OPERATING SYSTEM

NOTE:

MAKAUT course structure and syllabus of 5th semester has been changed from 2020. Previously **OPERATING SYSTEM** was in 3rd year both semester in CS & IT branch. Subject organization has been changed slightly in present curriculum. Taking special care of this matter we are providing chapterwise relevant MAKAUT university solutions and some model questions & answers for newly introduced topics, so that students can get an idea about university questions patterns.

INTRODUCTION

Multiple Choice Type Questions

1. SPOOLING stands for [WBUT 2009, 2013(IT), 2015(CS), 2015(IT)]
   a) Spontaneous Peripheral Operation Online
   b) Small Peripheral Operation Online
   c) Simultaneous Peripheral Operation Online
   d) None of these

Answer: (c)

2. What is a shell? [WBUT 2009, 2015(CS), 2017(CS)]
   a) it is a hardware component
   b) it is a command interpreter
   c) it is a part in compiler
   d) it is a tool in CPU scheduling

Answer: (b)

3. Which is not a layer of operating system? [WBUT 2010, 2014(IT), 2017(IT)]
   a) Kernel b) Shell c) Application program d) Critical section

Answer: (d)

4. The main purposes of OS is [WBUT 2012(IT), 2016(IT)]
   a) to provide users an environment to execute programs
   b) to manage computer resources
   c) both (a) & (b)
   d) none of these

Answer: (c)

5. ..... Provides an interface to the operating system for the user. [WBUT 2016(CS)]
   a) Kernel b) Micro-kernel c) Shell d) None of these

Answer: (c)

6. In a resident OS computer, which of the following systems must reside in the main memory under all situations? [WBUT 2016(IT)]
   a) Assembler b) Loader c) Linker d) Compiler

Answer: (d)

7. If UNIX command chmod 756 is applied to a file, them other will have [WBUT 2017(CS)]
   a) Read and write permission
   b) Read and execute permission
   c) Write and execute permission
   d) None of these

Answer: (d)

8. Which directory implementation is used in most of the Operating System? [WBUT 2018(CS)]
   a) Single level directory structure
   b) Two level directory structure
   c) Tree directory structure
   d) Acyclic directory structure

Answer: (c)

9. System calls are usually invoked by [WBUT 2018(IT)]
   a) a software interrupt b) polling
   c) an indirect jump d) a privileged instruction
Answer: (a)

10. Time Sharing Operating system has [WBUT 2018(IT)]
   a) high through put b) low execution time
   c) faster I/O d) None of these
Answer: (d)

11. For real time operating systems, interrupt latency should be [WBUT 2019(IT)]
   a) minimal b) maximum
   c) zero d) dependent on the scheduling
Answer: (a)

12. In distributed system, each processor has its own [WBUT 2019(IT)]
   a) local memory b) clock
   c) both local memory and clock d) none of the mentioned
Answer: (c)

13. Network operating system runs on [WBUT 2019(IT)]
   a) every system in the network b) server
   c) both server and every system in the network
   d) none of the mentioned
Answer: (b)

14. In distributed systems, link and site failure is detected by [WBUT 2019(IT)]
   a) polling b) handshaking
   c) token passing d) none of the mentioned
Answer: (b)

15. In real time operating system [WBUT 2019(IT)]
   a) all processes have the same priority
   b) a task must be serviced by its deadline period
   c) process scheduling can be done only once
   d) kernel is not required
Answer: (b)

16. A benefit of the microkernel organization is [WBUT 2019(IT)]
   a) extensibility b) portability c) flexibility d) all of these
Answer: (d)

Short Answer Type Questions

1. Mention one characteristic each of Time Sharing System and Batch Processing System. [WBUT 2009, 2012(CS), 2015(IT)]

Answer:
The characteristic of time sharing system is that the CPU is allocated to each user in turn for a small time-slice. As soon as the time slice is over, the CPU switches to the next user. This system uses CPU scheduling and multi programming whereas in Batch processing system, jobs were submitted in batches to the computer. The term "Batch" means the jobs were submitted in batches to the computer. In order to identity the various jobs and its function, special control cards, Job Control language Card were used. Scheduling of jobs in this system may be FCFS or SJN (Shortest Job Next). Here in this system, at most one program is in execution, no time division management is required. It has a better scope for resource utilization due to its serial processing.

2. Name one essential property of the following types of operating systems:
(a) Batch, (b) Interactive, (c) Time-sharing, (d) Real time, (e) Network.
[WBUT 2016(CS)]

Answer:
a) Batch: Jobs with similar needs are batched together and run through the computer as a group by an operator or automatic job sequencer. Performance is increased by attempting to keep CPU and I/O devices busy at all times through buffering, offline operation, spooling and multi-programming. Batch is good for executing large jobs that need little interaction; it can be submitted and picked up later.
b) Interactive: This system is composed of many short transactions where the results of the next transaction may be unpredictable. Response time needs to be short since the user submits and waits for the result.
c) Time-sharing: This system uses CPU scheduling and multi-programming to provide economical interactive use of a system. The CPU switches rapidly from one user to another. Instead of having a job defined by spooled card images, each program needs its next control card from the terminal and output is normally printed immediately to the screen.
d) Real time: It is often used in a dedicated application, this system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.
e) Network: It provides operating system features across a network such as file sharing.

3. What is the purpose of the command interpreter? Why is it usually separate from the kernel? **[WBUT 2016(CS)]**
Answer:
1st Part:
Command interpreter needs command from the user or from a file of commands and executes them, usually by turning them into one or more system calls. It is an interface of the operating system with the user.

2nd Part:
It is not usually a part of the kernel since the command interpreter is subject to change.

4. a) What do you mean by real time system? **[WBUT 2017(CS)]**

Answer:

Refer to Question No. 2(a) of Long Answer Type Questions.

b) Differentiate between soft and hard real time system. **[WBUT 2017(CS)]**

Answer:

In soft real time system it is considered undesirable, but not catastrophic if deadlines are occasionally missed. Most modern operating system can serve as the base for a soft real time system.

Example:

Multimedia transmission and reception, Compute games, networking and telecom networks, websites and services etc. A hard real time systems has time critical deadlines that must be met otherwise a catastrophic system failure can occur. It requires formal verifications of being to always meet to hard deadlines except for fatal errors.

Example:

Air traffic control, Nuclear power plant control, etc.

5. a) What is kernel? **[WBUT 2017(CS)]**

Answer:

A Kernel is a Central component of an operating system. It acts as an interface between the user applications and the hardware. The sole aim of the kernel is to manage the command caption between the software (user level) and the hardware (CPU, mouse etc.).

b) State the functions of system call. **[WBUT 2017(CS)]**

Answer:

Context switching is generally computationally intensive. That is, it requires considerable processor time, which can be on the order of nanoseconds for each of the tens or hundreds of switches per second. Thus context switching represents a substantial cost to the system in terms of CPU time and can, in fact, be the most costly operation in an operating system.

6. What is virtual machine? **[MODEL QUESTION]**

Answer:

Virtual machine creates an illusion of a real machine. The virtual machine creates this illusion by time multiplexing the system resources among all the users of the machine. A user can also run a single user OS on this virtual machine. Different OS can run on different virtual machine. Examples are: IBM 370 where VM/370 is on IBM 370. An illusion is created by a virtual machine OS that makes a single real machine to appear as several real machines.

7. What are the different types of virtualization? Explain in details. **[MODEL QUESTION]**

Answer:
The main types of virtualization include server, network, storage, application and desktop.
Application virtualization:
It increases the mobility of the applications and facilitates migration of the VMS with lesser down time.
Desktop virtualization:
It is the virtualization of the desktop so as to reduce the cost and increase of the service.
Network virtualization:
It completely reflects the physical network as the software-defined network.
Server virtualization:
The multiple operating systems and the physical server could be run on the single server.
Storage virtualization:
It gives the abstraction layer for the resources of physical storage to get optimized in virtual deployment.

8. What are the advantages of virtual machine? **[MODEL QUESTION]**

Answer:
Some of the advantages are:

- It allows concurrent running of dissimilar OS by different users.
- It eliminates certain conversion problem.
- Programs can be developed and debugged for machine configurations that is different from those of the host.
- The high degree of separation between independent virtual machine provides more security and privacy.

9. What are the applications of virtual machine system? **[MODEL QUESTION]**

Answer:
If we have a network of virtual machines, each of which can send information over virtual communication network then such a virtual machine is a perfect vehicle for OS research and development. Making changes to OS pointers is very dangerous. Therefore, the current system must be stopped and then taken out of use and only then changes should be made and tested. This is called as system development time. A virtual machine can eliminate much of this problem. System programmers are given their own virtual machine and system development is done on the virtual machine, instead of, on a physical machine. So, normal system operation is not disrupted for the system development.

10. How do you convert a physical machine into a virtual machine? **[MODEL QUESTION]**

Answer:
The physical machine could be converted to a virtual machine by the following ways:

- Installation of an agent to the physical machine.
- Installation of VI client with converter plug-in.

- A server placed to import or export the virtual machines.

11. What VMware tools? How do you use it? **[MODEL QUESTION]**

Answer:

1st part:

VMware tools represent a suite of utilities which enhance the performance of the virtual machines guest operating system and the VM management.

2nd part:

Installation of the VMware tool eliminates or enhances the issues like,

- Inadequate colour depth.
- Restricted mouse movement.
- Low video resolution.
- Incorrect network display speed
- Inability to drag-n-drop files or copy-paste
- Support the guest bound calls

12. What is hypervisor or virtual machine monitor? What are its types? **[MODEL QUESTION]**

Answer:

1st part:

It is nothing but the computer firmware, hardware or the software which creates and runs the virtual machines. The computer on which the hypervisor runs is called the host machine and the virtual machine is called the guest machine.

2nd part:

There are mainly of two types:

Type 1 or the native Bare metal hypervisor. It runs directly on the hardware of the host and manage the hardware and guest operating system.

Type 2 or the hosted hypervisor. It runs on the conventional OS.

13. What is fault tolerance in VMware? **[MODEL QUESTION]**

Answer:

VMware fault tolerance delivers continuous availability for the virtual machines even in an event of failure condition by creating and maintaining the secondary VM that is identical to the primary VM and is made available continuously. Fault tolerance could be enabled for most of the virtual machines.

Long Answer Type Questions

1. a) Discuss real time, parallel and distributed operating system. [WBUT 2019(IT)]

b) Discuss the advantages and disadvantages of Open sources OS.

c) What is / are main difficulty to write a program in real time operating system?

Answer:

a) A real time system is defined as a system in which the correctness of computations depends not only on the logical correctness of the computation but also on the time at which the result is produced. So, we can say that it has strict time constraints. The inputs, data or outputs. Should be available within this time period otherwise disasters can happen. Sensors bring data to the computer. The computer must analyze the data and adjusts controls to modify the sensor inputs.

Example: Some real time systems are like air traffic control, Industrial control systems, medical imaging systems. A system consisting of more than one processor that is tightly coupled (i.e. heavy sharing of resources like bus, clock, memory and I/O devices) is a parallel system. A single CPU may be slow so a possible solution is to have many CPUs put parallel to execute a single given problem. This is parallel processing and such system are also called as super computer. These systems will increase the CPU throughput (number of jobs per unit time), reduce time required for job execution. Distributed systems comprises of a large central computer to which a large number of remote terminals are attached.

Example: Consider that there are two sites like site-1 and site-2. These two sites are connected by some communication links and that the site-1 is having a printer. But site-2 does not have a printer. This site two can use the printer of site -1 without moving from site-2 to site-1 by resource sharing which is possible in a distributed system. In distributed systems communications are easy in these systems. E-mails can be sent Reliability of a distributed systems is good. Distributed processing is a form of online processing that allows a single transaction to be composed of application programs that access one or more databases on one or more computers across a network. This type of transaction in which multiple application programs cooperate is called as a distributed transaction.

b) The main advantages are:

- It is generally free and do not have to pay for using it.
- Most open source applications have their own communities which are constantly evolving the software thus improving its quality and security.
- Open sources software can be adopt for own business demand which cannot be done with proprietary systems. In addition it can also be modified for use with various systems.

Disadvantages:

- Most interfaces are not so user friendly and easy to use.
- Difficult to find drivers for some devices.
- Another disadvantage is the support device. In most cases, should have to rely on the corresponding community or pay for external supporting service.

c) The main difficulty is keeping the operating system within the fixed time contrasts of a real time system. If the system does not complete a task in a certain time frame it may cause a breakdown of a entire system it is running. Therefore when writing an operating system for a real time system, the writer must be sure that this scheduling schemes do not allow response time to execute the time constraints.

2. Write short notes on the following:

a) Real time systems **[WBUT 2012(IT), 2015(IT)]**

b) Time sharing OS or Batch processing OS **[WBUT 2013(IT)]**

c) Spooling **[WBUT 2015(CS)]**

Answer:

a) Real time systems:

A real time system is the one which must process information and produce a response within a specified time otherwise risk may occur. Any system in which the time at which the output is produced is significant, in that case real time system is applicable. The examples are aircraft control, ticket reservation system, temperature monitor in nuclear power station. Real time operating systems are designed for two general classes of applications such as event response and closed-loop control. Event response application such as automated visual inspection of assembly line parts, require a response to a stimulus in a certain amount of time. In contrast, closed loop control system, such as automotive cruise control system, continuously process feedback data to adjust one or more outputs.

b) Time sharing OS or Batch processing OS:

Refer to Question No. 1 of Short Answer Type Questions.

c) Spooling:

It is acronym as simultaneous peripheral operation on line. Spooling is used for data processing at remote sites. Spool is a buffer that holds output for a device, such as a printer, that cannot accept interleaved data streams. The spooling system copies the queued spool files to the printer one at a time. In some operating system, spooling is managed by a system daemon process. The operating system provided a control interface that enables user and system administrators to display the queue, to remove unwanted jobs before those jobs print, to suspend printing while the printer is serviced and so on.

PROCESSES, THREAD & SCHEDULING

Multiple Choice Type Questions

1. Context switching is [WBUT 2006, 2014(IT), 2017(IT)]
   a) part of the spooling
   b) part of polling
   c) part of interrupt handling
   d) part of interrupt servicing

Answer: (d)

2. The main advantage of interrupt concept is elimination of [WBUT 2006, 2014(IT), 2017(IT)]
   a) spooling
   b) polling
   c) job scheduling
   d) blocking the currently running process

Answer: (d)

3. A thread is a [WBUT 2007, 2012(IT), 2013(CS), 2013(IT), 2015(CS), 2016(CS), 2017(CS)]
   a) task b) process c) program d) light weight process

Answer: (d)

4. Which scheduling policy is most suitable for the time – sharing operating system? [WBUT 2007, 2016(CS), 2019(CS]
   a) Shortest job first
   b) Round Robin
   c) First come first serve
   d) Multilevel queue

Answer: (b)

5. The time spent by a process in the ready queue is [WBUT 2008, 2015(CS)]
   a) Waiting time
   b) Turnaround time
   c) Response time
   d) Throughput

Answer: (a)

6. Suppose that a process is in BLOCKED state waiting for some I/O service. When the service is completed, it goes to the [WBUT 2009, 2010, 2013(IT), 2018(IT)]
   a) RUNNING state
   b) READY state
   c) SUSPENDED state
   d) TERMINATED state

Answer: (b)

7. Scheduling a process from Ready Queue to CPU is done by [WBUT 2009, 2015(CS), 2019(CS)]
   a) Short Term Scheduler
   b) Middle Term Scheduler
   c) Long Term Scheduler
   d) Dispatcher

Answer: (a)

8. The scheduler which selects jobs from the pool of jobs and loads them to the ready queue is [WBUT 2010, 2016(CS)]
   a) long term b) short term c) medium term d) none of these

Answer: (a)

9. Which of the following reduces degree of multiprogramming?
[WBUT 2012(CS), 2016(IT)]

a) Long-term scheduler
b) Mid-term scheduler
c) Short-term scheduler
d) all of these

Answer: (a)

10. An address generated by the CPU is commonly referred to as
[WBUT 2012(CS), 2014(IT), 2015(IT), 2017(IT)]

a) logical address
b) physical address
c) relational address
d) virtual address

Answer: (a)

11. Address generated by CPU is generally referred to as
[WBUT 2013(CS), 2014(IT)]

a) logical
b) relational
c) virtual
d) physical

Answer: (a)

12. Which of the following is (are) non pre-emptive scheduling algorithm?
[WBUT 2014(IT), 2017(IT)]

a) FCFS
b) SJF
c) Round Robin
d) priority scheduling

Answer: (a)

13. In which of the following scheduling policies does context switching never take place?
[WBUT 2015(IT)]

a) Round Robin
b) Shortest Job First
c) Pre-emptive
d) First Come First Serve

Answer: (d)

14. Throughput is
[WBUT 2015(IT)]

a) number of processes competed per unit time
b) completion time of the whole process
c) time for waiting in ready queue
d) time waiting to get into the memory

Answer: (a)

15. Which one in the following is NOT shared by the threads of the same process?
[WBUT 2016(IT)]

a) Stack
b) File Descriptor Table
c) Address Space
d) Message

Answer: (a)

16. The number of processes completed per unit time is known as
[WBUT 2017(CS)]

a) Output
b) Throughput
c) Efficiency
d) Capacity

Answer: (b)

17. Round Robin scheduling is essentially the preemptive version of [WBUT 2017(CS)]
a) FIFO
b) Shortest Job First
c) Shortest Remaining Time First
d) Longest Time First
Answer: (a)

18. The average wait time for five processes P1-P5 with burst of 5, 19, 2, 16 and 7 milliseconds respectively, using SJF is [WBUT 2018(CS), 2019(CS)]
a) 5 ms
b) 9.8 ms
c) 28 ms
d) 10.6 ms
Answer: (d)

19. The processes that are residing in main memory and are ready and waiting to execute are kept on a list called [WBUT 2018(CS)]
a) job queue
b) ready queue
c) execution queue
d) process queue
Answer: (b)

20. The address of the next instruction to be executed by the current process is provided by the [WBUT 2018(CS)]
a) CPU registers
b) program counter
c) process stack
d) pipe
Answer: (b)

21. _____ are very effective because a mode switch is not required to switch from one thread to another. [WBUT 2018(CS)]
a) Kernel-level threads
b) Alterable threads
c) User-level threads
d) Application level threads
Answer: (c)

22. Which one of the following is not a valid state of a process? [WBUT 2018(IT)]
a) Load
b) Run
c) Wait
d) Terminate
Answer: (a)

23. Which module gives control of the CPU to the process selected by the short-term scheduler? [WBUT 2018(CS)]
a) Dispatcher
b) Interrupt
c) Scheduler
d) None of the mentioned
Answer: (a)

24. What are very effective because a mode switch is not required to switch from one thread to another? [WBUT 2019(CS)]
a) Kernel-level threads
b) Alterable threads
c) User-level threads
d) Application level threads
Answer: (c)

25. Time duration required for scheduling dispatcher to stop one process and start another is known as [WBUT 2019(IT)]

a) process latency b) dispatch latency
c) execution latency d) interrupt latency

Answer: (b)

26. The following program results in the creation of [WBUT 2019(IT)]

```
main()
{
if(fork()>0)
sleep(100);
}
```

a) an orphan process b) a zombie process
c) a process that executes forever d) none of the mentioned

Answer: (b)

27. How many times the following C program prints yes? [WBUT 2019(IT)]

```
main()
 {
  fork(); fork(); printf("yes")
 }
```

a) Only once b) Twice c) Four times d) Eight times

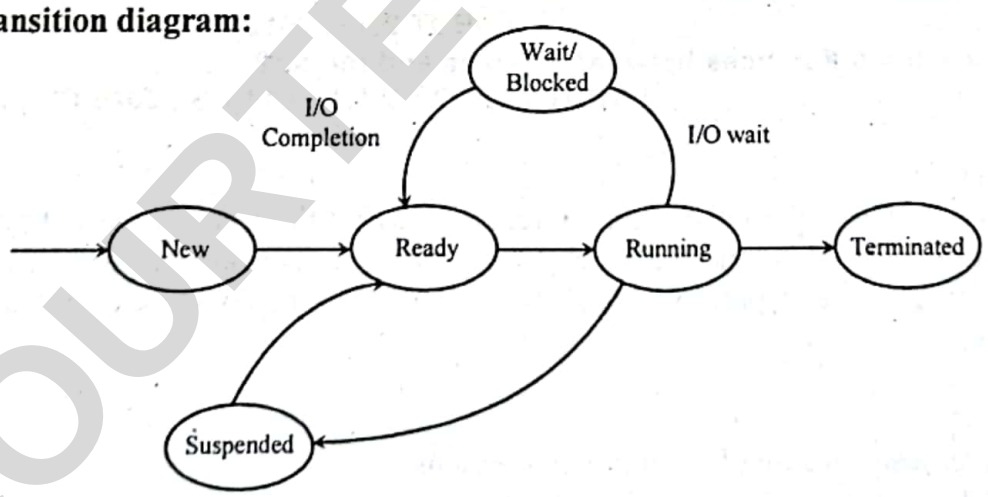
Answer: (c)

Short Answer Type Questions

1. With the help of a state transition diagram, explain various states of a process. [WBUT 2004, 2005, 2009, 2015(CS), 2017(CS)]

Answer:

State transition diagram:



2. How can context switch time be reduced? [WBUT 2007, 2019(CS)]

OR,

Why is context switching considered to be time consuming? [WBUT 2013(CS), 2017(IT)]

Answer:
Since the context of a process includes its state, values of CPU registers and memory management information etc, the context switching is significant. Depending upon the CPU speed and the number of registers to be saved/loaded, it varies from approx. 1 micro second to 1 millisecond. This time is a pure overhead, since CPU is not performing any useful task during this period. Why of reducing context switching time is by incorporating multiple sets of CPU registers. The context of currently active processes could be maintained in different register sets and a pointer could point to the register set of the currently running process. So, during context switching, the OS would need to switch on the pointer from the register set of one process to other; this reducing context switching time considerably.

3. What are co-operating processes? Discuss the advantages of co-operation processes. **[WBUT 2008, 2012(CS)]**

Answer:
Once we have multiply processes, it is likely that two or more of them will want to communicate with each other. Co-operating process can affect or be affected by the execution of another process including sharing of data.

Advantage of process co-operation are:

i) Information sharing
ii) Modularity
iii) Computation speed up
iv) Convenience

4. a) What is context switching? **[WBUT 2009, 2012(CS), 2013(IT), 2014(CS), 2015(CS)]**

Why is it considered to be an overhead? **[WBUT 2009, 2012(CS), 2014(CS), 2015(CS)]**

b) What are the differences between process and thread? **[WBUT 2009, 2012(IT), 2014(CS), 2015(CS), 2017(CS)]**

Answer:

a) 1st Part:
Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as context switch. Context switch time is pure overhead, because the system does not useful work while switching.

2nd Part:
It is an overhead because of the following reasons:

- The system does no useful work while switching.
- It spaced varies from machine to machine depending on the memory speed, number of registers etc.
- It depends on hardware support.

b) i) Process is a heavy weight. But thread is light weight
ii) Switching between processes incurs more overhead. But less overhead to thread.
iii) Communication between processes is expensive but on thread is less expensive.
iv) Processes are independent; threads are dependent.

5. Discuss the structure of Process Control Block. **[WBUT 2010]**

OR,

What is PCB? Mention its content. **[WBUT 2011]**

OR,

Describe process control block (PCB) in details. **[WBUT 2014(IT)]**

OR,

Mention the contents of PCB. **[WBUT 2015(CS)]**

OR,

What is Process Control Block? Discuss the structure of Process Control Block. **[WBUT 2015(IT)]**

OR,

What is process control block? **[WBUT 2018(IT)]**

Answer:

Process Control Block (PCB) is an operating systems data structure which contain the state information for each process (i.e., one PCB per process). State information is needed to suspend and correctly resume process execution when another process is scheduled to run. The structure of PCB is shown below:

Process Identifier (PID) which identifies process
User Identifier (UID) which identifies the user owning the process
CPU state i.e., Data register, Program Counter, Stack pointer etc.
Process scheduling control i.e., Priority, events pending etc.
Process accounting information i.e., ps command
Memory and I/O management i.e., location of all user data and files and devices currently opened

Content of a process:

In general, a computer system process consists of the following resources:

- An image of the executable machine code associated with a program.
- Memory (typically some region of virtual memory) which includes the executable code, process- specific data, a call stack and a heap to hold intermediate computation data generated during run time.
- Operating system descriptors of resources that are allocated to the process, such as file descriptors or handles and data sources and sinks.
- Security attributes such as the process owner and the process set of permissions.
- Processor state (context), such as the content of registers, physical memory addressing etc.

6. a) What are the contents of process control Block (PCB)?
b) Under what conditions the following state transition occurs with respect to a process?
i) Run to Ready; ii) Blocked (or wait) to Ready. **[WBUT 2013(CS), 2017(IT), 2018(CS)]**

Answer:

a) *Refer to Question No. 5 of Short Answer Type Questions.*

b) i) When an interrupt occurs. i.e., requires some I/O during execution, it backs to ready state. .

ii) A process is in waiting or blocked for some event to occur before it can continue execution. A waiting process lacks some resources other than CPU. It goes back to ready for I/O or event completion.

7. What are the relative advantages and disadvantages of user level thread and kernel level thread? **[WBUT 2013(CS), 2017(IT)]**

Answer:

User level thread	Kernel level thread
1. A user level thread maintains all its state in user space.	1. In kernel level thread the kernel does total work of thread movement.
2. Switching between threads can be done without changing address space and kernel information	2. It supports multiprogramming
3. It can run on any operating system and fast.	3. Kernel routines themselves can be multi-threaded.
4. It requires system call (non – blocking)	4. It is slow.

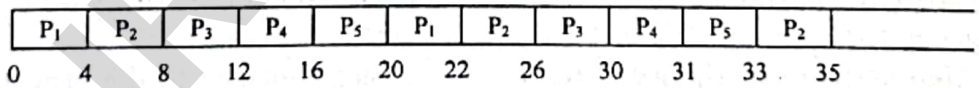
8. Consider the following set of processes with corresponding arrival times and burst times:

Process	Arrival Time (units)	CPU Burst Time (Units)
P1	0	6
P2	3	10
P3	5	8
P4	7	5
P5	10	6

Draw the Gantt chart considering Round Robin scheduling policy with time quantum = 4 units. Calculate individual turnaround time and average waiting time. **[WBUT 2013(CS), 2017(IT)]**

Answer:

1st Part:



2nd Part:

Turn around time = Finish time – arrival time

Turn around time for $P_1 = 22 - 0 = 22$

" " " $P_2 = 35 - 3 = 32$

" " " $P_3 = 30 - 5 = 25$

" " " $P_4 = 31 - 7 = 24$

" " " $P_5 = 33 - 10 = 23$

Waiting time = Turn around time – Processing time/Burst time

∴ Waiting time for $P_1 = 22 - 6 = 16$

" " " $P_2 = 32 - 10 = 22$

" " " $P_3 = 25 - 8 = 17$

" " " $P_4 = 24 - 5 = 19$

" " " $P_5 = 23 - 6 = 17$

∴ Average waiting time = (16+22+17+19+17)/5 = 91/5 = 18.2 units

9. What resources are used when a thread created? How do they differ from those when a process is created? **[WBUT 2014(CS)]**

OR,

What are the main reasons to use of Thread rather than process for different applications? **[WBUT 2019(CS)]**

Answer:

Thread creation typically uses fewer resources than process creation because a thread is smaller than a process. A process creation requires allocating a process control block (PCB), a rather large data structure. The PCB includes a memory map, list of open files and environment variables. Allocating and managing the memory map is typically the most time consuming activity. Creating either a user or kernel thread involves allocating a small data structure to hold a register set, stack and priority.

10. a) What do you mean by pre-emptive and non-pre-emptive scheduling? **[WBUT 2014(IT), 2017(CS)]**

b) What are the different scheduling criteria? **[WBUT 2014(IT)]**

Answer:

a) In non-preemptive scheduling, a process retains control of the CPU until the process is blocked or terminated and denies services to all other processes.

In preemptive scheduling, the scheduler may preempt (i.e. suspend a runnable process) before it is blocked/terminated in order to allocate CPU to another process.

b) Maximize throughput, CPU utilization, turn around time, response time, waiting time and fairness.

11. What is "Turn Around Time"? **[WBUT 2015(CS)]**

Answer:

It is defined as the total time elapsed from the time the job is submitted to the time the job is completed.

i.e. turnaround time = Process finish time – Process arrival time

12. Mention the basic principle of RR scheduling. Specify the impact of time quantum on its performance. **[WBUT 2011, 2014(IT), 2018(IT)]**

OR,

If time quantum I very less for Round Robin Algorithm, then what will be the problems. **[WBUT 2015(CS)]**

Answer:

1st Part:

The basic principle is to give response to the users (interactive systems) in a reasonable time i.e. time-sharing system. The algorithm is similar to FCFS but now it is a **preemptive FCFS** scheduling. The preemption takes place after a fixed interval of time called **time slice** or **time quantum.** Its implementations requires timer interrupts based on which preemption takes place.

Consider a set of processes and the processes are taken out of the ready queue in FCFS order for execution. The time-slice are given to the processes. This process may either finish up its execution before the timer goes off or CPU will be preempted from it after. The timer goes off and these causes interrupt to the operating system. At this time context switching will take place. The next process will be taken from the ready queue.

2nd Part:

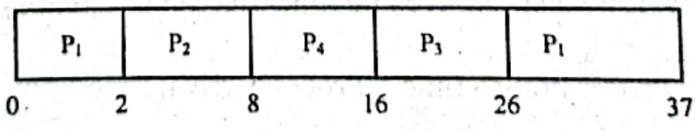
The performance of the RR scheduling algorithm depends on the size of the time quantum. If the quantum value is too large, it behaves as if it is FCFS scheduling algorithm. If the quantum value is too low, context switches are too frequent causing additional burden of workload on the CPU. But context switch time is purely an overhead because in this time period no purposeful computation is carried out for any process. The usual range for the quantum value is between 20 to 50 times of the context switch time. If both the quantum size and context switch overhead are close to zero then RR is called processor sharing because, in theory, it appears to the process that each has its own, though slower, CPU. All ready processes have equal share of CPU time and their speed is inversely proportional to the number of processes in the ready queue.

13. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds. The processes are assumed to have arrived in the order as shown below: Draw four Gantt chart for SRTF (shortest remaining time first) scheduling with time quantum = 5. **[WBUT 2015(IT)]**

Process	Arrival time	CPU Time
P1	0	13
P2	2	6
P3	3	10
P4	5	8

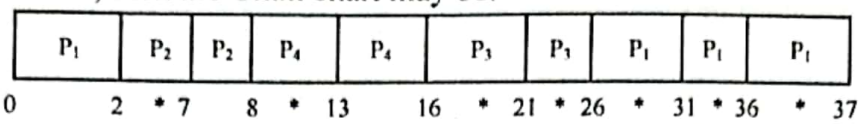
Answer:

Gnatt Chart without time quantum.



Process P_1 is started at time 0, since it is the only process in the queue. Process P_2 arrive at time 2. The remaining time for process P_1 (11 milliseconds) is longer than the time required by process P_2 (6 milliseconds). So process P_1 is preempted and the process P_2 is scheduled. Continuing this way, the above Gnatt chart is drawn.

If time quantum is 5, then the Gnatt chart may be:



The time quantum = 5 is reflected in star (*) position.

14. Given *n* processes to be scheduled on one processor; how many possible different schedules are there? Give a formula in terms of *n*. [WBUT 2016(CS)]

Answer:

1st Part: $n!$

2nd Part: $n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1$

15. What is the main objective of Multiprogramming? Draw and describe process state transitions. [WBUT 2016(IT)]

OR,

Explain the different states of a process using state transition diagram. [WBUT 2019(CS)]

Answer:

1st Part:

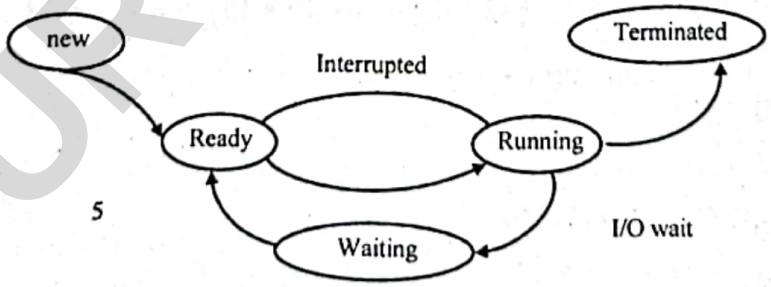
The main objective of multiprogramming is to have process running at all times. With this design, CPU utilization is said to be maximized.

2nd Part:

A program when needs to be executed goes through a process. This process has several state changes in the entire operation until termination of the program. Upon successful termination, the program would get useful results to the user. This entire process progression goes through state changes which are mention below in steps:

1. The process enters a state called NEW STATE.
2. The process then enters the READY STATE
3. The process then goes to an RUNNING STATE (Execution of the program starts here)
4. The process ends with the TERMINATED STATE.

The following PROCESS STATEDIAGRAM would show the entire operation.



Note that there is an intermediary state which is known to be the WAITING STATE. The program goes through this particular state when the CPU is busy with interaction with the I/O devices during I/O operation.

[WBUT 2017(CS)]

16. What is dispatcher?

Answer:

The dispatcher is a module that selects the process from the ready queue for allocating it the CPU. There is a switch associated during dispatching and that is the process status changes from ready to running. The dispatcher is placed in between the ready queen and processes scheduler.

[WBUT 2017(CS)]

17. a) What is Medium Term scheduler?

b) Describe the functions of short-term and long-term scheduler.

Answer:

Refer to Question No. 2(a) (2^{nd} Part) of Long Answer Type Questions.

18. Consider the following four processes, with the length of CPU-burst time given in milliseconds:

Processes	Arrival time	Burst time
P1	0	12
P2	0	10
P3	1	4
P4	4	10
P5	2	12

Draw the Gantt chart using RR scheduling with time slice 3ms. Calculate average waiting time and average turn around time. [WBUT 2017(CS)]

Answer:

The Gantt chart is should below.

P_1	P_2	P_3	P_4	P_5	P_1	P_2	P_3	P_4	P_5	P_1	P_2	P_4	P_5	P_1	P_2	P_5	P_5	
0	3	6	9	12	15	18	21	22	25	28	31	33	36	39	42	43	44	47

Turn around time = Finish time – Arrival time

$P_1 = 42 - 0 = 42$

$P_2 = 43 - 0 = 43$

$P_3 = 2 - 1 = 21$

$P_4 = 44 - 4 = 40$

$P_5 = 47 - 2 = 45$

So, Average turn around time $= (42 + 43 + 21 + 40 + 45)/5 = \frac{191}{5} = 38.2$

Waiting time = Turn around time – Burst time

$P_1 = 42 - 12 = 30$

$P_2 = 43 - 10 = 33$

$P_3 = 21 - 4 = 17$

$P_4 = 40 - 10 = 30$

$P_5 = 45 - 12 = 33$

So, Average waiting time $= (30 + 33 + 17 + 30 + 33)/5 = 143/5 = 28.6$

19. Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here:

i	T	(Pi) Arrival Time
0	80	0
1	20	10
2	10	10
3	20	80
4	50	85

(a) Suppose a system uses RR scheduling with a quantum of 15. Create a Gantt chart illustrating the execution of these processes?
(b) What is the turnaround time for process P_3?
(c) What is the average wait time for the processes? **[WBUT 2018(CS)]**

Answer:

a) As the Round-Robin scheduling follows a circular queue implementation. The Gnatt chart is as:

P_0	P_1	P_2	P_0	P_1	P_2	P_0	P_1	P_3	P_4	P_0	P_3	P_4	P_0	P_4

0 15 30 45 60 75 85 100 110 125 140 155 160 175 190 205

b) The turn around time for process P_3 is $= 160 - 80 = 80$ sec.

c) Average waiting time will be,
waiting time for process $P_0 = 0$ sec
waiting time for process $P_1 = 5$ sec
waiting time for process $P_2 = 20$ sec
waiting time for process $P_3 = 30$ sec
waiting time for process $P_4 = 40$ sec

So, average waiting time $= \frac{(0+5+20+30+40)}{5} = 19$ sec.

20. a) What are main features of Multiprocessor scheduling? **[WBUT 2018(IT)]**

Answer:

When processes can be easily classified into different groups then multi level. Queue scheduling is applied. It can be classified into different groups. For example interactive processes (foreground) and batch processes (background) are two types because of their different response time, scheduling needs and priorities. This algorithm partitions the ready queue into separate queues. Processes are permanently assigned to each queue. This is done based upon the properties such as memory size or process type. This is shown below.

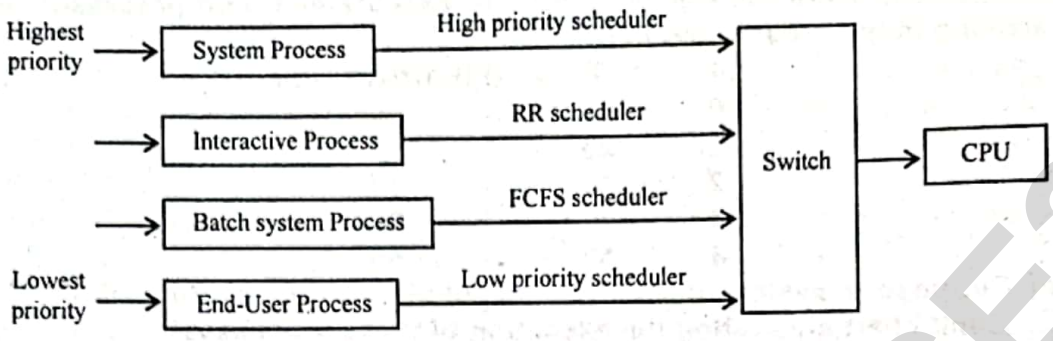


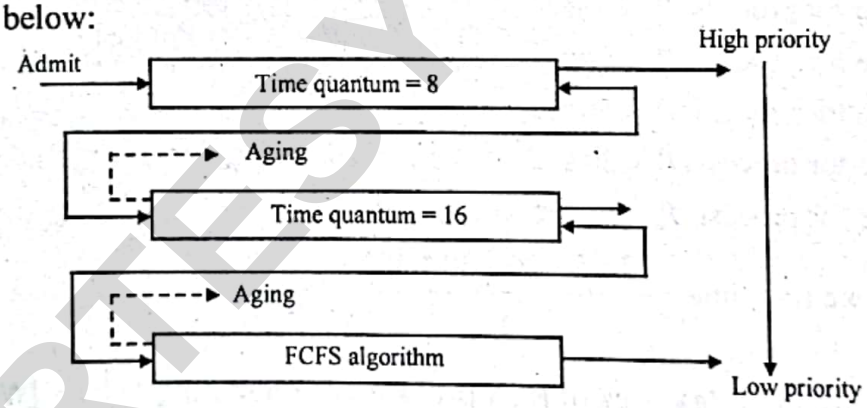
Fig: Multiprocessor scheduling

b) Briefly discuss Multiprocessor feedback queue scheduling. [WBUT 2018(IT)]

Answer:

It is enhancement of multiprocessor scheduling. The principle here is that processes can move between the queues now. In this scheduling, the ready queue is partitioned into multiple queues of different priorities. The processes are assigned to the queues by the system based on their CPU-burst characteristics. This means that if a process consumes too much of a CPU time, it is placed into a lower priority queue. So I/O bound and inter active processes stay in the higher priority queues and CPU bound processes move to the lower priority queues. However, these processes in the lower priority queues should be promoted to the next higher priority queue after a suitable time interval. This technique is known as aging.

This is shown below:



Multiprocessor Feedback Queue

21. Consider the following set of processes. The CPU burst time of them are given in milliseconds:

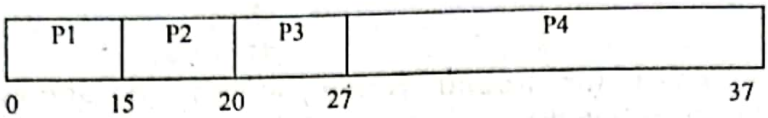
Process	CPU burst time
P1	15
P2	5
P3	7
P4	10

Draw the gnat chart for FCFS and RR Scheduling where time quantum q=5 milliseconds. Calculate the average waiting time. [WBUT 2018(IT)]

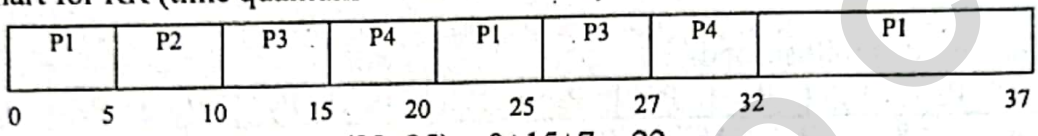
Answer:

Process	CPU burst time
P1	15
P2	5
P3	7
P4	10

Gnatt Chart for FCFS



Waiting time for P_1= 0; P_2= 15; P_3= 20; and P_4= 27
Average waiting time = (0+15+20+27)/4 = 62/4 = 15.5
Gnatt chart for RR (time quantum = 5 milliseconds)



Waiting time for P1= 0+(20–5)+(32–25) = 0+15+7 = 22
Waiting time for P2 =5
Waiting time for P3 = 10+ (25–15) = 10+10 = 20
Waiting time for P4 = 15+(27–20) = 15+7 = 22
Average waiting time = (22+5+20+22)/4 = 69/4 = 17.25

22. Explain the utility of Process Control Block (PCB) for a process? **[WBUT 2018(CS)]**

Answer:
Process Control Block is used for storing the collection of information about the processes and this is also called as the data structure which stores the information about the process. The information of the process is used by the CPU at the Run time. The various information which is stored into the PCB as follows:

- Name of the process
- State of the process i.e., ready, active, wait etc.
- Resource allocated to the process.
- Memory which is provided to the process.
- Scheduling information
- I/O devices used by the process
- Process ID or a identification number which is given by the CPU when a process request for a service.

23. Explain CPU scheduling criteria. **[WBUT 2019(CS)]**

Answer:
Refer to Question No. 2(a) of Long Answer Type Questions.

Long Answer Type Questions

1. Consider the following set of process. CPU burst time of them are given in milliseconds. **[WBUT 2009, 2015(CS)]**

Process	*CPU Burst Time (ms)*
P1	15
P2	5
P3	7
P4	10

Draw the Gantt chart for Round Robin scheduling where time quantum $q = 4$ milliseconds. Calculate the average waiting time and turnaround time. Mention the advantages and disadvantages of Round Robin scheduling.

Answer:

1st Part:

Time quantum = 4 milliseconds

P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_4	P_1

0 4 8 12 16 20 21 24 28 32 34 37

Waiting time for P_1 = (0–0) + (16–4) + (28–20) + (34–32) = 0 + 12 + 8 + 2 = 22 ms.

Waiting time for P_2 = (4–0) + (20–8) = 4 + 12 = 16 ms.

Waiting time for P_3 = (8–0) + (21–12) = 8 + 9 = 17 ms.

Waiting time for P_4 = (12–0) + (24–16) = (32–28) = 12 + 8 + 4 = 24 ms.

$$\therefore \text{Average waiting time} = \frac{22+16+17+24}{4} = \frac{79}{4} = 19.75 \text{ ms}$$

$$\text{Average turnaround time} = \frac{37+21+24+34}{4} = \frac{116}{4} = 29\text{ms}.$$

2nd Part:

If the time quantum of RR is small, it provides good response time but decreases efficiency as it increases number of process switching. If the quantum value is too large, it behaves as FCFS algorithm. If both the quantum size and context switch overhead are closed to zero then RR is called processor sharing.

2. a) What do you mean by scheduler? Explain different types of scheduler. **[WBUT 2016(CS), 2016(IT)]**

Explain CPU scheduling criteria. **[WBUT 2016(CS), 2016(IT), 2018(IT)]**

Answer:

1st Part:

Schedulers are special software (System) which handles process scheduling in various ways and its main task is to select the jobs to be submitted into the system and to decide which process to run.

2nd Part:

There are three types of scheduler namely, long-term, short-term and medium-term.

Long term scheduler is called job scheduler and it determines which programs are admitted to the system for processing. Job scheduler selects processes from the queue and loads them into memory for execution. The primary objective is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. **Short term scheduler** also called CPU scheduler. The main objective is to increase system performance is accordance with the chosen set of criteria. CPU scheduler selects process among the processes that are ready to execute and allocates CPU to one of them.

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

3rd Part:

The criteria are CPU utilization, throughput, turn-around time, waiting time and response time.

The key idea of CPU utilization is that if the CPU is busy all the time, the utilization factor of all the components of the system will be also high.

Throughput refers to the amount of work completed in unit time.

Turn-around time is defined as interval from the time of submission of a process to the time of its completion.

Waiting time may be defined by the difference of turn-around time and processing time.

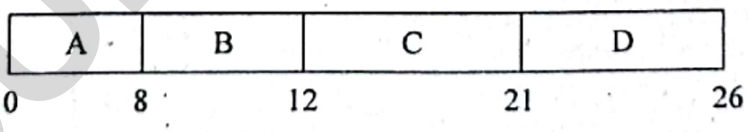
Response time in time sharing system is the interval from the time the last character of a command line of a program or transaction is entered to the time the last result appears on the terminal.

b) For the process listed in the table, draw a chart illustrating their execution using FCFS, SJF, SRTF (SRJF), Round Robin (Quantum = 2) and calculate average turn-around time and average waiting time. **[WBUT 2016(CS), 2016(IT)]**

Process	Arrival Time	Processing Time
A	0	8
B	1	4
C	2	9
D	3	5

Answer:

FCFS:



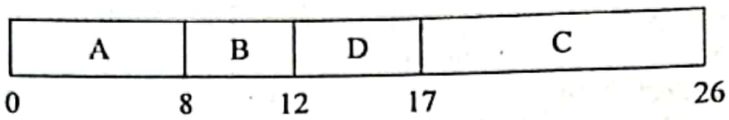
Average waiting time

$$= \frac{\{(0-0)+(8-1)+(12-2)+(21-3)\}}{4} = \frac{(0+7+10+18)}{4} = 8.75$$

Average turn-around time

$$= \frac{\{(8-0)+(12-1)+(21-2)+(26-3)\}}{4} = \frac{(8+11+19+23)}{4} = 15.25$$

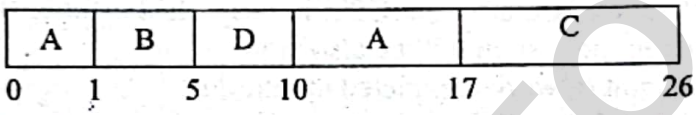
SSTF:



$$\text{Average waiting time} = \frac{\{(0-0)+(8-1)+(17-2)+(12-3)\}}{4} = 7.75$$

$$\text{Average turn-around time} = \frac{\{(8-0)+(12-1)+(26-2)+(17-3)\}}{4} = 14.25$$

SRTF:

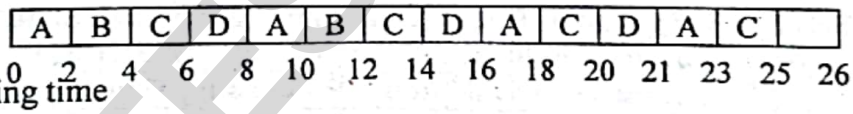


Average waiting time

$$= \frac{\{(0-0)+(10-1)+(1-1)+(17-2)+(5-3)\}}{4} = \frac{(9+0+15+2)}{4} = 6.5$$

$$\text{Average turn-around time} = \frac{\{(17-0)+(5-1)+(26-2)+(10-3)\}}{4} = \frac{52}{4} = 13$$

RR (Quantum = 2):



Average waiting time

$$= \left[\frac{\left\{\frac{0+(8-2)+(16-10)+(21-18)}{A}\right\} + \left\{\frac{2+(10-4)}{B}\right\} + \left\{\frac{4+(12-6)+(18-14)+(23-20)}{C}\right\} + \left\{\frac{6+(14-8)+(20-16)}{D}\right\}}{4} \right] = \frac{\{15+8+17+16\}}{4} = 14$$

Average turn-around time

$$= \frac{\{(23-0)+(12-1)+(26-2)+(21-3)\}}{4} = \frac{23+11+24+18}{4} = \frac{76}{4} = 19$$

3. a) What is thread? Draw and explain thread life cycle. **[WBUT 2017(CS)]**

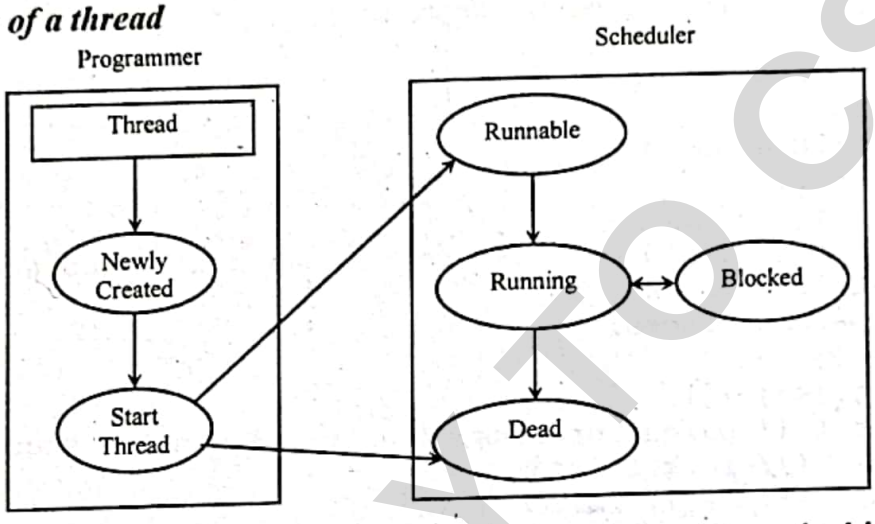
Answer:

1st Part: ***Refer to Question No. 11(a) of Long Answer Type Questions.***

2nd Part:

White a thread is alive, it is in one of several states. By invoking state () method, it does not mean that the thread has access to CPU and start executing straight away. Several factors determine how it will proceed.

Different states of a thread



- **New state:** After the creation of this state but before the start () method in vocation. At this point, the thread is considered not alive.
- **Runnable:** A thread start its life from runnable state. A thread first enters runnable state after the invoking of start () method but a thread can return to this state after either running waiting, sleeping or coming back from blocked state also. On this state a thread is waiting for a turn on the processor.
- **Running state:** A thread is in running state that means the thread is currently executing. There are several ways to enter in runnable state but there is only one way to enter in running state, the scheduler selects a thread from runnable pool.
- **Dead state:** A thread can be considered dead when its run () method completes. If any thread comes on this state that means it cannot ever run again.
- **Blocked state:** A thread can enter in this state because of waiting the resources that are hold by another thread.

b) Explain user and Kernel thread in detail. **[WBUT 2017(CS)]**

Answer:

Refer to Question No. 7 of Short Answer Type Questions.

4. Write a program using 'fork' to demonstrate the mother-child relationship of processes. **[WBUT 2017(CS)]**

Answer:

In a c-program, fork () can be used to create a new process, known as child process. This child is initially a copy of the parent, but can be used to run a different branch of the program or even execute a completely deferent program. After forking, child and parent process run in parallel. Any variables local to the parent process will have been copied for the child process, so up dating a variable in one process will not affect the other. Consider the following program.

```
# include <stdio.h>
# include <unistd.h>
int main (int argc, char ** arg v)
{print ("Beginning of program \n');
int counter =0;
pid = t pid = fork ();
it (pid = = 0)
{// child process
int i = 0;
for (j i <5j++i)
{print f (" child process: counter =% d\n"'++ counter);
{{else if (pid> 0)
{ // parent process
int j=0j
for (j,j<5j++j)
{print f (" parent process: counter =% d\n"'++ counter);
{{ eles {// fork faited
print f ("fork () failed ! \n")
return 1;
print f (" end of program\n");
return 0j
```

This program declares a counter Vendible, set to zero, before fork () ing. After the fark call, we home two process running in parallel, both incrementing their own version of counter. Each process will run to completion and exit. Because the process run in parallel, we home no way of knowing which will finish first. Running this program will print some thing similar to what is shown below, thought results may vary from one run to the next.

Beginning of program
Parent process: Counter = 1
Parent process: Counter = 2
Parent process: Counter = 3
Child process: Counter = 1
Parent process: Counter = 4
Child process: Counter = 2
Parent process: Counter = 5
Child process: Counter = 3
end of program

5. What is thread? Explain types of thread with example. What advantages do threads have over multiple processes? When a process is called a cooperating process? [WBUT 2018(CS)]

Answer:

1st & 2nd Part: ***Refer to Question No. 3(a) of Long Answer Type Questions.***

3rd Part:

A programmer can create multiple threads within a process

Threads executes currently

Threads improve the performance (throughput, speed, responsiveness) of a program or process.

Thread uses single address space.

4th Part: ***Refer to Question No. 3 of Short Answer Type Questions.***

6. What is meant by Process Control Block? Draw and explain different fields of PCB. Explain Context Switch by giving an example. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5 all at time 0.

(a) Draw four Gantt chart illustrating the execution of these processes using FCFS, SJF, a non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.

(b) What is the turnaround time of each process for each of the scheduling algorithms in part a?

(c) What is the waiting time of each process for each of the scheduling algorithms in part a?

(d) Which of the schedules in part a results in the minimal average waiting time (over all processes)? [WBUT 2018(CS)]

Answer:

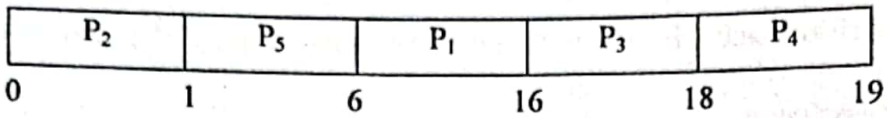
1st & 2nd Part: ***Refer to Question No. 5 of Short Answer Type Questions.***

3rd Part: ***Refer to Question No. 4(a) of Short Answer Type Questions.***

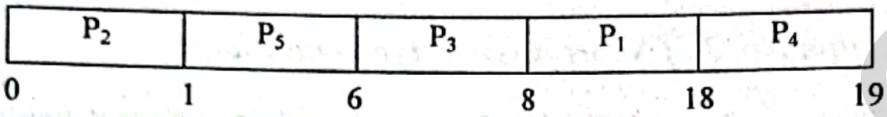
4th Part:

Process	BT	Priority	AT
P1	10	3	0
P2	1	1	0
P3	2	3	0
P4	1	4	0
P5	5	2	0

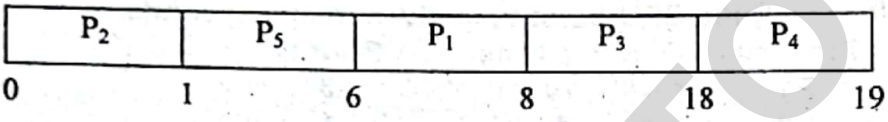
a) FCFS:



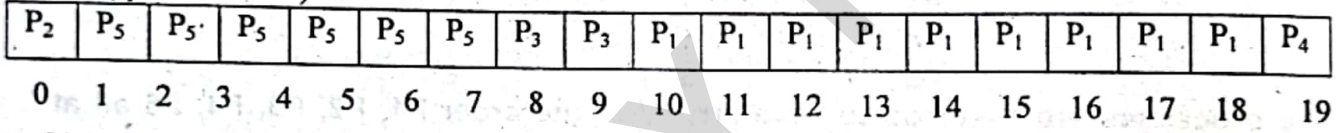
SJF



Non-preemptive priority



RR (quantum = 1)



b) Turn Around Time

FCFS	SJF	Non-preemptive priority	RR
TAT (P_1) = 10	TAT (P_1) = 10	TAT (P_1) = 10	TAT (P_1) = 10
TAT (P_2) = 1	TAT (P_2) = 1	TAT (P_2) = 1	TAT (P_2) = 1
TAT (P_3) = 2	TAT (P_3) = 2	TAT (P_3) = 2	TAT (P_3) = 2
TAT (P_4) = 1	TAT (P_4) = 1	TAT (P_4) = 1	TAT (P_4) = 1
TAT (P_5) = 5	TAT (P_5) = 5	TAT (P_5) = 5	TAT (P_5) = 5
Avg. TAT = 19/5 = 3.8	Avg. TAT = 19/5 = 3.8	Avg. TAT = 19/5 = 3.8	Avg. TAT = 19/5 = 3.8

c) Waiting Time

FCFS	SJF	Non-preemptive priority	RR
WT (P_1) = 6	WT (P_1) = 8	WT (P_1) = 6	WT (P_1) = 8
WT (P_2) = 0	WT (P_2) = 0	WT (P_2) = 0	WT (P_2) = 0
WT (P_3) = 16	WT (P_3) = 6	WT (P_3) = 16	WT (P_3) = 6
WT (P_4) = 18	WT (P_4) = 18	WT (P_4) = 18	WT (P_4) = 18
WT (P_5) = 1	WT (P_5) = 1	WT (P_5) = 1	WT (P_5) = 1
Avg. WT = 41/5 = 8.2	Avg. WT = 33/5 = 6.6	Avg. WT = 41/5 = 8.2	Avg. WT = 33/5 = 6.6

d) Among SJF, FCFS, non-preemptive priority and RR scheduling algorithms: Minimal average waiting time is shown for SJF and RR (with quantum 1) having the minimal average time of 6.6 unit.

7. Explain co-operating system. What is the purpose of interrupt? What are the differences between trap and interrupt? Can traps be generated intentionally by a user program? If so, for what purpose? **[WBUT 2019(IT)]**

Answer:

1st part: ***Refer to Question No. 3 of Short Answer Type Questions.***

2nd part:

The purpose of interrupts is to transfer program control to a specific address based on some event. In processors with a privileged mode of execution, the interrupt also causes a mode switch in to kernel mode.

3rd part:

A trap is a software generated interrupt while an interrupt is triggered by a signal to the processor. Traps are synchronous with the instruction stream: a direct result of the execution of a specific instruction while hardware interrupts are asynchronous.

4th & 5th part:

Yes. Traps can occur through exceptions or through explicit instruction in the program. The reason for this is to allow a user to force a mode switch to kernel mode.

8. What are the benefits of threads? What are user and kernel threads? **[WBUT 2019(IT)]**

Answer:

1st part: ***Refer to Question No. 9 (OR) of Short Answer Type Questions.***

2nd part:

There are two types of threads to be managed in a system – User thread and Kernel thread. User threads are supported above the kernel, without kernel support these are the threads that application programmers would put into their programs. Kernel threads are supported within the kernel of the OS itself. All modern OS support kernel level threads, allowing the kernel to perform multiple simultaneous tasks and/or to service kernel system calls simultaneously.

9. Consider the following set of processes: **[WBUT 2019(CS)]**

Process	CPU Burst Time	Priority	Arrival Time
P0	80	3	0
P1	20	1	10
P2	10	3	10
P3	20	4	80
P4	50	2	85

Draw the Gantt chart using RR (t_s = 15) and preemptive priority scheduling. Calculate average waiting time and total Turnaround time.

Answer:

1st part: *Refer to Question No. 19 of Short Answer Type Questions.*

2nd part:

Turnaround time for process $P_0 = (190 - 0) = 190$ sec

Turnaround time for process $P_1 = (110 - 10) = 100$ sec

Turnaround time for process $P_2 = (85 - 10) = 75$ sec

Turnaround time for process $P_3 = (160 - 80) = 80$ sec

Turnaround time for process $P_4 = (205 - 85) = 120$ sec

$\therefore$ Total Turnaround time = $(190 + 100 + 75 + 80 + 120) = 565$ sec

10. Discuss the CPU scheduling algorithms: FCFS, SJF, RR, priority.

[WBUT 2019(IT)]

Answer:

FCFS:

The basic principle of this algorithm is to allocate the CPU in the order which the processes arrive. For example, P_1, P_2, and P_3 processes arrive at time 0, i.e.

Process	CPU burst time
P_1	24
P_2	03
P_3	03

When arrive in order $P_1 \rightarrow P_2 \rightarrow P_3$, The Gnatt chart is

So, waiting time for P_1=0, for P_2=24, for P_3=27

Average waiting time = $\frac{(0 + 24 + 27)}{3} = 17$

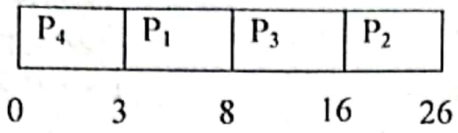
Average turn around time = $\frac{(24 + 27 + 30)}{3} = 27$

SJF:

The basic principle of this algorithm is to allocate the CPU to the process with least CPU burst time. Consider a set of process P_1, P_2, P_3 and P_4

Process	CPU burst time
P_1	05
P_2	10
P_3	08
P_4	03

Gnatt chart



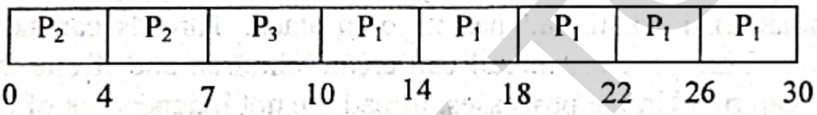
Average waiting time $= \frac{(0+3+8+16)}{4} = 6.75$

RR:

If there are n-processes in a ready queue and the time quantum is q time intervals, then each process gets $\frac{1}{n}$ of the CPU time in the chunks of at most 'q' units of time. Each process will have to wait for (n-1) × q time units until its next time quantum comes in. Consider the set of process at time quantum = 4

Process	CPU burst time
P_1	24
P_2	03
P_3	03

Gnatt chart



Average writing time $= \frac{(10+4+7)}{3} = \frac{21}{3} = 7$

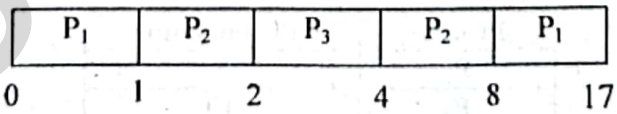
Priority:

In this algorithms, each process is given a priority. The schedule always picks up the highest priority process for its execution from the ready queue. Equal priority processes are scheduled on FCFs basis. The basic principle used here is that priority scheduling is a form of preemptive scheduling where priority is the basis of pre-emption.

Processes	CPU burst time	Priority	Arrival time
P_1	10	3	0
P_2	5	2	1
P_3	2	1	2

Where 1→ highest priority and 3→ lowest priority.

The Gnatt chart:



Average waiting time
= (0+(8-1))+(1+(4-2))+2 = 4

Here,

Waiting time for P_1=0+ (8-1) =7

Waiting time for P_2= 1+ (4-2)=3

Waiting time for P_3=2

11. Write short notes on the following: [WBUT 2008, 2012(IT)]

a) Thread [WBUT 2009, 2018(IT)]

b) Kernel level thread [WBUT 2009, 2015(CS)]

c) Shortest Job First (SJF) scheduling [WBUT 2018(CS), 2019(CS)]

OR,

Preemptive SJF scheduling [WBUT 2012(IT), 2016(IT), 2019(CS)]

d) Process Control Block [WBUT 2013(IT), 2015(IT)]

e) Multilevel feedback queue scheduling [WBUT 2015(CS), 2019(CS)]

f) Kernel level thread & User level thread [WBUT 2015(IT)]

g) Multithreading models [WBUT 2016(CS)]

h) Process life cycle [WBUT 2016(CS)]

i) Orphan process and Zombie process [WBUT 2016(IT)]

j) Aging Technique

Answer:

a) Thread:

Thread is a light weight process. In a process, thread allows multiple execution of stream. In many ways, threads are popular way to improve applications through parallelism. The CPU switches rapidly back and forth among the threads giving illusion that the threads are running in parallel. Like a process, a thread can only state (i.e. running, ready, blocked, terminated). Each thread has its own stack. Threads can take advantages of multi-processors. Like process, thread can create children and if one thread is blocked, another thread can run. Unlike processes, thread are not independent of one another.

b) Kernel level thread:

Kernel level thread can use all the privileges and facilities provided by processes themselves. Thus they can make system calls to convey this resource and I/O requirements to the operating system. The kernel creates appropriate data structures for the new thread and assign it an id. The call returns with the id of the thread. The process creating the thread can use this id for synchronization purpose. The kernel data structure for a thread would be a subset of the data structure for a process.

c) Shortest Job First (SJF) scheduling:

The basic principle of SJF algorithm is to allocate the CPU to the process with least CPU burst time. The processes are available in the ready queue. CPU is always assign to the process with least CPU burst time requirements.

Example

Process	CPU burst time
P_1	5
P_2	10
P_3	8
P_4	3

Since the process with smallest CPU burst time is executed first, so these processes would be scheduled in $P_4 \rightarrow P_1 \rightarrow P_3 \rightarrow P_2$ order.

So, average waiting time = 0+3+8+16/4 = 6.75ms.

SJF may be preemptive and non-preemptive. SJF is an optimal algorithm since this algorithm gives minimum average waiting time. The difficulties of this algorithm is that it cannot be implemented at the level of short term scheduling. It has aging problem.

d) Process Control Block:

Refer to Question No. 5 of Short Answer Type Questions.

e) Multilevel feedback queue scheduling:

Here, processes are not permanently assigned to a queue on entry to the system. Instead, they are allowed to move between queues. The idea is to separate processes with different CPU burst characteristics. If a process uses too much CPU time, it will be moved to a lower priority queue. Similarly, a process that waits too long in a low priority queue will be moved to a higher priority queue. This form of aging prevents starvation.

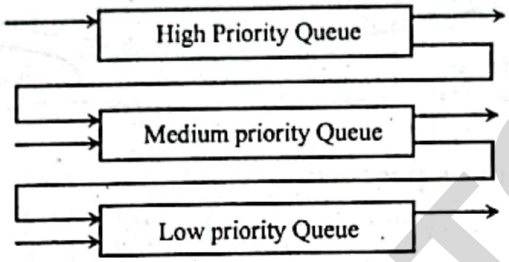


Fig: Multilevel Feedback Queue scheduling

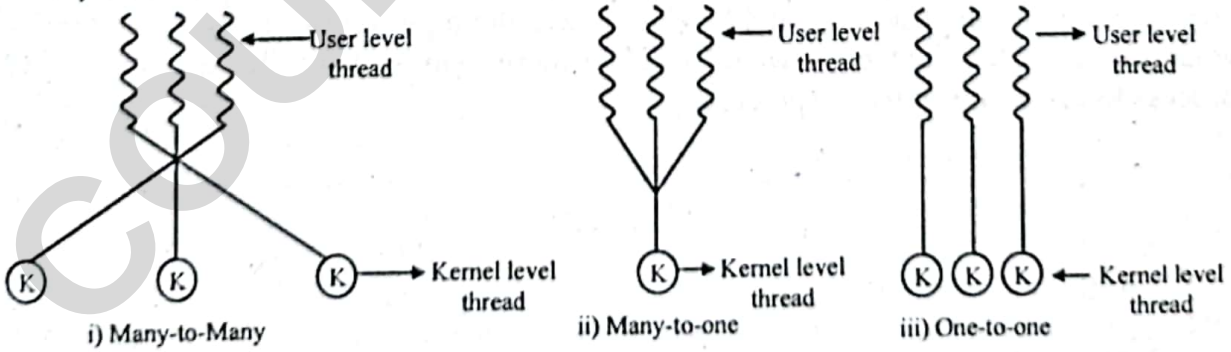
f) Kernel level thread & User level thread:

Refer to Question No. 9 of Short Answer Type Questions.

g) Multithreading models

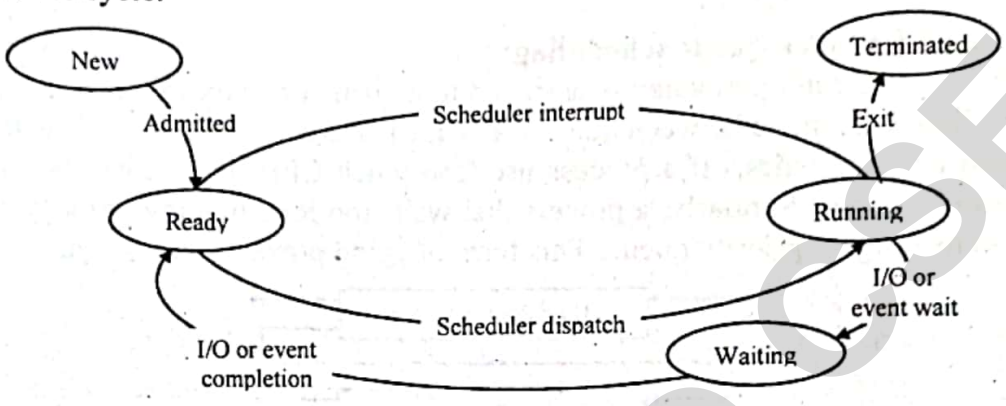
Some operating system provide a combined user level thread and kernel level thread facility. Solaris is a good example of this combined approach. In a combined system, multiple threads within the same application can run in parallel on multiple processors and a blocking system call need not block the entire process. Multithreading models are three types:

i) Many-to-Many
ii) Many-to-one
iii) One-to-one shown below.



h) Process life cycle:

Process life cycle is a manner of looking at process, in the context of their initial, maturing and final stages of evolution and growth. Understanding and analyzing processes in this manner helps to understand how they fit into a "system" of processes as well as how they change as a process natures within an organization. The figure shows the process life cycle.



i) Orphan process and Zombie process:

An orphan process is a computer process whose parent process has finished or terminated.

In Unix, any orphan process will be immediately adopted by the special "init" system process. This operation is called re-parenting and occurs automatically. Even though technically. The process has the "init" process as its parent, it is still called an orphan process since the process that originally created it no longer exists.

A **Zombie** is created when a parent process does not use the want system call after a child dies to need its exit status. The Zombie is not occupying any significant memory or resources, its only an exit status waiting to be delivered. Zombie processes routinely do things by spawning child processes and waiting for those child processes to complete.

j) Aging Technique:

A solution to the problem of indefinite blockage of low-priority processes is aging. Aging is a technique of gradually increasing the priority of processes that wait in the system for a long time. For example, if priority of processes range from 127 (low) to 0 (high), we could increase the priority of a waiting process by 1 every 15 minutes. Eventually, even a process with an initial priority of 127 would have the highest priority in the system and would be executed. In fact, it would take no more than 32 hours for a priority – 127 process to age to a priority – 0 process.

INTER-PROCESS COMMUNICATION

Multiple Choice Type Questions

1. Part of a program where the shared memory is accessed and which should be executed indivisibly, is called [WBUT 2011, 2016(CS)]
   a) semaphores
   b) directory
   c) critical section
   d) mutual exclusion

Answer: (c)

2. A situation where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which access takes place is called: [WBUT 2014(CS), 2018(CS), 2019(CS)]
   a) data consistency
   b) race condition
   c) aging
   d) Starvation

Answer: (b)

3. Mutual exclusion problem occurs between [WBUT 2015(IT)]
   a) two adjacent processes that do not interact
   b) processes that share resources
   c) processes that do not use the same resources
   d) none of these

Answer: (c)

4. In order to implement mutual exclusion on a critical resource for competing processes, only one program at a time should be allowed [WBUT 2016(IT)]
   a) in the critical section of the program
   b) to perform message passing
   c) to exhibit cooperation
   d) none of these

Answer: (a)

5. In order to allow only one process to enter its critical section, binary semaphores are initialized to [WBUT 2017(CS)]
   a) 0
   b) 1
   c) 2
   d) 3

Answer: (b)

6. To avoid race condition the maximum number of processes that may simultaneously be inside the critical section is [WBUT 2018(IT)]
   a) hundred
   b) one
   c) two
   d) three

Answer: (b)

Short Answer Type Questions

1. Describe the two basic operations on semaphore. [WBUT 2011, 2015(CS)]
   Explain whether any integer variable with similar operations can act as semaphore or not. [WBUT 2011, 2015(CS), 2018(IT)]

Answer:

1st Part:

A semaphore supports two basic operations Wait (s) and Signal (s).

Wait(s) operation decrements the value of the semaphore by 1. If the value is already zero, the operation blocks until the value of the semaphore becomes positive (due to the action of some other thread). When the semaphore's value becomes positive, it is decremented by 1 and the wait operation returns.

Signal (s) (sometimes called up) operations increments the value of the semaphore by 1. If the semaphore was previously zero and other threads are blocked in a wait(s) operation on that semaphore, one of those threads is unblocked and its wait operation completes (which brings the semaphore value back to zero).

Consider the counter 'C'. These two operations can be defined atomically as:

```
Signal (s) {
      C ++ ;
  {
Wait(s) {
        While true {
           if ( C > 0) {
              C - - ;
                break ;
                          }
                        }
                    }
```

2nd Part:

Any integer variable (non-negative) can act as semaphore. Semaphores are used for counting tasks such as creating a critical region that allows a specified number of threads to enter. For example, if we want at most four threads to be able to enter a section, we could protect it with a semaphore and initialize that semaphore to four. The first four threads will be able to decrement the semaphore and enter the region, but at that point, the semaphore will be zero and any other threads will block outside the critical region until one of the current threads leaves and signals the semaphore. This type of semaphore (which can take on integer value > 1) called counting semaphore.

2. a) What is race condition? [WBUT 2012(CS), 2013(IT), 2019(IT)]

b) Explain Peterson solution for avoiding race condition. [WBUT 2012(CS)]

OR,

What are the conditions for solution to Mutual Exclusion problem? [WBUT 2013(IT)]

Answer:

a) Race condition is a situation in which multiple processes read and write a shared data item and the final result depends on the relative timing of their execution.

b) In 1981, Peterson discovered a simple way to achieve mutual exclusion. The algorithm is as follows:

```
# define FALSE  0
# define TRUE   1
# define    N     2       /*No. of processes */
int turn;  /* whose turn is it? */
int interested [N]; /* all values initially 0 */
void enter-region (int process) / *Process is 0 or 1 */
{int other; /* No. of the other process */
other = 1 – process; /*the opposite of process */
interested [process] = TRUE;
Turn = process; /* Set flag */
while (turn = = process && interested [other] = = TRUE)
}
Void leave – region (int process) /*process who is leaving */
{ interested [Process] = FALSE; /*Indicate departure from critical region */
}
```

Before using the shared variables i.e. before entering its critical region, each process calls enter-region with its own process number 0 or 1 as the parameter. This call will cause it to wait, if need be, until it is safe to enter. After it has finished with the shared variables, the process calls leave-region to indicate that it is done and to allow the other process to enter, if it so desires.

Let us see how this solution works. Initially, neither process is in its critical region. Now process 0 calls enter-region. It indicates its interest by setting its array element and sets turn to 0. Since process 1 is not interested, 'enter-region' returns immediately. If process 1 now calls enter-region, it will hang there until interested [0] goes to FALSE, an event that only happens when process 0 calls leave-region to exit the critical region. Now, consider the case both processes call enter-region almost simultaneously. Both will store their process number in turn. Whichever store is done last is the one that counts; the first one is lost. Suppose that process 1 stores last, so turn is 1. When both processes come to the 'while statement', process 0 executes it zero times and enter its critical region. Process 1 loops and does not enter its critical region 2.

3. What is critical section problem? What are the requirements those are to be met by a solution to the critical section problem? [WBUT 2012(IT), 2016(CS)]

OR,

What is critical section problem? What are the requirements a critical section problem must satisfy? [WBUT 2016(IT)]

OR,

What are the criteria to the solutions of the critical section problem? [WBUT 2019(IT)]

Answer:

1st Part:

The key to preventing trouble involving shared storage is find some way to prohibit more than one process form reading and writing the shared data simultaneously. That part of the program where the shared memory is accessed is called the critical section.

2nd Part:

To solve the critical section problem, three criteria are there like,

- ***Mutual exclusion:*** If process P_i is executing in its critical section, then no other process can be executing in their critical sections.
- ***Progress:*** If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely.
- ***Bounded waiting:*** A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

4. a) What are the operations on a semaphore? [WBUT 2013(CS), 2017(IT), 2018(IT)]
b) What are the problems with these operations if these follow the classical definition? [WBUT 2013(CS), 2017(IT)]
c) What is the possible remedy to the above problem? [WBUT 2013(CS), 2017(IT)]

Answer:

a) The semaphore has only two operations P ( ) and V ( ).
P ( ) – waits until value > 0 then decrement.
V ( ) – increment, waiting up a thread waiting in P ( ) if necessary.

b) The problem with these operations is that it requires busy waiting which wastes CPU cycle that some other process might be able to use productively.

c) The possible remedy is monitor. Monitors have a property for achieving mutual exclusion i.e. the process only can be active in a monitor at any instant. Consequently, the programmer does not need to code this synchronization constraint explicitly.

5. a) What is semaphore? [WBUT 2014(IT), 2019(IT)]

Answer:

A semaphore is a variable which accepts non-negative integer values and except for initialization may be accessed and manipulated only through two standard atomic operations wait and signal. The two primitives take only argument as the semaphore variable.

b) Write Peterson algorithm for two process critical section problem? [WBUT 2014(IT)]

Answer:

Peterson algorithm is a simple algorithm that can be run by two processes to ensure mutual exclusion for one resource. It does not require any special hardware and it uses busy waiting (a spinlock). Shared variables are created and initialized before either process starts. The shared variable flag [0] and flag [1] are initialized to FALSE because neither process is yet interested in the critical section. The shared variable turn is set to either 0 or 1 randomly.

var flag : array[0..1] of boolean;
turn : 0..1;
flag[0] : = FALSE;
flag[1] : = FALSE;
turn : = random(0..1)
After initialization, each process, which is called process i in the code (the other process is process j) runs the following code.

repeat
flag [i]:= TRUE
turn : = j;
white (Flag [j] and turn = j) do no-op;

Critical section
flag [i] : = FALSE;

Remainder section
until FALSE;

Information common to both processes
turn = 0
flag [0] = FALSE
flag [1] = FALSE

6. State producer-consumer problem. Give a solution to the producer-consumer problem using semaphore. Justify your solution guarantees Mutual Exclusion.
[WBUT 2016(CS), 2016(IT)]

Answer:
Given a set of cooperating processes, some of which "produce" data items (producers) to be "consumed" by others (consumers), with possible disparity between production and consumption rates.
Now, we have to produce a synchronization protocol that allows both producers and consumers to operate concurrently at their respective service rates in such a way that produced items are consumed in the exact order in which they are produced.
Semaphore used: mutex = 1
Empty = N;
Full = 0;

Implementation using semaphore

```
void producer()
    { int item;
     while (true)
          { produce_item (&item);
            wait (empty);
            wait (mutex);
```

```
                enter_item (item);
                signal (mutex);
                signal (full);}
        }
void consumer()
      { while (true)
        { wait (full);
           wait (mutex);
            remove (&item);
            signal (mutex);
            signal (empty);
            consume_item (item);
        }
      }
```

7. a) Explain Race condition in context of process synchronization. **[WBUT 2017(CS)]**

Answer:

Refer to Question No. 2(a) of Short Answer Type Questions.

b) What are semaphore and mutex? **[WBUT 2017(CS)]**

Answer:

1st Part: ***Refer to Question No. 5(a) of Short Answer Type Questions.***

2nd Part:

Mutex is a program object that allows multiple program threads to share the same resource, such as tile access, but not simultaneously. When a program is started a mutex is created with a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished.

8. How to implement a solution to the Readers-Writers Problem with the use of semaphores? **[WBUT 2018(CS), 2019(CS)]**

Answer:

By using the definition of semaphore, the Readers-Writers problem can be solved in the following way:

integer n Readers: $=0$
Semaphore SemReaders: $=1$
Semaphore SemWriters: $=1$

Reader $[i:1..m]$:: do true $\rightarrow$
P (SemReaders)
n Readers: $= n$ Readers $+1$
if n Readers $=1 \rightarrow P$ (SemWriters) fi
V (SemReaders)
read the database

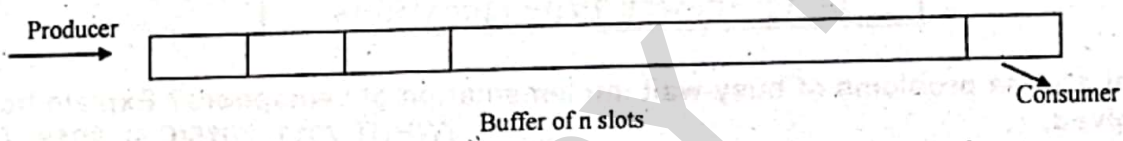
```
                P (SemReaders)
nReaders := nReaders – 1
                if nReaders = 0 → V (SemWriters) fi
        od      V (SemReaders)
Writer [j : 1..n] ::
        do true →
        P (SemWriters)
        write the database
        V (SemWriters)
        od
```

Clearly this solution gives readers preference over writers once one reader is accessing the database, all readers are allowed to read the database without performing the P or V operation on SemWriters.

9. What is Bounder Buffer Problem? Explain with the solution. [WBUT 2019(CS)]

Answer:

Bounded buffer problem, which is also called producer Consumer problem. There is a buffer of n slots and each slot is capable of storing are unit of data. There are two processes running, namely producer and consumer which are operating on the buffer.



Buffer of n slots

A producer tried to insert data into an empty buffer. A consumer tries to remove data from a filled slot in the buffer. Those two processes would not produce expected output if they are execute concurrently. There needs to make them an independent manner. Then what is the solution.

Solution:

One solution of this problem is to use semaphore.

- A binary semaphore which is used to acquire and release the lock.
- Empty, a counting semaphore where initial value is the number of slots in the buffer since initially all slots are empty.
- Full, a counting semaphore where initial value is Q. At any instant, the current value of empty represents the number of empty slots and full represents the occupied slot number.

10. Explain Critical-section problem. How semaphore can be used to solved it? [WBUT 2019(CS)]

Answer:

1st part: ***Refer to Question No. 3 of Short Answer Type Questions.***

2nd part:
The simplest method is to prevent any change of processor control inside the critical section. On uni-processor system this can be done by disabling interrupts on entry into the critical section, avoiding system calls that can cause a context switch while inside the section and restoring interrupts to their previous state on exit. Any thread of execution entering any critical section anywhere in the system will with this implementation, prevent any other thread, including an interrupt, from getting the CPU and therefore from entering any other critical section or, indeed, any code what so ever, until the original thread leaves its critical section. This brute-force approach can be improved upon by **using semaphores.** To enter a critical section, a thread must obtain a semaphore, which it releases on leaving the sections. Other threads are prevented from entering the critical section at the same time as the original thread, but are free to gain control of the CPU and execute other code, including other critical sections that are protected by different semaphores.

11. How semaphore can be used for Dining Philosophers' problem?
[WBUT 2019(IT)]

Answer:
Refer to Question No. 2 of Long Answer Type Questions.

Long Answer Type Questions

1. What are the problems of busy-wait implementation of semaphore? Explain how it is solved. **[WBUT 2011, 2015(CS), 2018(IT)]**

Answer:
The main disadvantage of the semaphore is that requires busy waiting. While a process is in its critical section, any other process that tries to enter its critical section must loop continuously in the entry code. This continual looping is clearly a problem in a real multiprogramming system, where a single CPU is shared among many processes. Busy waiting wastes CPU cycles that some other process might be able to use productively. This type of semaphore is also called a **spin lock** because the process "Spins" while waiting for the lock. To overcome the need for busy waiting, we can modify the definition of wait ( ) and signal ( ) semaphore operations. When a process executes the wait ( ) operation and finds that the semaphore value is not positive, it must wait. However, rather than engaging in busy waiting the process can block itself. The block operation places a process into a waiting queue associated with the semaphore and the state of the process is switched to the waiting state. Then control is transferred to the CPU scheduler, which selects another process to execute. A process that is blocked, waiting on a semaphore 'S' should be restarted when some other process executes a signal ( ) operation. The process is restarted by a wakeup ( ) operation, which changes the process from the waiting state to the ready state. The process is then placed in the ready queue.

2. What is Dining philosopher's problem? Derive an algorithm to solve the problem using semaphore? **[WBUT 2012(CS)]**

OR,

Discuss dinning philosopher problem with the solution. **[WBUT 2014(CS)]**

Answer:

In computer science, the dining philosophers problem is an illustrative example of a common computing problem in concurrency. It is a classic multiprocessor synchronization problem. It is summarized as five philosophers sitting at a table with a large bowl of spaghetti in the center. A fork is placed in between each pair of adjacent philosophers and as such, each philosopher has one fork to his left and one fork to his right. A dangerous possibility of deadlock occurs when every philosopher holds a left fork and waits perpetually for a right fork or vice-versa.

Algorithm to solve dining philosopher problem using semaphore:

```
/ * Program dining philosophers.* /
 Semaphore fork [5] = {1};
 Semaphore room = {4};
 Void philosopher (int i) {
       While (true) {
       Think ( );
       Wait (room);
       Wait (fork [i]);
       Wait (form [(i+1) mod 5]);
            eat ( );
            Signal (fork[(i+1)mod 5 ] );
            Signal (fork [i]);
            Signal (room);
            }
            }
 Void main ( ) {
Parbegin (Philosopher (0), Philosopher (1), Philosopher (2),
                           Philosopher (3), Philosopher (4));
 }
```

3. What is semaphore? Differentiate between binary and counting semaphore. **[WBUT 2012(IT), 2016(IT), 2018(IT)]**

Answer:

1st Part: ***Refer to Question No. 5(a) of Short Answer Type Questions.***

2nd Part:

A Binary semaphore is an integer variable which can be accessed by a cooperating process through the use of two primitives wait & signal. A binary semaphore is initialized by the OS to 1 and it can assume only one of the two values i.e. S = 0 or 1.

The implementation of binary semaphore is extremely simple. It does not meet the requirement of bounded waiting. A counting semaphore can be implemented using binary semaphore. Binary semaphores are called as spin locks.

On the other hand counting semaphore are free of inherent disadvantage of binary semaphore. It comprises of an integer variable that is initialized to a value $k \geq 0$. During operation it can assume any value less than K.
CPU cycles are saved in counting semaphore as a waiting process does not perform any busy waiting.
A counting semaphore is more complex to implement, since it involves implementation of FCFS queue.

4. What do you mean by race condition? What is semaphore? What is counting semaphore? What is the advantage of using a counting semaphore than a binary semaphore? What is Readers-Writers problem? How it can be solved using semaphore? Explain with algorithm. **[WBUT 2015(IT)]**

Answer:

1st Part: ***Refer to Question No. 2(a) of Short Answer Type Questions.***

2nd Part: ***Refer to Question No. 5(a) of Short Answer Type Questions.***

3rd Part:

Basically the semaphore are of two types: **Binary** and **Counting** semaphores. Counting semaphore may have value to be greater than one, typically used to allocate resources from a pool of identical resources.

4th Part:

i) Because the waiting processes will be permitted to enter their critical section in a FCFS order, so the requirement of bounded waiting is fully met.
ii) CPU cycles are saved here as a waiting process does not perform any busy waiting.

5th & 6th Part:

Another famous IPC problem is the Readers Writers problem, which models access to a database. Imagine, for example, an airline reservation system, with many competing processes wishing to read and write it. It is acceptable to have multiple processes reading the database at the same time, but one process is updating (writing) the database. No other process may have access to the database, not even readers. This is the problem of 'readers and writers'.

```
Type def int semaphore, /* use your imagination*/
Semaphore mutex = 1;
Semaphore db = 1/*controls access to the database*/
int rc=0;
void reader(void)
{while (TRUE){
Down(&mutex)
rc = rc+1;
        if(rc==1)down(&db)
        up(&mutex);
```

```
read_data_base();
down(&mutex);
rc = rc – 1;
if(rc==0) up(&db);
up(&mutex);
use_data_read()
}}
void writer(void)
{while(TRUE){
think_up_data();
down(&db);
void write(void)
{while(TRUE){
think_up_data()
down(&db);
write_data_base();
up(&db);}}
```

In this solution, the first reader to get access to the database does a 'down' on the semaphore db. Subsequent readers merely increment a counter rc. As readers leave, they decrement the counter and the last one out does an 'up' on the semaphore allowing a blocked writer, if there is one, to get in. Suppose that while a reader is using the database another reader comes along. Since having two reader at the same time is not a problem, the second reader is admitted. A third and subsequent readers can also be admitted if they come along. But only one writer allowed to write on the database.

5. Write a program using 'signal' to demonstrate a race condition.

[WBUT 2017(CS)]

Answer:

The following code registers a signal handler with multiple signals in order to log when a specific event occurs and to free associated memes before exiting.

```
# include <Signal. h>
# include <Syslog. h>
# include <String. h>
# include <Stdlib. h>
Void * global 1, * global 2;
char * what;
Void sh (int dummy){" %s/n", what);
syslog (LOG-NOTICE,
face (global 2);
face (global 1);
/* Sleep statements added to expand timing window for race condition*/
sleep(10);
exit(0);
int main (int argc, char* argv [])
```

```
{what = argv [1];
global 1 = strdup (argv[2]);
global 2 = malloc (340);
signal (SIGHUP, Sh);
signal (SIGTERM, SH);
/* sleep statement added to expand timing window for race condition */
sleep(10);
exit (0);
```

However, the following sequence of events may result in a double free.

1. a SIGHUP is delivered to the process
2. sh () is invoked to process the SIGHUP
3. This first invocation of sh () reaches the point where global 1 is freed.
4. At this point, a STGTERM is sent to the process.
5. The second invocation of sh () might do and there free of global1

This is gust one possible exploitation of the above code. As another example, the syslog call may use malloc calls. Which are not async-signal sate. this could cause corruption of the heap management structures.

6. Explain Critical-Section problem. Explain Counting Semaphore and binary semaphore with algorithm. **[WBUT 2018(CS)]**

Answer:

1st Part: ***Refer to Question No. 3(1st part) of Short Answer Type Questions.***

2nd Part: ***Refer to Question No. 3(a) (2nd part) of Long Answer Type Questions.***

7. Write short notes on the following:

a) Dining-Philosopher Problem **[WBUT 2015(CS)]**
b) Peterson solution for CS **[WBUT 2016(CS)]**
c) Semaphore **[WBUT 2018(CS)]**

Answer:

a) Dining-Philosopher Problem:

Refer to Question No. 2 of Long Answer Type Questions.

b) Peterson solution for CS:

Refer to Question No. 5(b) of Short Answer Type Questions.

c) Semaphore:

Refer to Question No. 5(a) of Short Answer Type Questions.

DEADLOCK

Multiple Choice Type Questions

1. Banker's Algorithm for Resource Allocation deals with [WBUT 2007, 2015(CS), 2017(CS)]

a) deadlock prevention b) deadlock avoidance
c) deadlock recovery d) mutual exclusion

Answer: (b)

2. The default remedy of starvation is [WBUT 2007, 2013(IT), 2015(CS), 2019(CS)]

a) Ageing b) Critical Section c) Mutual Exclusion d) All of these

Answer: (a)

3. Banker's algorithm solves the problem of [WBUT 2012(CS), 2016(CS)]

a) deadlock avoidance b) context switching
c) deadlock recovery d) mutual exclusive

Answer: (a)

4. Which of the following resources can cause deadlocks? [WBUT 2017(CS)]

a) Read only files b) Shares programs
c) Printers d) All of these

Answer: (b)

5. A computer system has 9 tape drives, with *n* processes competing for them. Each process may need 3 tape drives. The maximum value of *n* for which the system is guaranteed to be deadlock free is [WBUT 2019(CS)]

a) 9 b) 7 c) 8 d) 6

Answer: (d)

Short Answer Type Questions

1. Differentiate between "starvation" & "deadlock" with suitable example. [WBUT 2007, 2008, 2012(CS)]

OR,

What is the difference between starvation and deadlock? [WBUT 2013(CS), 2017(IT)]

Answer:

i) In deadlock, none of the involved processes can possibly make progress. In a starvation, a process is ready to execute, but it is not being allowed to execute.
ii) Deadlock always leads to starvation but vice-versa is not always true.
iii) In deadlock, a process must be able to acquire a resource at first and then go into deadlock. In starvation, the request may be deferred infinitely.

iv) A deadlock with two or more of the processes involvement become struck indefinitely. Starvation is indefinite postponement of entry for a process that has requested it.

2. What is Deadlock? **[WBUT 2007, 2011]**
Explain the Banker's algorithm and show how it is helpful to overcome deadlock. **[WBUT 2007]**

OR,

What is Banker's safety algorithm? **[WBUT 2008, 2014(CS)]**

Answer:

1st Part:

Deadlock can be defined as permanent blocking of a set of processes that either compete for system resources or communicate with each other.

2nd Part:

Banker's Algorithm

This algorithm is used as the technique of a bank, a bank, which offers loans and receives payments to / from customers. The bankers will only grant a loan if he can still meet the needs of other customers. Banker's algorithm is a deadlock avoidance algorithm; the name was chosen because the bank never allocates its available cash.

Several data structure is used to implement the banker's algorithm like,

A vector of length m indicates the number of **available** resources of each type. If **available [j] = k** there are k instances of resource type R_j **available.**

n = no. of processes, m = no. of resource types

An n x m matrix defines the **maximum demand** of each process. If Max [i j] = k, Then process P_i may request at most k instances of resource type R_j.

As n × m matrix defines the number of resources of each type currently **allocated** to each process. If **allocation** [i,j] = k, then process P_i is currently **allocated** k instances of resource type R_j

An n x m matrix indicates the remaining resource need of each process. If **Need** [i,j]=k, then process p_i may **Need** k more instances of resource type R_j to complete its task. So, Need [i,j] = Max[i,j] – Allocation [i,j].

Safety Algorithm

This algorithm determines whether a system is in safe state or not.

1. Assume that work and finish be vectors of length m and n respectively. Initialize work = Available and Finish [i]= false for i= 0,1,2,----n
2. Find an i such that both
   Finish [i]= false
   Need [i] ≤ work
   If no such i exists, go to 4
3. Work = Work + Allocation [i]
   Finish [i] = true
   goto 2
4. If finish [i] = true for all i, then the system is in safe state.

3. "All unsafe states may not lead to deadlock." – Why or why not?
[WBUT 2009, 2012(CS), 2014(CS), 2019(CS)]

Answer:
It is unworthy that a process may be in an unsafe state but not result in a deadlock. For example, if a process request A which would result in an unsafe state, but release B which would prevent circular wait, then the state is unsafe but the system is not in deadlock.

4. What is deadlock? Describe the necessary and sufficient conditions for the occurrence of deadlock. **[WBUT 2015(CS), 2016(CS), 2016(IT)]**

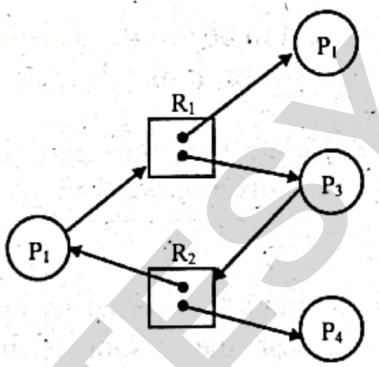
Answer:
1st Part: ***Refer to Question No. 2(1st Part) of Short Answer Type Questions.***

2nd Part: ***Refer to Question No. 1(a) of Long Answer Type Questions.***

5. What is deadlock? Critically comment on the following topic: Cycle in resource allocation graph does not always imply the occurrence of deadlock.
[WBUT 2015(IT), 2017(CS)]

Answer:
1st Part: ***Refer to Question No. 2(1st Part) of Short Answer Type Questions.***

2nd Part:



Graph with a cycle but no deadlock cycle is: $P_1 \rightarrow R_1 \rightarrow P_3 \rightarrow R_2 \rightarrow P_1$ Here is no deadlock as P_4 can release R_2 which can then be allocated to P_3 breaking the cycle.

6. Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free. **[WBUT 2016(CS)]**

Answer:
The system is deadlock-free.
Proof: Suppose the system is deadlocked. This implies that each process is holding one resource and is waiting for one more. Since there are three processes and four resources, one process must be able to obtain two resources. This process requires no more resources and therefore it will return its resources when done.

7. What is the optimistic assumption made in the deadlock detection algorithm? How can this assumption be violated? **[WBUT 2018(CS)]**

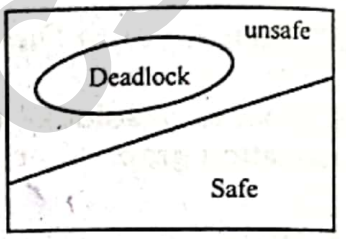
Answer:

The optimistic assumption is that there will not be any form of circular wait in terms of resources allocated and processes making requests for them. This assumption could be violated if a circular wait does indeed occur in practice.

8. With example describe if resources are not properly allocated to the processes it may lead from a safe state to an unsafe state. **[WBUT 2018(IT)]**

Answer:

A System is in safe state only if there exists a safe sequence. A sequence of processes $\langle P_1, P_2 ..... P_n \rangle$ is a safe sequence for the Current allocation state if for each P_i, the resources that P_i can still request can be satisfied by the currently available resources plus the resources held by all the P_j, such that j<i. In this situation, if the resources that process P_i needs are not immediately available then P_i can wait until all P_j have finished. When they have finished, P_i can obtain all of the needed resources, complete its designated task, return its allocated resources and terminate. When P_i terminates, P_{i+1} can obtain its needed resources and so on. If no such sequence exists, then the system state is said to be unsafe. This is shown in the figure.



But not all unsafe states are deadlocks. However, it may lead to a deadlock.

9. Prove that linear ordering for denying the "circular wait" condition actually prevents circuits from developing in resource allocation graphs. **[WBUT 2019(CS)]**

Answer:

The Circular wait can be denied by imposing a total ordering on all of the resource types and then forcing all processes to request the resources in order (increasing and decreasing). This strategy impose a total ordering of all resources typed, and to require that each process requests resources in a numerical order (increasing or decreasing) of enumeration. With this rule, The resources allocation graph can never have a cycle. For example, provide a global numbering of all of the resources, as shown below:

1= Card reader 2= Printer 3=Plotter 4= Tape drive 5= Card punch; Now the rule is that the process can request resources whenever they want to but all request must be made in numerical order.

10. What are the methods for handling deadlock? **[WBUT 2019(IT)]**

Answer:

Methods for handling deadlock:

Avoiding Mutual Exclusion

It means that only one process can use the resource at time i.e. resources are not shared by the number of processes at a time. A process never needs to wait for a sharable resource.

Avoiding Hold & Wait

It means that each & every process in the deadlock state, must hold at least one resource and wait for at least one resource. We can avoid this Condition with a protocol
This protocol has some drawbacks, because it is very expensive and time consuming.

No Preemption

The third necessary condition "no preemption" means resources are not released in the middle of the processing. There are two possible techniques which could obviate this. First, if a process already holding some resources requests an additional measure which is held by another process, the requesting process can be forced to give up the resources currently held. It would then need to wait for all the necessary resources. The second possibility is that a resource R required by one process and held by a second, could be forcibly removed from the second. This approach is not feasible for serially reasonable resources, but it can be used for resources such as memory space, registers etc which can be saved and restore later.

Avoiding Circular Wait

The circular wait condition can be prevented if resources are organized into a particular order and that resource requests have to follow this order.

Deadlock Avoidance

It is one of the methods of dynamically or online escaping from the deadlocks. In this method, if a process request for resources, the avoidance algorithm checks about the state of the system before the allocation of resources so that there can never be a circular – wait condition. The resource allocation state is defined by the number of available and allocated resources and the maximum demands of the processes.

The most quoted avoidance strategy of this type is known Banker's algorithm.
For that, ***refer to Question No. 2 (2^{nd} part) of Short Answer Type Questions.***

Long Answer Type Questions

1. a) State four necessary conditions for deadlock.
[WBUT 2007, 2008, 2011, 2013(CS), 2014(CS)]

OR,

What are the criterions for a system to be Deadlock? [WBUT 2012(IT)]

OR,

Write down all the necessary conditions of Deadlock. [WBUT 2013(IT)]

OR,

What are the necessary conditions for deadlock? [WBUT 2015(IT), 2019(IT)]

OR,

What are the four necessary conditions for deadlock to occur in a system? Explain. [WBUT 2017(IT)]

OR,

What are the necessary and sufficient conditions for deadlock to occur? [WBUT 2018(IT)]

Answer:

There are four conditions for deadlock:

i) **Mutual exclusion:** A resource can be used only by one process at a time. If another process request that resource, process must be wait until it has been released.

ii) **Hold-and-wait:** Some processes must be holding some resources in a non-sharable mile and at the same time must be waiting to acquire some more resources, which are currently held by other processes.

iii) **No preemption:** Resources granted to a process can be released back to the system only as a result of the voluntary action of that process, after the process has completed its task.

iv) **Circular wait:** Deadlocked processes are involved in a circular chain such that each process holds one or more resources being requested by the next process in the chain.

b) Consider the following snapshot of a system:

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

Answer the following question using the Banker's algorithm:

i) **What is the content of the matrix need?**

ii) **Is the system a safe state?**

iii) **If a request from process P1 arrives for (0, 4, 2, 0), can the request be granted immediately?**

[WBUT 2007, 2008, 2013(CS), 2013(IT), 2014(IT), 2017(IT), 2019(CS)]

Answer:

i) Need = Max – Allocation

The content of the need matrix = $\begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 6 & 4 & 2 \end{vmatrix}$

ii) Yes, the sequence $< P_0, P_2, P_1, P_3, P_4 >$ satisfies the safety sequence.

iii) Yes, since

(0, 4, 2, 0) ≤ Available (i.e. 1, 5, 2, 0)

(0, 4, 2, 0) ≤ Max (i.e. 1, 7, 5, 0)

2. Consider the following snapshot of a system where $r_i\,(i=1..4)$ denote resource types and P_1 to P_5 denote processes. The vector 'Available' has usual meaning.

Available

r1	r2	r3	r4
2	1	0	0

	Current allocation:				*maximum demand:*			
Process	r1	r2	r3	r4	r1	r2	r3	r4
P1	0	0	1	2	0	0	1	2
P2	2	0	0	0	2	7	5	0
P3	0	0	3	4	6	6	5	6
P4	2	3	5	4	4	3	5	6
P5	0	3	3	2	0	6	5	2

i) Is this system currently in a safe state? Justify your answer.

ii) If a request from P_3 arrives for (0, 1, 0, 0), can that request be safely granted immediately? **[WBUT 2009, 2014(CS), 2015(CS)]**

Answer:

Need matrix

$$= \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 6 & 6 & 2 & 2 \\ 2 & 0 & 0 & 2 \\ 0 & 3 & 2 & 0 \end{matrix}$$

Step-1 P_1 is selected

Since, $(0,0,0,0) \le (2,1,0,0)$

Then, available matrix = (2, 1, 0, 0) + (0, 0, 1, 2) = (2, 1, 1, 2)

Step-2 P_4 is selected

Since, $(2,0,0,2) \le (2,1,1,2)$

Then, available matrix = (2, 1, 1, 2) + (2, 3, 5, 4) = (4, 4, 6, 6)

Step-3 P_5 is selected

Since, $(0,3,2,0) \le (4,4,6,6)$

Then, available matrix = (0, 3, 3, 2) + (4, 4, 6, 6) = (4, 7, 9, 8)

Step-4 P_2 is selected

Since, $(0,7,5,0) \le (4,7,9,8)$

Then, available matrix = (4, 7, 9, 8) + (2, 0, 0, 0) = (6, 7, 9, 8)

Step-5 P_3 is selected

Since, $(6,6,2,2) \le (6,7,9,8)$

Then, available matrix = (6, 7, 9, 8) + (0, 0, 3, 4) = (6, 7, 12, 12)

Then the system sequence is $(P_1, P_4, P_5, P_2, P_3) = (6,7,12,12)$

3. Consider a system with five processes P_0 through P_4 and have three resource types A, B, C. Find out the number of instances of each resource type and retrieve the safe sequence. [WBUT 2016(CS), 2016(IT)]

	MAX			NEED			AVAILABLE		
	A	B	C	A	B	C	A	B	C
P_0	7	5	3	7	4	3	2	3	0
P_2	3	2	2	0	2	0			
P_2	9	0	2	6	0	0			
P_3	2	2	2	0	1	1			
P_4	4	3	3	4	1	1			

Answer:

	Max			Need			Available		
	A	B	C	A	B	C	A	B	C
P_0	7	5	3	7	4	3	2	3	0
P_1	3	2	2	0	2	0			
P_2	9	0	2	6	0	0			
P_3	2	2	2	0	1	1			
P_4	4	3	3	4	3	1			

Now, allocation will be = Max – Need

	A	B	C
P_0	0	1	0
P_1	3	0	2
P_2	3	0	2
P_3	2	1	1
P_4	0	0	2

We must determine whether this new system state is safe. To do so, we execute our safety algorithm and find that the sequence $<P_1, P_3, P_4, P_0, P_2>$ satisfies our safety requirement. Hence, we can immediately grant the request of process P_1.

The need of P_0 is 7, 4, 3 which is too great for the 2, 3, 0 that is available. Therefore, the safety algorithm would skip over P_0 in its search for the first process to have its finish simulated. The need of P_1 is 0, 2, 0 which nowhere exceeds 2, 3, 0. Thus, P_1 can be given its need and made to finish.

4. Explain Circular-wait for the occurrence of deadlock and how it could be prevented. Consider a system consisting of m resources of the same type being shared by n processes only one at a time. Show that the system is deadlock free if the following condition holds:

(a) The maximum need of each process is between 1 and m resources.

(b) The sum of all maximum needs is less than $m+n$.

Consider the following snapshot of a system:

Process	Allocation	Max	Available
P0	ABCD	ABCD	ABCD
P1	0012	0012	1520
P2	1000	1750	
P3	1354	2356	
P4	0632	0652	
P5	0014	0656	

Answer the following question using the Banker's algorithm:

1. **What is the content of matrix "Need"?**
2. **Is the System in a safe state?**
3. **If a request from process P1 arrives for (0, 4, 2, 0) can the request be granted immediately?** **[WBUT 2018(CS)]**

Answer:

1st Part:

Suppose N = Sum of all needi, A = Sum of all allocation, M = Sum of all Maxi. Use contradiction to prove.

Assume this system is not deadlock free. If there exists a deadlock state, them $A = m$ because there is only one kind of resource and resources can be requested and released only one at a time. From condition (b), $N + A = M < m + n$. So, we get $N + m < m + n$. So we get $N < n$. It shows that at least one process i that needi = 0. From condition (a), P_i can release at least 1 resource. So there are $n - 1$ processes sharing m resources now, condition (a) and (b) still hold. Go on the argument, no process will wait permanently, so there is no deadlock.

2nd Part: ***Refer to Question No. 1(b)(iii) of Long Answer Type Questions.***

5. Write short note on Starvation. **[WBUT 2019(CS)]**

Answer:

Starvation is a condition where a process does not get the resources it needs for a long time because the resources are being allocated to the other processes. It generally occurs in a priority based scheduling system starvation is the name given to the indefinite postponement of a process because it requires some resource is never allocated to the process.

In starvation, a process is ready to execute but it is not being allowed to execute. Here request may be differed infinitely.

MEMORY MANAGEMENT

Multiple Choice Type Questions

1. Thrashing [WBUT 2007, 2014(CS), 2016(CS)]
   a) reduces page I/O
   b) decreases the degree of multiprogramming
   c) implies excessive page I/O
   d) improves the system performance

Answer: (c)

2. Which of the following is false? [WBUT 2007, 2014(CS), 2015(IT), 2019(CS)]
   a) Segmentation suffers from external fragmentation
   b) Paging suffers from internal fragmentation
   c) Virtual memory is used only in multi-user system
   d) Segmented memory can be paged

Answer: (c)

3. Which of the following page replacement algorithms suffers from Belady's anomaly? [WBUT 2007, 2012(CS), 2013(IT), 2014(CS), 2015(CS), 2017(CS), 2019(CS)]
   a) Optimal replacement b) LRU
   c) FIFO d) Both (a) and (c)

Answer: (c)

4. Variable partition memory allocation can lead to [WBUT 2008, 2009, 2015(CS)]
   a) External fragmentation b) Internal fragmentation
   c) Both of these d) None of these

Answer: (a)

5. Virtual memory concept is supported by [WBUT 2008, 2014(CS)]
   a) demand paging b) simple segmentation
   c) simple page allocation d) both (a) and (c)

Answer: (d)

6. If a process has 32 k bytes logical address space and the page size is 2048 bytes then the number of frames of that process is [WBUT 2009, 2019(CS)]
   a) 4 b) 8 c) 16 d) 32

Answer: (c)

7. Page fault occurs when [WBUT 2009, 2015(CS), 2015(IT), 2017(CS)]
   a) the page in corrupted by application software
   b) the page is not in main memory
   c) the page is in main memory
   d) one tries to divide a number by 0

Answer: (b)

8. Which of the following schemes suffers from External Fragmentation? [WBUT 2011, 2013(IT), 2018(IT)]

a) Segmentation
b) Paging
c) Paged segmentation
d) All of these

Answer: (a)

9. To enable a process to be larger than the amount of memory allocated to it, one can use [WBUT 2011, 2014(IT), 2017(IT)]

a) Overlays b) Paging c) Compaction d) Swapping

Answer: (a)

10. TLB is a kind of [WBUT 2014(IT), 2016(IT), 2017(IT)]

a) virtual memory b) interrupt c) cache d) main memory

Answer: (c)

11. With a segmentation, if there are 64 segments, and the maximum segment size is 512 words, the length of the logical address in bits is [WBUT 2015(IT)]

a) 12 b) 14 c) 15 d) 16

Answer: (c)

12. Compaction is used to solve the problem of [WBUT 2016(CS), 2018(IT)]

a) external fragmentation
b) internal fragmentation
c) both (a) and (b)
d) none of these

Answer: (a)

13. With segmentation, if there are 64 segments and maximum segment size is 512 words, the length of bits in logical address is [WBUT 2016(IT)]

a) 12 b) 15 c) 14 d) 16

Answer: (b)

14. A memory page containing a heavily used variable that was initialized very early and is in constant use is removed when ..... page replacement algorithm is used. [WBUT 2016(IT)]

a) LRU b) LFU c) FIFO d) none of these

Answer: (c)

15. Which is the fastest of the following? [WBUT 2017(CS)]

a) Cache memory b) RAM c) CD-ROM d) Register

Answer: (d)

16. The mechanism that brings a page into memory only when it is needed, is called [WBUT 2017(CS)]

a) Segmentation
b) Fragmentation
c) Demand paging
d) page and replacement

Answer: (c)

17. Page fault frequency in an operating system is reduced when [WBUT 2018(CS)]
    a) processes tend to be of an equal ratio of the I/O-bound and CPU-bound
    b) size of pages is increased
    c) locality of reference is applicable to the process
    d) processes tend to be CPU-bound

Answer: (c)

18. _____ is a high speed cache used to hold recently referenced page table entries a part of paged virtual memory. [WBUT 2018(CS)]
    a) Translation looks a side buffer
    b) Inverse page table
    c) Segmented page table
    d) Indexed page table

Answer: (a)

19. For 3 page frames, the following is the reference string:
    7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 [WBUT 2018(CS)]
    a) 10 b) 15 c) 11 d) 12

Answer: (d)

20. Which page replacement algorithm gives the lowest Page fault rate?
    a) LRU
    b) FIFO [WBUT 2018(IT)]
    c) Optimal page replacement
    d) None of these

Answer: (a)

21. If there are 32 segments, each of size 1K, than the logical address should have [WBUT 2018(IT)]
    a) 10 bits b) 14 bits c) 15 bits d) 16 bits

Answer: (c)

22. In DMA transfer [WBUT 2019(CS)]
    a) CPU is involved actively during data transfer
    b) CPU is involved partially during data transfer
    c) DMA controller is actively involved during data transfer
    d) Both (b) and (c)

Answer: (a)

23. Page stealing is [WBUT 2019(CS)]
    a) a sign of efficient system
    b) taking larger space's for pages paged out
    c) taking page frames from other working sets
    d) one of the tuning goals

Answer: (c)

24. Where does swap space reside? [WBUT 2019(IT)]
    a) RAM b) ROM c) Disk d) Cache

Answer: (c)

Short Answer Type Questions

1. What is difference between logical address and physical address?
[WBUT 2003, 2004, 2008, 2009, 2016(IT), 2018(IT]

Answer:
Logical addresses are generated by the CPU also referred to as virtual addresses. Physical addresses are seen by memory unit. Logical and physical addresses are same in compile time and load time address binding scheme; but they are differ in execution time address binding scheme.

2. What is fragmentation? **[WBUT 2007, 2008, 2014(CS)]**
What is the problem of fragmentation and how can it be solved?
[WBUT 2007, 2008, 2016(IT)]

Answer:
1st Part:
Fragmentation happened in a dynamic memory allocation system when many of the free blocks of the memory space are too small to satisfy any request. Fragmentation are of two types: internal and external.
External fragmentation can be solved with paging. Paging avoids the considerable problem of fitting the varying sized memory chunks onto the backing store, from which most of the previous memory management schemes suffered.
Another possible solution to the problem of external fragmentation is compaction, where the goal is to shuffle the memory contents to place all free memory together in one large block.

2nd Part:
The problem of fragmentation is as follows:
In the worst situation, we could have a block of free (wasted) memory between every two processes. If all these memory were in one big free block, then we might be able to run several more processes. The selection of first-fit versus best-bit can affect the amount of fragmentation (first fit is better for some systems and best-bit is better for others).
Another factor is important that which end of a free block is allocated. No matter which algorithms are used, however, external fragmentation will be a problem. Depending on the total amount of memory storage and the average process size, external fragmentation may be either a minor or a major problem. Another problem that arises with the multiple partition allocation scheme (internal fragmentation).
The problem of external fragmentation can be solved easily through compaction. The goal is to shuffle the memory contents to place all free memory together in one large block but compaction is not always possible. Another possible rotation to the external fragmentation problem is to permit the physical address space of a process to be noncontiguous, thus allowing a process to be allocated physical memory whenever the latter is available. One way of implementing this solution is through the use of a paging scheme. Paging avoids considerable problem of fitting the varying-sized memory chunks onto the backing store.

3. Given memory partitions of 100k, 500k, 200k, 300k & 600k (in order): How would each of the first-fit, best-fit & worst-fit algorithms place processes of 212k, 417k, 112k & 426k (in order)? Which algorithm makes the most efficient use of memory? **[WBUT 2007, 2012(CS), 2013(IT), 2015(IT)]**

OR,

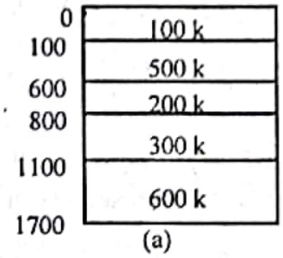
Compare best fit and first fit algorithm for memory allocation. **[WBUT 2011]**

OR,

How would each of the First Fit, Best Fit and Worst Fit algorithms place processes of 212KB, 417KB, 112KB and 426KB (in order). Which algorithm makes the most efficient use of memory? **[WBUT 2014(CS), 2015(CS)]**

Answer:

Consider the memory partitions which is shown below:



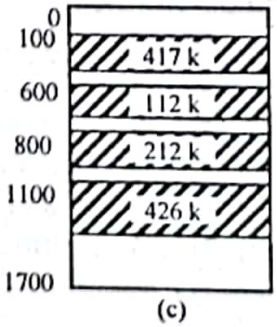
(a)

In first fit, the first hole which is large enough for use to satisfy the request so that the first process of 212k is allocated the block of 500k.

(b)

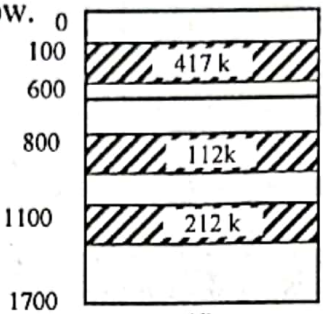
Second process of 417k has been placed in the block of 600k, the third process of 112k has been placed in the block of 200k and the last request which is 426k came in the system, then the condition will not be satisfied since there is no block which is large enough to place it.

The allocation is shown in above figure. In best fit, the smallest hole, i.e., large enough to satisfy the request is allocated so far. So, if the best fit is used 212k process which is allocated the block of 300k, 417k process has been placed in 500k, 112k process is placed in 200k block and finally, the 426k process is placed in 600k block. The memory allocation is shown below.



(c)

Finally, if worst-fit is used, the largest hole has been allocated to the requesting process. So, the request of 212k is satisfied by placing it on 600k block, 417k process is placed on the block of 500k, 112k is placed on 300k block. But 426k process cannot be allocated in the desired memory as there is no block large enough space to hold the request. The memory allocation is shown below.



(d)

The algorithm best fit is the most efficient algorithm for allocation of memory. This method creates the smallest possible hole.

4. Briefly explain Belady's anomaly.
[WBUT 2007, 2009, 2010, 2011, 2014(IT), 2017(CS), 2019(CS)]

Answer:
Belady's anomaly states that it is possible to have more page faults when increasing the no. of page frames while using FIFO method of frame management. In computer memory, information is loaded in specific sized chunks. Each chunk is referred to as a page. It requires a frame for each page it can load. A page fault occurs when a page must be loaded from memory. The following is an example of Belady's anomaly. Using 3 page frames, 9 page faults occur.

0 means	Page request	3	2	1	0	3	2	4	3	2	1	0	4
Page fault = 9	Frame 1	(3)	3	3	(0)	0	0	(4)	4	4	4	4	4
	Frame 2		(2)	2	2	(3)	3	3	3	3	(1)	1	1
	Frame 3			(1)	1	1	(2)	2	2	2	2	(0)	0

5. Give details of how paging is implemented in hardware. Explain what a Translation Lookaside Buffer (TLB) is and give details of how it works.
[WBUT 2008, 2010, 2014(CS)]

OR,

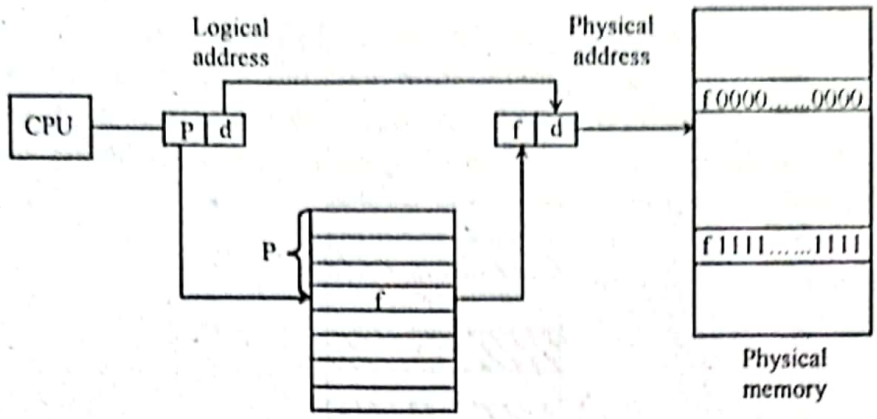
What are TLB? Draw the diagram of paging hardware with TLB. [WBUT 2012(CS)]

OR,

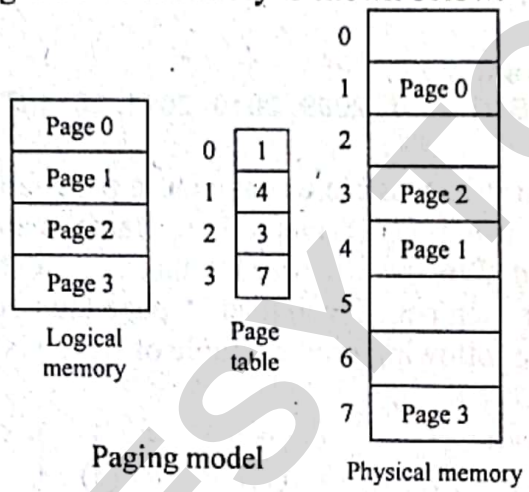
What is translation look aside buffer? Why is it used? [WBUT 2013(IT)]

Answer:

1st Part:
Traditionally, support for paging has been handled by hardware. However recent designs have implemented paging by closely integrating the hardware and operating system. The hardware support for paging is shown below.

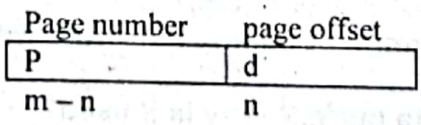


Every address generated by the CPU is divided into two parts: a page number (P) and a page offset (d). The page number is used as an index into a page table. The page table contains the base address of each page in physical memory. This base address is combined with the page offset to define the physical memory address that is sent to the memory unit. The paging model of memory is shown below:



Paging model

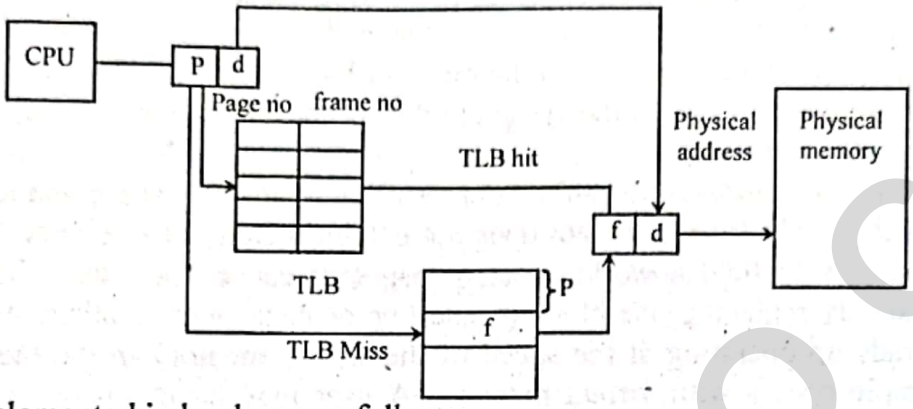
The page size is defined by the hardware. The size of a page is typically a power of 2. The selection of power of 2 as a page size makes the translation of a logical address into a page number and a page offset. If the size of the logical address space is 2^m and a page size is 2^n addressing units, then the high-order m–n bits of a logical address designate the page numbers, and the n low order bits designate the page offset.
Thus logical address is shown as:



2nd Part:

The TLB is associative high speed memory. Each entry in the TLB consists of two parts: a key (or tag) and a value. When the associative memory is presented with an item, the item is compared with all keys simultaneously. If the item is found, the corresponding value field is returned. Here the search is fast, the hardware however is expensive. Typically the number of entries in the TLB is small often numbering between 64 and 1024. The TLB is used with page tables in the following way. The TLB contains only a

few of the page-table entries. When a logical address is generated by the CPU its page number is presented to the TLB. If the page number is found, its frame number is immediately available and is used to access memory, the whole task may take less than 10% longer than it would if an unmapped memory reference were used. If the page number is not in the TLB, a memory reference to the page table must be made, known as TLB Miss. When the frame number is obtained, we can use it to access memory, known as TLB hit, this is shown below:



TLB is implemented in hardware as follows:

i) It should be represented as an array where each element of the array i.e. each entry of TLB contains a valid bit, a virtual page number, an M bit, an R bit and a page frame number.

ii) Upon every instruction generated by the CPU, the MMU will call TLB lookup, if there is a TLB entry for the virtual page should set R and M bit and return the page frame.

6. What are the advantages and disadvantages of having unequal size partitions in fixed partitioning scheme? [WBUT 2009, 2015(IT)]

Answer:

The advantages are: minimizes memory waste within a partition.

The disadvantages are:

Some queues may be empty, whereas then queues are long. A preferable approach is to employ a single queue for all processes.

The number of partitions is predefined and limits the total number of active processes in the system. Partition sizes are preset and small jobs do not nun efficiently.

7. Why are page sizes always power of 2?
[WBUT 2009, 2012(CS), 2013(CS), 2015(IT), 2016(IT), 2017(IT)]

Answer:

We know that paging is implemented by breaking up an address into a page and offset number. It is most efficient to break. The address into X page bits and Y offset bits, rather than perform arithmetic on the address to calculate the page number and offset. Because each bit position represents a power of 2, splitting an address between bits results in a page size that is a power of 2.

8. What is thrashing?
[WBUT 2009, 2010, 2013(CS), 2013(IT), 2014(IT), 2017(CS), 2017(IT), 2018(IT), 2019(CS)]

OR,

What is Thrashing? What is the cause of Thrashing? [WBUT 2015(IT)]

Answer:

Thrashing is a degenerate case that occurs when there is insufficient memory at one level in the memory hierarchy to properly contain the working set required by the upper levels of the memory hierarchy. This can result in the overall performance of the system dropping to the speed of a lower level in the memory hierarchy. Therefore, thrashing can quickly reduce the performance of the system to the speed of main memory or, worse yet, the speed of the disk drive.

There are two primary causes of thrashing: (1) insufficient memory at a given level in the memory hierarchy, and (2) the program does not exhibit locality of reference. If there is insufficient memory to hold a working set of pages or cache lines, then the memory system is constantly replacing one block (cache line or page) with another. As a result, the system winds up operating at the speed of the slower memory in the hierarchy. A common example occurs with virtual memory. A user may have several applications running at the same time and the sum total of these program's working sets is greater than all of physical memory available to the program. As a result, as the operating system switches between the applications it has to copy each application's data to and from disk and it may also have to copy the code from disk to memory. Since a context switch between programs is often much faster than retrieving data from the disk, this slows the programs down by a tremendous factor since thrashing slows the context switch down to the speed of swapping the applications to and from disk.

If the program does not exhibit locality of reference and the lower memory subsystems are not fully associative, then thrashing can occur even if there is free memory at the current level in the memory hierarchy. For example, suppose an eight kilobyte L1 caching system uses a direct-mapped cache with 16-byte cache lines (i.e., 512 cache lines). If a program references data objects 8K apart on each access then the system will have to replace the same line in the cache over and over again with each access. This occurs even though the other 511 cache lines are currently unused.

If insufficient memory is the cause of thrashing, an easy solution is to add more memory (if possible, it is rather hard to add more L1 cache when the cache is on the same chip as the processor). Another alternative is to run fewer processes concurrently or modify the program so that it references less memory over a given time period. If lack of locality of reference is causing the problem, then you should restructure your program and its data structures to make references local to one another.

9. What is dynamic loading? What is dynamic linking? How are they related? [WBUT 2012(CS), 2013(IT)] [WBUT 2013(IT)]

Answer:

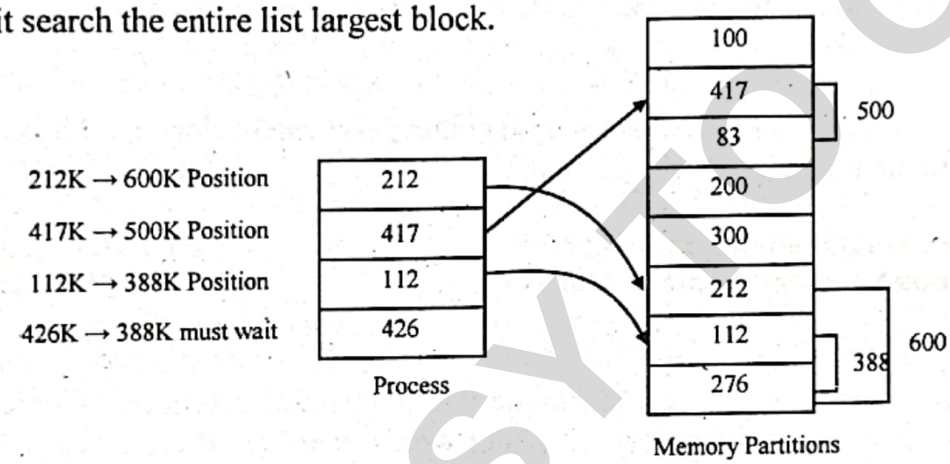
Dynamic loading means loading the library into the memory during load or run time. It can be imagined to be similar to plugins, i.e., exec., can actually execute before the

dynamic loading happens. It retrieves the addresses of functions and variables contained in the library, execute those functions or access those variables and unload the library from memory.

Dynamic linking refers to the linking that is done during load or run-time and not when the exec. is created. In case of dynamic linking the linker while creating the exec., does minimal work. For the dynamic linker to work it actually has to load the libraries too. Here it is also called linking loader.
Dynamic loading can be done at any point in program execution, but linking is performed at the time of program loading only.

10. Explain the worst fit algorithm for memory management. What are its benefits? [WBUT 2013(CS), 2017(IT)]

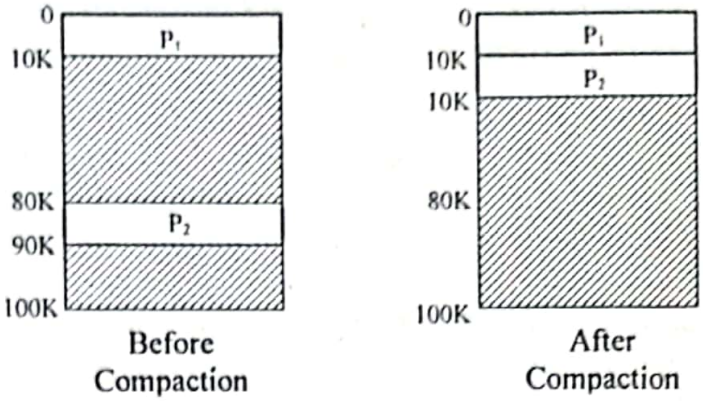
Answer:
Worst – fit search the entire list largest block.



The benefits is that the leftover block produced would be longer and potentially more useful than that produced by the best fit approach.

11. What is compaction? What are its overheads? [WBUT 2013(CS), 2016(IT)]
OR,
What is compaction? What are the drawbacks of compaction? [WBUT 2018(IT)]

Answer:
1st Part:
Compaction is a technique to place all free memory together in one large block and to reduce external fragmentation. Compaction is possible only if relocation is dynamic and is done at execution time. This is shown below:



Here P_1 and P_2 are the processes occupied. Others are holes (▨). Two holes can be compacted into one hole of size 80K. Compaction is not always possible. For processes to be able to execute in their new locations all addresses must be relocated. If relocation is static compaction cannot be done.

2nd Part:

Compaction is an overhead process because nothing else can be done until it is done. Jobs are limited to memory size.

12. What is virtual memory concept? How is it supported and implemented? **[WBUT 2013(IT), 2014(CS)] [WBUT 2013(IT)]**

Answer:

1st Part:

Virtual storage management is a technique that allows the execution of processes that may not be completely in memory. In either words, it refers to the concept whereby a process with a larger size than available memory can be loaded and executed by loading the process in parts. The main advantage of this concept is that programs can be larger than the physical memory. As well as it abstracts main memory into an extremely large, uniform array of storage, separating logical memory from physical memory. Because of this separation, the programmer needs to be aware of only the logical memory space while the operating system maintains two or more levels of physical memory space. This techniques frees programmers from concern over memory storage limitations. But the concept of virtual management technique is not easy to implement and also may decrease performance if it is used carelessly. Virtual memory is the separation of user logical memory from physical memory. This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available. Virtual memory is commonly implemented by demand paging.

2nd Part:

Virtual memory technique is implemented by Demand paging. When a process is swapped-in, the pager loads into the physical memory a set of its pages, which may be predicted to be initially needed by the process. The secondary memory holds retaining pages of the process. Subsequently, when the process needs some pages that may not be

in memory resident, the pager will load those pages into the RAM. The technique is called demand paging. For its implementation, OS needs some hardware support.

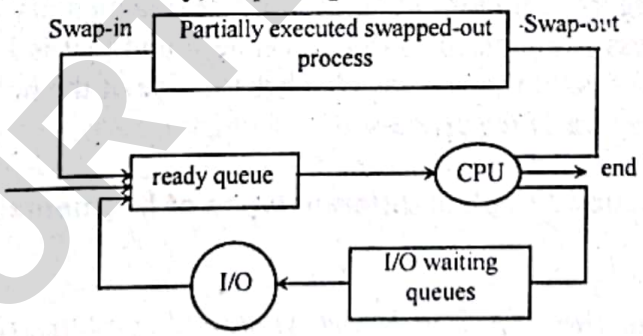
13. Why are segmentation and paging sometimes combined into one scheme?
[WBUT 2015(IT)]

Answer:
Segmentation and paging are often combined in order to improve upon each other. Segmented paging is helpful when the page table becomes very large. A large contiguous section of the page table that is unused can be collapsed into a single segment table entry with a page table address of zero. Paged segmentation handles the case of having very long segments that require a lot of time for allocation. By paging the segments, we reduce wasted memory due to external fragmentation as well as simplify the allocation.

14. The list of all passwords is kept within the operating system. Thus, if a user manages to read this list password protection is no longer provided. Suggest a scheme that will avoid this problem. What is Swap-In and Swap-Out?
[WBUT 2015(IT)]

Answer:
1st Part:
Encrypt the password internally so that they can only be accessed in coded form. The only person with access or knowledge of decoding should be the system operator.

2nd Part:
It can be advantageous to remove processes from memory and thus reduce the degree of multiprogramming. Later, the process can be reintroduced into memory, and its execution can be continued where it is left off. This scheme is called swapping. The process is swapped out and it is later swapped in, by the medium term scheduler. Swapping may be necessary to improve the process mix or because change in memory requirements has over committed available memory, requiring memory to be freed up.



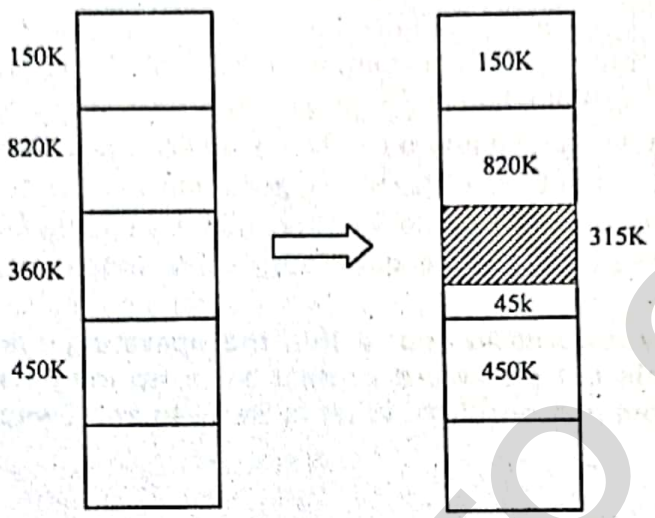
15. What is a multilevel paging? What is DMA? How does DMA increase system concurrency?
[WBUT 2015(IT)]

Answer:
Refer to Question No. 14 (a) & (c) of Long Answer Type Questions.

16. Different memory partitions of 150K, 820 K, 360 K and 450 K (in the given order) are present. Explain how best fit algorithm can be used to place a process of 315 K. **[WBUT 2018(IT)]**

Answer:

1st part:



In best fit algorithm it will search nearer to 315k i.e. 360k. But in the worst fit algorithm, the 315k will be fit in 820k. So memory space will be lost. So external fragmentation will occur.

2nd Part:

Best fit searches the entire list of holes to find the smallest hole whose size is greater than or equal to the size of the process.

In situations where best fit finds an almost perfect match, the hole that remains is virtually useless because it is so small. To prevent the creation of these useless holes, worst fit works the opposite of best fit; it always picks the largest remaining holes.

In first fit, the operating system looks at all sections of free memory compared to large or small holes. The process is allocated to the first hole found that is larger than the size of the process. Unless the size of the process matches the size of the hole, the hole continues to exist, reduced by the size of the process.

17. What is fragmentation? Explain different types of fragmentation. **[WBUT 2019(CS)]**

Answer:

1st Part: ***Refer to Question No. 2 of Short Answer Type Questions.***

2nd Part: ***Refer to Question No. 4(c) [1st part] of Long Answer Type Questions.***

18. What are the advantages and disadvantages of Paging and Segmentation? What are the difference between a page and a frame? **[WBUT 2019(CS)]**

Answer:

1st part:

Advantages and disadvantages of paging:

- No external fragmentation.
- Simply memory management algorithm.
- Swapping is easy because of equal sized pages and page frame.

Disadvantages of paging:

- Internal fragmentation.
- Page table may consume more space.

Advantages and disadvantages of segmentation:

- No internal fragmentation
- Segment tables consume less memory than page tables (only one entry per actual segment as opposed to one entry per page in Paging method).
- Because of the small segment table, memory reference is easy.
- Lends itself to sharing data among processes.
- Lends itself to protection.
- As the individual lines of a page do not form one logical unit, it is not possible to set a particular access right to a page.
- Considerable compaction overhead is incurred in order to support dynamic segment growth.
- There is difficulty in managing variable size segments on secondary storage.
- The maximum size of a segment is limited by the size of memory.
- It is necessary to develop technical or constraints to present segment thrashing.

Disadvantages of segmentation:

- Considerable compaction overhead is incurred in order to support dynamic segment growth.
- There is difficulty in managing variable size segments on secondary storage.
- The maximum size of a segment is limited by the size of memory.
- It is necessary to develop technical or constraints to present segment thrashing.

2nd part:

The logical address generated by the CPU is stored in a set of blocks known as pages and the physical address generated by the memory is stored in frames. In a paging system, programs and data stored on disk are divided into equal, fixed sized blocks called pages and main memory is divided into blocks of the same site called frames. Exactly one page can fit in one frame.

19. Write down the merits and demerits of a virtual memory system.

[WBUT 2019(CS)]

Answer:

Advantages:

i. Allocating memory is easy and chip.
ii. Any free package is ok, operating system can take first one out of list it keeps.
iii. Eliminates external fragmentation
iv. Allows demand paging and pre-paging
v. No need for consideration about fragmentation

Disadvantage:

i. Longer memory access times
ii. Inverted page tables
iii. Internal fragmentation
iv. Can be improved using TLB
v. Page table length register to limit virtual memory size.

20. Compare best fit and first fit algorithm for memory allocation. [WBUT 2019(CS)]

Answer:

First Fit:

In the first fit approach is to allocate the first free partition or hole large enough which can accommodate the process. It finishes after finding the first suitable free partition.

Advantages:

- Fastest algorithm because it searches as little as possible.

Disadvantages:

- The remaining unused memory areas left after allocation become waste if it is too smaller. Thus request for larger memory requirement cannot be accomplished.

Best Fit:

The best fit deals with allocating the smallest free partition which meets the requirement of the requesting process. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

Advantages:

- Memory utilization is much better than first fit as it searches the smallest free partition first available.

Disadvantages:

- It is slower and may even tend to fill up memory with tiny useless holes.

21. What are internal fragmentation and external fragmentation? [WBUT 2019(IT)]

Answer:

Refer to Question No. 4(c) of Long Answer Type Questions.

22. Discuss in brief: segmentation with paging. [WBUT 2019(IT)]

Answer:

Refer to Question No. 13 of Short Answer Type Questions.

23. What do you mean by locality of reference? **[MODEL QUESTION]**

Answer:

The principle of locality of reference can be stated as follows:
A logical address generated while executing an instruction in a process is likely to be in proximity of logical addresses generated during the previous few executed instructions of the process.

24. When processes exhibit good locality of reference? **[MODEL QUESTION]**

Answer:

Execution of a program is mostly sequential in nature since only 10–20% instructions are branch instruction. Thus, the next instruction to be executed typically follows the previously executed instruction in the logical address space. Reference to non-scalar data, e.g., arrays also tend to be in close proximity of previous references because processes tend to perform similar operations on several elements of an array.

25. What is the principle of locality of reference? **[MODEL QUESTION]**

Answer:

Principle of locality indicates that the logical address used in an instruction is likely to refer to a page that is in the current locality. Given this likelihood we can expect that too many page faults may not occur during operation of a process if pages in its current locality are present in memory. Thus, one can expect demand paged virtual memory systems to operate with reasonable levels of efficiency.

26. What is dirty bit? Explain it in details. **[MODEL QUESTION]**

Answer:

In order to reduce the page fault service time, a special bit called the dirty bit can be associated with each page. The dirty bit is set to 1 by the hardware whenever the page is modified (written into). When we select a victim by using a page replacement algorithm, we examine its dirty sit. If it is set, that means the page has been modified since it was swapped in. In this case we have to write that page into the backing store. However, if the dirty bit is reset, that means that page has not been modified since it was swapped in. So we do not have to write it into the backing store. The copy in the backing store is valid. Let the probability of a page being dirty be 'd'. in this case, effective access time (eat) becomes

$$eat = p * (1- d) * swap\_in + d * (swap\_in + swap\_out) + (1 - p) * emat$$

27. A demand paging system takes 100 time units to service a page fault and 300 time units to replace a dirty page. Memory access time is 11 time unit. The probability of a page fault is P. In case of a page fault, the probability of page being dirty is also P. it is observed that the average access time is 33 time units. What is the value of P then? **[MODEL QUESTION]**

Answer:

$$P\left(P\times 300+(1-P)\times 100\right)+(1-P)\times 1=3 P\left(P\times 300+(1-P)\times 100\right)+(1-P)\times 1=3$$

$$\Rightarrow P(300P+100-100P)+1-P=3 \Rightarrow P(300P+100-100P)+1-P=3$$

$\Rightarrow 200P^2 + 99P - 2 = 0$

After solving this $P \approx 0.0194$

28. If an instruction takes *i* microseconds and a page fault takes an additional *j* microseconds, what will be the effective instruction time if on the average a page fault occurs every *k* instructions. **[MODEL QUESTION]**

Answer:

Given, page fault service time = j and average memory access time = i

One page fault occurs every k instruction. Thus, page fault rate $= \frac{1}{k}$

So, effective instruction time with page fault will be

$$= \left(\frac{1}{k}\right) \times \{i + j\} + \left(1 - \frac{1}{k}\right) \times \{i\} = \frac{j}{k} + i = i + \frac{j}{k}$$

29. A demand paging system with page table held in registers, takes 5 ms to service a page fault, if empty page is to be replaced is not dirty. It takes 15ms of the replaced page is dirty. Memory access time is 1 ms. Assume we want an effective access time 2 ms and that the page to be replaced is dirty 60% of the time what is the approximate page fault rate to meet this access time requirement? **[MODEL QUESTION]**

Answer:

A demand paging system with page table held in registers takes 5 ms to service a page fault, if empty page is available or if the page is to replaced is not dirty. It takes 15 ms if the replaced page is dirty. Memory access time is 1 ms. Assume are want an effective access time 2 ms and that the page to be replaced is dirty 60% of the time. The approximate page fault rate to meet this access time requirement is 0.01%. Let the hit ratio for main memory =P, then desired time = $P\times$ main memory access time $+(1-P)$ page fault time.

$\Rightarrow 2 = P \times 1 + (1-P)(0.6 \times 15 + 0.4 \times 5) \times 10^3$

$\Rightarrow 2 = P + (1-P)(9+2) \times 10^3$

$P = \left(1/\left(11 \times 10^3\right)\right) = 0.01\%$

30. When is the page considered dirty? When is the dirty bit cleared on a page? **[MODEL QUESTION]**

Answer:

1st part:

When the page is first written i.e., when the copy in memory does not match the version on disk.

2nd part:

When the dirty page is written out again to disk i.e., after it has been replaced by another page.

31. Why is it useful to know that a page is dirty? **[MODEL QUESTION]**

Answer:

It is cheaper to pick a clean page for replacement than a dirty page. The clean page can do simply discarded when it is replaced, whereas the dirty page must be written back to disk. Therefore, the page replacement policy will prefer to replace a clean pages over a dirty page

Long Answer Type Questions

1. a) What is swapping? What is its purpose? **[WBUT 2008, 2014(CS)]**

Answer:

To replace pages or segments of data in memory, swapping is a useful technique. That enables a computer to execute programme and manipulates data files easier than main memory. Swapping is particularly useful in multiprogramming environment with a round – robin CPU- scheduling algorithm. When a quantum expires memory manager swap out the process that just finished & swap in another process to memory space that has been freed.

b) Consider the following sequence of memory references generated by a single program in a pure paging system:

10, 11, 104, 104, 170, 173, 177, 309, 245, 246, 247, 458, 364.

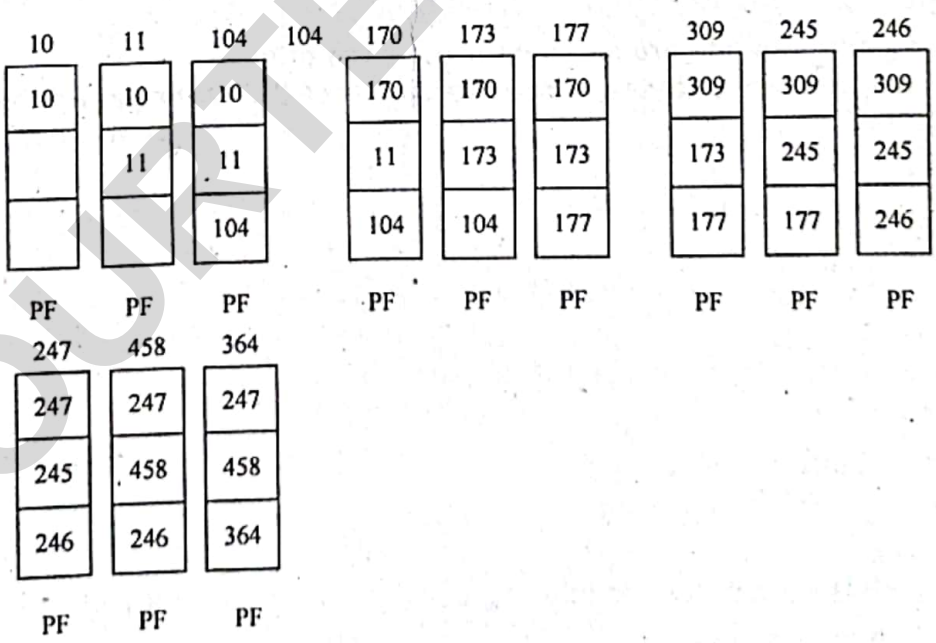
Determine the no. of page faults for each of the following page replacement policies assuming three (3) page frames are available and all are initially empty. The size of a page is 100 words:

i) LRU

ii) FIFO

iii) Optimal page replacement **[WBUT 2008, 2014(CS), 2019(CS)]**

Answer:



Page fault for FIFO = 12

Similarly for LRU = 12
Page fault for optimal page replacement = 12

c) What is effective memory access time? A paging system with the table stored in the memory.
i) If memory reference takes 200 ns, how long does a paged memory reference take?
ii) If we add TLBs and 75% hit is successful, what is the effective memory reference time (Assume that finding page-table entry in the TLBs take zero time, if the entry is there.) **[WBUT 2008]**

OR,

i) Consider a paging system with the page table stored a paged memory reference take.
ii) If we add TLBs and 75 per cent of all page-table references are found in the TLBs what is the effective memory reference time? (Assume that finding a page table Entry in the TLBs takes zero time, if the entry is there.) **[WBUT 2012(CS)]**

Answer:

Effective memory access time will be:
i) 400 nanosecond; i.e., 200 nanoseconds to access the page table and 200 nanoseconds to access the word in memory.

ii) Effective memory reference time
$= (.75 \times 200 + .25 \times 400)$ nanoseconds
$= (150 + 100) = 250$ nanoseconds.

2. a) Consider a system with a 32-bit logical address space, a two-level paging scheme, 4 byte page table entries, 1 kB pages and a 4 entry TLB. The page-table base register access time is 0 ns, TLB access time is 10 ns and memory access time is 100 ns.
i) How many address bits are needed for the page offset?
ii) How much memory in bytes is required to store the outer page table entirely in main memory? **[WBUT 2009, 2015(CS)]**

Answer:

i) $1\text{KB} = 2^{10} = \dfrac{2^{32}}{\text{Page size}}$

$\therefore \text{Page size} = \dfrac{2^{32}}{2^{10}} = 2^{22}$

$\therefore \text{Page size} = 2^{22}$ bytes

$\text{Page offset} = 22\text{bits}$

$\therefore \text{Page No} = 32 - 22 = 10$ bits

ii) The page no is further divided into two equal parts, so 5 bit for page no. and 5 bit for offset. So the logical address.

Page no		Page offset
P_1	P_2	d
5	5	22

So, the memory required = $2^5 \times 4 = 2^7$ bytes.

b) i) Given references to the following pages by a program,
0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7.
How many page faults will occur if the program has three (3) page frames available to it and uses both FIFO replacement strategy and LRU replacement strategy?
[WBUT 2009, 2015(CS), 2018(IT)]
ii) Which replacement strategy in the above performs better and why?
[WBUT 2009, 2015(CS)]

Answer:
i) FIFO strategy

Frame	0	9	0	1	8	1	8	7	8	7	1	2	8	2	7
0	0*	0	0	0	8*	8	8	8	8	8	8	8	8	8	8
1		9*	9	9	9	9	9	7*	7	7	7	7	7	7	7
2				1*	1	1	1	1	1	1	1	2*	2	2	2
	✓	✓	×	✓	✓	×	×	✓	×	×	×	✓	×	×	×

∴ Page fault = 6

LRU strategy

Frames	0	9	0	1	8	1	8	7	8	7	1	2	8	2	7
0	0*	0	0	0	8	8	8	8	8	8	1	1	8	8	8
1		9*	9	9	9	9	9	9	9	9	9	2	2	2	2
2				1	1	1	1	7	7	7	7	7	7	7	7
	✓	✓	×	✓	✓	×	×	✓	×	×	✓	✓	✓	×	×

∴ Page fault = 8
In this scheme FIFO is better.

3. a) What are the two major differences between segmentation and paging?
b) What is internal fragmentation? **[WBUT 2009, 2015(CS)]**

Answer:
a) Segment is a logical size. Visible to the user program and is of arbitrary size.
A page is a physical unit invisible to the user program and is of fixed size.
Segment size can become enormous paging has the problem of internal fragmentation.
Segment need to search for the proper hole in the main memory.
In paging scheme, since all pages have the same size, it can place the new page into any free slot.
In segmentation, placement policy is similar to variable partitioning allocation.
It uses best fit, first fit and worst fit.

b) In internal fragmentation, the space wasted internal to the allocated memory regions. Allocated memory may be slightly longer than requested memory; this size difference is wasted memory internal to a portion.

4. a) What is overlays? **[WBUT 2010, 2015(IT), 2017(CS)]**

Answer:

In operating systems, an overlay is when a process replaces itself with the code of another program. On Unix system, this is accomplished with the exec( ) system call. In Unix, the only way to run new programs is to fork the running process and then overlay the new program on top of the child. This is known as the fork-exec technique.

b) What are the advantages of segmentation over paging?
[WBUT 2010, 2015(IT), 2017(CS), 2019(CS)]

Answer:

Advantages:

- External fragmentation is eliminated
- Allocation/De allocation of fixed size units such as pages, is easier.
- Can share code and data with other registers tightly, so that only the desired portions are shared.
- Can associate protection with critical segments without protecting other items.

c) Explain the difference between internal fragmentation and external fragmentation.
[WBUT 2010, 2015(IT), 2017(CS), 2018(IT), 2019(CS)]

Which one occurs in paging system?
[WBUT 2010, 2015(IT), 2017(CS), 2018(IT), 2019(CS)]

How the problem of external fragmentation be solved?
[WBUT 2010, 2015(IT), 2017(CS), 2019(CS)]

Answer:

1st Part:

External fragmentation happens when a dynamic memory allocation algorithm allocates some memory and a small piece is leftover that cannot be effectively used. If too much external fragmentation occurs, the amount of usable memory is drastically reduced. Total memory space exists to satisfy a request, but it is not contiguous.

Internal fragmentation is the space wasted inside of allocated memory blocks because of restriction on the allowed sizes of allocated blocks. Allocated memory may be slightly larger than requested memory, this size difference is memory internal to a partition, but not being used.

2nd Part:

In paging scheme, there is no external fragmentation only have some internal fragmentation. Note that frames are allocated as units. If the memory requirement of a process do not happen to coincide with page boundaries, the last frame allocated may not be completely fall. For example, if page size is 2048 bytes, a process of 72, 766 bytes would need 35 pages plus 1,086 bytes. It would be allocated 36 frames, resulting an internal fragmentation of 2048 – 1086 = 962 bytes. In worst case, a process would need n pages plus 1byte. It would be allocated n+1 frames, resulting in an internal fragmentation of almost an entire frame.

3rd Part:

The solution to the problem of external fragmentation is compaction. The goal is to shuffle the memory contents so as to place all free memory together in one large blocks.

d) State the advantages and disadvantages of single contiguous memory allocation. **[WBUT 2010, 2015(IT), 2017(CS)]**

Answer:

Advantages:

- OS can easily move a process during execution
- OS can allow a process to grow over times
- Simple, fast hardware: two special register an add and a compare.

Disadvantages:

- Slows down hardware due to the add on every memory reference
- Can't share memory (such as program text) between process
- Process is still limited to physical memory size
- Degree of multiprogramming is very limited since all memory of all active processes must fit in memory
- Complicates memory management.

5. a) What is the purpose of modify bit in page table?

b) Consider the following page reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

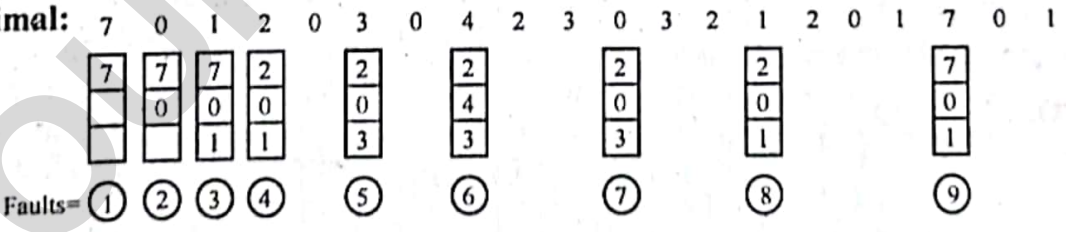
How many page faults would occur for the following replacement algorithms? Assuming 3 frames are available. Also assume that initially none of pages in main memory.

(i) Optimal replacement, (ii) FIFO replacement. **[WBUT 2010, 2014(IT), 2017(CS)]**

Answer:

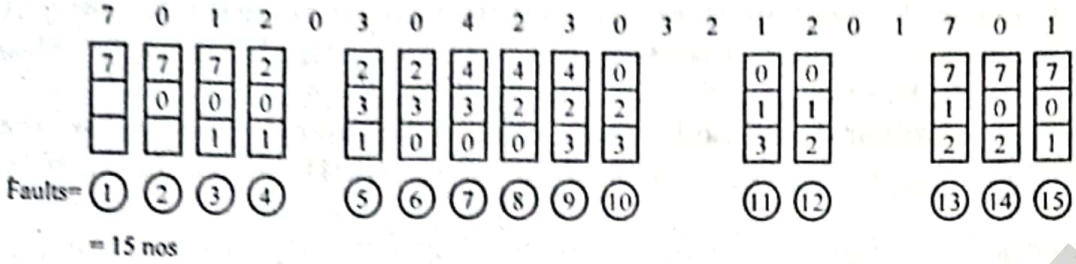
a) The modify bit is set by the memory management hardware when there is a write – memory reference to that page. When a page is first loaded into memory, the modify or dirty bit is cleared. If the dirty bit for a page selected for replacement is not set, the page has not been modified since it was loaded into memory. Only replaced pages that have been modified need be written back out to the swapping store.

b) i) Optimal:



Page faults = 09

ii) FIFO:



= 15 nos

6. a) Under what circumstances do page faults occur? [WBUT 2012(IT), 2018(CS)]

b) Describe the actions taken by the operating system when a page fault occurs. [WBUT 2012(IT), 2018(CS)]

c) Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

How many page faults would occur for the following replacement algorithms assuming four and six frames respectively? All frames are initially empty:

- **LRU replacement**
- **FIFO replacement**
- **Optimal replacement.**

[WBUT 2012(IT), 2015(IT)]

Answer:

a) A page fault occurs when an access to a page that has not been brought into main memory takes place. This situation is said to be page fault.

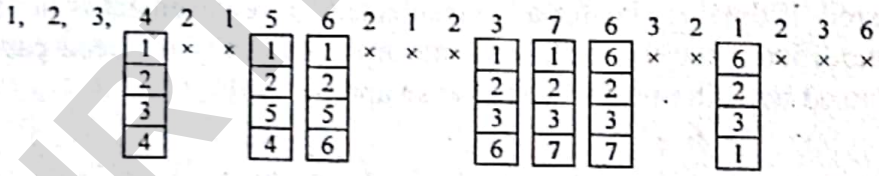
b) The operating system verifies the memory access, aborting the program if it is invalid. If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted.

c) Reference string

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

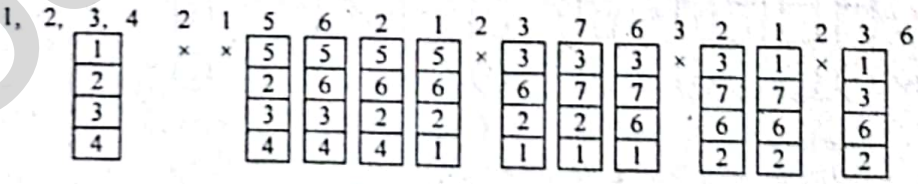
For the frames:

LRO:



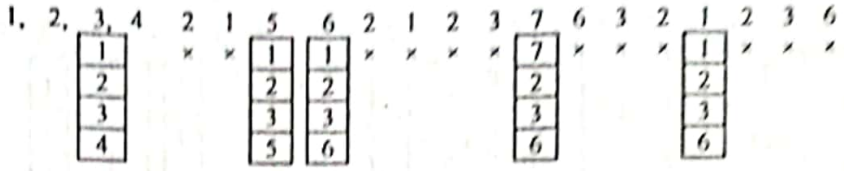
Page fault = 10

FIFO:



Page fault = 14

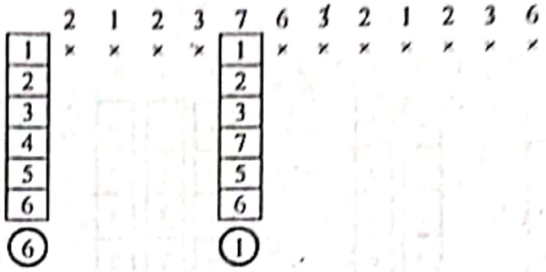
Optimal:



Page fault = 8

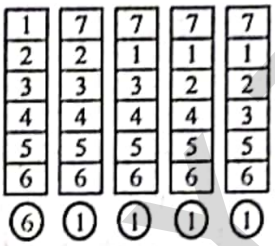
For Six (6) Frames:

LRU



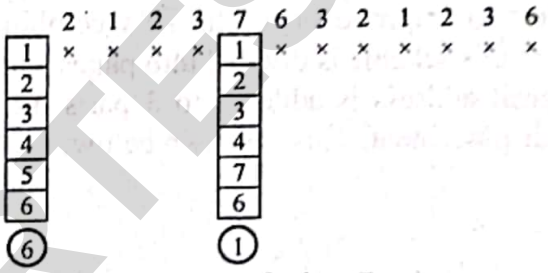
Page fault = 7

EIFO:



Page fault = 10

Optimal

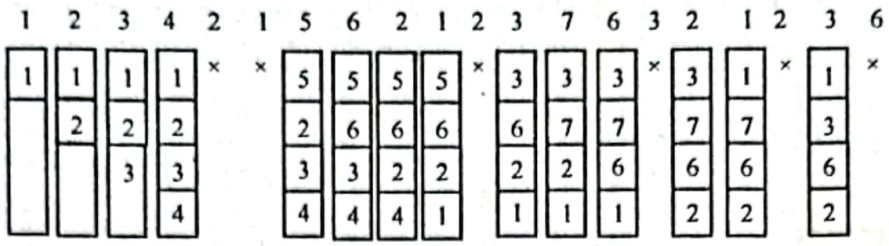


Page fault = 7

7. a) Consider the following page reference string and a memory consisting of 4 frames: 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6
Find the number of page faults considering
i) FIFO page replacement strategy
ii) LRU page replacement strategy.
Comment on the results obtained.
b) What are the disadvantages of segmentation memory management technique? How can these disadvantages be avoided if segmentation with paging is used?
[WBUT 2013(CS), 2017(IT)]

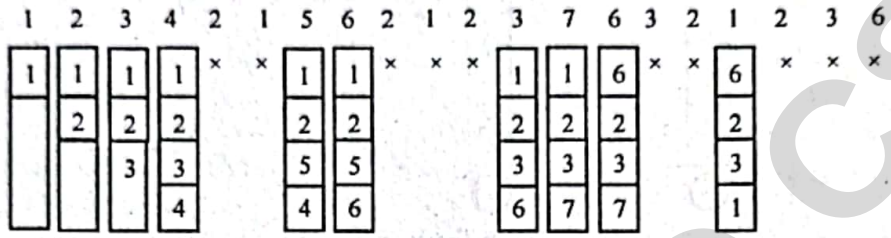
Answer:

a) i) FIFO:



Page fault = 14

ii) LRU:



∴ Page fault = 10

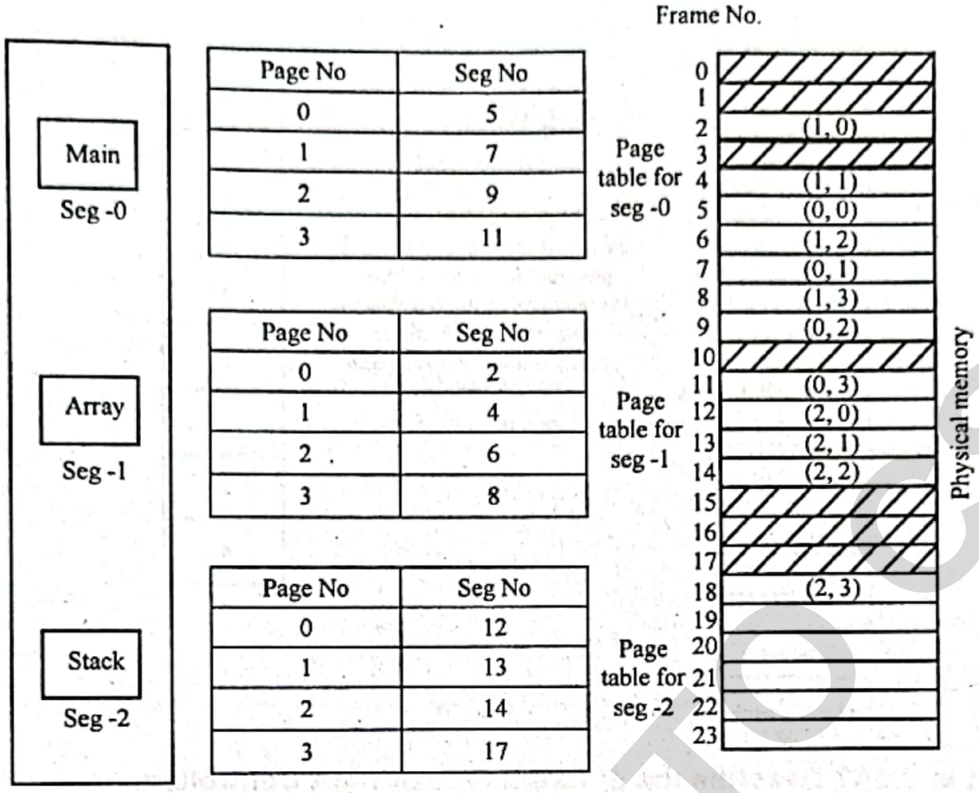
b) The disadvantages of segmentation in memory management technique are:

i) As with paging, this mapping requires two memory references per logical address. This slows the system by a factor of 2.

ii) When the no. of segments is large, then the size of the segment table will also grow. So, it cannot be accommodated in any of the registers and has to be kept in memory.

iii) Segmentation is prone to external fragmentation. This may occur when all the blocks in the memory are too small to accommodate a segment.

Both paging and segmentation have their advantages and disadvantages. So it is better to combine this two schemes to improve on each. This combined scheme is **page the segments.** Each segment in this scheme is divided into pages and each segment maintains a page table. So, the logical address is added into 3 parts like segment number, page number and the offset or displacement. This is shown below:



Logical address space

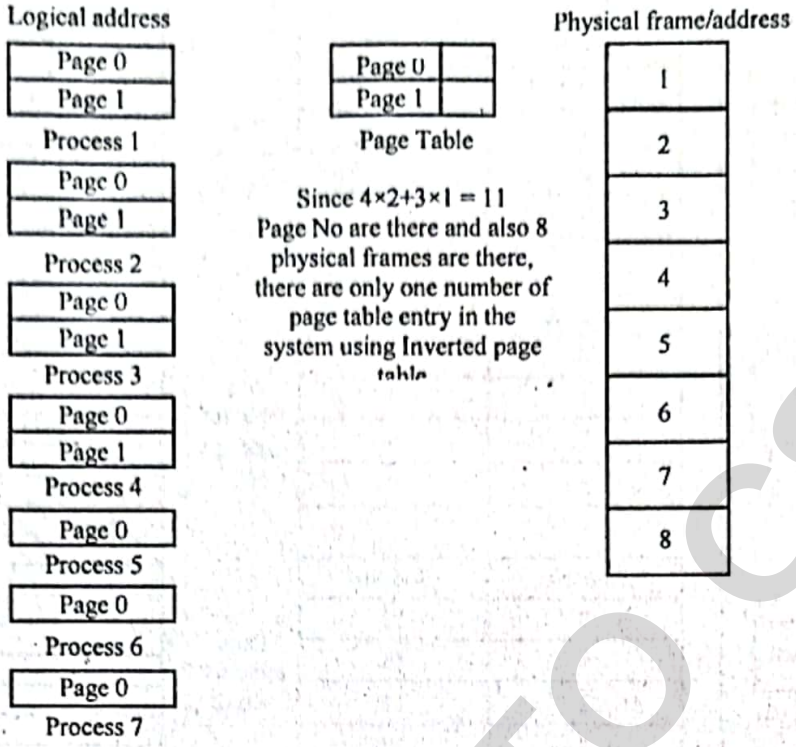
8. a) What is Direct Memory Access? How is it performed? What are its benefits? **[WBUT 2013(CS), 2014(CS), 2017(IT)]**

b) A system has 8 physical frames. There are 7 processes in the system of which 4 processes have 2 pages each and 3 processes have 1 page each. The system uses inverted page table. Find the total number of page table entries in the system. Justify your answer. **[WBUT 2013(CS), 2017(IT)]**

Answer:

a) *Refer to Question No. 14(a) of Long Answer Type Questions.*

b)



9. a) What is DMA? Describe the different type of DMA Controllers.
b) What is thrashing? How can one detect the thrashing? [WBUT 2015(CS)]

Answer:

a) 1st Part: ***Refer to Question No. 14(a) of Long Answer Type Questions.***

2nd Part:

A simple DMA controller is a standard component in PCs and bus mastering I/O boards for the PC usually contain their own high speed DMA hardware.

b) ***Refer to Question No. 9 of Short Answer Type Questions.***

10. a) What is paging? Differentiate between internal and external fragmentations. What is thrashing? [WBUT 2016(CS), 2016(IT)]

Answer:

1st Part:

Paging is a memory management technique that permits a program memory (contiguous) to be non-contiguous into physical memory and thereby allowing a program to be allocated physical memory wherever it is possible. The physical is conceptually divided into a number of fixed size blocks called as frames or page frames. The virtual address space or logical memory of a process is also broken into blocks of the same size called as pages. Wherever a program is to be executed its pages are loaded into any frame from the disks, page map table (PMT) or simply page table which contains the starting address or base address of each page stored in physical memory. If the no. of entries in PMT is quite small then registers can be used to reduce access time.

2nd Part: *Refer to Question No. 4(c) of Long Answer Type Questions.*

3rd Part: *Refer to Question No. 9 of Short Answer Type Questions.*

b) What is TLB? What do you mean by 'Belady's Anomaly'?

[WBUT 2016(CS), 2016(IT)]

Answer:

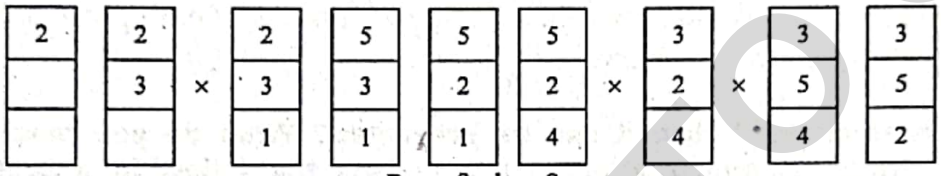
Refer to Question No. 4 & 5 of Short Answer Type Questions.

c) Having 3 physical memory frames show the behavior of LRU and FIFO and optimal page replacement algorithm for the page address string like 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2. **[WBUT 2016(CS), 2016(IT)]**

Answer:

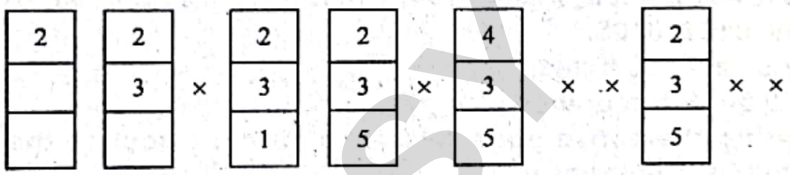
FIFO:

2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2



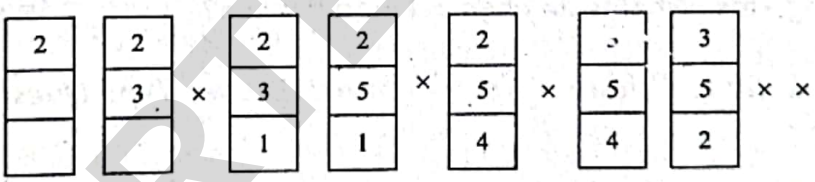
Page fault = 9

Optimal:



Page fault = 6

LRU:



Page fault = 7

11. Give memory partition of 100K, 500K, 200K, 300K and 600K (in order). How would each of the first fit, best fit and worst fit algorithm place process of 212K, 417K, 112K and 426K (in order)? Which algorithm makes the most efficient use of memory? Compare the following main memory organization schemes contiguous memory allocation, pure segmentation and pure paging with respect to the following issues:

(a) External fragmentation

(b) Internal fragmentation

(c) Ability to share code across processes **[WBUT 2018(CS)]**

Answer:

1st Part: ***Refer to Question No. 3 of Short Answer Type Questions.***

2nd Part:

Contiguous allocation with fixed sized partitions does not suffer from external fragmentation, but contiguous allocation with variable size partitions does. Pure paging does not suffer from external fragmentation, since partitions and pages are fixed in size. Segmentation does suffer from external fragmentation.

(a) Segmentation and Variable-sized partitions do not suffer from internal fragmentation, since by definition, a segment / partition is exactly as large as it needs to be. However, contiguous allocation with fixed size partitions and paging both completely filled.

(b) Contiguous allocation provides no support for code sharing. In segmentation, as long as the segments of a process do not mix text and data, we can easily share code between processes. We simply adjust the segment tables of each process. To point to the segment of code in memory.

12. What is thrashing? How it can be prevented? What do you mean by page reference string? Suppose a process accesses the following addresses at a particular time interval:
0100, 0432, 0101, 0612, 0102, 0103, 0104, 0101, 0611, 0102, 0103, 0104, 0101, 0601, 0101, 0102, 0609, 0102, 0105.
Assume a page size = 100 bytes.

(a) What will be the reference string?

(b) Considering the above page reference string, calculate the page fault rate for the following algorithms:

(i) LRU replacement.

(ii) Optimal replacement. Assume that number of frames = 3.

Explain working sets. What is its physical significance? **[WBUT 2018(CS)]**

Answer:

1st & 2nd Part: ***Refer to Question No. 9 of Short Answer Type Questions.***

3rd Part:

We evaluate an algorithm by running it on a particular string of memory references and computing the number of page faults. The string of memory references is called a page reference string.

4th Part:

Consider the Address sequence
0100, 0432, 0101, 0612, 0102, 0103, 0104, 0101, 0611, 0102, 0103, 0104, 0101, 0601, 0101, 0102, 0609, 0102, 0105
Since page size = 100 bytes
The reference in the reference string is obtained by dividing (integer division) each address reference by the page size i.e., 100 bytes.

So, the generated page reference string will be

1416111161111611611

Now, consecutive occurrences of the same reference are replaced by a single reference.

So, new page reference string will be

1 4 1 6 1 6 1 6 1 6 1

Optimal:

1	4	1	6	1	6	1	6	1	6	1
1	1	×	1	×	×	×	×	×	×	×
	4		4							
			6							

Page fault = 3

LRU:

1	4	1	6	1	6	1	6	1	6	1
1	1	×	1	×	×	×	×	×	×	×
	4		4							
			6							

Page fault = 3

5th Part:

The working set is the set of pages that are physically in memory at any one time. So the working set is a number of pages a process demands to be loaded from memory during execution. If the entire working set is in memory, the process will run without causing many faults until it moves into another execution phase. If the available memory is too small to hold the entire working set, the process will cause many page faults and run slowly since executing an instruction takes a few nanoseconds and reading in a page from the disk typically takes 10 milliseconds. At a rate of one or two instructions per 10 milliseconds, it will take ages to finish.

Working set is discussed because the CPU will automatically store accessed memory in cache, close to the processor. The working set is a nice way to describe the memory you want stored. If it is small enough, it can all fit in the cache and your algorithm will run very fast. On the OS level, the kernel has to tell the CPU where to find the physical memory your application is using (resolving virtual addresses) every time you access a new page (typically 4k in size) so also you want to avoid that hit as much as possible.

13. Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load. [WBUT 2019(IT)]

a) How does the CPU interface with the device to coordinate the transfer?

b) How does the CPU know when the memory operations are complete?

c) The CPU is allowed to execute other program while the DMA controller is transferring data. Does this process interface with the execution of the user program. If so, describe what forms of interference are caused?

Answer:

a) The device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU. To initiate a DMA transfer, The CPU first sets up the DMA registers, which contain a pointer to the source of a transfer, a pointer to the destination of the transfer and a counter of the number of bytes to be transferred. Then the DMA controller proceeds to place addressed on the bus to perform transfers while the CPU is available to accomplish other work.

b) One interrupt is generated per block to tell the device driver. That the operation has completed, i.e. The DMA controller interrupts the CPU.

c) Yes. Once DMA processes are done, an interrupt is generated which adds so the main processes lead. Thus affecting the sequence of queuing process.

14. Write short notes on the following:

a) DMA and its utility **[WBUT 2008, 2012(CS), 2013(IT)]**
b) Segmentation **[WBUT 2012(IT), 2016(CS), 2018(IT)]**
c) Multi level paging **[WBUT 2012(IT)]**
d) Belady's Anomaly **[WBUT 2015(CS), 2018(IT)]**
e) Hierarchical page table **[WBUT 2015(IT)]**
f) External Fragmentation & Internal Fragmentation **[WBUT 2016(IT)]**

Answer:

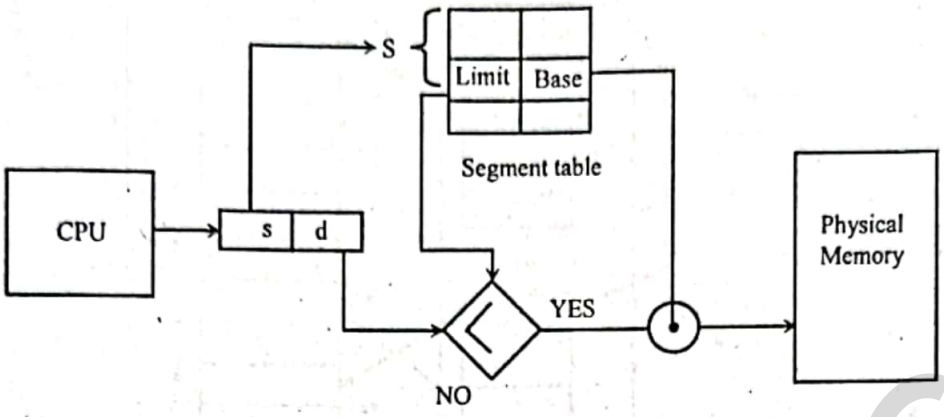
a) DMA & its utility:

DMA is a technique which used to transfer data between memory and I/O devices with less effort on behalf of the operating system. The DMA unit has access to the data bus and can transfer data autonomously in and out of memory. In practice, a program would instruct the DMA unit to, say, transfer a specified block of data from memory to a peripheral device. The DMA unit operates by suspending the CPU and accessing the memory system itself to obtain the data required. This technique is called Cycle stealing, because machine cycles are effectively stolen from the CPU and used by the DMA unit to transfer data along the data bus. Note that, DMA is not an interrupt; the current program's context is not saved and the CPU does not do something else.

b) Segmentation:

Generally a user prefers to view system memory as a collection of variable-sized segments rather than a linear array of words. Segmentation is a memory management scheme that supports the user view of memory. A logical address space is a collection of segments. Each segment has a name and a length. Each address specify both segment name and the offset. We now define a mechanism to map two-dimensional user-defined addresses into one-dimensional physical addresses, and is done in form of the segment table. A logical address consists of a segment number S and an offset d. The segment number S is used as an index into the segment table. Each entry in the segment table has a segment base, which points where the physical memory begins and a segment limit which points where the physical memory ends. Therefore, the offset d must be between 0

and the limit; otherwise we trap to the operating system indicating that the logical addressing attempts beyond the end of the physical memory segment. This is shown below.



Trap; addressing error

c) Multi level paging:

The solution to the large memory requirements of page tables is to use multilevel paging. In this method a virtual address is divided into three or more sections, with all but the last section being page numbers in different page table, and the last one being the offset. In two-level paging, the first section is the page number in a top level page table and is used to look up a second level page table. The second sections is the page number in the second level table and is used to look up the physical page number.

d) Belady's Anomaly: ***Refer to Question No. 4 of Short Answer Type Questions.***

e) Hierarchical page table.

Hierarchical page tables break up the logical address space into multiple page tables. A simple technique is a two level page table.

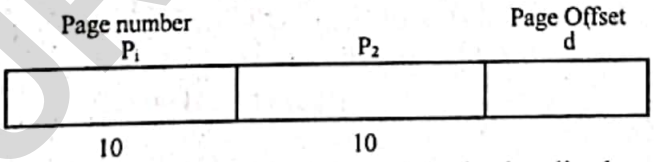
Example:

A logical address (on 32-bit machine with 4k page size) is divided into

i) a page number consisting of 20 bits

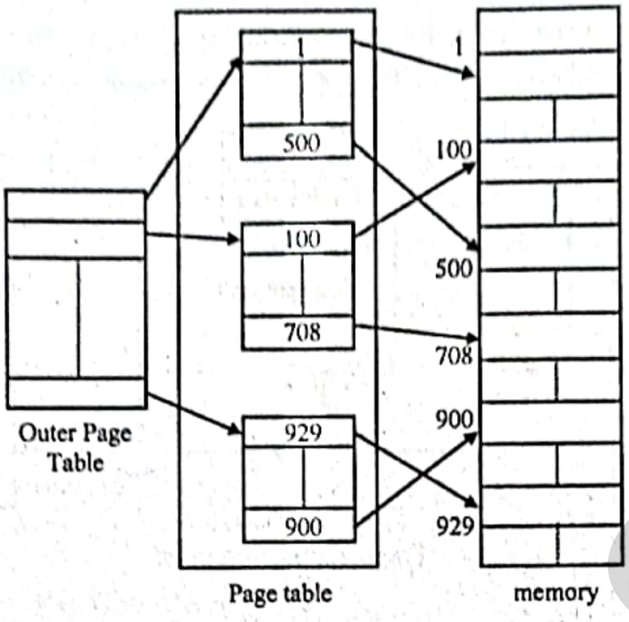
ii) a page offset consisting of 12 bits.

Since the page table is paged, the page number is further divided into a 10 bit page number and a 10 bit offset. Thus, a logical address is as



Where P_i is an index into the outer page table and P_2 is the displacement within the page of the outer page table.

Two level page table:



f) External Fragmentation & Internal Fragmentation:

Refer to Question No. 4(c) of Long Answer Type Questions.

INPUT / OUTPUT HARDWARE

Multiple Choice Type Questions

1. Disk I/O generally done in terms of **[WBUT 20013(CS)]**

a) sectors b) bytes c) blocks d) bits

Answer: (c)

2. Which of the following system call is used for opening or creating a file? **[WBUT 2019(IT)]**

a) Read b) Write c) Open d) Close

Answer: (c)

Short Answer Type Questions

1. Differentiate between Blocking *vs.* Non-Blocking input-output. **[WBUT 2013(CS)]**

OR,

Differentiate between Blocking Non-Blocking input-output. **[WBUT 2017(IT)]**

Answer:

Blocking- Here, process suspended until I/O completed.

It is easy to use and understand. It is insufficient for some needs Nonblocking- Here I/O call returns as much as available.

- It is implemented via multithreading
- It returns quickly with count of bytes read or written.

2. What are the difference between a trap and an interrupt? What is the use of each function? **[WBUT 2016(CS)]**

Answer:

1st Part:

An interrupt is generally initiated by an I/O device, and causes the CPU to stop what it is doing, save its context, jump to the appropriate interrupt service routine, complete it, restore the context, and continue execution.

A trap is usually initiated by the CPU hardware. Whenever the trap condition occurs, the CPU stops what it is doing, saves the context, jumps to the appropriate trap routine, complete it, restores the context and continue execution.

2nd Part:

Interrupt are used for external devices to signal the processor to execute some service code.

Traps are used both for control transformation between the user code and the operating system via system calls and for error handling.

The CPU uses this to get the serial device interrupt service routine, which is then executes as

Interrupts are hardware interrupts, while traps are software invoked interrupts.
In some usages, the term trap refers specifically to an interrupt intended to initiates a context switch to a monitor program.

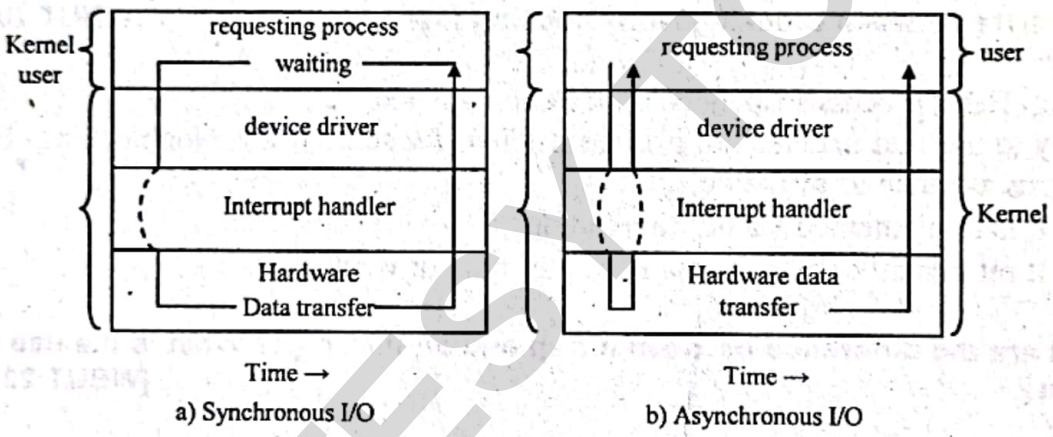
Long Answer Type Questions

1. Write a short note on Blocking and Non-blocking I/O. **[WBUT 2013(IT)]**

Answer:

With blocking I/O, a process is moved to the wait queue when an I/O request is made and moved back to the ready queue when the request completes, allowing other process to rum in the machine.

With non-blocking I/O, the I/O request returns immediately, whether the requested I/O operation has occurred or not. This allows the process to check for available data without getting hung completely if it is not there. One approach for programmers to implement non-blocking I/O is to have a multithread application, in which one thread makes blocking I/O calls, while other threads continue to update the screen or perform other tasks. Two I/O methods are shown below.

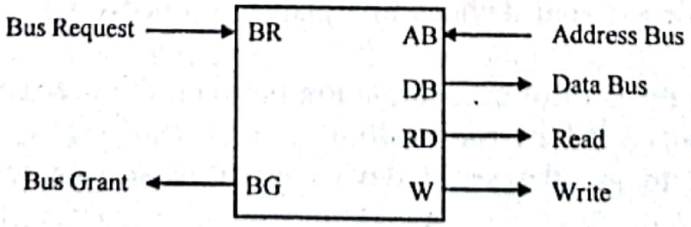


2. Write short note on Data transfer method of I/O devices. **[WBUT 2019(CS)]**

Answer:

Data transfer between CPU and the I/O devices may be done in different modes. Steps involved are:

- Buffer the byte into the buffer.
- Inform the CPU that the device has the 1 byte to transfer (i.e. bus grant request)
- Transfer the byte (at system bus speed)
- Replace the control often bus back to CPU. Fig. shows-



FILE MANAGEMENT

Multiple Choice Type Questions

1. In a tree-structured directory, the series of directory names that culminates in a file name is referred to as the **[WBUT 2016(IT)]**

a) Path name
b) Working Directory
c) Symbolic name
d) None of these

Answer: (a)

Short Answer Type Questions

1. Explain with diagram different File Allocation methods. **[WBUT 2014(CS)]**
Suppose a disk has size 128 GB, and blocks are of size 64 KB. If all block numbers are stored as 4-byte integers, how large must a main-memory file-allocation table (FAT) be?

Answer:

1st Part:

One main problem in file management is how to allocate space for files so that disk space is utilized effectively and files can be accessed quickly. Three major methods of allocating disk space are:

- Contiguous Allocation
- Linked Allocation
- Indexed Allocation.

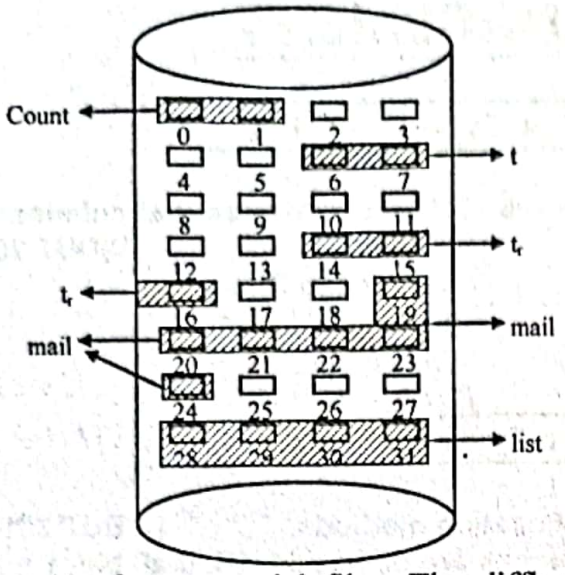
Each method has its advantages and disadvantages. Accordingly, some systems support all three (e.g. Data General's RDOS). More commonly, a system will use one particular method for all files.

In contiguous allocation each file occupies a set of consecutive addresses on disk and each directory entry contains file name, starting address of the first block, length in blocks etc. It has usual dynamic storage application problem.

e.g.,

Only starting location and length (no of blocks) is required.

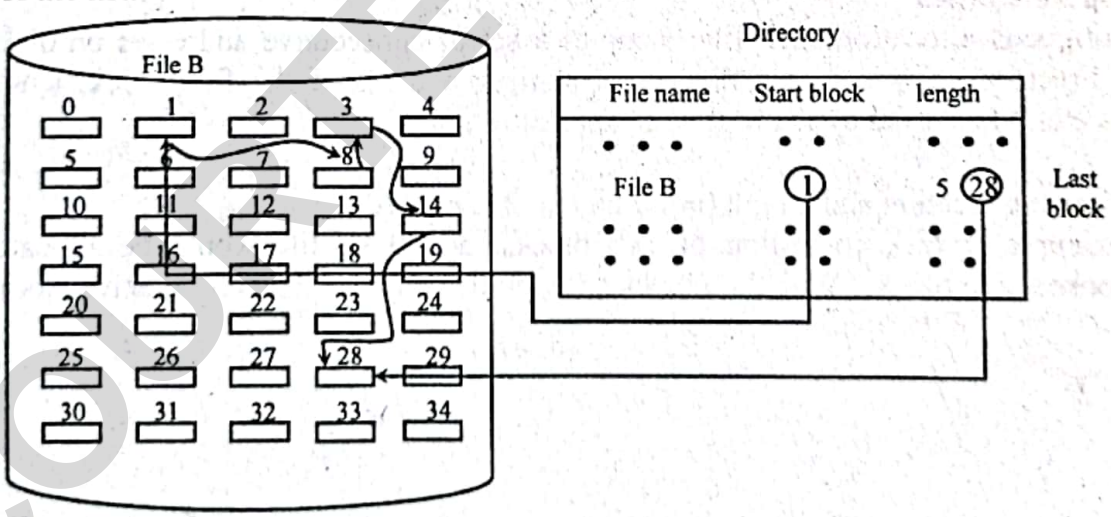
For example, a disk consisting of 1kb blocks, a 100 kb file would be allocated in 100 consecutive blocks. With 2 kb blocks, it would be allocated 50 consecutive blocks.



Directory		
File	start	length
Count	0	2
t_r	14	3
mail	19	6
list	28	4
t	6	2

It is suitable for sequential files. The difficulty is that it is not suitable to find the contiguous free blocks in the disk. Other problem is external fragmentation – which means that some free blocks could happen between two files.
It complicates "mapping around" the bad block, which can also continue in fragmentation by breaking the contiguity of disk areas. Moreover, the method is very simple to implement as contiguous chunk of space is avoidable easily.

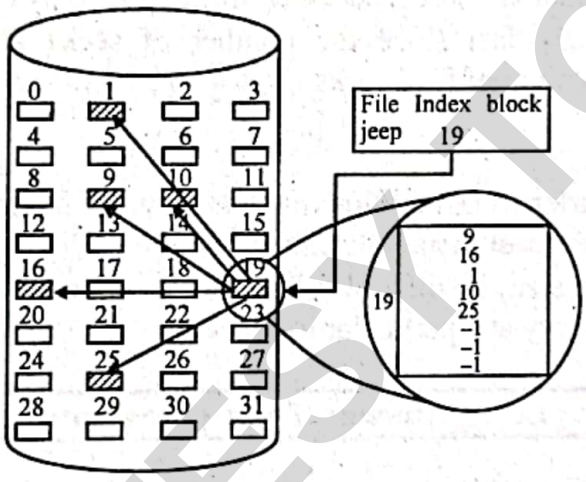
Linked file allocation technique:
With the linked allocation approach, disk blocks of a file are chained together with a linked list. The directory entry of a file contains a pointer to the first block and a pointer to the last block. To create a file, we create a new directory entry and the pointers are initialized to nil. When a write occurs, a new disk block is allocated and chained to the end of the list.



File blocks are chained in a linked list. The directory entry has pointers to the first and last file blocks. Append is difficult to do without the last pointer.

Advantages:
File size does not have to be specified. No external fragmentation.
Disadvantages:
It does sequential access efficiently and is not for direct access. Each block contains a pointer, wasting space. Blocks scatter everywhere and a large number of disk seeks may be necessary.

Indexed allocation method of disk space
Linked allocation solves the external fragmentation and size declaration problems of contiguous allocation. However, in the absence of a FAT, linked allocation cannot support efficient direct access, since the pointers to the blocks are scattered with the blocks themselves all over, the disk and musk be retrieved in order. Indexed allocation solves this problem by bringing all the pointers together into one location i.e. index block each file has its own index block, which is an array of disk block addresses. The i^{th} entry in the index block points to i^{th} block of the file. The directory contains the address of the index block shown below.



To find and read the i^{th} block, we use the pointer in the i^{th} index block entry. This scheme is similar to the paging scheme. When the file is created, all pointers in the index block are set to nil. When the i^{th} block is first written, a block is obtained from the free space manager, and its address is put in the i^{th} index block entry. Index allocation supports direct access without suffering from external fragmentation. Index allocation does suffer from wasted space, however.

2^{nd} Part:
The number of table entries is 128 GB divided by 64 KB, and each takes 4 bytes. Thus the amount of main memory space required by the FAT is 4×2 = 8MB

2. Describe structure of FAT file system. **[WBUT 2018(IT)]**
Answer:
File Allocation Table (FAT) file system is a simple file system originally designed for small disks and simple folder structures. The FAT file system is named for its method of organization. The file allocation table, which resides at the beginning of the volume.

Structure of FAT file system:
The fig. below illustrates how the FAT file system organizes a volume.

Partition Boot sector	FAT 1	FAT 2 duplicate	Root folder	Other folder and all files.

3. Explain what is Indexed Allocation of file space on disk. What are the advantages and disadvantages of contiguous allocation? [WBUT 2019(CS)]

Answer:

1st Part:

Linked allocation does not support random access of files, since each block can only be found from the previous. Indexed allocation solves this problem by bringing all the pointers together into an index block.

2nd Part:

Advantages:

- Both the sequential and direct accesses are supported by this.
- This is extremely fast since the number of seeks are minimal because of contiguous allocation of file blocks.

Disadvantages:

- This method suffer from both internal and external fragmentation. This makes it in-efficient in terms of memory utilization.
- Increasing file size is difficult, because it depends on the availability of contiguous memory at a particular instance.

Long Answer Type Questions

1. a) Briefly explain different free space management techniques. [WBUT 2010, 2014(IT)]

b) If the size of each data block is 512 bytes in Unix file system, assuming the size of a pointer is 4 bytes. Find the maximum size of a file when Inode block contains 10 direct pointers, 1 single indirect pointer, 1 double indirect pointer and 1 triple indirect pointer. [WBUT 2010, 2014(IT)]

Answer:

a) Free space management is a technique to reuse the space from deleted files for new files, if possible. To keep track of free disk space, the system maintains a free space list. The free space list records all free disk blocks when a file is deleted, its disk space is added to the free space list.

Frequently, the free space list is implemented as a **bit map** or **bit vector**. Each block is represented by 1 bit. If the block is free, the bit is 1 and if the block is allocated, the bit is 0. For example, consider a disk where blocks 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 17, 18, 25, 26 and 27 are free and the rest of the blocks are allocated. The free space bit map would be 00111100111111000110000001110000000.... . The main advantage of this approach is its relative simplicity and its efficiency in finding free blocks on the disk. Unfortunately, bit

vectors are inefficient unless the entire vector is kept in main memory. Another approach to free space management is to **link together** all the free disk blocks. Keeping a **pointer** to the first free block in a special location on the disk and caching it in memory. The first block contains a pointer to the next free disk block and so on. However, this scheme is not efficient; to traverse the list, we must read each block which requires substantial I/O time. Fortunately, traversing the free list is not a frequent action.

b) Since each i-node contains 13 block points i.e., 10 direct + 1 single indirect + 1 double indirect + 1 triple indirect pointer.

Step 1:
10^{th} Pointer: First 10 pointers point to a data block (each 512 bytes) of a file, i.e., 512×10 = 5120 bytes.

Step 2:
If the file is bigger than 10 blocks (5120 bytes), the 11^{th} pointers points to a single indirect block, which contains 128 pointers to 128 more data blocks i.e.,

$$5120+\frac{512}{4}\times 512 \qquad (\because \text{ size of a pointer is 4 bytes})$$

$$=5120+65536=70,656$$

So it can support files upto 70,656 bytes. If the file is bigger that, the 12^{th} pointer points to a double indirect block, which contains 128 pointers to 128 more single indirect blocks.

Step 3:

$$70656+\frac{512}{4}\times\frac{512}{4}\times 512=70656+8388608=8,459,264 \text{ byte}$$

If the file is bigger than that, the 13^{th} pointer points to a triple indirect block which contains 128 pointers to 128 more double indirect blocks so, **Step 4** will be

$$=8459264+\frac{512}{4}\times\frac{512}{4}\times\frac{512}{4}+512$$

$$=8459264+1073741824$$

$$=1,082,201,088 \textit{ bytes}$$

2. a) What are the advantages and disadvantages of linked file Allocation Technique?
b) How does Indexed file Allocation Technique overcome the above disadvantages? **[WBUT 2013(CS), 2017(IT)]**

Answer:

a) ***Advantages:***

i) No external fragmentation with linked allocation.
ii) Any free block can be used to satisfy a request.

iii) There is no need to declare the size of a file when that file is created.
iv) A file an continue to grow as long as there are free blocks.

Disadvantages:
i) It is inefficient to support direct access.
ii) It is effective only for sequential access files.
iii) Loss reliability.

b) The indexed file allocation method is the solution to the problem of both contiguous and linked allocation. This is done by bringing all the pointers together into one location called the index block. It supports direct access i.e. the ith entry in the index block points to the i^{th} sector of the file. It is reliable. Any free block anywhere on the disk may satisfy a request for more space.

3. What is a record in a file? For a file having multiple records what are the different indexing strategies there? Explain each strategy very briefly with relative advantages and disadvantages. **[WBUT 2016(CS), 2016(IT)]**

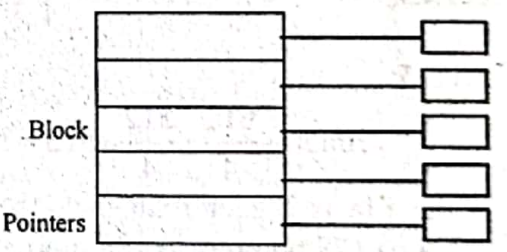
Answer:

1st Part:

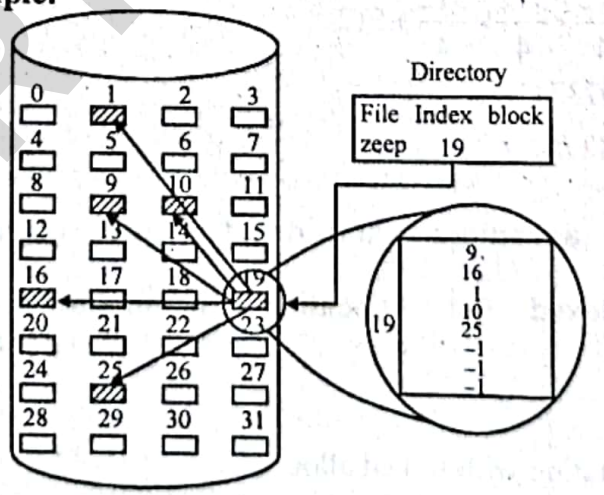
Records in file are made up with field, each of which contains one item of information. A set of records constitutes a file. For example, a personal file contains records such as name, address, pin code, phone no, etc.

2nd Part:

File has array of pointers (index) to block.
Allocate block pointers contiguously to metadata and must set max length when file created. Allocate pointers at creation, allocate blocks on demand. Maintain multiple tests of block pointers and last entry points to next block of pointers.



Index allocation example:



Advantages:

- Easy to implement
- No external fragmentation
- Files can be easily grown with the limit of the array size.
- Fast random access.

Disadvantages:

- Large space overhead (index)
- Sequential access may be slow.
- Must allocate contiguous block for fast access.

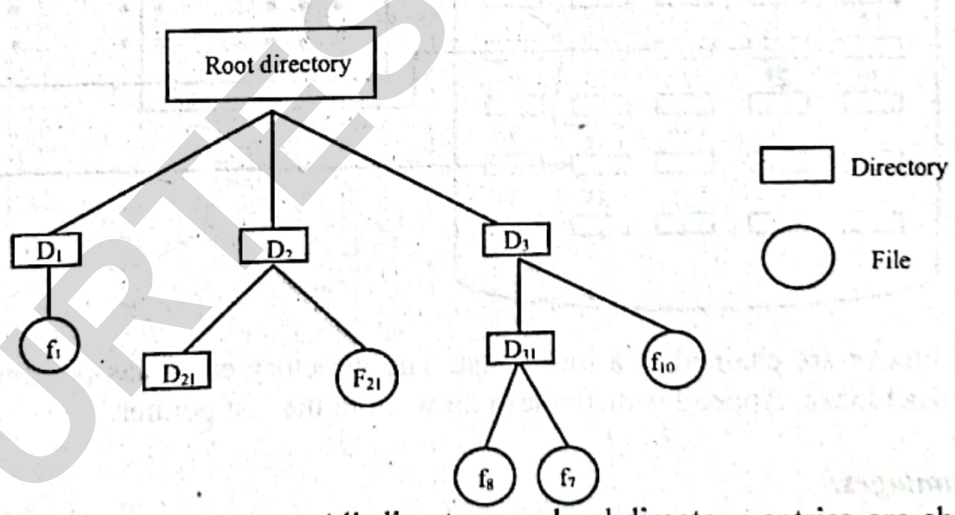
4. Discuss file structure and directory structure. Discuss various techniques for directory implementation. Discuss the file allocation methods. [WBUT 2019(IT)]

Answer:

1st Part:

File structure refers to logical structuring of the record determined by the sequence in which they are accessed. Logical definition of a file is that it is a set of logical records. It is allocated storage space on a disk in terms of physical blocks. Normally, the block size is of 512 bytes. All basic read/write operations are in terms of blocks. In a directory structure, a directory groups the files owned by a user. It contains information about files and thus is central to the functioning of the file system. It contains information like file attributes, its location and its ownership. It can be viewed as a symbol table that translates the filenames into their directory entries.

2nd part:



A hierarchical directories are shown. All directory and subdirectory entries are shown, which define the totality of system files. This is a better way to organize files. Directories are stored on disks, so the efficiency of directory operations is affected by the number of disk accessed required to complete a search, to append or remove an entry. Hence directories are organized in a way that minimize the search, addition and deletion time. Various system calls are used for directory implementation like create a file where a

corresponding entry is made in the directory, list directory knows the list of files in the directory, close directory, after reading a directory it should be closed. Directories can be renamed.

3rd part: ***Refer to Question No. 1(1st Part) of Short Answer Type Questions.***

5. Write short notes on the following:

a) Linked file allocation technique **[WBUT 2009, 2012(CS), 2013(IT), 2018(CS), 2018(IT)]**

b) Free-space management technique (any *one*) of disk **[WBUT 2013(IT)]**

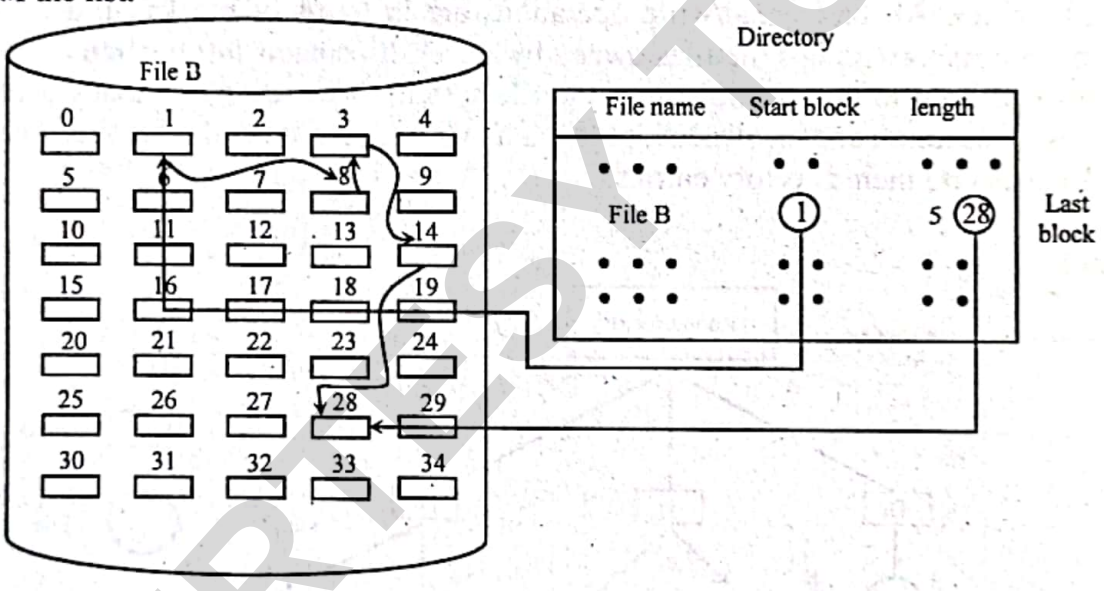
c) Indexed allocation method of disk space **[WBUT 2015(IT)]**

d) *i*-node **[WBUT 2016(CS), 2018(IT)]**

Answer:

a) Linked file allocation technique:

With the linked allocation approach, disk blocks of a file are chained together with a linked list. The directory entry of a file contains a pointer to the first block and a pointer to the last block. To create a file, we create a new directory entry and the pointers are initialized to nil. When a write occurs, a new disk block is allocated and chained to the end of the list.



File blocks are chained in a linked list. The directory entry has pointers to the first and last file blocks. Append is difficult to do without the last pointer.

Advantages:

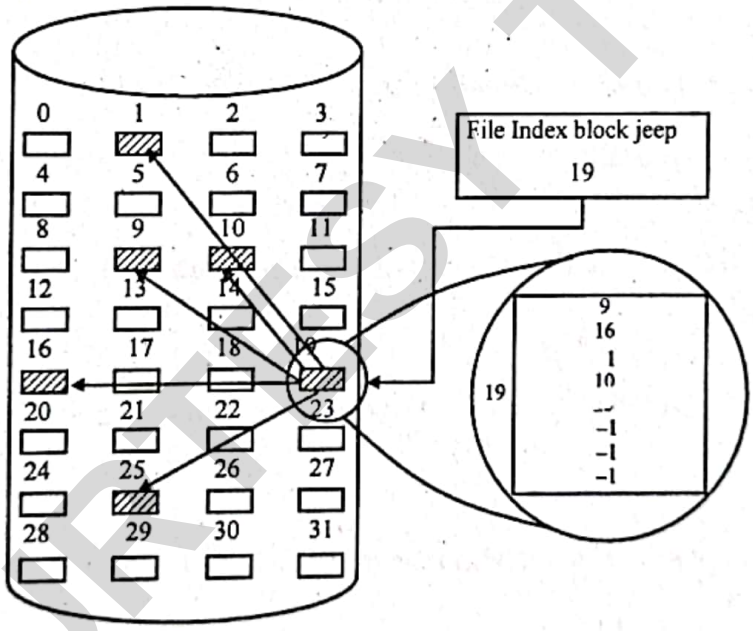
File size does not have to be specified. No external fragmentation.

Disadvantages:

It does sequential access efficiently and is not for direct access. Each block contains a pointer, wasting space. Blocks scatter everywhere and a large number of disk seeks may be necessary.

b) Free-space management technique (any *one*) of disk:
Refer to Question No. 1(a) of Long Answer Type Questions.

c) Indexed allocation method of disk space:
Linked allocation solves the external fragmentation and size declaration problems of contiguous allocation. However, in the absence of a FAT, linked allocation cannot support efficient direct access, since the pointers to the blocks are scattered with the blocks themselves all over, the disk and musk be retrieved in order. Indexed allocation solves this problem by bringing all the pointers together into one location i.e. index block each file has its own index block, which is an array of disk block addresses. The i^{th} entry in the index block points to i^{th} block of the file. The directory contains the address of the index block shown below.
To find and read the i^{th} block, we use the pointer in the i^{th} index block entry. This scheme is similar to the paging scheme. When the file is created, all pointers in the index block are set to nil. When the i^{th} block is first written, a block is obtained from the free space manager, and its address is put in the i^{th} index block entry. Index allocation supports direct access without suffering from external fragmentation. Index allocation does suffer from wasted space, however.



d) *i*-node:
An i-node is a data structure on a file system on LINUX and other UNIX like operating systems that stores all the information about a file except its name and its actual data. When a file is created, it is assigned both a name and an i-node number, which is an integer that is unique within the file system. Both the file names and their corresponding i-node numbers are stored as entries in the directory that appears to the user to contain the files. i.e. the directory associates file names with i-nodes. Whenever a user or a program refers to a file by name, the operating system uses that name to look up the corresponding i-node. The i-node numbers and their corresponding i-nodes are held in i-node tables.

The concept of i-nodes is particularly important to the recovery of damaged file systems. When parts of the i-nodes are lost, they appear in the lost & found directory within the partition in which they once existed. The operating system obtain a files i-node number and information through the use of system call named **stat.** A file's i-node number can easily be found by using the **ls** command.

DISK MANAGEMENT

Multiple Choice Type Questions

1. RAID configuration disks are used to provide [WBUT 2007, 2014(CS), 2016(CS)]
   a) fault tolerance
   b) nearest cylinder next
   c) high data density
   d) none of these

Answer: (a)

2. The time to move the disk arm to be desired cylinder in a hard disk is known as [WBUT 2012(CS), 2016(CS)]
   a) Rotainaly latency
   b) Positioning time
   c) Indexed
   d) Hashed

Answer: (a)

3. Which of the following is crucial time while accessing data on the disk? [WBUT 2013(IT)]
   a) seek time
   b) rotational time
   c) transmission time
   d) waiting time

Answer: (a)

4. Time required of read-write head to move to desired cylinder is [WBUT 2013(IT)]
   a) transfer time
   b) seek time
   c) rotational latency
   d) none of these

Answer: (b)

5. Where does the swap space reside? [WBUT 2014(IT), 2017(IT), 2018(IT)]
   a) RAM
   b) DISK
   c) ROM
   d) on-chip cache

Answer: (b)

6. The smallest possible unit of disk storage is [WBUT 2014(IT), 2017(IT)]
   a) word
   b) segment
   c) block
   d) extent

Answer: (c)

7. The total time to prepare a disk drive mechanism for a block of data to be read from is its [WBUT 2018(CS)]
   a) Access time
   b) Seek time
   c) Latency plus seek time
   d) Access time plus seek time plus transmission time

Answer: (c)

Short Answer Type Questions

1. a) What is seek time? What is rotational latency?
b) What are the advantages of SCAN disk scheduling technique over circular SCAN disk scheduling technique? **[WBUT 2013(CS), 2017(IT)]**

Answer:

a) 1st Part:

Seek time is defined as the time required to move the disk arm to the required track. It consists of two key components:

i) The initial startup time

ii) The time taken to traverse the tracks that have to be crossed once the access arm is up to speed.

2nd Part:

Rotational latency is defined as the time required to reach the desired sector by the read / write head. Magnetic disks have rotational speed in the range 5400 to 10,000 r.p.m.

b) i) The average head movements in scan is less than c-scan scheduling algorithm.

ii) The c-scan increases the total seek time but scan does not.

2. Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current head position of the disk is at 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170
Calculate the average cylinder movements for the following algorithms:
i) SSTF
ii) C-SCAN
iii) SCAN **[WBUT 2019(CS)]**

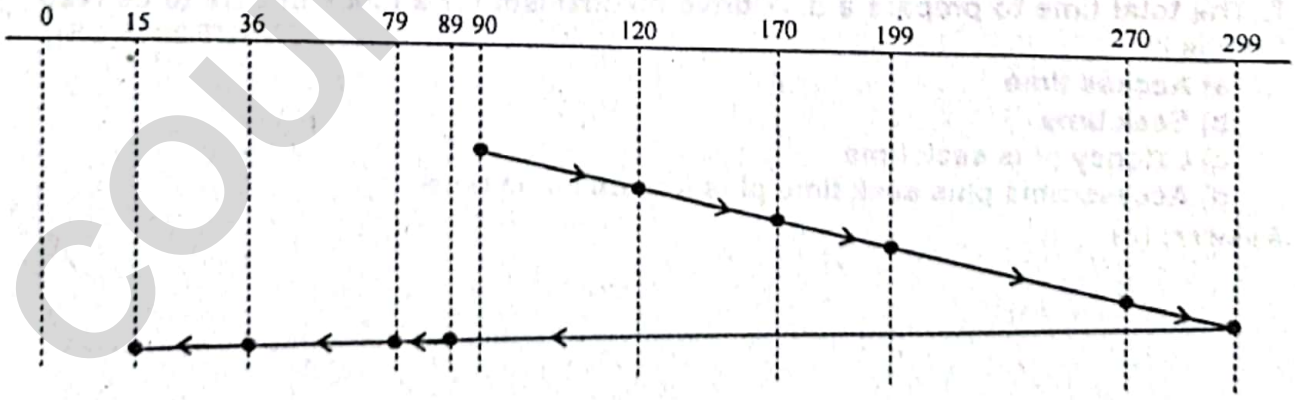
Answer:

i) and ii): ***Refer to Question No. 1 of Long Answer Type Questions.***

iii) Scan:

Given queue = 36, 79, 15, 120, 199, 270, 89, 170

Current head position = 90



Cylinder movements

$(120-90)+(170-120)+(199-170)+(270-199)$

$+(299-270)+(299-89)+(89-79)+(79-36)+(36-15)$

$=30+50+29+71+29+210+10+43+21$

$= 493$

∴ Average cylinder movement for scan will be

$=\frac{493}{10}=49.3$

Long Answer Type Questions

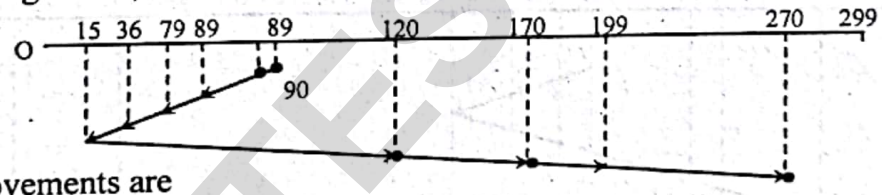
1. Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current head position of the disk is at 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170.
Calculate the average cylinder movements for the following algorithms:
i) SSTF; ii) C–SCAN. [WBUT 2007, 2010, 2013(IT)]

OR,

Suppose a disk drive has 300 cylinders, numbered 0 to 229. The current head position of the disk is at 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170.
Calculate the average cylinder movements for Shortest Seek Time First (SSTF) algorithm. Mention any one disadvantage of SSTF. [WBUT 2009, 2015(CS)]

Answer:

i) For SSTF algorithm,



The head movements are

$= (90 - 89) + (89 - 79) + (120 - 79) + (170 - 120) + (199 - 170)$
$+ (270 - 199) + (270 - 36) + (36 - 15)$

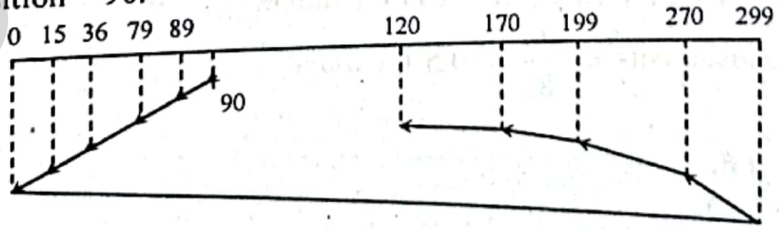
$= 1 + 10 + 41 + 50 + 29 + 71 + 234 + 21 = 457$

The average cylinder movements will be = 457/8 = 57.125

ii) C- SCAN:

Given queue = 36, 79, 15, 120, 199, 270, 89, 170

Current head position = 90.



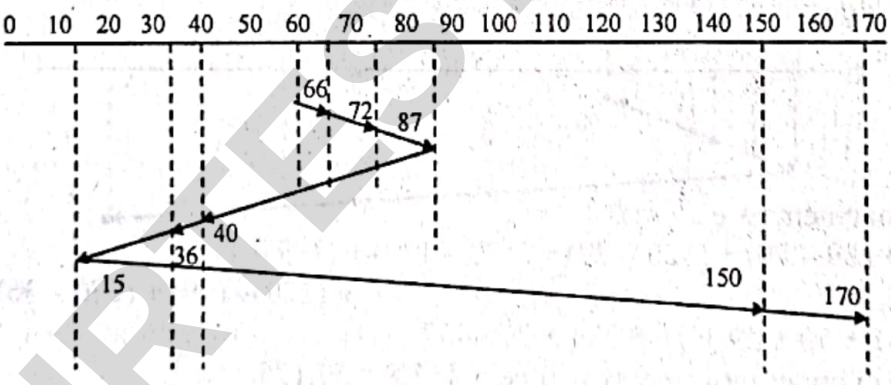
Cylinder movements = (90 – 89) + (89 – 79) + (79 – 36) + (36 – 15) + (15 – 0)
+ (0 – 299) + (299 – 270) + (270 – 199) + (199 – 170) + (170 – 120)
= 1+10+43+21+15+200+71+29+50= 568

$\therefore$ Average cylinder movement for SSTF = $\frac{568}{10} = 56.8$

Last Part:
Although the SSTF algorithm is a substantial improvement over the FCFS, but it not optimal.

2. a) Explain the working of Shortest Seek Time First (SSTF) disk scheduling policy. What are its advantages and disadvantages?
b) Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current position of the desk arm is 90. The queue of pending requests, in FIFO order is 36, 79, 75, 120, 199, 270, 89, 170. Calculate the average movements for the following algorithms:
i) FCFS ii) SSTF. **[WBUT 2013(CS), 2017(IT)]**

Answer:
a) This algorithm works on this principle "When a disk operation finishes, choose the request that is closest to the current head position or choose the request that has minimum seek time from the current head position".
Consider the disk Queue, (87, 170, 40, 150, 36, 72; 66. 15).
The initial head position is say 60. Now, closet to the head position is the request at cylinder 66. Then the closet to 66 is 72, and then 87 and so on.



Total head movement in SSTF are:
= (66 – 60) + (72 – 66) + (87 –72) + (87 – 40) + (40 – 36)
+ (36 – 15) + (150 –15) + (170 – 150)
= 6 + 6 + 15 + 37 + 4 + 21 + 135 + 20 = 244 Cylinders

$\therefore$ Average head movements = $\frac{244}{8} = 30.5$ Cylinder

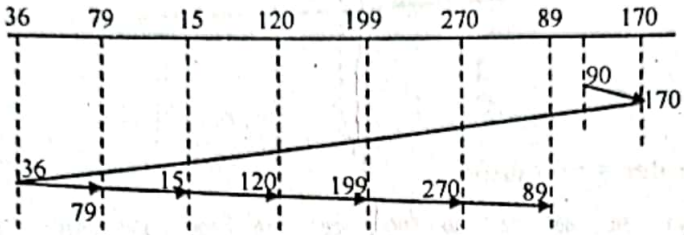
2nd Part:
Advantages of SSTF:
i) It minimizes latency
ii) Better though put than F1F0.

Disadvantages:

i) Starvation may occur.

ii) SSTF services requests for those tracks which are highly localized. So, the innermost and outermost tracks receive poor services as compared to the midrange tracks.

b) FCFS:



Total head movements =

$(170 - 90) + (170 - 36) + (79 - 36) + (79 - 15) + (120 - 15) + (199 - 120) + (270 - 99) + (270 - 89)$

$= 80 + 134 + 43 + 64 + 105 + 79 + 71 + 181 = 757$

$\therefore$ Average head movements $\frac{757}{8} = 94.62$

SSTF: ***Refer to Question No. 1 of Long Answer Type Questions.***

Average head movements i.e. $\frac{457}{8} = 57.12$

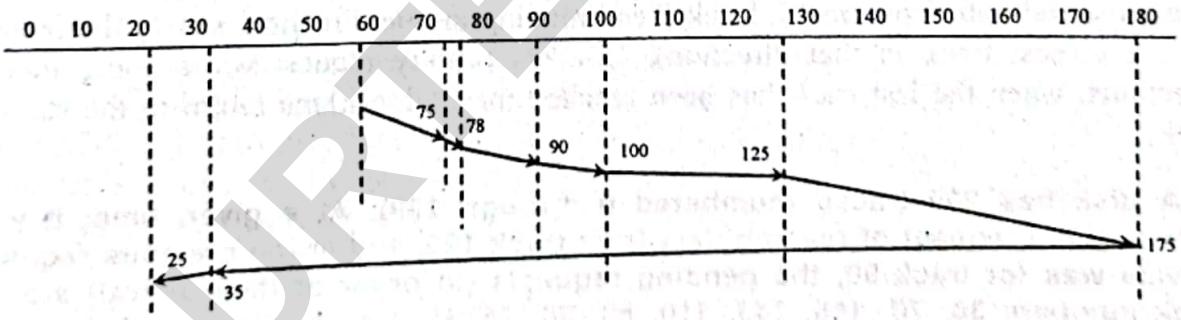
3. Draw the disk read/write head movement diagram for SSTF, SCAN, C-SCAN and FIFO, for the track requests as **[WBUT 2016(CS), 2016(IT)]**

25, 75, 35, 100, 95, 175, 78, 125, 90, 35

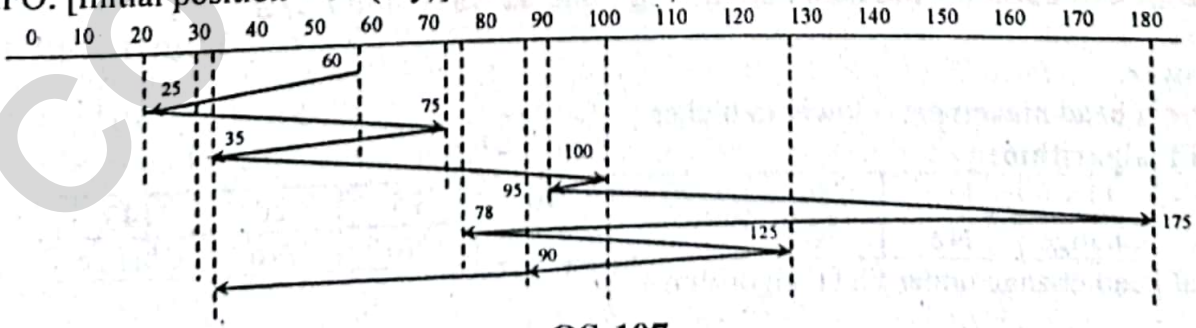
Answer:

SSTF: [Initial position = 60 (say)]

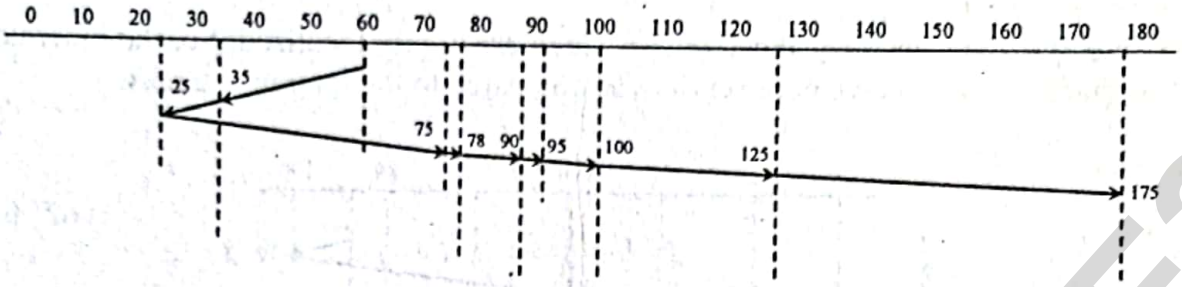
25, 75, 35, 100, 95, 175, 78, 125, 90, 35



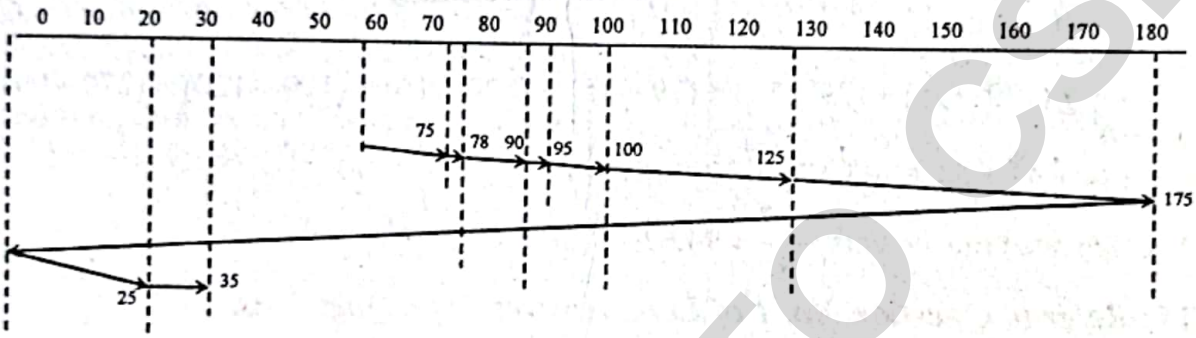
FIFO: [Initial position = 60 (say)]



SCAN [Initial pointer = 60 (say)]



C–SCAN [Initial pointer = 60 (say)]



4. a) What are the essential goals of disk scheduling? Why is each important?

[WBUT 2018(IT)]

Answer:

To satisfy a series of I/O request which can affect the performance, disk scheduling is required. These are no. of algorithms for scheduling disk I/O requests, like FIFO (service requests in the order they are received), priority (According to the priority associated with the requesting process), SSTF (Services the request whose track position is closest to the current track position), SCAN (Move the read/write head back and forth between the innermost and outermost track), Look (head moving in one direction, satisfy the request for the closest track in that direction), CSCAN (satisfy request while going in one directions, when the last track has been reached these algorithms return to the starting track).

b) A disk has 200 tracks (numbered 0 through 199). At a given time, it was servicing the request of reading data from track 120, and at the previous request, service was for track 90, the pending requests (in order of their arrival) are for track numbers 30, 70, 115, 143, 110, 80, 20, 25. How many times will the head change direction for the disk scheduling policies SSTF and FCFS?

[WBUT 2018(IT)]

Answer:

Current head movement = lower to higher

SSTF algorithm:

115	110	80	70	30	25	20	143
Change	No	No	No	No	No	No	Change

Total head change under SSTF algorithm = 2

FCFS algorithm:

30	70	115	143	110	80	20	25
Change	Change	No	No	Change	No	No	Change

Total head change under FCFS algorithm = 4

5. a) What are seek time and latency time? **[WBUT 2018(IT)]**

Answer:

Seek time:

It is defined as the time required to move the disk arm to the required track.

i.e. $T_s = m * n + S$

Where T_s = Estimated seek time
n = No. of tracks reversed
m = Constant that depends on disk drive
S = Start uptime

Latency time:

It is defined as the time required to reach the defined sector by the read/write head.

Latency can be computed by dividing the number of revolution per minute, R, into 30

i.e. $t = \frac{30}{R}$

b) Describe physical and logical formatting of disk. **[WBUT 2018(IT)]**

Answer:

A hard disk must be physically formatted before it can be logically formatted. A hard disk's physical formatting is usually performed by the manufacturer. Physical formatting divides the hard disk platters into their basic physical elements: Tracks, sectors and cylinders. These elements define the way in which data is physically recorded on and read from the disk. Tracks are concentric circular paths written on each ride of a platter and it is identified by number starting with track zero.

Tracks are divided into smaller areas called sectors which are used to store a fixed amount of data. It is usually formatted to contain 512 bytes of data.

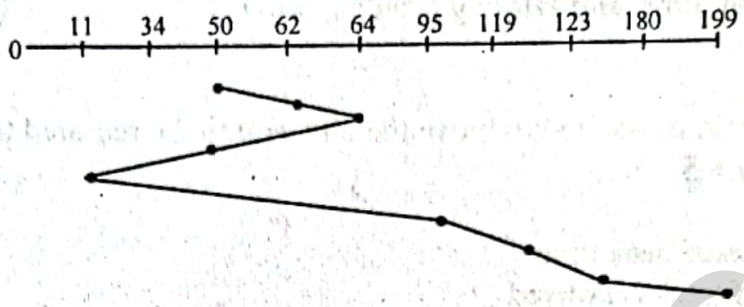
A cylinder is combined of a set of tracks that lie at the same distance from the spindle on all sides of all the platter. For example, track three on every side of every platter is located at the same distance from the spindle.

After a hard disk has been physically formatted, it must also be logically formatted. Logical formatting places a file system on the disk, allowing an operating system (such as DOS, windows, linux) to use the available disk space to store and retrieve files. Different OS use different file system. So logical file systems depends on OS formatting your entire hard disk with one file system necessarily limits the number and types of OS. But there is a solution to the above problem. Before a disk is logically formatted, it can be divided into partitions. Each partition can be formatted with a different file system, allowing one to install multiple OS.

c) Compare SSTF and C-SCAN algorithm in the context of disk scheduling. **[WBUT 2018(IT)]**

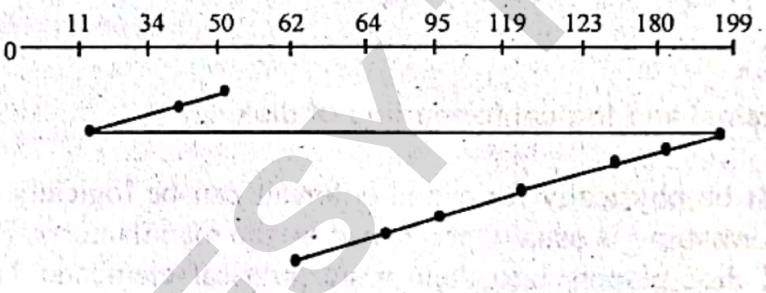
Answer:

In shortest Seek Time First (SSTF) algorithm request is serviced according to next shortest distance. For example,



Starting at 50, the next shortest distance is 62 instead of 34 since it is only 12 tracks away from 62 to 64 instead of 34 since there are only 2 tracks between them and not 11 if it were to go the another way. Then it is 34 and then 11 so on from 11 it can moves to 95 and so on.

But in C-scan (Circular Scan) algorithm it works just like the elevator to some extent but it begins the scan towards the nearest end like 50 to 34,then 11 and then 199, 180, 123, 119, 95, 64, 62.



6. Write short notes on the following:

a) Boot block and Bad block **[WBUT 2008, 2012(CS)]**

b) Scan and C-Scan algorithm **[WBUT 2016(IT)]**

c) Scan disk scheduling algorithm **[WBUT 2018(CS)]**

Answer:

a) Boot Block

When a computer is switched on an initial program called boot strap programme, which initializes the CPU registers, device controllers and other contents of main memory and then starts the operating system by finding out O.S. kernel on the disk, loading the kernel into memory and begins the operating system execution from a fixed initial address. Most system stores a tiny boot strap loader program in the boot ROM which in turn invokes a full boot strap program from disk that is stored in a partition at a fixed location on the disk called the boot block. A disk having a boot partition is called a boot disk or system disk.

Bad Block

Disks have generally tendency to become a failure. Sometimes one or more sectors become defective. A disk block with one or more bad sectors are called bad blocks.

MS-DOS format command, as a part of formatting process, scans the disk to find bad blocks and writes a special value in the corresponding FAT entry to specify to the allocation routine not to allocate that block. Data that reside on the bad block are lost.

The disk controller can be told to replace each bad sector logically with some spare sector that was set aside during low-level formatting process. This scheme is known as sector sparing or forwarding.

Some controllers can be instructed to replace a bad block by sector slipping process.

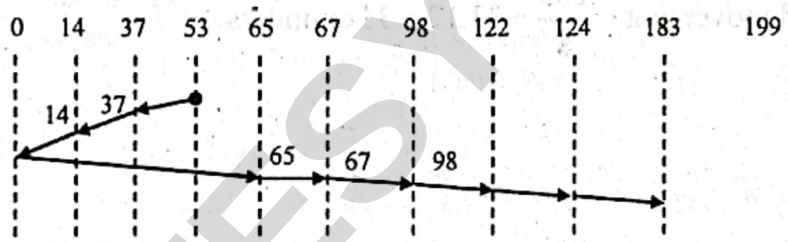
b) Scan and C-Scan algorithm:

In the **scan algorithm,** the disk arm starts at one end of the disk and moves towards the other end, servicing requests as it reaches each cylinder, until it gets to the other end of the disk. At the other end, the direction of head movement is reversed and servicing continuous. The head continuously scans back and forth across the disk. The scan algorithm is sometimes called **elevator** algorithm, since the disk arm behaves just like an elevator in a building, first servicing all request going up and then reversing to service requests the other way.

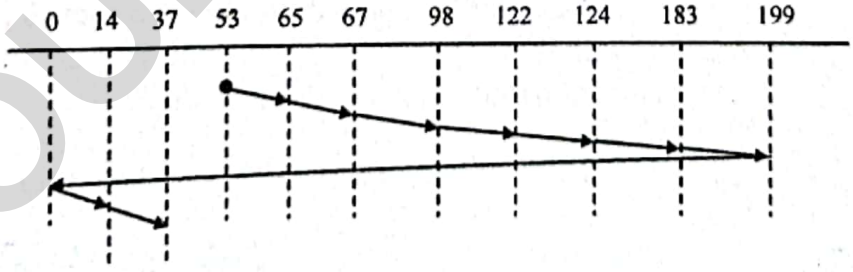
Example:

98, 183, 37, 122, 14, 124, 65, 67

Head starts at = 53.



C-SCAN (Circular Scan) Scheduling is a variant of SCAN designed to provide a more uniform wait time. Like scan, C-SCAN moves the head from one end of the disk to the other servicing request along the way. When the head reaches the other end, however, it immediately returns to the beginning of the disk without servicing any request on the return trip.



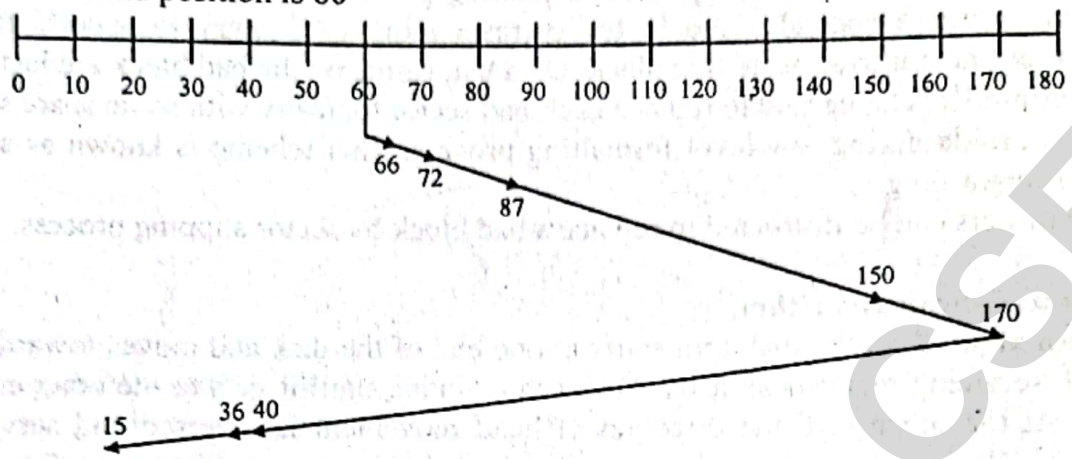
c) Scan disk scheduling algorithm:

This algorithm sometimes called **Elevator Algorithm**, where the disk arm starts at one end of the disk and moves towards the other end. As the head gets to each track, all

outstanding requests are satisfied for that track. At the other end, the direction of the head movement is reversed and this back and forth movement continues. Consider the queue

87, 170, 40, 150, 36, 72, 66, 15

The initial head position is 60



The total head movement

$$= (60 \text{ to } 66) + (66 \text{ to } 72) + (72 \text{ to } 87) + (87 \text{ to } 150) + (150 \text{ to } 170)$$
$$+ (170 \text{ to } 40) + (40 \text{ to } 36) + (36 \text{ to } 15)$$
$$= 6 + 6 + 15 + 63 + 20 + 130 + 4 + 21 = 265$$

Average head movement $= \frac{265}{8} = 33.12 = 33$ cylinders

QUESTION 2015 [6th-CS]

GROUP – A
(Multiple Choice Type Questions)

1. Answer all questions:

i) A thread is a
a) task b) program c) process ✓d) lightweight process

ii) Banker's algorithm for resource allocation is used for
✓a) deadlock avoidance b) deadlock prevention
c) deadlock recovery d) mutual exclusion

iii) The time spent by a process in ready queue is called
✓a) waiting time b) turnaround time
c) response time d) none of these

iv) The shell is
a) a hardware component ✓b) a command interpreter
c) a part in compiler d) a tool in CPU scheduling

v) Variable partition memory allocation can lead to
✓a) external fragmentation b) internal fragmentation
c) both (a) and (b) d) none of these

vi) SPOOLING stands for
a) Spontaneous Peripheral Operation Online b) Small Peripheral Operation Online
✓c) Simultaneous Peripheral Operation Online d) none of these

vii) Page fault occurs when
a) the page is corrupted by application software ✓b) the page is not in main memory
c) the page is in main memory d) one tries to divide a number by 0

viii) Scheduling a process from Ready Queue to CPU is done by
✓a) Short Term Scheduler b) Middle Term Scheduler
c) Long Term Scheduler . d) dispatcher

ix) Which page replacement algorithm suffers from Belady's anomaly?
a) LRU ✓b) FIFO
c) optimal page replacement d) none of these

x) The default remedy of starvation is
✓a) ageing b) critical section c) mutual exclusion d) all of these

GROUP – B
(Short Answer Type Questions)

2. a) What is "Turn Around Time"?
b) With the help of a state transition diagram, explain various states of a process.
c) If time quantum I very less for Round Robin Algorithm, then what will be the problems.

a) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 11.
b) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 1.
c) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 12.

3. Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current head position of the disk is at 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170. Calculate the average cylinder movements for Shortest-Seek Time First (SSTF) algorithm. Mention the disadvantages of SSTF.

See Topic: DISK MANAGEMENT, Long Answer Type Question No. 1.

4. Describe the two basic operations on semaphore. Explain whether any integer variable with the similar operation can act as semaphore or not.

See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 1.

5. How would each of the First Fit, Best Fit and Worst Fit algorithms place processes of 212KB, 417KB, 112KB and 426KB (in order). Which algorithm makes the most efficient use of memory?

See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 3.

6. What is deadlock? Describe the necessary and sufficient conditions for the occurrence of deadlock.

See Topic: DEADLOCK, Short Answer Type Question No. 4.

GROUP – C
(Long Answer Type Questions)

7. a) Consider the following snapshot of a system where $r_i (i = 1,2,...,4)$ denote resource types and $P_i (i = 1,2,...,5)$ denote processes. The vector 'Available' has usual meaning. Available Matrix $(r_1 = 2, r_2 = 1, r_3 = 0, r_4 = 0)$.

Process	Current Allocation:				Maximum demand:			
	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
P_1	0	0	1	2	0	0	1	2
P_2	2	0	0	0	2	7	5	0
P_3	0	0	3	4	6	6	5	6
P_4	2	3	5	4	4	3	5	6
P_5	0	3	3	2	0	6	5	2

(i) Is this system currently in a safe state? Justify your answer.
(ii) If a request from P_3 arrives for (0, 1, 0, 0), can that request be safely granted immediately?
b) Consider the following set of process. CPU burst times of them are given in milliseconds.

Process	CPU Burst Time (ms)
P1	15
P2	5
P3	7
P4	10

Draw the Gantt chart for Round Robin scheduling where time quantum $q = 4$ milliseconds. Calculate the average waiting time and average turn-around time. Mention the advantages and disadvantages of Round Robin scheduling.

a) See Topic: DEADLOCK, Long Answer Type Question No. 2.

b) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 1.

8. a) Consider a system with a 32-bit logical address space, a two-level paging scheme, 4 byte page table entries, 1kB pages and a 4 entry TLB. The page-table base register access time is 0 ns, TLB access time is 10 ns and memory access time is 100 ns.

(i) How many address bits are needed for the page offset?

(ii) How much memory in bytes I required to store the outer page table entry in main memory?

b) (i) Given references to the following pages by a program.

0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7.

How many page faults will occur of the program has three (3) page frames available to it and uses both FIFO replacement strategy and LRU replacement strategy.

(ii) Which replacement strategy in the above performs better and why?

See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 2.

9. a) What is DMA? Describe the different type of DMA Controllers.

b) What are the two major differences between segmentation and paging?

c) What is thrashing? How can one detect the thrashing?

d) What is internal fragmentation?

a) & c) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 9(a) & (b).

b) & d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 3(a) & (b).

10. a) What is context switching? Why is it considered to be an overhead?

b) What are the differences between process and thread?

c) What are the problems of busy-wait implementation of semaphore? Explain how it is solved.

d) Mention the contents of PCB.

a) & b) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 4.

c) See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 1.

d) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 5.

11. Write short notes on any *three* of the following:

a) Kernel level thread & User level thread

b) Dining-Philosopher Problem

c) Spooling

d) Belady's anomaly

e) Shortest Remaining Time First CPU scheduling.

a) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(f).
b) See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 7(a).
c) See Topic: INTRODUCTION, Long Answer Type Question No. 2(c).
d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(d).
e) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(c).

QUESTION 2015 [5th-IT]

GROUP – A
(Multiple Choice Type Questions)

1. Answer all questions.

i) With a segmentation, if there are 64 segments, and the maximum segment size is 512 words, the length of the logical address in bits is

a) 12 b) 14 ✓c) 15 d) 16

ii) Which of the following statements is false?

a) segmentation suffers from external fragmentation
b) paging suffers from internal fragmentation
c) segmented memory cab be paged
✓d) virtual memory is used only in multi-ser systems

iii) The full form of SPOOL is

a) shared processor object oriented language b) special purpose object oriented language
✓c) simultaneous peripheral operations online d) none of these

iv) An address generated by the CPU is commonly referred to as

✓a) logical address b) physical address
c) relation address d) virtual address

v) Throughput is

✓a) number of processes competed per unit time
b) completion time of the whole process
c) time for waiting in ready queue
d) time waiting to get into the memory

vi) A system program that sets up an executable program in main memory ready for execution

a) assembler b) linker ✓c) loader d) text editor

vii) Mutual exclusion problem occurs between

a) two adjacent processes that do not interact
b) processes that share resources
✓c) processes that do not use the same resources
d) none of these

viii) In which of the following scheduling policies does context switching never take place?

a) Round Robin
b) Shortest Job First
c) Pre-emptive
✓d) First Come First Serve

ix) Page fault occurs when

a) the page is corrupted by application software
b) the page is in main memory
✓c) the page is not in main memory
d) one tries to divide a number by zero

x) Safety algorithm may require an order of _____ operations to determine whether a state is safe (where, m is the number of resource type and n is the number of processes).

a) $m \times n^2$
✓b) $m \times n$
c) $m^2 \times n$
d) none of these

GROUP – B

(Short Answer Type Questions)

2. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds. The processes are assumed to have arrived in the order as shown below: Draw four Gantt chart for SRTF (shortest remaining time first) scheduling with time quantum = 5.

Process	Arrival time	CPU Time
P1	0	13
P2	2	6
P3	3	10
P4	5	8

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 13.

3. a) What are the necessary conditions for deadlock?
b) Write short note on Kernel Level Thread.

a) See Topic: DEADLOCK, Long Answer Type Question No. 1(a).
b) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(b).

4. What is Process Control Block? Discuss the structure of Process Control Block.

See Topic: PRO PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 5.

5. a) Mention one characteristic each of Time Sharing System and Batch Processing System.
b) What are the advantages and disadvantages of having unequal size partitions in fixed partition scheme?

a) See Topic: INTRODUCTION, Short Answer Type Question No. 1.
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 6.

6. What is deadlock? Critically comment on the following topic: Cycle in resource allocation graph does not always imply the occurrence of deadlock.

See Topic: DEADLOCK, Long Answer Type Question No. 5.

GROUP – C
(Long Answer Type Questions)

7. a) What are overlays?
b) What are the advantages of segmentation over paging?
c) Explain the difference between internal fragmentation and external fragmentation. Which one occurs in paging system? How can the problem of external fragmentation be solved?
d) Why are segmentation and paging sometimes combined into one scheme?
e) State the advantages and disadvantages of single contiguous memory allocation.

a), b), c), & e) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 4.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 13.

8. Given memory partition 100k, 500k, 200k, 300k and 600k (in order), how would each of the First-fit, Best-fit, Worst fit algorithm place processes of 212k, 417k, 112k, and 426k? Which algorithm makes the most efficient use of memory? Why is page size always power of 2? What is a multilevel paging? What is DMA? How does DMA increase system concurrency?

1st & 2nd part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 3.
3rd part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7.
4th & 5th part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 15.

9. Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 6, 6, 2, 1, 2, 3, 6. How many page faults would occur for the following replacement algorithms, assuming four frames. LRU replacement, FIFO replacement, Optimal replacement. The list of all passwords is kept within the operating system. Thus, if a user manages to read this list password protection is no longer provided. Suggest a scheme that will avoid this problem. What is Thrashing? What is the cause of Thrashing? What is Swap-In and Swap-Out?

1st part: See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 6(c).
2nd & 5th part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 14.
3rd & 4th part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 8.

10. What do you mean by race condition? What is semaphore? What is counting semaphore? What is the advantage of using a counting semaphore than a binary semaphore? What is Readers-Writers problem? How it can be solved using semaphore? Explain with algorithm.

See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 4.

11. Write the short notes any *three* of the following:
   a) Multilevel feedback queue scheduling
   b) Real-time systems
   c) Indexed allocation method of disk space
   d) Multithreading models
   e) Hierarchical page table.

a) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(e).
b) See Topic: INTRODUCTION, Long Answer Type Question No. 2(a).
c) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 5(c).
d) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(g).
e) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(e).

QUESTION 2016 [6th-CS]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following:

i) Banker's algorithm solves the problem of
- ✓a) deadlock avoidance
- b) deadlock recovery
- c) deadlock prevention
- d) mutual exclusion.

ii) A thread is a
- a) task
- b) process
- c) program
- ✓d) light weight process

iii) The time to move the disk arm to the desired cylinder in hard disk is known as
- ✓a) rotational latency
- b) seek time
- c) positional time
- d) disk time

iv) Thrashing
- a) reduces page I/O
- b) decreases the degree of multiprogramming
- ✓c) implies excessive page I/O
- d) improves the system performance

v) ..... Provides an interface to the operating system for the user.
- a) Kernel
- b) Micro-kernel
- ✓c) Shell
- d) None of these

vi) Which scheduling policy is most suitable for a time-shared operating system?
- a) Shortest job first
- ✓b) Round Robin
- c) First come first serve
- d) Priority

vii) Compaction is used to solve the problem of
- ✓a) external fragmentation
- b) internal fragmentation
- c) both (a) and (b)
- d) none of these

viii) RAID configuration disk is used to provide
- ✓a) fault tolerance
- b) nearest cylinder next
- c) high data density
- d) none of these

ix) The scheduler which selects jobs from the pool of jobs and loads to the ready queue is
- ✓a) long term
- b) short term
- c) medium term
- d) none of these

x) Part of the program where the shared memory accessed and which should be executed indivisibly, is called
- a) semaphores
- b) directory
- ✓c) critical section
- d) mutual exclusion

Group – B
(Short Answer Type Questions)

2. Name one essential property of the following types of operating systems:
(a) Batch, (b) Interactive, (c) Time-sharing, (d) Real time, (e) Network.

See Topic: INTRODUCTION, Short Answer Type Question No. 2.

3. What are the difference between a trap and an interrupt? What is the use of each function?

See Topic: INPUT/ OUTPUT HARDWARE, Short Answer Type Question No. 2.

4. What is the purpose of the command interpreter? Why is it usually separate from the kernel?

See Topic: INTRODUCTION, Short Answer Type Question No. 3.

5. Given *n* processes to be scheduled on one processor; how many possible different schedules are there? Give a formula in terms of *n*.

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 14.

6. Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free.

See Topic: DEADLOCK, Short Answer Type Question No. 6.

Group – C
(Long Answer Type Questions)

7. a) What do you mean by scheduler? Explain different types of scheduler. Explain CPU scheduling criteria.

b) For the process listed in the table, draw a chart illustrating their execution using FCFS, SJF, SRTF (SRJF), Round Robin (Quantum = 2) and calculate average turn-around time and average waiting time.

Process	Arrival Time	Processing Time
A	0	8
B	1	4
C	2	9
D	3	5

See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 2.

8. a) What is critical section problem? What are the requirements a critical section problem must satisfy?

b) What is deadlock? What are the necessary conditions for deadlock to occur?

c) Consider a system with five processes P_0 through P_4 and have three resource types A, B, C. Find out the number of instances of each resource type and retrieve the safe sequence.

	MAX			NEED			AVAILABLE		
	A	B	C	A	B	C	A	B	C
P_0	7	5	3	7	4	3	2	3	0
P_2	3	2	2	0	2	0			
P_2	9	0	2	6	0	0			
P_3	2	2	2	0	1	1			
P_4	4	3	3	4	1	1			

a) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 3.
b) See Topic: DEADLOCK, Short Answer Type Question No. 4.
c) See Topic: DEADLOCK, Long Answer Type Question No. 3.

9. a) State producer-consumer problem. Give a solution to the producer-consumer problem using semaphore. Justify your solution guarantees Mutual Exclusion.
b) What is paging? Differentiate between internal and external fragmentations. What is thrashing?
c) What is TLB? What do you mean by 'Belady's Anomaly'?
d) Having 3 physical memory frames show the behavior of LRU and FIFO and optimal page replacement algorithm for the page address string like 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2.
a) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 6.
b), c) & d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 10.

10. a) What is a record in a file? For a file having multiple records what are the different indexing strategies there? Explain each strategy very briefly with relative advantages and disadvantages.
b) Draw the disk read/write head movement diagram for SSTF, SCAN, C-SCAN and FIFO, for the track requests as
25, 75, 35, 100, 95, 175, 78, 125, 90, 35
a) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 3.
b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 3.

11. Write short notes on any three of the following:
a) Process life cycle
b) Orphan process and Zombie process
c) *i*-node
d) Segmentation
e) Peterson solution for CS
a) See Topic: PROCESSES, THREAD & SCHEDULING Long Answer Type Question No. 11(h).
b) See Topic: PROCESSES, THREAD & SCHEDULING Long Answer Type Question No. 11(i).
c) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 5(d).
d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(b).
e) See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 7(b).

QUESTION 2016 [5th-IT]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following:
i) Which one in the following is NOT shared by the threads of the same process?
✓a) Stack
b) File Descriptor Table
c) Address Space
d) Message

ii) Important objectives of computer security include
a) Confidentiality
b) Integrity
c) Availability
✓d) All of these

iii) Which of the following reduces degree of multiprogramming?

✓a) Long Term Scheduler b) Medium Term Scheduler

c) Short Term Scheduler d) All of these

iv) In a resident OS computer, which of the following systems must reside in the main memory under all situations?

a) Assembler b) Loader c) Linker ✓d) Compiler

v) In order to implement mutual exclusion on a critical resource for competing processes, only one program at a time should be allowed

✓a) in the critical section of the program b) to perform message passing

c) to exhibit cooperation d) none of these

vi) In a tree-structured directory, the series of directory names that culminates in a file name is referred to as the

✓a) Path name b) Working Directory c) Symbolic name d) None of these

vii) A memory page containing a heavily used variable that was initialized very early and is in constant use is removed when ..... page replacement algorithm is used.

a) LRU b) LFU ✓c) FIFO d) none of these

viii) The main purpose of OS is

a) to provide users an environment to execute programs

b) to manage computer resources

✓c) both (a) & (b) d) none of these

ix) With segmentation, if there are 64 segments and maximum segment size is 512 words, the length of bits in logical address is

a) 12 ✓b) 15 c) 14 d) 16

x) TLB is a kind of

a) Virtual memory b) Interrupt ✓c) Cache d) Main memory.

Group – B
(Short Answer Type Questions)

2. What is Semaphore? Differentiate between Binary & Counting Semaphores.

See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 3.

3. What is the main objective of Multiprogramming? Draw and describe process state transitions.

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 15.

4. Categorize different types of attackers in the context of security.

OUT OF SYLLABUS

5. What is the problem of fragmentation and how can it be solved?
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 2.

6. Why are page sizes always powers of 2? What is the difference between logical and physical addresses?
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7 & 1.

Group – C
(Long Answer Type Questions)

7. a) What do you mean by scheduler? Explain different types of scheduler. Explain CPU scheduling criteria.
b) For the process listed in the table, draw a chart illustrating their execution using FCFS. SJF, SRTF (SRJF). Round Robin (quantum = 2) and calculate average turn-around time and average waiting time:

Process	Arrival Time	Processing Time
A	0	8
B	1	4
C	2	9
D	3	5

See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 2.

8. a) What is critical section problem? What are the requirements a critical section problem must satisfy?
b) What is deadlock? What are the necessary conditions for deadlock to occur?
c) Consider a system with five processes P0 through P4 and have three resource types A, B, C. Find out the number of instances of each resource type and retrieve the safe sequence where:

	MAX			NEED			AVAILABLE		
	A	B	C	S	B	C	A	B	C
P0	7	5	3	7	4	3	2	3	0
P1	3	2	2	0	2	0			
P2	9	0	2	6	0	0			
P3	2	2	2	0	1	1			
P4	4	3	3	4	3	1			

a) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 3.
b) See Topic: DEADLOCK, Short Answer Type Question No. 4.
c) See Topic: DEADLOCK, Long Answer Type Question No. 3.

9. a) State producer-consumer problem. Give a solution to the producer-consumer problem using semaphore. Justify your solution guarantees Mutual Exclusion.
b) What is paging? Differentiate between internal and external fragmentations. What is thrashing?
c) What is TLB? What do you mean by "Belady's Anomaly"?
d) Having 3 physical memory frames show the behavior of LRU and FIFO and optimal page replacement algorithm for the page address string like:
2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2.

a) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 6.
b), c) & d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 10.

10. a) What is a record in a file? For a file having multiple records what are the different indexing strategies there? Explain each strategy very briefly with relative advantages and disadvantages.
b) Draw the disk read/write head movement diagram for SSTF, SCAN, CSCAN and FIFO, for the track requests as:

25, 75, 35, 100, 95, 175, 78, 125, 90, 25

a) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 3.
b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 3.

11. Write short notes on any three of the following:
a) Process Control Block
b) Scan and C-Scan algorithm
c) Aging Technique
d) External Fragmentation & Internal Fragmentation.

a) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(d).
b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 6(b).
c) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(j).
d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(f).

QUESTION 2017 [6th-CS]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for any *ten* of the following:

i) A page fault occurs
✓a) when the page is not in the memory
b) when the is in the memory
c) when the process enters the blocked state
d) when the process enters the ready state

ii) Which is the fastest of the following?
a) Cache memory
b) RAM
c) CD-ROM
✓d) Register

iii) What is a shell?
a) It is a hardware component
✓b) It is a command interpreter
c) It is a part in compiler
d) It is a tool in CPU scheduling

iv) A thread is a
a) Task
b) Program
c) Process
✓d) Lightweight process

v) Round Robin scheduling is essentially the preemptive version of
✓a) FIFO
b) Shortest Job First
c) Shortest Remaining Time First
d) Longest Time First

vi) In order to allow only one process to enter its critical section, binary semaphores are initialized to

a) 0 ✓b) 1 c) 2 d) 3

vii) Banker's algorithm for resource allocation deals with

a) Deadlock prevention ✓b) Deadlock avoidance
c) Deadlock recovery d) Mutual exclusion

viii) Which of the following page replacement algorithms suffers from Belady's anomaly?

a) Optimal b) LRU ✓c) FIFO d) Both (a) and (b)

ix) The mechanism that brings a page into memory only when it is needed, is called

a) Segmentation b) Fragmentation
✓c) Demand paging d) page and replacement

x) If UNIX command chmod 756 is applied to a file, them other will have

a) Read and write permission b) Read and execute permission
c) Write and execute permission ✓d) None of these

xi) Which of the following resources can cause deadlocks?

a) Read only files ✓b) Shares programs c) Printers d) All of these

xii) The number of processes completed per unit time is known as

a) Output ✓b) Throughput c) Efficiency d) Capacity.

Group – B
(Short Answer Type Questions)

2. a) What is kernel?
b) State the functions of system call.
See Topic: INTRODUCTION, Short Answer Type Question No. 5

3. a) What do you mean by real time system?
b) Differentiate between soft and hard real time system.
See Topic: INTRODUCTION, Short Answer Type Question No. 4.

4. a) What is Medium Term scheduler?
b) Describe the functions of short-term and long-term scheduler.
See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 17.

5. a) What is deadlock?
b) Justify the following statement:
"cycle in resource allocation graph does not always imply the occurrence of deadlock."
See Topic: DEADLOCK, Short Answer Type Question No. 5.

6. a) Explain Race condition in context of process synchronization.
b) What are semaphore and mutex?
See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 7.

Group – C
(Long Answer Type Questions)

7. a) What is thread? Draw and explain thread life cycle.
b) Differentiate between process and thread.
c) Explain user and Kernel thread in detail.
a) & c) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 3.
b) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 4(b).

8. a) Explain different states of a process using state transition diagram.
b) What do you mean by preemptive and non- preemptive scheduling?
c) What is dispatcher?
d) Consider the following four processes, with the length of CPU-burst time given in milliseconds:

Processes	Arrival time	Burst time
P1	0	12
P2	0	10
P3	1	4
P4	4	10
P5	2	12

Draw the Gantt chart using RR scheduling with time slice 3ms. Calculate average waiting time and average turn around time.
a) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 1.
b) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 10(a).
c) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 16.
d) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 18.

9. Write a program using 'signal' to demonstrate a race condition.
See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 5.

10. Write a program using 'fork' to demonstrate the mother-child relationship of processes.
See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 4.

11. a) What is overlay?
b) What are the advantages of segmentation over paging?
c) Explain the difference between internal fragmentation and external fragmentation. Which one occurs in paging system? How the problem of external fragmentation be solved?
d) State the advantages and disadvantages of single contiguous memory allocation.
See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 4.

12. a) What is the purpose of modify bit in page table?
b) Consider the following page reference string:

7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

How many page faults would occur for the following replacement algorithms, assuming 3 frames are available and initially none of pages in main memory?

i) Optimal replacement
ii) FIFO replacement

c) What is Thrashing?
d) Explain Belady's anomaly.

a) & b) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 5.
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 8.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 4.

QUESTION 2017 [5th-IT]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following:

i) Which of the following is (are) non-pre-emptive scheduling algorithm?
✓a) FCFS b) SJF c) Round Robin d) Priority Scheduling

ii) Which of the following is not the layer of operating system?
a) Kernel b) Shell
c) Application program ✓d) Critical section

iii) Where does not swap space reside?
a) RAM ✓b) DISK c) ROM d) On-chip cache

iv) An address generated by the CPU is commonly referred to as
✓a) logical address b) physical address c) relational address d) virtual address

v) Cryptography technique is used in
a) polling b) job scheduling ✓c) protection d) file management

vi) TLB is a kind of
a) Virtual Memory b) Interrupt ✓c) Cache d) Main memory

vii) The smallest possible unit of disk storage is
a) Word b) Segment ✓c) Block d) Extent

viii) The main advantage of the interrupt concept is elimination of
a) Spooling b) Polling
c) Job scheduling ✓d) Blocking the current running process

ix) Context switching is

a) part of the spooling
b) part of polling
c) part of interrupt handling
✓d) part of interrupt servicing

x) To enable a process to be larger than the amount of memory allocated to it, one can use

✓a) overlays b) paging c) compaction d) swapping

Group – B
(Short Answer Type Questions)

2. a) What are the operations on a semaphore?
b) What are the problems with these operations if these follow the classical definition?
c) What is the possible remedy to the above problem?

See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 4.

3. Consider the following set of processes with corresponding arrival times and burst times:

Process	Arrival Time (units)	CPU Burst Time (units)
P1	0	6
P2	3	10
P3	5	8
P4	7	5
P5	10	6

Draw the Gantt chart considering Round Robin scheduling policy with time quantum = 4 units. Calculate individual turnaround time and average waiting time.

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 8.

4. a) What are the contents of process control block (PCB)?
b) Under what conditions do the following state transitions occur with respect to a process?
(i) Run to Ready.
(ii) Blocked (or Wait) to Ready.

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 6.

5. a) What are the relative advantages and disadvantages of user level thread and kernel level thread?
b) What is thrashing?

a) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 7.
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 8.

6. a) What is seek time? What is rotational latency?
b) What are the advantages of SCAN disk scheduling technique over circular SCAN disk scheduling technique?

See Topic: DISK MANAGEMENT, Short Answer Type Question No. 1.

Group – C
(Long Answer Type Questions)

7. a) Consider the following page reference string and a memory consisting of a 4 frames:
1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6.
Find the number of page faults considering
(i) FIFO page replacement strategy
(ii) LRU page replacement strategy.
Comment on the results obtained.
b) What are the disadvantages of segmentation memory management technique? How can these disadvantages be avoided if segmentation with paging is used?
c) Why are page sizes always powers of 2?

a) & b) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 7.
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7.

8. a) Consider the following snapshot of a system:

Process	Allocation	Max	Available
	ABCD	ABCD	ABCD
P0	0012	0012	1520
P1	1000	1750	
P2	1354	2356	
P3	0632	0652	
P4	0014	0656	

Answer the following questions using banker's algorithm:
(i) What is the content of need Matrix?
(ii) Is the system in safe state?
(iii) If the request P1 arrives for (0, 4, 2, 0) can the request be granted immediately?
b) What are the four necessary conditions for deadlock to occur in a system? Explain.

a) See Topic: DEADLOCK, Long Answer Type Question No. 1(b).
b) See Topic: DEADLOCK, Long Answer Type Question No. 1(a).

9. a) Differentiate between Bloc' ing Non-Blocking input-output.
b) What is Direct Memory Access? How is it performed? What are its benefits?
c) A system has 8 physical frames. There are 7 processes in the system of which 4 processes have 2 pages each and 3 processes have 1 page each. The system uses inverted page table. Find the total number of page table entries in the system. Justify your answer.
d) Why is context switching considered to be time consuming?

a) See Topic: INPUT/OUTPUT HARDWARE, Short Answer Type Question No. 1.
b) & c) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 8.
d) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 2.

10. a) Explain the working of Shortest Seek Time First (SSTF) disk scheduling policy. What are its advantages and disadvantages?
b) Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current position of the disk arm is 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170. Calculate the average movements for the following algorithms:

i) FCFS
ii) SSTF.
c) Explain the worst fit algorithm for memory management. What are its benefits?
a) & b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 2.
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 10.

11. a) Explain any one technique adopted by operating systems for protection of objects in the system.
b) What are the advantages and disadvantages of linked file Allocation Technique?
c) How does Indexed file Allocation Technique overcome the above disadvantages?
d) What is compaction? What is its overhead?
e) What is the difference between starvation and deadlock?
a) OUT OF SYLLABUS
b) & c) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 2.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 11.
e) See Topic: DEADLOCK, Short Answer Type Question No. 1.

QUESTION 2018 [6th-CS]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for any *ten* of the following:
i) Which module gives control of the CPU to the process selected by the short-term scheduler?
✓a) Dispatcher
b) Interrupt
c) Scheduler
d) None of the mentioned

ii) The processes that are residing in main memory and are ready and waiting to execute are kept on a list called
a) job queue
✓b) ready queue
c) execution queue
d) process queue

iii) Page fault frequency in an operating system is reduced when
a) processes tend to be of an equal ratio of the I/O-bound and CPU-bound
b) size of pages is increased
✓c) locality of reference is applicable to the process
d) processes tend to be CPU-bound

iv) _____ is a high speed cache used to hold recently referenced page table entries a part of paged virtual memory.
✓a) Translation looks a side buffer
b) Inverse page table
c) Segmented page table
d) Indexed page table

v) Which directory implementation is used in most of the Operating System?

a) Single level directory structure
b) Two level directory structure
✓c) Tree directory structure
d) Acyclic directory structure

vi) The total time to prepare a disk drive mechanism for a block of data to be read from is its

a) Access time
b) Seek time
✓c) Latency plus seek time
d) Access time plus seek time plus transmission time

vii) The average wait time for five processes P1-P5 with burst of 5, 19, 2, 16 and 7 milliseconds respectively, using SJF is

a) 5 ms
b) 9.8 ms
c) 28 ms
✓d) 10.6 ms

viii) A situation where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which access takes place is called

✓a) race condition
b) data inconsistency
c) starvation
d) fatal error

ix) The address of the next instruction to be executed by the current process is provided by the

a) CPU registers
✓b) program counter
c) process stack
d) pipe

x) Banker's algorithm for resource allocation deals with

✓a) Deadlock avoidance
b) Deadlock prevention
c) Deadlock recovery
d) Deadlock detection

xi) _____ are very effective because a mode switch is not required to switch from one thread to another.

a) Kernel-level threads
b) Alterable threads
✓c) User-level threads
d) Application level threads

xii) For 3 page frames, the following is the reference string:

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

a) 10
b) 15
c) 11
✓d) 12

Group – B

(Short Answer Type Questions)

2. a) Explain the utility of Process Control Block (PCB) for a process?

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 22.

b) Under what conditions the following state transition occurs with respect to a process?

(i) Run to ready,
(ii) Blocked (or wait) to ready

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 6(b).

3. Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here:

i	T	(Pi)Arrival Time
0	80	0
1	20	10
2	10	10
3	20	80
4	50	85

(a) Suppose a system uses RR scheduling with a quantum of 15. Create a Gantt chart illustrating the execution of these processes?
(b) What is the turnaround time for process P^3?
(c) What is the average wait time for the processes?

See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 19.

4. a) What are the operations on semaphores?

See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 4(a).

b) How to implement a solution to the Readers-Writers Problem with the use of semaphores?

See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 8.

5. What is the optimistic assumption made in the deadlock detection algorithm? How can this assumption be violated?

See Topic: DEADLOCK, Long Answer Type Question No. 7.

6. Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 6(a), (b).

Group – C
(Long Answer Type Questions)

7. Explain Circular-wait for the occurrence of deadlock and how it could be prevented. Consider a system consisting of m resources of the same type being shared by n processes only one at a time. Show that the system is deadlock free if the following condition holds:

(a) The maximum need of each process is between 1 and m resources.
(b) The sum of all maximum needs is less than $m+n$.

Consider the following snapshot of a system:

Process	Allocation	Max	Available
P0	ABCD	ABCD	ABCD
P1	0012	0012	1520
P2	1000	1750	
P3	1354	2356	
P4	0632	0652	
P5	0014	0656	

Answer the following question using the Banker's algorithm:

1. What is the content of matrix "Need"?
2. Is the System in a safe state?
3. If a request from process P1 arrives for (0, 4, 2, 0) can the request be granted immediately?

See Topic: DEADLOCK, Long Answer Type Question No. 4.

8. What is thread? Explain types of thread with example. What advantages do threads have over multiple processes? When a process is called a cooperating process? Explain Critical-Section problem. Explain Counting Semaphore and binary semaphore with algorithm.

1st, 2nd, 3rd, & 4th Part: See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 5.

5th & 6th Part: See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 6.

9. Give memory partition of 100K, 500K, 200K, 300K and 600K (in order). How would each of the first fit, best fit and worst fit algorithm place process of 212K, 417K, 112K and 426K (in order)? Which algorithm makes the most efficient use of memory? Compare the following main memory organization schemes contiguous memory allocation, pure segmentation and pure paging with respect to the following issues:

(a) External fragmentation
(b) Internal fragmentation
(c) Ability to share code across processes

See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 11.

10. What is meant by Process Control Block? Draw and explain different fields of PCB. Explain Context Switch by giving an example. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5 all at time 0.

(a) Draw four Gantt chart illustrating the execution of these processes using FCFS, SJF, a non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.
(b) What is the turnaround time of each process for each of the scheduling algorithms in part a?
(c) What is the waiting time of each process for each of the scheduling algorithms in part a?
(d) Which of the schedules in part a results in the minimal average waiting time (over all processes)?

See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 6.

11. What is thrashing? How it can be prevented? What do you mean by page reference string? Suppose a process accesses the following addresses at a particular time interval:
0100, 0432, 0101, 0612, 0102, 0103, 0104, 0101, 0611, 0102, 0103, 0104, 0101, 0601, 0101, 0102, 0609, 0102, 0105.
Assume a page size = 100 bytes.

(a) What will be the reference string?
(b) Considering the above page reference string, calculate the page fault rate for the following algorithms:
  (i) LRU replacement.
  (ii) Optimal replacement. Assume that number of frames = 3.

Explain working sets. What is its physical significance?

See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 12.

12. Write short notes on the following (*any three*):

a) Semaphore
b) Scan disk scheduling algorithm
c) Linked file allocation technique
d) Belady's anomaly
e) Preemptive SJF scheduling
f) Security and Protection.

a) See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 7(c).
b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 6(c).
c) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 5(a).
d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(d).
e) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(c).
f) OUT OF SYLLABUS

QUESTION 2018 [5th-IT]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives from any ten of the following:

i) Suppose that a process is in BLOCKED state waiting from some I/O service. When the service is completed, it goes to the

a) RUNNING state
✓b) READY state
c) SUSPENDED state
d) TERMINATED state

ii) Which of the following schemes suffers from external fragmentation?

✓a) Segmentation
b) Paging
c) Spooling
d) Buffering

iii) Where does the Swap space reside?

a) RAM
✓b) Disk
c) ROM
d) On-Chip Cache

iv) System calls are usually invoked by

✓a) a software interrupt
b) polling
c) an indirect jump
d) a privileged instruction

v) Time Sharing Operating system has

a) high through put
b) low execution time
c) faster I/O
✓d) None of these

vi) Which page replacement algorithm gives the lowest Page fault rate?

✓a) LRU
b) FIFO
c) Optimal page replacement
d) None of these

vii) If there are 32 segments, each of size 1K, than the logical address should have

a) 10 bits
b) 14 bits
✓c) 15 bits
d) 16 bits

viii) Which one of the following is not a valid state of a process?

✓a) Load
b) Run
c) Wait
d) Terminate

ix) Compaction is used to solve the problem of

✓a) external fragmentation
b) internal fragmentation
c) starvation
d) thrashing

x) To avoid race condition the maximum number of processes that may simultaneously be inside the critical section is

a) hundred
✓b) one
c) two
d) three

Group – B
(Short Answer Type Questions)

2. a) What are the necessary and sufficient conditions for deadlock to occur?
b) What is thrashing?

a) See Topic: DEADLOCK, Long Answer Type Question No. 1(a).
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 8.

3. Different memory partitions of 150K, 820 K, 360 K and 450 K (in the given order) are present. Explain how best fit algorithm can be used to place a process of 315 K.

See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 16.

4. With example describe if resources are not properly allocated to the processes it may lead from a safe state to an unsafe state.

See Topic: DEADLOCK, Short Answer Type Question No. 8.

5. a) What is process control block?
b) Explain whether any integer variable with similar Operations can act as semaphore or not.

a) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 5.
b) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 1(2nd Part).

6. a) What are main features of Multiprocessor scheduling?
b) Briefly discuss Multiprocessor feedback queue scheduling.
See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No.20(a) & (b).

Group – C
(Long Answer Type Questions)

7. a) Mention the basic principle of RR Scheduling. Specify the impact of time quantum on its performance.
See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 12.

b) Consider the following set of processes. The CPU burst time of them are given in milliseconds:

Process	CPU burst time
P1	15
P2	5
P3	7
P4	10

Draw the gnat chart for FCFS and RR Scheduling where time quantum q=5 milliseconds. Calculate the average waiting time.
See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 21.

c) Given reference to the following pages by a program,
0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7
How many page fault will occur if the program has 3 page frames available in it when it uses
i) FIFO replacement, and
ii) LRU replacement?
See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 2(b) (i).

d) Explain CPU scheduling criteria.
See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 2(a) (3rd Part).

8. a) What is semaphore? Differentiate between binary and counting semaphore.
See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 3.

b) What are the problems of busy-wait implementation of semaphore? Explain how it is solved.
See Topic: INTER-PROCESS COMMUNICATION, Long Answer Type Question No. 1.

c) Explain the difference between external fragmentation and internal fragmentation. Which one occurs in paging system?
See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 4(c)(1st & 2nd Part).

9. a) What are the essential goals of disk scheduling? Why is each important?
See Topic: DISK MANAGEMENT, Long Answer Type Question No. 4(a).

b) A disk has 200 tracks (numbered 0 through 199). At a given time, it was servicing the request of reading data from track 120, and at the previous request, service was for track 90, the pending requests (in order of their arrival) are for track numbers 30, 70, 115, 143, 110, 80, 20, 25. How many times will the head change direction for the disk scheduling policies SSTF and FCFS?
See Topic: DISK MANAGEMENT, Long Answer Type Question No. 4(b).

c) What is the difference between logical and physical address?
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 1.

d) What is compaction? What are the drawbacks of compaction?
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 11.

10. a) What are seek time and latency time?
b) Describe physical and logical formatting of disk.
c) Compare SSTF and C-SCAN algorithm in the context of disk scheduling.
d) Describe structure of FAT file system.
a), b) & c) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 5(a), (b) & (c).
d) See Topic: FILE MANAGEMENT, Short Answer Type Question No. 2.

11. Write short notes on any three of the following:
i) Linked File Allocation
ii) Segmentation
iii) i-node
iv) Belady's Anomaly
v) Kernel Level Thread
i) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 5(a).
ii) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(b).
iii) See Topic: FILE MANAGEMENT, Long Answer Type Question No. 5(d).
iv) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(d).
v) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(f).

QUESTION 2019 [6th-CS]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following:
i) A computer system has 9 tape drives, with *n* processes competing for them. Each process may need 3 tape drives. The maximum value of *n* for which the system is guaranteed to be deadlock free is
a) 9 b) 7 c) 8 ✓d) 6

ii) The average wait time for five processes P1-P5 with burst of 5, 19, 2, 16 and 7 milliseconds respectively, using SJF is

a) 5 ms b) 9.8 ms c) 28 ms ✓d) 10.6 ms

iii) A situation where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which access takes places is called

✓a) race condition b) data inconsistency
c) starvation d) fatal error

iv) What are very effective because a mode switch is not required to switch from one thread to another?

a) Kernel-level threads b) Alterable threads
✓ c) User-level threads d) Application level threads

v) The default remedy of starvation is

✓a) Ageing b) Critical section c) Mutual exclusion d) All of these

vi) Which of the following page replacement algorithms suffers from Belady's anomaly?

a) Optimal replacement b) LRU ✓c) FIFO d) Both (a) & (c)

vii) Which of the following is false?

a) Segmentation suffers from external fragmentation
b) Paging suffers from internal fragmentation
✓c) Virtual memory is used only in multi-user system
d) Segmented memory can be paged

viii) If a process has 32 k bytes logical address space and the page size is 2048 bytes then the number of frames of that process is

a) 4 b) 8 ✓c) 16 d) 32

ix) In DMA transfer

✓a) CPU is involved actively during data transfer
b) CPU is involved partially during data transfer
c) DMA controller is actively involved during data transfer
d) Both (b) and (c)

x) Page stealing is

a) a sign of efficient system b) taking larger space's for pages paged out
✓c) taking page frames from other working sets d) one of the tuning goals

xi) Scheduling a process from Ready Queue to CPU is done by

✓ a) Short term scheduler b) Middle term scheduler
c) Long term scheduler d) Dispatcher

xii) Which scheduling policy is most suitable for the time-sharing operating system?

a) Shortest job first ✓ b) Round robin

c) First come first serve d) Multilevel queue

Group – B
(Short Answer Type Questions)

2. Prove that linear ordering for denying the "circular wait" condition actually prevents circuits from developing in resource allocation graphs. How can context switch time be reduced?

1st Part: See Topic: DEADLOCK, Short Answer Type Question No. 9.

2nd Part: See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 2.

3. What is fragmentation? Explain different types of fragmentation.

See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 17.

4. How to implement a solution to the Readers-Writers problem with the use of semaphores?

See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 8.

5. Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current head position of the disk is at 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170 Calculate the average cylinder movements for the following algorithms:

i) SSTF
ii) C-SCAN
iii) SCAN

See Topic: DISK MANAGEMENT, Short Answer Type Question No. 2.

6. What is Bounder Buffer Problem? Explain with the solution.

See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 9.

7. What are the advantages and disadvantages of Paging and Segmentation? What are the difference between a page and a frame?

See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 18.

Group – C
(Long Answer Type Questions)

8. a) Explain CPU scheduling criteria.

b) Explain the different states of a process using state transition diagram.

c) What are the main reasons to use of Thread rather than process for different applications?

d) Consider the following set of processes:

Process	CPU Burst Time	Priority	Arrival time
PO	80	3	0
P1	20	1	10
P2	10	3	10
P3	20	4	80
P4	50	2	85

Draw the Gantt chart using RR (t_s = 15) and preemptive priority scheduling. Calculate average waiting time and total Turnaround time.

a) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 23.
b) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 15.
c) See Topic: PROCESSES, THREAD & SCHEDULING, Short Answer Type Question No. 9.
d) See Topic: CPU SCHEDULING, Long Answer Type Question No. 9.

9. a) Explain what is Indexed Allocation of file-space on disk. What are the advantages and disadvantages of contiguous allocation?
b) Write down the merits and demerits of a virtual memory system.
c) Explain Belady's anomaly.
d) Compare best fit and first fit algorithm for memory allocation.

a) See Topic: FILE MANAGEMENT, Short Answer Type Question No. 3.
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 19.
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 4.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 20.

10. a) "All unsafe states may not lead to deadlock". – Why or why not?
b) Explain Critical-section problem. How semaphore can be used to solved it?
c) Consider the following snapshot of a system:

Process	Allocation	Max	Available
	ABCD	ABCD	ABCD
PO	0012	0012	1520
P1	1000	1750	
P2	1354	2356	
P3	0632	0652	
P4	0014	0656	

Answer the following questions using the Banker's algorithm:
i) What is the content of the matrix need?
ii) Is the system in a safe state? Justify.
iii) If a request from process P1 arrives for (0, 4, 2, 0) can the request be granted immediately? Answer with justification.

a) See Topic: DEADLOCK, Short Answer Type Question No. 3.
b) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 10.
c) See Topic: DEADLOCK, Long Answer Type Question No. 1(b).

11. a) What are the advantages of segmentation over paging?
b) Explain the difference between Internal and External fragmentation. Which one occurs in paging system? How the problem of external fragmentation be solved?
c) What is thrashing?
d) Consider the following sequence of memory references generated by a single program in a pure paging system:
10, 11, 104, 104, 170, 173, 177, 309, 245, 246, 247, 458, 364.

Determine the number of page faults for each of the following page replacement policies assuming 3(three) page frames are available and all are initially empty.
The size of a page is 100 words:

i) LRU
ii) FIFO
iii) Optimal page replacement.

a) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 4(b).
b) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 4(c).
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 9.
d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 1(b).

12. Write short notes on any three of the following:

a) Data transfer method of I/O devices.
b) Process control block (PCB)
c) Preemptive SJF scheduling
d) Starvation
e) Kernel-Level Thread and User-Level Thread
f) Security and Protection

a) See Topic: INPUT OUTPUT HARDWARE, Long Answer Type Question No. 2.
b) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(d).
c) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(c).
d) See Topic: DEADLOCK, Long Answer Type Question No. 5.
e) See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 11(f).
f) OUT OF SYLLABUS

QUESTION 2019 [5th-IT]

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for any *ten* of the following:

i) Internet provides …… For remote login.
✓a) telnet b) http c) ftp d) rpc

ii) For real time operating systems, interrupt latency should be
✓a) minimal b) maximum
c) zero d) dependent on the scheduling

iii) In distributed system, each processor has its own
a) local memory b) clock
✓c) both local memory and clock d) none of the mentioned

iv) A benefit of the microkernel organization is
a) extensibility b) portability c) flexibility ✓d) all of these

v) Time duration required for scheduling dispatcher to stop one process and start another is known as

a) process latency
✓b) dispatch latency
c) execution latency
d) interrupt latency

vi) Which of the following system call is used for opening or creating a file?

a) Read
b) Write
✓c) Open
d) Close

vii) Where does swap space reside?

a) RAM
b) ROM
✓c) Disk
d) Cache

viii) Network operating system runs on

a) every system in the network
✓b) server
c) both server and every system in the network
d) none of the mentioned

ix) The following program results in the creation of

```
main()
{
if(fork()>0)
sleep(100);
}
```

a) an orphan process
✓b) a zombie process
c) a process that executes forever
d) none of the mentioned

x) How many times the following C program prints yes?

```
main()
 {
  fork(); fork(); printf("yes")
 }
```

a) Only once
b) Twice
✓c) Four times
d) Eight times

xi) In distributed systems, link and site failure is detected by

a) polling
✓b) handshaking
c) token passing
d) none of the mentioned

xii) In real time operating system

a) all processes have the same priority
✓b) a task must be serviced by its deadline period
c) process scheduling can be done only once
d) kernel is not required

Group – B
(Short Answer Type Questions)

2. What are internal fragmentation and external fragmentation?
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 21.

3. a) What is Race condition?
b) What are the criteria to the solutions of the critical section problem?
a) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 2(a).
b) See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 3.

4. What is semaphore? How semaphore can be used for Dining Philosophers' problem?
1st part: See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 5(a).
2nd part: See Topic: INTER-PROCESS COMMUNICATION, Short Answer Type Question No. 11.

5. Discuss the necessary conditions for deadlock. What are the methods for handling deadlock?
1st part: See Topic: DEADLOCK, Long Answer Type Question No. 1(a).
2nd part: See Topic: DEADLOCK, Short Answer Type Question No. 10.

6. Discuss in brief: segmentation with paging.
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 22.

Group – C
(Long Answer Type Questions)

7. Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.
   a) How does the CPU interface with the device to coordinate the transfer?
   b) How does the CPU know when the memory operations are complete?
   c) The CPU is allowed to execute other program while the DMA controller is transferring data. Does this process interface with the execution of the user program If so, describe what forms of interference are caused?

See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 13.

8. Explain co-operating system. What is the purpose of interrupt? What are the differences between trap and interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?
See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 7.

9. a) Discuss real time, parallel and distributed operating system.
b) Discuss the advantages and disadvantages of Open sources OS.
c) What is / are main difficulty to write a program in real time operating system?
See Topic: INTRODUCTION, Long Answer Type Question No. 1.

10. Discuss file structure and directory structure. Discuss various techniques for directory implementation. Discuss the file allocation methods.
See Topic: FILE MANAGEMENT, Long Answer Type Question No. 4.

11. Discuss the CPU scheduling algorithms: FCFS, SJF, RR, priority. What are the benefits of threads? What are user and kernel threads?

1st part: See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 10.
2nd & 3rd part: See Topic: PROCESSES, THREAD & SCHEDULING, Long Answer Type Question No. 8.