit

Running: ready: when the processes is interrupted

Chapter 5

1. Explain the differences in how much the following scheduling algorithms discriminate in favour of short processes:
a. FCFS
b. RR
c. Multilevel feedback queues
Ans
a. FCFS—discriminates against short jobs since any short jobs arriving after long jobs will have a longer waiting time.
b. RR—treats all jobs equally (giving them equal bursts of CPU time) so short jobs will be able to leave the system faster since they will finish first.
c. Multilevel feedback queues work similar to the RR algorithm— they discriminate favourably toward short jobs.

2. What is the drawback of priority scheduling? How to overcome this.
Ans
Characteristics of Round-Robin Scheduling
Here are the important characteristics of Round-Robin Scheduling:

- Round robin is a pre-emptive algorithm
- The CPU is shifted to the next process after fixed interval time, which is called time quantum/time slice.
- The process that is preempted is added to the end of the queue.
- Round robin is a hybrid model which is clock-driven
- Time slice should be minimum, which is assigned for a specific task that needs to be processed. However, it may differ OS to OS.
- It is a real time algorithm which responds to the event within a specific time limit.
- Round robin is one of the oldest, fairest, and easiest algorithm.
- Widely used scheduling method in traditional OS.

3. An operating system uses Shortest Remaining Time first (SRT) process scheduling algorithm. Consider the arrival times and execution times for the following processes:

Process	Execution time	Arrival time
P1	20	0
P2	25	15
P3	10	30
P4	15	45

What is the total waiting time for process P2?
Ans: 15

4. What are the characteristics of Round-Robin Scheduling algorithm?

5. Name the criterias used to judge the scheduling algorithms.

CPU utilization – keep the CPU as busy as possible
Throughput – # of processes that complete their execution per time unit
Turnaround time – amount of time to execute a particular process
Waiting time – amount of time a process has been waiting in the ready queue

	Response time – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)
6.	What are the advantages and disadvantages of Round-robin Scheduling? **Advantages** 1. Round robin algorithm doesn't suffer from convoy effect since each process is preempted after the time quantum expires. 2. Round robin algorithm doesn't suffer from starvation since there is no priority associated with each process. 3. Since round robin algorithms is FCFS algorithm with time quantum, the response time is better. 4. Implementation is very simple using a FIFO queue. **Disadvantages** 1. Average waiting time can be high since each process has to wait in ready queue after time quantum expires. More the burst time, higher the waiting time. 2. Low value for time quantum can lead to frequent context switches. 3. High value of time quantum can lead to poor response times and make this algorithm behave as that of FCFS. 4. Same quantum is given to both IO bound and CPU bound process which is not fair. That's why modern operating system use a combination of round robin and priority based scheduling.
	IPC
1.	What are the limitations of pipes? Ans: Pipes have two limitations: 1. Historically, they have been half duplex (data flows in only one direction). Some systems now provide full-duplex pipes, but for maximum portability, we should never assume that this is the case. 2. Pipes can be used only between processes that have a common ancestor. Normally, a pipe is created by a process, that process calls `fork`, and the pipe is used between the parent and the child.
2.	What are named pipes? FIFOs are sometimes called named pipes. With FIFOs, however, unrelated processes can exchange data.