a. See Topic: KNOWLEDGE & REASONING (USING LOGIC / GRAPHS / FRAMES / SCRIPTS / RULE), Long Answer Type Question No. 16.b).
b. See Topic: KNOWLEDGE & REASONING (USING LOGIC / GRAPHS / FRAMES / SCRIPTS / RULE), Long Answer Type Question No. 16.c).
c. See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 20.d).
d. See Topic: KNOWLEDGE & REASONING (USING LOGIC / GRAPHS / FRAMES / SCRIPTS / RULE), Long Answer Type Question No. 16.d).
e. See Topic: EXPERT SYSTEMS, Long Answer Type Question No. 1.

DISTRIBUTED OPERATING SYSTEM

NOTE:

WBUT course structure and syllabus of 7th Semester has been changed from 2013. **DISTRIBUTED OPERATING SYSTEM [CS 704A]** has been introduced as a new subject in present curriculum. Taking special care of this matter we are providing chapterwise model questions & answers along with the complete solutions of new university papers, so that students can get an idea about university questions patterns.

11. Write the short notes any three of the following:
    a) DQDB
    b) FTP
    c) Cryptography
    d) DNS
    e) ICMP
    f) HTTP

a) See Topic: APPLICATION LAYER, Long Answer Type Question No. 3(j).
b) See Topic: APPLICATION LAYER, Long Answer Type Question No. 3(f).
c) See Topic: APPLICATION LAYER, Long Answer Type Question No. 3(h).
d) See Topic: APPLICATION LAYER, Long Answer Type Question No. 3(c).
e) See Topic: NETWORK LAYER, Long Answer Type Question No. 17(g).
f) See Topic: APPLICATION LAYER, Long Answer Type Question No. 3(k).

OPERATING SYSTEM

NOTE:

WBUT course structure and syllabus of 3rd Year has been changed from 2012. Previously **Operating System [CS 501]** was in 5th semester, CS & IT branch. The Syllabus of this subject is almost same in **5th Sem IT & 6th Sem CS** as an optional Paper. Taking special care of this matter we are providing the relevant WBUT questions and solutions of **Operating System [CS 501]** papers from 2003 to 2012 & new university solved papers of both semesters. So that students can get an idea about university questions patterns.

INTRODUCTION AND SYSTEM STRUCTURE

Multiple Choice Type Questions

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

2. 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)

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. A multi-user, multiprocessing operating system cannot be implemented on hardware that does not support [WBUT 2012(CS)]
   a) address translation
   b) DMA for disk transfer
   c) at least two modes of CPU execution (privileged and non-privileged)
   d) demand paging
Answer: (d)

5. A benefit of the microkernel organization is [WBUT 2012(CS)]
   a) extensibility b) portability c) flexibility d) all of these
Answer: (d)

6. 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)

7. 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)

8. Which of the following is true about Multiprogramming? [WBUT 2012(IT)]
   a) Increase efficiency by overlapping CPU burst and I/O burst of different processes
   b) Allow users to write multiple programs
   c) Allow multiple users to use the computer
   d) All of these
Answer: (d)

9. Monolithic kernel is a characteristic of [WBUT 2012(IT)]
   a) Windows NT b) Linux c) DOS d) All of these
Answer: (d)

10. Micro kernel is characteristics of [WBUT 2012(IT)]
   a) Windows NT b) Linux c) DOS d) All of these
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. CPU generates [WBUT 2013(IT)]
   a) logical address b) physical address c) both (a) & (b) d) none of these
Answer: (a)

13. Which of the following loaders is executed when a system in first turned on or restarted? [WBUT 2013(IT)]
   a) compile and go loader b) boot strap loader
   c) relative loader d) absolute loader and relative loader
Answer: (b)

14. ..... 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)

15. 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)

Short Answer Type Questions

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

3. Mention one characteristic of distributed system. **[WBUT 2012(CS)]**

Answer:

Characteristics of distributed system are: Resource sharing, openness, concurrency, scalability, fault tolerance etc.

4. Differentiate between multiprogramming and multitasking OS. **[WBUT 2012(CS)]**

Answer:

Multiprogramming is a technique to execute number of programs simultaneously by a single processor.

Multitasking is a logical extension of multiprogramming only. It handles many tasks simultaneously.

Efficient memory utilization is in multi programming. It supports multiple simultaneous interactive user (terminals).

A multitasking system allow many users to share the computer simultaneously. Multitasking solves the problem by scheduling the tasks instructions.

5. 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.

6. What is the difference between Time Sharing and Batch system? **[WBUT 2012(IT)]**

Answer:

In time sharing system the tasks or processes are given specific time and after completion of time, operating system switches between different tasks. The switching is so fast that the computer user does not see any changes in the program. On the other hand, in batch system the job is keep in order and are run one after other. It is good for large job. The system is always busy by using buffering, spooling and multiprogramming. In batch system user interaction is involved in the processing. The job is bitter managed in time sharing system rather than batch processing.

7. What is a multiuser, multiprogram operating system? **[WBUT 2013(IT)]**

Answer:

A multiprogramming OS permits multiple programs to be loaded into the memory and execute the programs concurrently.

A multiuser OS support simultaneous interaction with multiple users. It also support user authentication, resource usage accounting for the interactive users and protection of user's environments.

8. What is boot strapping? Distinguish between multiprogramming and multithreading OS? State the function of batch processing system. **[WBUT 2014(CS)]**

Answer:

1st Part:

Bootstrapping is the process of loading a set of instructions when a computer is first turned on or booted. During the start up process, diagnostic tests are performed, such as power-on-self test or check configurations for devices and implement routine testing for the connection of peripherals, hardware and external memory devices. The bootstrap loader or program is than load to initialize OS.

2nd Part: ***Refer to Question No. 4 of Short Answer Type Questions.***

Last Part:
A batch operating system normally read a stream of separate jobs, each with its own control cards (called job control cards) that predefine what the job does. When the job is complete, its output is usually printed. The important feature of a batch system is lack of interaction between the user and the job while that job is being executed. Scheduling of jobs may be FCFS or SJN (Shortest job next).

9. 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.

10. 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 operati system via system calls and for error handling.
The CPU uses this to get the serial device interrupt service routine, which is th 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.

11. What is the purpose of the command interpreter? Why is it usually separa from the kernel? **[WBUT 2016(CS)]**

Answer:
1st Part:
Command interpreter needs command from the user or from a file of commands an 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.

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

Answer:
Refer to Question No. 1(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 ar 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 forma verifications of being to always meet to hard deadlines except for fatal errors.
Example:
Air traffic control, Nuclear power plant control, etc.

Long Answer Type Questions

1. Write short notes on the following:
a) **Real time systems** **[WBUT 2012(IT), 2015(IT)]**
b) **Dual mode operation** **[WBUT 2012(IT)]**
c) **Time sharing OS or Batch processing OS** **[WBUT 2013(IT)]**
d) **Spooling** **[WBUT 2015(CS)]**

Answer:

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.

Dual mode operation:

Dual mode of operation is the distinction between execution of user mode and kernel or supervisor mode. A mode bit is added to the hardware to indicate the current mode i.e. kernel (0) or user (1). Dual mode of operation provide us with more protection to the operating system. For example, if a kernel instruction is executed under user mode, the hardware does not execute it, although identifies it as an illegal execution and traps it to the operating system.

Time sharing OS or Batch processing OS:

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

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.

PROCESS MANAGEMENT

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. 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)

5. 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)

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)]

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)]

a) Short Term Scheduler b) Middle Term Scheduler
c) Long Term Scheduler d) Dispatcher

Answer: (a)

8. A process is [WBUT 2012(IT)]

a) a program b) a file
c) a program in execution d) a function of computer

Answer: (c)

9. Context switch is involved in [WBUT 2012(IT)]
   a) Switching computer on
   b) Switching CPU from one process to another
   c) Switching a process on
   d) None of these
Answer: (d)

10. Which of the following is true about multithreading? [WBUT 2012(IT)]
   a) It is a programming technique
   b) It is provided by linker
   c) It is an obsolete ideal now
   d) It increases responsiveness and efficiency of a program
Answer: (d)

11. Dispatcher is responsible [WBUT 2012(IT)]
   a) Dispatching files
   b) Giving control of CPU to a process selected by short time scheduler
   c) Increasing the throughput of the system
   d) None of these
Answer: (b)

12. Which is *not* a valid process state? [WBUT 2013(IT)]
   a) new b) ready c) run d) load
Answer: (d)

13. 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)

14. 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)

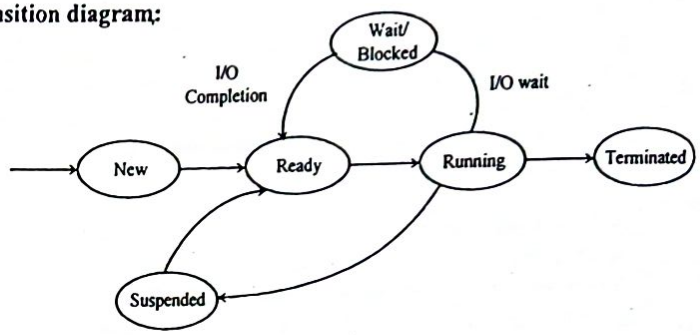
15. The number of processes completed per unit time is known as [WBUT 2017(CS)]
   a) Output b) Throughput c) Efficiency d) Capacity
Answer: (b)

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. 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

3. 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.

Part:

. 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.

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

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

OR,

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

OR,

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

OR,

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

OR,

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

swer:

's Control Block (PCB) is an operating systems data structure which contain the information for each process (i.e., one PCB per process). State information is ded to suspend and correctly resume process execution when another process is led 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

nt of a process:

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.

5. What are the advantages of using Thread over Process? What is System Call? **[WBUT 2012(IT)]**

Answer:

1st Part:

1) A programmer can create multiple threads within a process
2) Threads executes currently
3) Threads improve the performance (throughput, speed, responsiveness) of a program or process.
4) Thread uses single address space.

Last Part:

The system call is the request for service that a program makes of the kernel. The service is generally something that only the kernel has privilege to do such as I/O 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)]**

Answer:

a) *Refer to Question No. 4 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.

3. bw can context switch time be reduced? [WBUT 2007]
OR,
N is context switching considered to be time consuming? 92
[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.

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

Answer:
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.

11. What is a Process? Draw and explain the process state diagram? [WBUT 2014(IT)]

Answer:
1st Part:
Process is a dynamic entity i.e., it is an instance of a program in execution. It is represented by the 5-tuple (process id, code, data, register values, pc value).

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

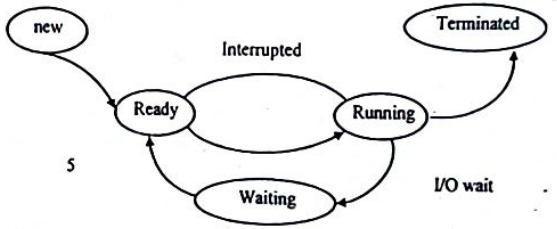
12. What is the main objective of Multiprogramming? Draw and describe process state transitions. [WBUT 2016(IT)]

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.

13. 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.

14. What is dispatcher? [WBUT 2017(CS)]

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.

Long Answer Type Questions

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

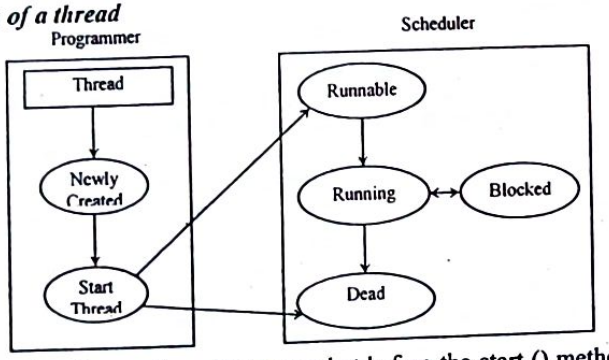
Answer:

1st Part: *Refer to Question No. 2(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.

2. Write short notes on the following:

a) Thread [WBUT 2008, 2012(IT)]
b) Kernel level thread [WBUT 2009, 2015(IT)]
c) Process Control Block [WBUT 2012(IT), 2016(IT)]
d) Kernel level thread & User level thread [WBUT 2015(CS)]
e) Multithreading models [WBUT 2015(IT)]
f) Process life cycle [WBUT 2016(CS)]
g) Orphan process and Zombie process [WBUT 2016(CS)]

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) Process Control Block:

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

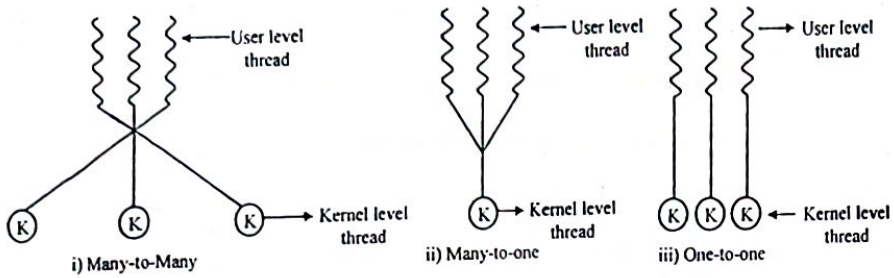
d) Kernel level thread & User level thread:

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

e) Multithreading models

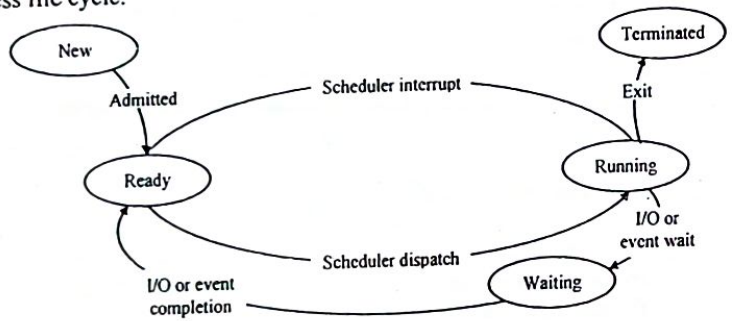
Some operating system provide a combined user level thread and kernel level thread facility. Solaris is a good example of this combined approach. In a combined system, multiple threads within the same application can run in parallel on multiple processors and a blocking system call need not block the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

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



f) 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.



g) 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.

CPU SCHEDULING

Multiple Choice Type Questions

1. The scheduler which selects jobs from the pool of jobs and loads them to th ready queue is [WBUT 2010, 2016(CS)]

a) long term b) short term c) medium term d) none of thes

Answer: (a)

2. 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)

3. Preemptive SJF is known as [WBUT 2012(IT)]

a) Shorted I/O Burst First b) Shortest CPU Burst First
c) Round Robin d) Shortest Remaining Time First

Answer: (d)

4. The short-term scheduler is responsible for: [WBUT 2014(CS)]

a) Selecting the shortest jobs to enter the system
b) Discarding job groups that should be removed from the system
c) Selecting which process should be allocated to the CPU next
d) Swapping jobs out of memory
e) None of the above

Answer: (c)

5. The average wait time for five processes PI-P5 with burst of 5, 19, 2, 16 and 7 milliseconds respectively, using SJF is: [WBUT 2014(CS)]

a) 5 milliseconds b) 10.6 milliseconds
c) 28.25 milliseconds d) 9.8 milliseconds
e) none of the above

Answer: (b)

6. Suppose that the operating system is running a non-preemptive scheduler and that process p is currently running. A context switch can occur: [WBUT 2014(CS)]

a) When p terminates process or blocks
b) When another process unblocks
c) When another process enters
d) When the time quantum is exhausted
e) When the priority of some other process exceeds the priority of p

Answer: (a)

1. 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)

1. 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)

1. 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)

Short Answer Type Questions

1. What is main feature of Multiprocessor scheduling? [WBUT 2012(IT)]
Briefly discuss Multilevel feedback queue scheduling?

Answer:

1st Part:

Multiprocessor scheduling is better than uniprocessor scheduling in the ways that which process to run and on which CPU and whether processes are unrelated or they come in groups. The simplest way of scheduling is that of time sharing by maintaining a global ready queue. It provides automatic load balancing because it can never happen that one CPU is idle while others are overloaded. The one approach to CPU scheduling in a multi-processor system has all scheduling decision, I/O processing and other system activities handled by a single processor called master server. The other processor executes only user code.

2nd Part:

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.

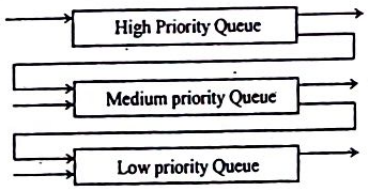


Fig: Multilevel Feedback Queue scheduling

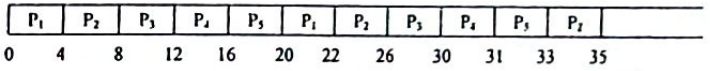
2. Consider the following set of processes with corresponding arrival times and burst times:

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 $P_1 = 22 - 6 = 16$

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

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

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

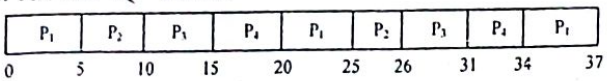
" " " $P_5 = 23 - 6 = 17$

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

3. Consider the following set of processes with their respective CPU execution times. Assume that they have arrived in the order shown. Draw the Gantt chart for Round Robin scheduling with time quantum = 5 time units. Calculate turnaround time of each process. [WBUT 2013(IT)]

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

Answer:
Gnatt chart for RR with TQ = 5 units.



Turn-around time
for, $P_1 = 37 - 0 = 37$ units
$P_2 = 26 - 2 = 24$ units
$P_3 = 31 - 3 = 28$ units
$P_4 = 34 - 5 = 29$ units

4. 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.

5. What is dispatch latency? [WBUT 2014(IT)]

Answer:
Dispatch latency is the time it takes for the dispatcher to stop one process and start another running. With a scheduler written specifically to honour application priorities real time applications can be developed with a bounded dispatch latency.

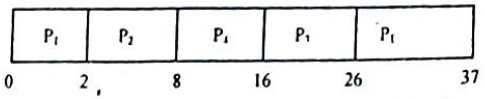
6. What is "Turn Around Time"? [WBUT 2015(CS)]

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

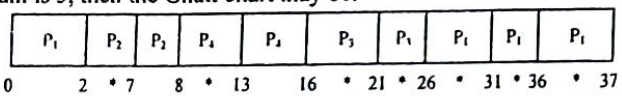
7. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds. The processes are assumed to have arrived in the order as shown below: Draw four Gantt chart for SRTF (shortest remaining time first) scheduling with time quantum = 5. [WBUT 2015(IT)]

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.

8. 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: $n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1$

9. a) What is Medium Term scheduler? [WBUT 2017(CS)]
b) Describe the functions of short-term and long-term scheduler.

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

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

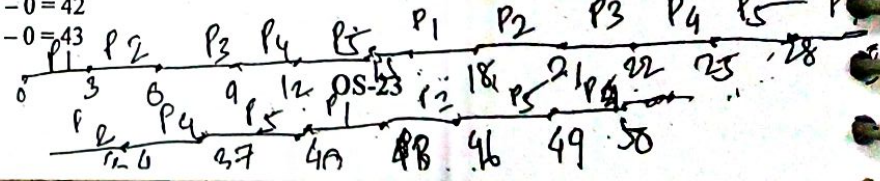
Processes	Arrival time	Burst time
P1	0	12
P2	0	10
P3	1	4
P4	4	10
P5	2	12

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 $= (30 + 33 + 17 + 30 + 33)/5 = 143/5 = 28.6$

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.

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

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

Processes	Burst Time	Arrival Time
P1	10	0
P2	5	1
P3	2	3
P4	7	5
P5	9	7

Deduce the Average Turn Around Time and Average Waiting Time using Shortest Remaining Time First and Round Robin Scheduling. **[WBUT 2014(IT)]**

Answer:

SRTF:

The Gnatt chart will be

P_1	P_2	P_3	P_2	P_4	P_1	P_5

0 1 3 5 8 15 24 33

Average turn around time

for $P_1 = 24 - 0 = 24$
$P_2 = 8 - 1 = 7$
$P_3 = 5 - 3 = 2$
$P_4 = 15 - 5 = 10$
$P_5 = 33 - 7 = 26$

A.T.T. $= 69/5 = 13.8$

Similarly, average waiting time $= \frac{15 + 3 + 0 + 3 + 17}{5} = \frac{38}{5} = 7.6$

RR: - (For Time Quantum = 5)

P_1	P_2	P_3	P_4	P_5	P_1	P_4	P_5

0 5 10 12 17 22 27 29 33

Average waiting time $= \frac{(22 + 4 + 7 + 10 + 5)}{5} = \frac{48}{5} = 9.6$

Average turn around time $= \frac{(27 + 9 + 9 + 24 + 26)}{5} = \frac{95}{5} = 19$

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

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:

a) 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.

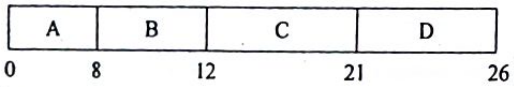
Throughput refers to the amount of work completed in unit time.

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) FCFS:



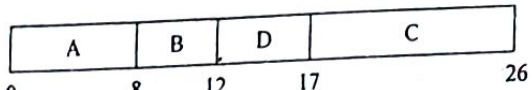
Average waiting time

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

Average turn-around time

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

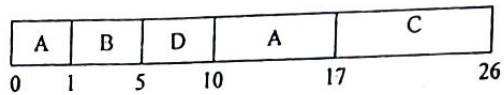
SSTF:



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

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

SRTF:

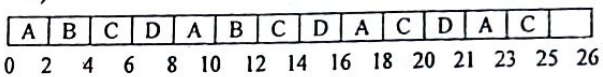


Average waiting time

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

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

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

4. Write short notes on the following:

a) Shortest Job First (SJF) scheduling **[WBUT 2009, 2015(CS)]**

b) Multilevel feedback queue scheduling **[WBUT 2013(IT), 2015(IT)]**

c) Aging Technique **[WBUT 2016(IT)]**

Answer:

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.

It 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.

b) Multilevel feedback queue scheduling:

Refer to Question No. 1(2nd Part) of Short Answer Type Questions.

c) 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 could be executed. In fact, it would take no more than 32 hours for a priority – 127 process to age to a priority – 0 process.

PROCESS SYNCHRONIZATION

Multiple Choice Type Questions

1. IPC stands for [WBUT 2006, 2013(IT)]
   a) Internal Program Controller b) Internal Process Controller
   c) Interprocess Communication d) None of these

Answer: (c)

2. Concurrent process are those who [WBUT 2010, 2013(CS)]
   a) do not overlap time b) overlap in time
   c) are exerted by a process at the same time d) none of these

Answer: (b)

3. 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)

4. Which of the following is a Inter Process Communication (IPC) mechanism
   a) PIPE b) Message Queue [WBUT 2012(IT)]
   c) Shared Memory d) All of these

Answer: (d)

5. Inter Process Communication (IPC) is used for [WBUT 2012(IT)]
   a) Communication between computer and printer
   b) Communication between computer and keyboard
   c) Communication between two separate processes
   d) None of these

Answer: (d)

6. Critical section is segment of code, which [WBUT 2012(IT)]
   a) Defines critical variables b) defines critical functions
   c) accesses shared resources d) both (a) and (b)

Answer: (c)

7. A semaphore is [WBUT 2012(IT)]
   a) a file b) a processes synchronization tool
   c) a memory management tool d) a disk management tool

Answer: (b)

8. 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)]
   a) data consistency b) race condition
   c) aging d) Starvation

Answer: (b)

9. 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)

10. 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)

11. 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)

Short Answer Type Questions

1. Mention the basic principle of RR scheduling. Specify the impact of time quantum on its performance. [WBUT 2011, 2014(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.

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

...l 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, ...nsider 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 ...n is lost. Suppose that process 1 stores last, so turn is 1. When both processes come to ...e 'while statement', process 0 executes it zero times and enter its critical region. Process 1 loops and does not enter its critical region 2.

...What is Message Passing and why it is used? What do you mean by direct and indirect communication? [WBUT 2012(IT)]

...nswer:

...Part:

In ...nessage passing, the information to be shared is physically copied from the sender ...ocess space to the address space of all the receiver process and this is done by ...ransmitting the data to be copied in the form of messages (block of information).

It enables processes to communicate by exchanging messages and allows programs to be ...itten by using simple communication primitives such as **send** and **receive**, i.e.

```
Send(destination, & message);
Receive (Source, & message);
```

...s efficient and simple to use.

...nd Part:

...direct communication (message), process B and A are the identities of the receiver and the source respectively i.e.

```
...ocess A;
----------
Send (B, message);
--------
Process B;
----------
...ceive (A, message);
```

The link is automatically established between the paths of processes that want to ...mmunicate. The link may be unidirectional, but is usually bi-directional.

...indirect communication (message), first process sends the message into the **mail box** and the second process removes a message from the mailbox, i.e.

```
...cess A;
---------
Send (mailbox 1, message);

Process B;
---------
...ceive (mailbox 1, message);
```

A link is established between a pair of processes only if they have a common mailbox. A ...ink can be associated with more than two processes and can be either unidirectional or

bi-directional. Message exchange between two processes, transfers the contents of message from the sender's to the receiver's addressing space.

4. Prove that this mechanism actually to prevents deadlock. [WBUT 2012(IT)]

Answer:

In the case of a dining philosopher problem, four conditions for deadlock are there

i) Mutual exclusion – Only one philosopher can use a fork at any one time

ii) Hold and wait - Deadlock only occurs when philosophers hold one fork and are waiting for the second

iii) No preemption - If one philosopher could take a from his neighbour then the problem is sorted out.

iv) Circular wait – The circular table shows how the philosopher chain his double back on itself to form the circular chain.

There are two cases to be considered

i) Odd number of philosopher (n) - if n is odd then there is no way in which to specify the order of each philosopher to pick up the forks, so that deadlock can never occur. Thus, there is no correct way to solve this

ii) If n is even then the problem can be solved by writing two different methods. If all the odd philosophers were to pick up the fork on their left and then one on their right and all even philosophers were to pick up the fork on their right and then one on their left, then no deadlock would occur.

5. 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)]

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.

6. 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? [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.

7. What is Semaphore? What are the operations on it? [WBUT 2013(IT)]
Answer:
1st Part: *Refer to Question No. 3(a) (1st Part) of Long Answer Type Questions.*
2nd Part: *Refer to Question No. 6(a) of Short Answer Type Questions.*

8. a) What is semaphore? [WBUT 2014(IT)]
Answer:
Refer to Question No. 3(a) (1st Part) of Long Answer Type Questions.

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
```

9. Explain RPC. [WBUT 2014(IT)]
Answer:
Remote Procedure Call (RPC) is a powerful technology for creating distributed client/server program. RPC is an interprocess communication technique that allows client and server software to communicate. The Microsoft RPC facility is compatible with the open groups distributed computing environment specification for remote procedure calls and is interoperable with other DCE-based RPC system. RPC serves as a go-between for client / server communications. RPC is designed to make client / server interaction easier and safer by factoring out common tasks, such as security, synchronization and data flow handling. RPC services are:
Background intelligent transfer service, cryptographic services, DNS server, error reporting services etc.

10. 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);
        }
   }
```

11 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. 3(a)(1st Part) of Long 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.

Long Answer Type Questions

1. Write down the Readers – Writers algorithm using. Mention the difference between First – Reader / Writer and Second–reader / Writer problem. [WBUT 2007]

Answer:

1st Part:

In this problem, a number of concurrent processes require access to some object (such as a file). Some processes extract information from the object and are called Readers; others change or insert information in the object and are called writers. The Bernstem conditions state that many readers may access the object concurrently, but if a writer is accessing the object, no other processes (readers or writers) may access the object.

Structure of a Writer Process:

```
While (true)
{wait (wrt);
        // writing is performed //
signal (wrt);
}
```

Structure of a Reader Process:

```
While (true)
{Wait (mutex);
read count ++;
if (read count) =1) Wait (wrt)
        signal (mutex);
                // reading is performed //
        Wait (mutex);
        read count – –;
        if (read count =0) signal (wrt);
                Signal (mutex);
    }
```

2nd Part:

The readers – writers problem has several variations, all involving priorities. The simplest one referred to as the first readers – writers problem; requires that no reader will be kept waiting unless a writer has already obtained permission to use the shared object. In other words, no reader should wait for other readers to finish simply because a writer is writing. The second readers – writer problem requires that, once a writer is ready, the writer performs its write as soon as possible. In other words, if a writer is waiting to access the object no new readers may start reading.

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

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.

3. a) What is semaphore? What are the different types of semaphore? **[WBUT 2012(CS)]**

Answer:

1st Part:

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.

2nd Part:

Basically the semaphore are of two types: **Binary** and **Counting** semaphores. A binary semaphore is an integer variable which can be accessed by a cooperating process through the use of two primitives i.e. wait and signal. Counting semaphore may have value to be greater than one, typically used to allocate resources from a pool of identical resources.

b) 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));
}
```

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

b) What is monitor? Give a solution of bounded buffer problem with monitor. **[WBUT 2012(IT)]**

c) Describe Dining Philosopher problem. **[WBUT 2012(IT)]**

Answer:

a) 1st Part: *Refer to Question No. 3(a) of Long 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.

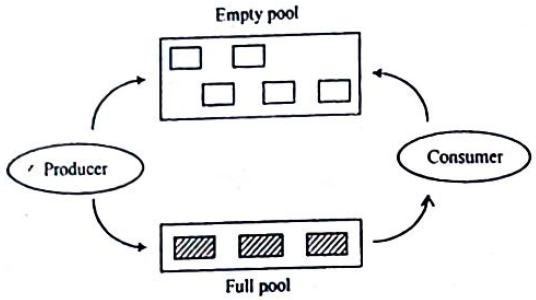
CPU cycles are saved in counting semaphore as a waiting process does not perform any busy waiting.

A counting semaphore is more complex to implement, since it involves implementation of FCFS queue.

b) A monitor is defined as a collection of procedures, variable and data structures that are all grouped together in a special kind of module or package. It is a high level tool for process synchronization. Bounded buffer problem is also called the producers and consumer problem. A finite supply of containers is available.

Producers take an empty container and fill it with a product. Consumers take a full container, consume the product and leave an empty container. The main complexity of this problem is that we must maintain the count for both the number of empty and full containers that are available.

Producers produce a product and consumers consume the product, but both use of one of the containers each time.



The solution to the bounded buffer problem with N containers using a monitor, is given below.

```
Import Threading
Class pool (object) ;
def– unit – (self, start);
self. items = start
self. monitor = threading . condition ( )
def get (self) ;
self. monitor . acquire ( )
While (Self . items = = 0) ;
If. monitor. Wait ( )
self. items - = - 1
self. monitor . release ( )
def put (Self) ;
self. monitor . acquire ( )
If. item + = 1
self. monitor . notify ( )
self. monitor . release ( )
def producer ( ) ;
global full
global empty
while True ;
empty . get ( )
" Fill the container with product

Full . put ( )
def consumer ( ) ;
global full
global empty
while True ;
full . get ( )
" Consume the product from the monitor

empty . put ( )
```



```
# The global lode
#
full = pool ( 0 )
empty = pool ( N )
```

c) *Refer to Question No. 3(b) of Long Answer Type Questions.*

5. a) Define the critical section problem. **[WBUT 2014(CS)]**
b) Identify the requirements to be satisfied to solve the critical section problem.
c) What is semaphore? How is it used to overcome critical section problem?

Answer:

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

c) 1st Part: *Refer to Question No. 3(a)(1st Part) of Long 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.

6. a) How mutual exclusion, hold & wait and circular wait are different from each other? **[WBUT 2014(CS)]**

Answer:

In mutual exclusion, at least one resource must be held in a non-sharable mode, but in hold & wait, a process must be simultaneously holding at least one resource and waiting for at least one resource that is currently being held by some other process. Whereas in circular wait, a closed chain of processes exists such that each process holds at least one resource needed by the next process in the chain.

b) How does the monitor give the solution for synchronization problem? What is binary semaphore? **[WBUT 2014(CS)]**

Answer:
1st Part:
A high level abstraction that provides a convenient and effective mechanism for process synchronization is monitor.
Only one process may be active within the monitor at a time. High level synchronization construct the allows the safe sharing of an abstract data type among concurrent processes.

```
Monitor monitor-name
Shared variable declarations
  Procedure body P1 (- - - -){
} - - - -
Procedure body P2 (- - - -){
} - - - -
Procedure body Pn (- - - -){
} - - - -
{ Initialization code
}
}
```

Process synchronization allow a process to wait within the monitor, a condition variable must be declared as condition x, y;
Condition variable can only be used with the operation wait and signal.

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

7. 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 & 3rd Part: *Refer to Question No. 3(a) of Long Answer Type Questions.*

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 Part & 6th Part:
Refer to Question No. 1(1st part) of Long Answer Type Questions.

8. 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 the heap management structures.

9. Write short notes on the following:
a) Dining-Philosopher Problem **[WBUT 2015(CS)]**
b) Peterson solution for CS **[WBUT 2016(CS)]**
Answer:
a) Dining-Philosopher Problem:
Refer to Question No. 3(b) of Long Answer Type Questions.

b) Peterson solution for CS:
Refer to Question No. 8(b) 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)]
   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)

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)]

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

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.

5. Describe a deadlock prevention mechanism. **[WBUT 2012(IT)]**

Answer:

Deadlock might be prevented by denying any one of the conditions below:

- **Elimination of mutual exclusion** – The mutual condition must hold for non-sharable resources, i.e. several processes cannot simultaneously share a single resource. Thus

condition is difficult to eliminate because some resources such as tape drive and printer are inherently non-sharable. Note that sharable resources like read only file do not require mutually exclusive access and thus cannot be involved in deadlock.

- **Elimination of hold and wait** – There are two possibilities for elimination of 2nd condition. The first alternative is that a process request be granted all of the resources it needs at once, prior to execution. The second alternative is to disallow a process from requesting resources whenever it has previously allocated resources. This strategy requires that all of the resources a process will need must be requested at once.
- **Elimination of "No-preemption" condition** – Forcing a process waiting for a resource that can not immediately be allocated to relinquish all of its currently held resources, so that other processes may use them to finish, can alleviate the no-preemption condition.
- **Elimination of "Circular Wait" condition** – 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 or decreasing). This strategy imposes a total ordering of all resources types, and to require that each process request resources in a numerical order of enumeration with this rule, the resource allocation graph can never have a cycle.

6. 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.*

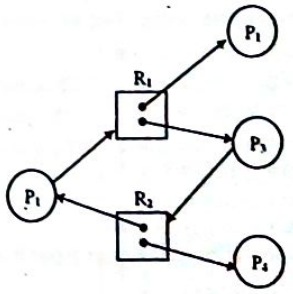
2nd Part: *Refer to Question No. 1(a) of Long Answer Type Questions.*

7. 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.

8 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.

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)]**

OR,

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

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)]

Answer:

a) 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) 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
$(0, 4, 2, 0) \leq$ Available (i.e. 1, 5, 2, 0)
$(0, 4, 2, 0) \leq$ 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_i to P_5 denote processes. The vector 'Available' has usual meaning.

Available

r1	r2	r3	r4
2	1	0	0

Process	*Current allocation:* r1	r2	r3	r4	*maximum demand:* 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, $(0,0,0,0) \leq (2,1,0,0)$

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

Step-2 P_4 is selected

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

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

Step-3 P_5 is selected

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

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

Step-4 P_2 is selected

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

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

Step-5 P_3 is selected

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

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. a) What is the difference between Deadlock avoidance and Deadlock prevention techniques? **[WBUT 2013(IT)]**

Answer:

Deadlock prevention	Deadlock avoidance
Preventing deadlock by constraining how requests for resources can be made in the system and how they are handled. The goal is to ensure that a least one of the necessary condition for deadlock can never hold.	The system dynamically considers every request and decides whether it is safe to grant it at this point. The system requires additional apriori information regarding the overall potential use of each resources for each process. Allows more concurrency

b) Discuss any one Deadlock prevention technique with its disadvantages and advantages. **[WBUT 2013(IT)]**

Answer:

Advantages: *Refer to Question No. 3(2nd Part) of Short Answer Type Questions.*

Disadvantages:

i) Requires that there be a fixed number of resource to allocate.
ii) Requires that the population of processes to be fixed.
iii) Requires that the banker to grant all requests within finite time
iv) Requires that clients repay all loans within finite time.
v) Requires that processes to state maximum needs in advance.

4. a) What is Deadlock? List four necessary conditions for the occurrence of deadlock. **[WBUT 2014(IT)]**

Answer:

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

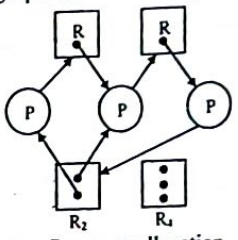
2nd Part: *Refer to Question No. 1(a) of Long Answer Type Questions.*

b) Does presence of cycle in a resource allocation graph necessarily create deadlock. **[WBUT 2014(IT)]**

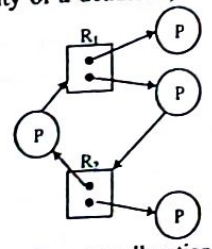
Answer:

If a resource-allocation graph does contain cycles and each resource category contains only a single instance, then a deadlock exists.

If a resource category contains more than one instance, then the presence of a cycle in the resource allocation graph indicates the possibility of a deadlock, but does not guarantee one i.e.



Resource allocation graph with a deadlock



Resource allocation graph with a cycle but not deadlock

5. 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.

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)]
- 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)]
- 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. Virtual memory means [WBUT 2008]
- a) the job size is not bounded by physical memory limit
- b) the job size is bounded by the physical memory limit
- c) independent of physical memory limit
- d) none of these

Answer: (a)

7. 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]
- a) 4
- b) 8
- c) 16
- d) 32

Answer: (c)

8. 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)

9. Which of the following schemes suffers from External Fragmentation? [WBUT 2011, 2013(IT)]
   a) Segmentation b) Paging
   c) Paged segmentation d) All of these

Answer: (a)

10. 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)

11. Translation look aside Buffer is a kind of [WBUT 2013(CS)]
   a) interrupt b) cache c) virtual memory d) i/o device

Answer: (b)

12. Paging suffers from [WBUT 2013(CS)]
   a) internal fragmentation b) external fragmentation
   c) both (a) & (b) d) none of these

Answer: (a)

13. Which of the following algorithm generally suffers from Belady's anomaly
   a) optimal b) FIFO c) LRU d) all of these [WBUT 2013(CS)]

Answer: (b)

14. Swap space generally resides on [WBUT 2013(CS)]
   a) main memory b) files c) programs d) disk

Answer: (d)

15. TLB is a kind of [WBUT 2014(IT), 2016(IT), 2017(IT)]
   a) virtual memory b) interrupt c) cache d) main memory

Answer: (c)

16. 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)

17. Compaction is used to solve the problem of [WBUT 2016(CS)]
   a) external fragmentation b) internal fragmentation
   c) both (a) and (b) d) none of these

Answer: (a)

18. 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)

19. Which is the fastest of the following? [WBUT 2017(CS)]
   a) Cache memory b) RAM c) CD-ROM d) Register

Answer: (d)

20. The mechanism that brings a page into memory only when it is needed, called [WBUT 2017(CS)]
   a) Segmentation b) Fragmentation
   c) Demand paging d) page and replacement

Answer: (c)

21. Where does not swap space reside? [WBUT 2017(IT)]
   a) RAM b) DISK c) ROM d) On-chip cache

Answer: (b)

Short Answer Type Questions

1. What is fragmentation? [WBUT 2007, 2008, 2014(CS)]
How can it be overcome? [WBUT 2008]

OR,

What is the problem of fragmentation and how can it be solved? [WBUT 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.

2. 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,

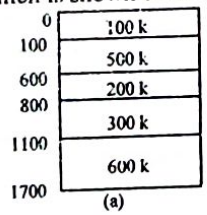
Compare best fit and first fit algorithm for memory allocation. [WBUT 2011]

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)]

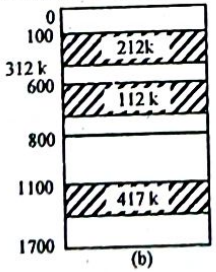
Answer:

Consider the memory partitions which is shown below:



(a)

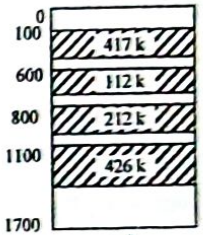
In first fit, the first hole which is large enough for use to satisfy the request so that the first process of 212k is allocated the block of 500k.



(b)

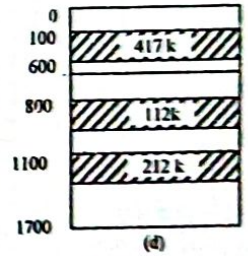
Second process of 417k has been placed in the block of 600k, the third process of 1 has been placed in the block of 200k and the last request which is 426k came in system, then the condition will not be satisfied since there is no block which is la enough to place it.

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



(c)

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



(d)

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

3. Briefly explain Belady's anomaly.
[WBUT 2007, 2009, 2010, 2011, 2014(IT), 2017(C

Answer:

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

1. 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)]**

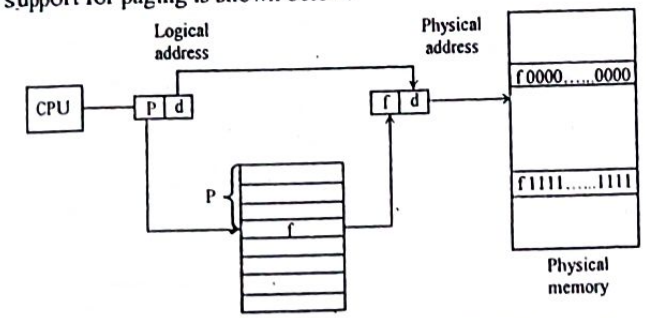
OR,

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

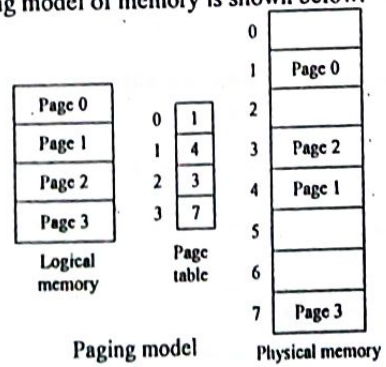
Answer:

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.

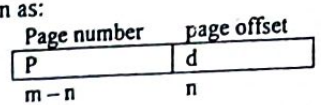


Every address generated by the CPU is divided into two parts: a page number (P) and a page offset (d). The page number is used as an index into a page table. The page table contains the base address of each page in physical memory. This base address is combined with the page offset to define the physical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



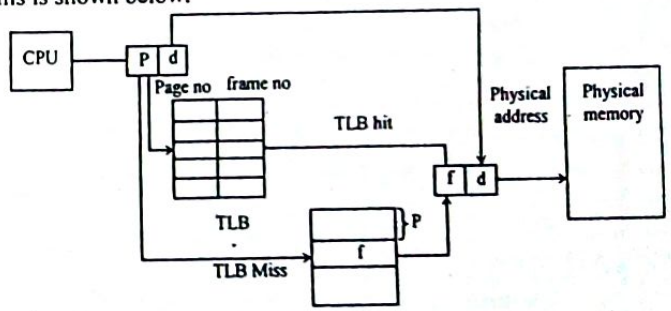
The page size is defined by the hardware. The size of a page is typically a power The selection of power of 2 as a page size makes the translation of a logical address into page number and a page offset. If the size of the logical address space is 2^m and a size is 2^n addressing units, then the high-order m–n bits of a logical address designa 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 part a key (or tag) and a value. When the associative memory is presented with an iter item is compared with all keys simultaneously. If the item is found, the correspondi value field is returned. Here the search is fast, the hardware however is expe Typically the number of entries in the TLB is small often numbering between 64 a 1024. The TLB is used with page tables in the following way. The TLB contains few of the page-table entries. When a logical address is generated by the CPU its pa number is presented to the TLB. If the page number is found, its frame num immediately available and is used to access memory, the whole task may take less th 10% longer than it would if an unmapped memory reference were used. If the number is not in the TLB, a memory reference to the page table must be made, known TLB Miss. When the frame number is obtained, we can use it to access memory, k 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 TLB contains a valid bit, a virtual page number, an M bit, an R bit and a page number.

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

5. What are the advantages and disadvantages of having unequal size partitions 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.

6. 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.

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

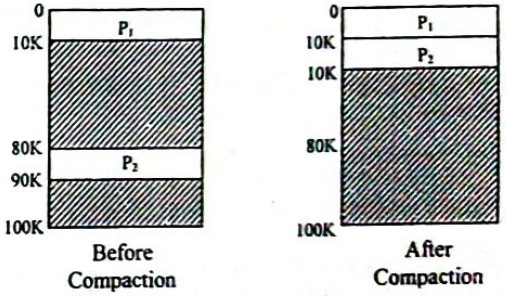
caching system uses a direct-mapped cache with 16-byte cache lines (i.e., 512 cacl lines). If a program references data objects 8K apart on each access then the system w have to replace the same line in the cache over and over again with each access. Th occurs even though the other 511 cache lines are currently unused.

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

8. Explain compaction. **[WBUT 2010, 2014(IT**

Answer:

Compaction is a technique to place all free memory together in one large block and t reduce external fragmentation. Compaction is possible only if relocation is dynamic an 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 b compacted into one hole of size 80K. Compaction is not always possible. For processe to be able to execute in their new locations all addresses must be relocated. If relocatio is static compaction cannot be done.

9. What is the main disadvantage of allocating memory contiguously?
[WBUT 2012(IT)

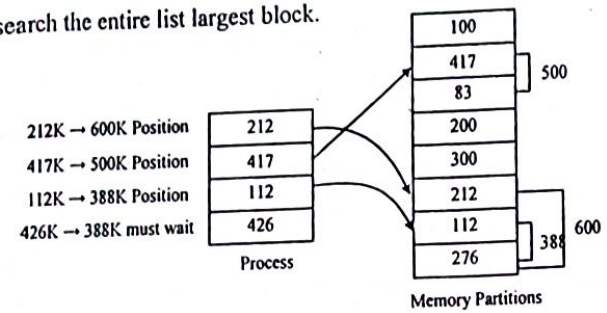
Answer:

It is suitable for sequential files. The difficulty is that it is not suitable to find th contiguous free blocks in the disk. Other problem is external fragmentation – whicl means that some free blocks could happen between two files.

It complicates "mapping around" the bad block, which can also continue in fragmentatio by breaking the contiguity of disk areas. Moreover, the method is very simple t implement as contiguous chunk of space is avoidable easily.

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), 2017(IT)]**

Answer:
1st Part: *Refer to Question No. 8 of Short Answer Type Questions.*

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 predicted to be initially needed by the process. The secondary memory holds resident pages of the process. Subsequently, when the process needs some pages that may not in memory resident, the pager will load those pages into the RAM. The technique called demand paging. For its implementation, OS needs some hardware support.

13. What is demand paging? What are its advantages and disadvantages? **[WBUT 2013(I**

Answer:
1st Part:
Virtual memory is commonly implemented by demand paging. With demand-paged virtual memory pages are only loaded when they are demanded during program execution; pages that are never accessed are thus never loaded into physical memory. A demand-paging system is similar to a paging system with swapping where process resides in secondary memory usually a disk. When we want to execute a process, we swap it into memory.

2nd Part:
Advantages:
i) Lesser I/O is needed
ii) less memory is needed
iii) faster response time

Disadvantages:
i) Increased load time when a page is loaded
ii) Increased vulnerability to timing attacks.

14. What is the cause of thrashing? How does the system detect trashing? Once detects thrashing, what can the system do to eliminate this problem? Given memory partitions of 100 KB, 500 KB, 300 KB, and 600 KB (in order) **[WBUT 2014(CS**

Answer:
1st Part: *Refer to Question No. 14(b) of Long Answer Type Questions.*
Last Part: *Refer to Question No. 2 of Short Answer Type Questions.*

15. Explain different types of fragmentation. Which one may occur in paging system? **[WBUT 2014(CS**

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

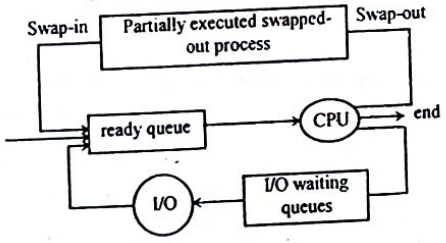
16. Why are segmentation and paging sometimes combined into one scheme? **[WBUT 2015**

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

7. The list of all passwords is kept within the operating system. Thus, if a user nanages 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.



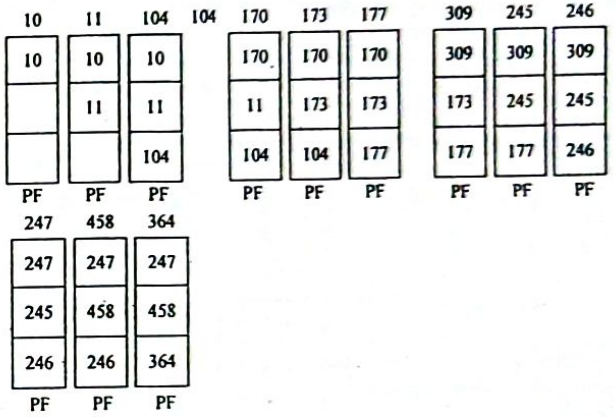
Long Answer Type Questions

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 algo. 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 sing 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 replacemei 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

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 i the memory.
i) If memory reference takes 200 ns, how long does a paged memory referenc take?
ii) If we add TLBs and 75% hit is successful, what is the effective memor reference time (Assume that finding page-table entry in the TLBs take zero time, the entry is there.) [WBUT 2008

OR,

i) Consider a paging system with the page table stored a paged memory referenc take.
ii) If we add TLBs and 75 per cent of all page-table references are found in th TLBs what is the effective memory reference time? (Assume that finding a pag 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 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?
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?
ii) Which replacement strategy in the above performs better and why?
[WBUT 2009, 2015(CS)]

Answer:

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

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

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

Page offset = 22bits

$\therefore$ 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) 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
	✓	✓	×	✓	✓	×	×	✓	×	×	×	✓	×	×	×

$\therefore$ 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
	✓	✓	×	✓	✓	×	×	✓	×	×	✓	✓	✓	×	×

$\therefore$ 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?
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. **[WBUT 2010, 2015(IT), 2017(CS)]**
Answer:
a) 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) 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) 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.

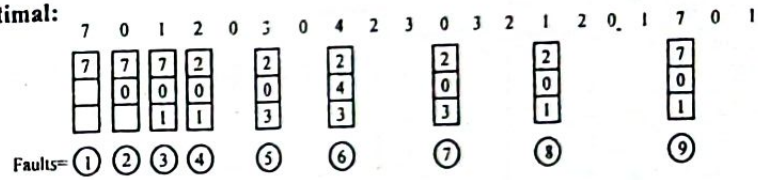
d) *Advantages:*
- OS can easily move a process during execution
- OS can allow a process to grow over times
- Simple, fast hardware: two special register an add and a compare.

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)]

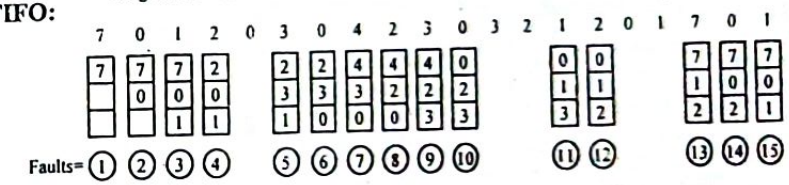
Answer:
a) The modify bit is set by the memory management hardware when there is a write - memory reference to that page. When a page is first loaded into memory, the modify or dirty bit is cleared. If the dirty bit for a page selected for replacement is not set, the page has not been modified since 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)]
b) Describe the actions taken by the operating system when a page fault occurs. [WBUT 2012(IT)]

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)]**

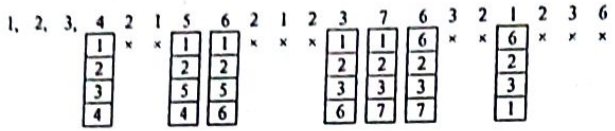
Answer:
a) A page fault occurs when an access to a page that has not been brought into main memory takes place. This situation is said to be page fault.

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

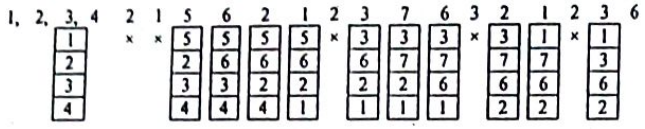
For the frames:

LRO:



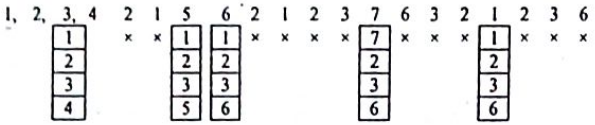
Page fault = 10

FIFO:



Page fault = 14

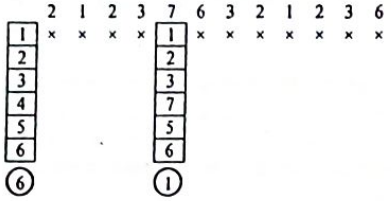
Optimal:



Page fault = 8

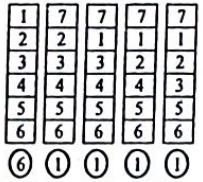
For Six (6) Frames:

LRU



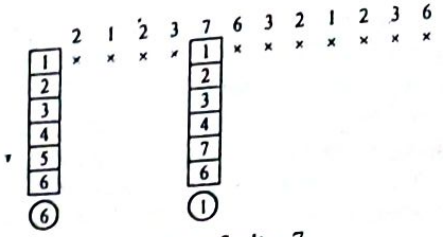
Page fault = 7

EIFO:



Page fault = 10

Optimal

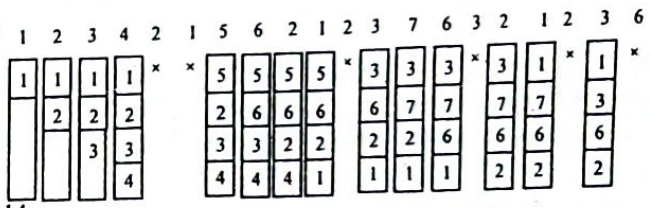


Page fault = 7

7. a) Consider the following page reference string and a memory consisting of 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 techniq
How can these disadvantages be avoided if segmentation with paging is used?
[WBUT 2013(CS), 2017(I

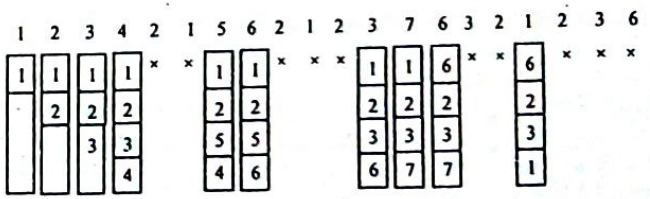
Answer:

a) i) FIFO:



Page fault = 14

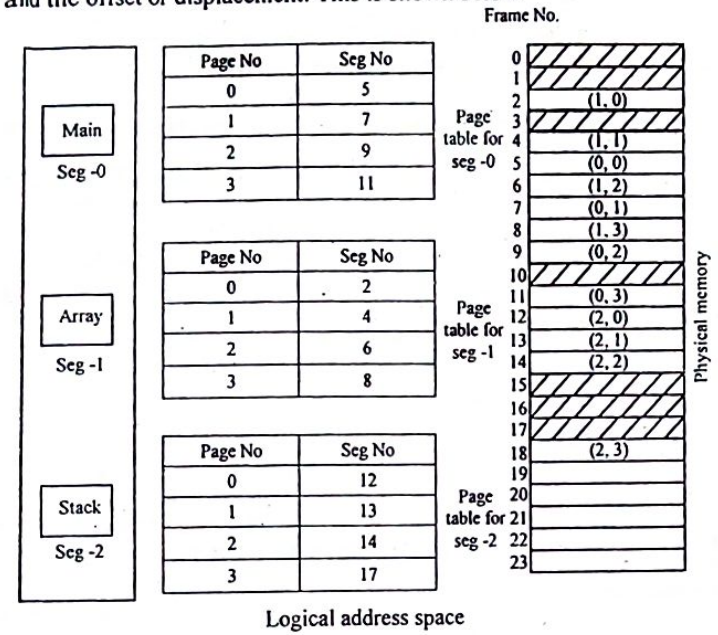
ii) LRU:



∴ 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. T 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 gr 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 blo 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:



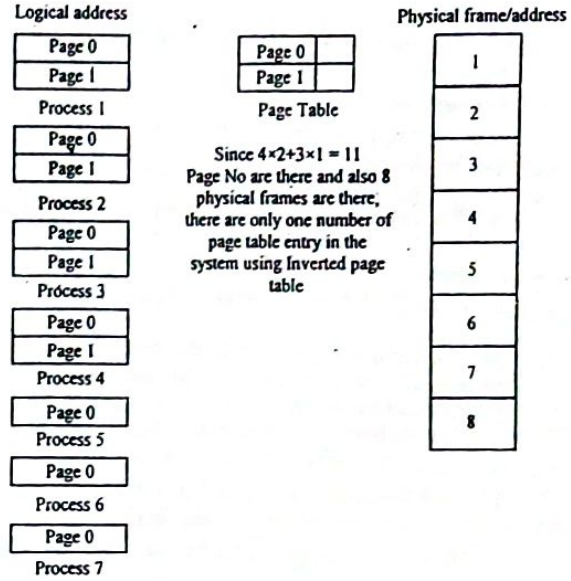
8. a) What is Direct Memory Access? How is it performed? What are its benefits? **[WBUT 2013(CS), 2014(CS), 2017(IT)]**

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) Discuss the relative advantages and disadvantages of segmentation and paging. **[WBUT 2013(IT)]**

Answer:

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

b) Consider the following page reference string:-
1, 2, 3, 4, 21, 56, 21, 2, 3, 7, 6, 3, 2, 1, 2, 36
How many page faults would occur for the following replacement algorithms? Assuming 4.5 frames are available.

i) LRU replacement.
ii) FIFO replacement. **[WBUT 2013(IT)]**

Answer:

Refer to Question No. 7(a) of Long Answer Type Questions (similar type).

10. a) What are the advantages and disadvantages of the Best Fit algorithm? **[WBUT 2013(IT)]**

Answer:

Advantages:

i) Minimal External fragmentation.
ii) Makes the best use of memory spaces.

Disadvantages:

i) Many small spaces.
ii) Slower in making allocation.

b) What are the advantages of the First fit algorithm? [WBUT 2013(IT)]

Answer:

Advantages:

i) Quite fast, good for recommending holes.

ii) It is faster than best fit.

c) What are the differences between two types of fragmentation in memory? [WBUT 2013(IT)]

Answer:

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

11. What is paging? Explain the hardware for paging? How does paging differ from segmentation with respect to hardware? What is Virtual memory? How can segmentation be done with the concept of virtual memory? What is External fragmentation? [WBUT 2014(IT)]

Answer:

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

2nd Part: *Refer to Question No. 4(1st Part) of Short Answer Type Questions.*

3rd Part:

In segmentation, the main memory is partitioned into segments whereas in paging it is partitioned into frames. In segmentation logical address is divided into segment whereas in paging it is divided into pages. Segmentation suffers external fragmentation whereas paging suffers internal fragmentation. Segmentation support user's view of memory but paging does not.

4th Part: *Refer to Question No. 12(1st Part) of Short Answer Type Questions.*

5th Part:

Segments may also be used to implement virtual memory. In this case, each segment has an associated flag indicating whether it is present in main memory or not. If a segment is accessed that is not present in main memory, an exception is raised and the operating system will read the segment into memory from secondary storage. Some systems, such as Burroughs B5500 use segmentation instead of paging, dividing virtual address spaces into variable length segments. A virtual address here consist of a segment number and an offset within the segment.

6th Part:

External fragmentation happens with a dynamic memory allocation algorithm allocates some memory and a small piece is left over that cannot be effectively used. If too much external fragmentation occurs, the amount of usable memory is drastically reduced.

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

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 for the PC usually contain their own high speed DMA hardware.

b) *Refer to Question No. 7 of Short Answer Type Questions.*

13. a) What is paging? Differentiate between internal and external fragmentation What is thrashing? [WBUT 2016(CS), 201

Answer:

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

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

3rd Part: *Refer to Question No. 7 of Short Answer Type Questions.*

b) What is TLB? What do you mean by 'Belady's Anomaly'? [WBUT 2016(CS), 20

Answer:

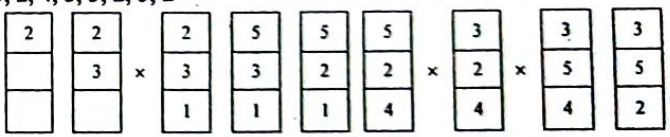
Refer to Question No. 3 & 4 of Short Answer Type Questions.

c) Having 3 physical memory frames show the behavior of LRU and FIF optimal page replacement algorithm for the page address string like 2, 3, 2, 1, 5, 4, 5, 3, 2, 5, 2. [WBUT 2016(CS), 201

Answer:

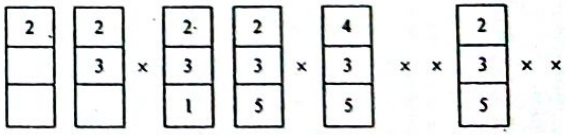
FIFO:

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



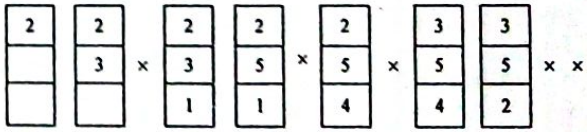
Page fault = 9

Optimal:



Page fault = 6

LRU:



Page fault = 7

14. Write short notes on the following:
a) DMA and its utility [WBUT 2008, 2012(CS), 2013(IT), 2015(IT)]
b) Paging [WBUT 2012(IT)]
c) Segmentation [WBUT 2012(IT), 2016(CS)]
d) Multi level paging [WBUT 2012(IT), 2015(IT)]
e) Belady's Anomaly [WBUT 2015(CS)]
f) Hierarchical page table [WBUT 2015(IT)]
g) External Fragmentation & Internal Fragmentation [WBUT 2016(IT)]

Answer:

a) DMA & its utility:

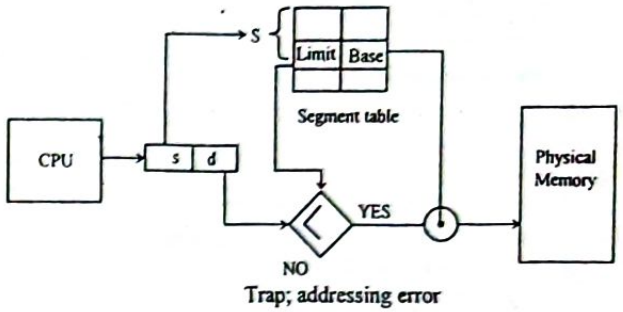
DMA is a technique which used to transfer data between memory and I/O devices with less effort on behalf of the operating system. The DMA unit has access to the data bus and can transfer data autonomously in and out of memory. In practice, a program would instruct the DMA unit to, say, transfer a specified 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) Paging:

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.

c) 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.



d) Multi level paging:

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

e) Belady's Anomaly:

Refer to Question No. 3 of Short Answer Type Questions.

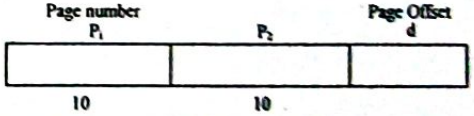
f) Hierarchical page table.

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

Example:

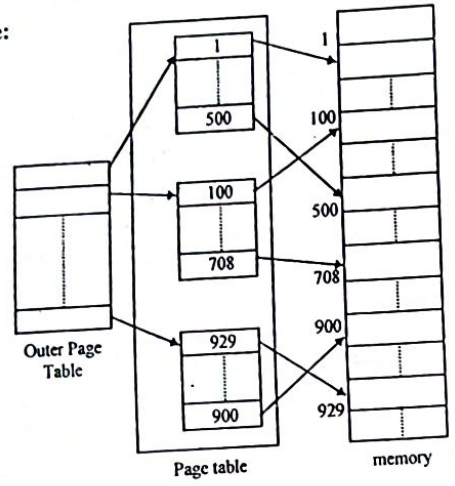
A logical address (on 32-bit machine with 4k page size) is divided into
i) a page number consisting of 20 bits
ii) a page offset consisting of 12 bits.

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_1 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:



g) External Fragmentation & Internal Fragmentation.

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

INPUT / OUTPUT MANAGEMENT

Multiple Choice Type Questions

1. Disk I/O generally done in terms of [WBUT 2013(CS)]
   a) sectors b) bytes c) blocks d) bits

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/C call returns as much as available.

- It is implemented via multithreading
- It returns quickly with count of bytes read or written.

Long Answer Type Questions

1. 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.

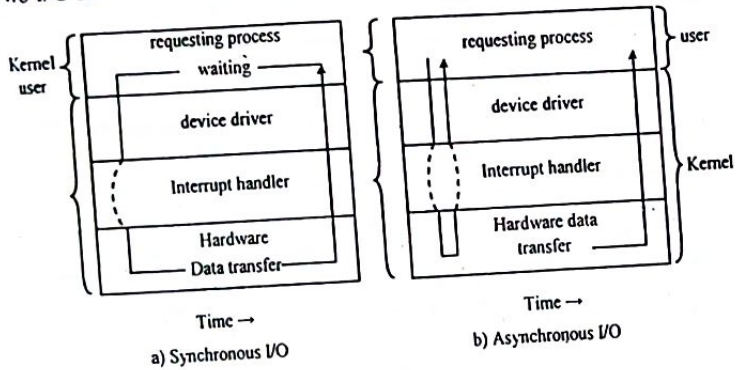
b) The indexed file allocation method is the solution to the problem of both contiguous and linked allocation. This is done by bringing all the pointers together into one location called the index block. It supports direct access i.e. the ith entry in the index block points to the i^{th} sector of the file. It is reliable. Any free block anywhere on the disk may satisfy a request for more space.

2. 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.



FILE SYSTEM

Multiple Choice Type Questions

1. The first block of a file system is **[WBUT 2013(CS)]**

a) superblock b) inode blocks c) data block d) boot block

Answer: (d)

2. 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)

3. 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)

4. 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. Mention any two file access method? **[WBUT 2012(IT)]**

Answer:

There are several files access methods:

- Read
- Write
- Execute
- Append
- Delete
- List

2. 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.

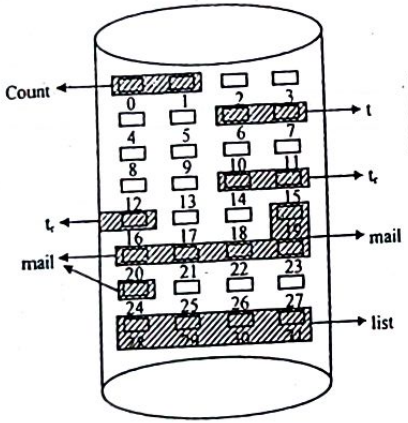
Each method has its advantages and disadvantages. Accordingly, some systems support all three (e.g. Data General's RDOS). More commonly, a system will use one particular method for all files.

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
t4	14	3
mail	19	6
list	28	4
t	6	2

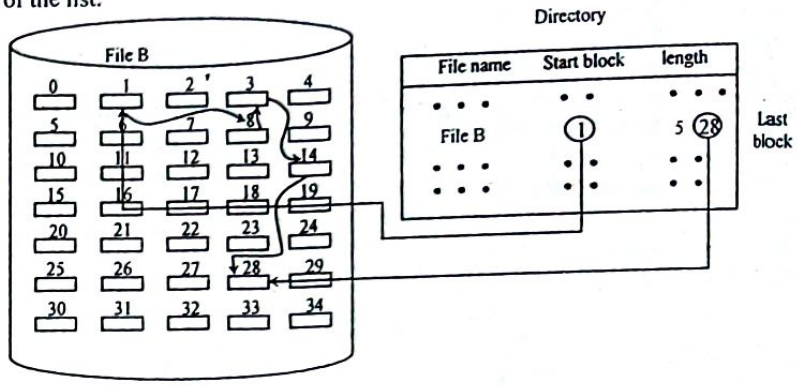
It is suitable for sequential files. The difficulty is that it is not suitable to find the contiguous free blocks in the disk. Other problem is external fragmentation – which means that some free blocks could happen between two files.

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 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:

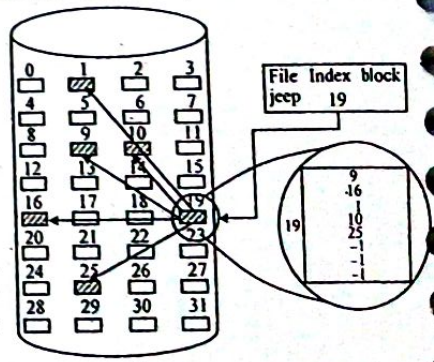
File size does not have to be specified. No external fragmentation.

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.

2nd 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

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:
10th 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 11th 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 12th pointer points to a double indirect block, which contains 128 pointers to 128 more single indirect blocks i.e.,

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 13th 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 is seek time and latency time? **[WBUT 2012(IT)]**

b) A disk has 200 track's (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 request (in order of their arrival) is for track numbers

30 70 115 130 110 80 20 25

How many times will the head change its direction for the disk scheduling policies SSTF (Shortest Seek Time First) one FCFS (First Come First Serve)?

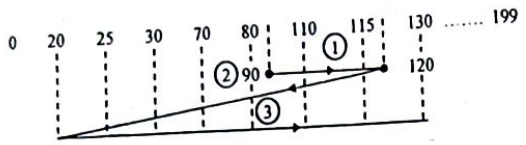
Answer:

a) Seek time:
It is defined as the time required to move the disk arm to the required track. It consists of two key components i) the initial start up time and ii) the time taken to traverse the tracks that have to be crossed once the access arm is up to speed. The linear formula for the seek time is $T_S = m*n+S$

where, T_S = Estimated seek time
m = Constant that depends on the disk drive
n = No. of tracks traversed
S = Start up time.

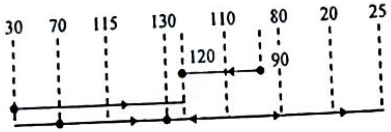
Latency time:
It is defined as the time required to reach the desired sector by the read/write head. Magnetic disks have rotational speed in the range of 5400 to 10,000 rpm. Thus, at 100,000 r.p.m. the average rotational delay will be 3 ms. However, our floppy disks typically rotates between 300 and 600 rpm. Thus, now the average delay will be between 100 and 200 ms.

b) SSTF:



The head change its direction in SSTF = 3

FCFS:



The head change its direction in
FCFS = 4

3. a) What is an *i*-node? Describe different fields in an *i*-node. [WBUT 2012(IT)]
b) Describe structure of FAT file system.

Answer:

a) 1st Part:

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.

2nd Part:

There are seven fields of the i-mode in UNIX operating system, these are, File owner identifier, File type, File access permission, File access time, No. of links to the file, table of contents and File size.

There are three types of user in the UNIX operating system such as owner, group and other. The owner has rights to access all files in UNIX operating system. The owner is otherwise called as super user.

UNIX OS has four different types of files such as ordinary, directory, special file, FIFO or PIPE.

File access permissions specifies the access rights to the three types of user to read, write and execute.

File access time gives the information about the last modification and last access time an the last modification of the i-node.

No. of links represent the number of names of file has in the UNIX directory hierarchy. Table of contents has the disk addresses of data in the file. File size is the no. of byte from the beginning of the file.

b) FAT is a simple but efficient method of disk space allocation which is used by the DOS and OS/2 operating system. A section of disk at the beginning of each partition set aside to contain the table. The table has one entry for each disk block, and is in by block number. The FAT is used much as it is a linked list. The directory entry contair the block number of the first block of the file. The table entry indexed by that number then contains the block number of the next block in the file. This chain continue until the last block, which has a special end-of-file value as the table entry. U blocks are indicated by a 0 table value. By locating a new block to a file is a simpl matter of finding the first 0-valued table entry, and replacing the previous end-o value with the address of the new block. The zero is then replaced with the end-of-fil value. FAT allocation scheme can result in a significant number of head seeks, unle FAT is cached. The disk head must move to the start of the partition to read the FAT an find the location of the block in question, then move to the location of the block itse the worst case, both moves occur for each block. Here random access is optimize because the disk head can find the location of any block by reading the information FAT.

4. What is a record in a file? For a file having multiple records what are th different indexing strategies there? Explain each strategy very briefly with rel advantages and disadvantages. [WBUT 2016(CS), 2016(IT)

Answer:

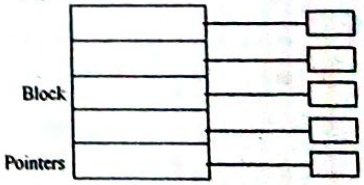
1st Part:

Records in file are made up with field, each of which contains one item of informati set of records constitutes a file. For example, a personal file contains records such a name, address, pin code, phone no, etc.

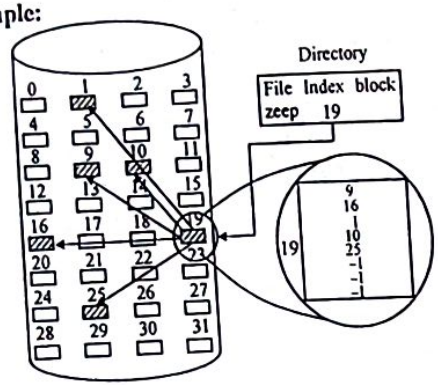
2nd Part:

File has array of pointers (index) to block.

Allocate block pointers contiguously to metadata and must set max length when created. Allocate pointers at creation, allocate blocks on demand. Maintain multiple test 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.

5. Write short notes on the following:

a) Linked file allocation technique **[WBUT 2009, 2012(CS), 2013(IT)]**

b) Free-space management technique (any *one*) of disk **[WBUT 2013(IT)]**

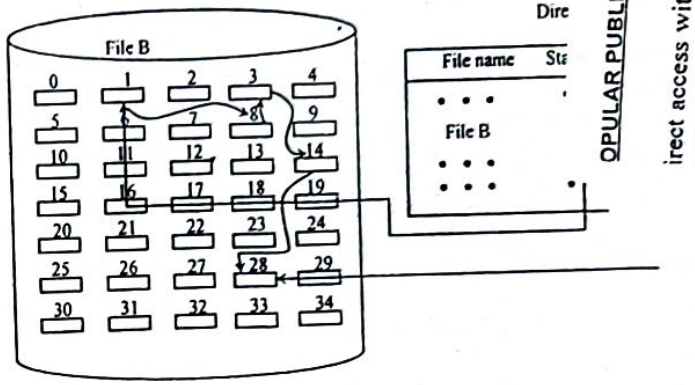
c) Indexed allocation method of disk space **[WBUT 2015(IT)]**

d) *i*-node **[WBUT 2016(CS)]**

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:

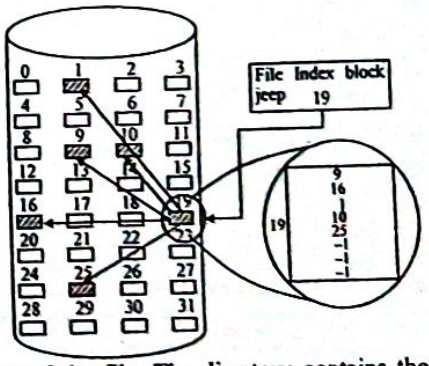
It does sequential access efficiently and is not for direct access. Each block contains a pointer, wasting space. Blocks scatter everywhere and a large number of disk seeks may be necessary.

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

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. Which of the following RAID levels implements some form of parity calculation to introduce redundancy? [WBUT 2012(CS
   a) RAID Level 2 b) RAID Level 4
   c) RAID Level 6 d) all of these

Answer: (d)

3. The time to move the disk arm to be desired cylinder in a hard disk is known [WBUT 2012(CS), 2016(CS
   a) Rotainaly latency b) Positioning time
   c) Indexed d) Hashed

Answer: (a)

4. Where does the swap space reside? [WBUT 2014(IT
   a) RAM b) DISK c) ROM d) on-chip cach

Answer: (b)

5. The smallest possible unit of disk storage is [WBUT 2014(IT), 2017(IT
   a) word b) segment c) block d) extent

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 SCA disk scheduling technique? [WBUT 2013(CS), 201

Answer:

a) 1st Part:

Seek time is defined as the time required to move the disk arm to the required track. 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 to speed.

2nd Part:

Rotational latency is defined as the time required to reach the desired sector by the 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.

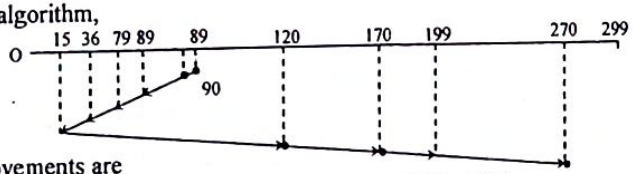
Long Answer Type Questions

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,

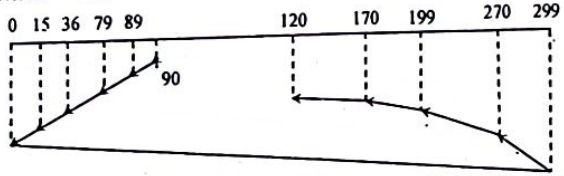
Suppose a disk drive has 300 cylinders, numbered 0 to 229. The current head position of the disk is at 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170.
Calculate the average cylinder movements for Shortest Seek Time First (SSTF) algorithm. Mention any one disadvantage 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.



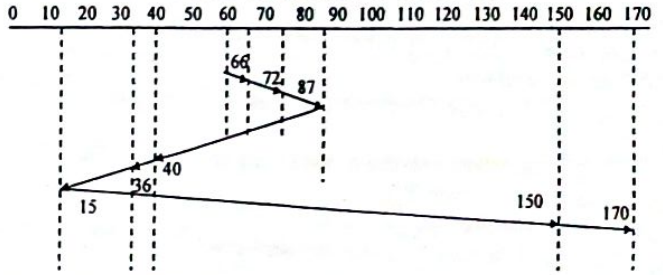
Cylinder movements = (90 – 89) + (89 – 79) + (79 – 36) + (36 – 15) + (15 – 0)
+ (0 – 299) + (299 – 270) + (270 – 199) + (199 – 170) + (170 – 120)
= 1+10+43+21+15+200+71+29+50= 568

∴ 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 a cylinder 66. Then the closet to 66 is 72, and then 87 and so on.

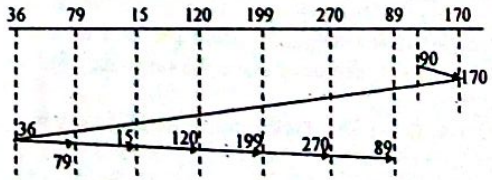


Total head movement in SSTF are:
= (66 – 60) + (72 – 66) + (87 –72) + (87 – 40) + (40 – 36)
+ (36 – 15) + (150 –15) + (170 – 150
= 6 + 6 + 15 + 37 + 4 + 21 + 135 + 20 = 244 Cylinders

∴ 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. a) What is rotational latency? **[WBUT 2013(IT)]**

Answer:

Refer to Question No. 1(a) (2^{nd} Part) of Short Answer Type Questions.

b) What are the advantages of Shortest Seek Time First and Scan disk scheduling algorithms? **[WBUT 2013(IT)]**

Answer:

Advantages of shortest-seek-time-First (SSTF):

i) It minimizes latency.
ii) Better throughput than FIFO.

Advantages of scan-disk scheduling algorithm:

i) The throughput is better than FIFO.
ii) It has been the basis of most disk scheduling strategies.
iii) No starvation problems.
iv) It eliminates the discrepancy inherent in SSTF schemes.

c) What are the disadvantages of First Come First Served disk scheduling algorithm? **[WBUT 2013(IT)]**

Answer:

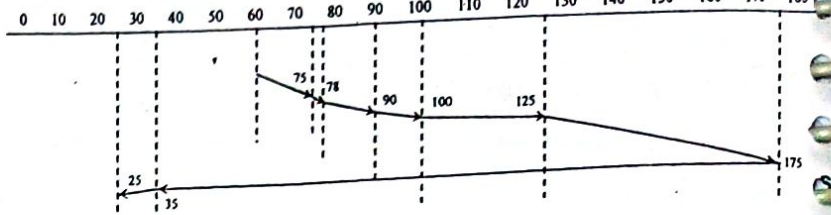
Disadvantages of FCFS:

i) It is very simple algorithm, as a result it involves a lot of random head movements and disk rotation.
ii) Throughput is not efficient.
iii) It is used in small system only where I/O efficiency is not very important.
iv) It is acceptable when the load on a disk is light. As the load grows, FCFS tends to saturate the device and the response time become longer.

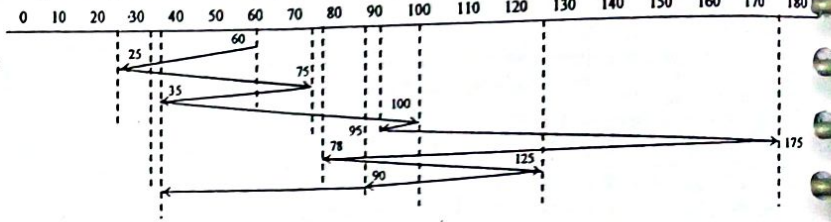
4. Draw the disk read/write head movement diagram for SSTF, SCAN, C-SCAN and FIFO, for the track requests as **[WBUT 2016(CS), 2016(IT)]**
25, 75, 35, 100, 95, 175, 78, 125, 90, 35

Answer:

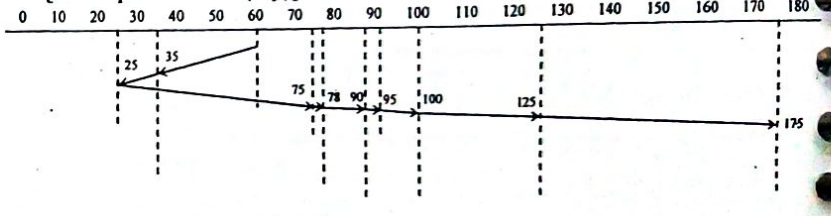
SSTF: [Initial position = 60 (say)]
25, 75, 35, 100, 95, 175, 78, 125, 90, 35



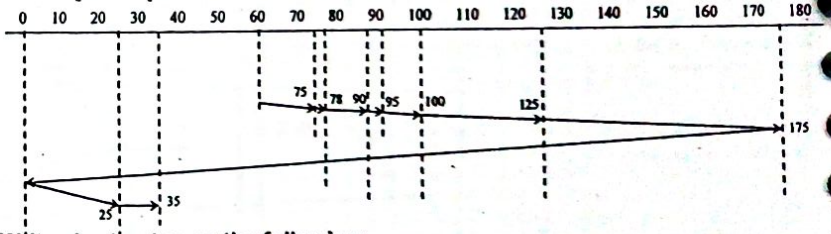
FIFO: [Initial position = 60 (say)]



SCAN [Initial pointer = 60 (say)]



C–SCAN [Initial pointer = 60 (say)]



5. 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)]**

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.

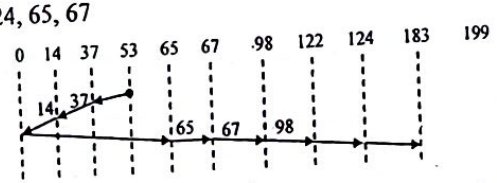
b) Scan and C-Scan algorithm:

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.

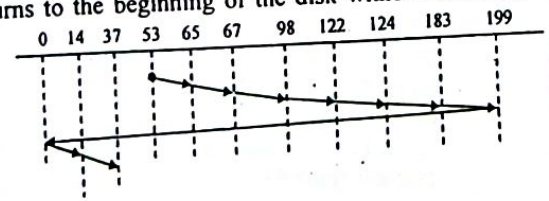
Example:

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.



PROTECTION & SECURITY

Multiple Choice Type Questions

1. Encryption is the process of **[WBUT 2013(CS)]**

a) hiding information b) authenticating information

c) both (a) & (b) d) none of these

Answer: (c)

2. Cryptography technique is used in **[WBUT 2014(IT), 2017(IT)]**

a) polling b) job scheduling c) protection d) file management

Answer: (c)

3. 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). **[WBUT 2015(IT)]**

a) $m \times n^2$ b) $m \times n$ c) $m^2 \times n$ d) none of these

Answer: (b)

4. Important objectives of computer security include **[WBUT 2016(IT)]**

a) Confidentiality b) Integrity c) Availability d) All of these

Answer: (d)

Short Answer Type Questions

1. Explain any one technique adopted by operating systems for protection of objects in the system. **[WBUT 2013(CS), 2017(IT)]**

Answer:

Protection is a mechanism that prevent accidental or intentional misuse of a system. There are three aspects to a protection mechanism like authentication, authorization and access enforcement. **Authentication** typically done with passwords, a secret piece of information used to establish identity of a user. Passwords should be relatively long and obscure.

In access enforcement, some part of the system must be responsible for enforcing access controls and protecting authentication information. Security kernel is an inner layer of the operating system that enforces security.

2. Categorize different types of attackers in the context of security.[WBUT 2016(IT)]

Answer:

Different types of security attacks are:

- DOS – Denial of service
- Trojan Horse – Comes with other software
- Virus – Reproduces itself by attaching to other executable files.

- Worms – Self producing program
- Spyware

A **denial-of-service (DOS)** attack is a special form of cyber attack that focuses on the interruption of a network service.
In the list of computer attacks, **Trojan horses** rank right after the viruses. They often disguise themselves in a piece of software, in screen server, or in a game which appears to work normally.
Most famous computer attacks are viruses, which have been around for longest time. They install themselves onto the computers and spread to the other files on the system.
Worms can be called the cousins of viruses. The difference between viruses and worms is that worms infect the system without any kind of assistance from the user.
Spyware is a malware which is designed to spy on the victim's computer. Certain activity will be spied by the spy ware.

Long Answer Type Questions

1. Write short note on Security and Protection in Operation System. [WBUT 2012(IT)]

Security and protection in operating system must involve:
(i) Confidentiality or privacy i.e. users must be authorized user so that primary can be maintained. Interruption occurs when an unauthorized users are going to access computer systems.
(ii) Integrity – computer system resources can be modified only by authorized parties
(iii) Availability – The requirement that a computer system resources can be modified only by authorized parties.
(iv) Authenticity – Identification of the user.

MISCELLANEOUS

Multiple Choice Type Questions

1. The segment of code in which the process may change common varia update tables, write into files is known as [WBUT 20..4(
a) Program b) critical section
c) non-critical section d) Synchronizing
Answer: (a)

2. In a resident OS computer, which of the following systems must reside ir main memory under all situations? [WBUT 2..
a) Assembler b) Loader c) Linker d) Compile
Answer: (d)

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

Long Answer Type Questions

1. Write a program using 'fork' to demonstrate the mother-child relationshi processes. [WBUT 201..(
Answer:
In a c-program, fork () can be used to create a new process, known as child proc... child is initially a copy of the parent,. but can be used to run a different branch o program or even execute a completely deferent program. After forking, child an.. process run in parallel. Any variables local to the parent process will have been co for the child process, so up dating a variable in one process will not affect th... 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
```


QUESTION 2013 [6th-CS]

GROUP – A

(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following:

i) Translation look aside Buffer is a kind of

a) interrupt ✓b) cache c) virtual memory d) i/o device

ii) Address generated by CPU is generally referred to as

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

iii) Paging suffers from

✓a) internal fragmentation b) external fragmentation

c) both (a) & (b) d) none of these

iv) Which of the following algorithm generally suffers from Belady's anomaly

a) optimal ✓b) FIFO c) LRU d) all of these

v) Concurrent processes

a) overlap in space b) do not overlap in time

✓c) overlap in time d) both (a) and (c)

vi) Thread is referred to as

✓a) lightweight process b) process c) program d) set of proces

vii) Swap space generally resides on

a) main memory b) files c) programs ✓d) disk

viii) Disk I/O generally done in terms of

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

ix) The first block of a file system is

a) superblock b) inode blocks c) data block ✓d) boot block

x) Encryption is the process of

a) hiding information b) authenticating information

✓c) both (a) & (b) d) none of these

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: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 6.

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: CPU SCHEDULING, Short Answer Type Question No. 2.

4. 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.

See Topic: PROCESS MANAGEMENT, 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: PROCESS MANAGEMENT, Short Answer Type Question No. 7.

b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7.

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 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. 6.

8. a) 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 questions using banker's algorithm:

i) What is the content of need Matrix?

ii) Is the system in safe state?

iii) If the request for 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 Blocking vs 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 MANAGEMENT, Short Answer Type Question No. 1.

b) & c) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 8.

d) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 8.

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, 75, 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 a 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 are its overheads?

e) What is the difference between starvation and deadlock?

) See Topic: PROTECTION AND SECURITY, Short Answer Type Question No. 1.
) & c) See Topic: INPUT/OUTPUT MANAGEMENT, Long Answer Type Question No. 1.
) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 11.
) See Topic: DEADLOCK, Short Answer Type Question No. 1.

QUESTION 2013 [5th-IT]

GROUP – A
(Multiple Choice Type Questions)

I. Choose the correct alternatives for any ten of the following:

) A thread is referred to as
a) process ✓b) light weight process c) task d) program

i) CPU generates
✓a) logical address b) physical address c) both (a) & (b) d) none of these

iii) Which page replacement algorithm suffers from Belady's anamoly?
a) LRU b) optimal ✓c) FIFO d) all of these

iv) Which of the following loaders is executed when a system in first turned on or restarted?
a) compile and go loader ✓b) boot strap loader
c) relative loader d) absolute loader and relative loader

v) IPC stands for
a) Internal Program Controller ✓b) Inter Process Communication
c) Internal Process Controller d) none of these

vi) Suppose that a process is in BLOCKED state waiting for some I/O service. When the service is completed, it goes to the
a) RUNNING b) SUSPENDED ✓c) READY d) TERMINATED

vii) Which is *not* a valid process state?
a) new b) ready c) run ✓d) load

viii) Which of the following memory allocation schemes suffers from external fragmentation?
✓a) segmentation b) swapping c) paging d) none of these

ix) The default remedy of starvation is
✓a) ageing b) critical section c) mutual exclusion d) all of these

x) Which of the following is crucial time while accessing data on the disk?
✓a) seek time b) rotational time
c) transmission time d) waiting time

xi) 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

xii) Time required of read-write head to move to desired cylinder is
a) transfer time ✓b) seek time c) rotational latency d) none of these

GROUP – B
(Short Answer Type Questions)

2. Write down all the necessary conditions of Deadlock.
See Topic: DEADLOCK, Long Answer Type Question No. 1(a).

3. a) What is context switch?
b) What is translation look aside buffer? Why is it used?
a) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 3(a) (1st Part).
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 4(2nd Part).

4. Consider the following set of processes with their respective CPU execution times. Assume that they have arrived in the order shown. Draw the Gantt chart for Round Robin scheduling with time quantum = 5 time units. Calculate turnaround time of each process.

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

See Topic: CPU SCHEDULING, Short Answer Type Question No. 3.

5. a) What do you understand by Race condition?
b) What are the conditions for solution to Mutual Exclusion problem?
See Topic: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 2.

6. a) What is a multiuser, multiprogram operating system?
b) What is virtual memory concept? How is it supported and implemented?
a) See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Short Answer Type Question No. 7.
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 12.

GROUP – C
(Long Answer Type Questions)

7. a) Discuss the relative advantages and disadvantages of segmentation and paging.
b) Consider the following page reference string:
1, 2, 3, 4, 21, 56, 21, 2, 3, 7, 6, 3, 2, 1, 2, 36
How many page faults would occur for the following replacement algorithms? Assume 4 frames are available.
i) LRU replacement ii) FIFO replacement.
c) What is thrashing?

a) & b) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 9.
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7.

8. a) What is rotational latency?
b) What are the advantages of Shortest Seek Time First and Scan disk scheduling algorithms?
c) What are the disadvantages of First Come First Served disk scheduling algorithm?
d) Suppose a disk drive has 300 cylinders numbered 0-299. The current position of the arm is at 90. The queue of the pending request in FIFO order is 36, 79, 15, 120, 199, 270, 89, 170.
Calculate the average movements for the following algorithm:
C-SCAN (arm is moving towards 299).

a), b) & c) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 3.
d) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 1.

9. a) Consider the following snapshot of a system where P0, ... , P4 are the processes and A, B, C, D are resource types.

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

i) What is the content of need matrix?
ii) Is the system in safe state?
iii) If a request from process P1 arrives for 0, 4, 2, 0, can the request be granted immediately?
b) What is the difference between Deadlock avoidance and Deadlock prevention techniques?
c) Discuss any one Deadlock prevention technique with its disadvantages and advantages.

a) See Topic: DEADLOCK, Long Answer Type Question No. 1(b).
b) & c) See Topic: DEADLOCK, Long Answer Type Question No. 3.

10. a) What are the advantages and disadvantages of the Best Fit algorithm?
b) What are the advantages of the First fit algorithm?
c) What are the differences between two types of fragmentation in memory?

d) Given memory partitions of 100 kB, 500 kB, 300 kB and 600 kB (in order):
i) How would each of the First Fit, Best Fit, Worst Fit algorithms (consider separately) place processes of 212 kB, 417 kB, 112 kB and 426 kB (in order)?
ii) Which algorithm makes the most efficient use of memory?

a), b) & c) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 10.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 2.

11. a) What is dynamic loading and what is dynamic linking? How are they related?
b) What is demand paging? What are its advantages and disadvantages?
c) What is Semaphore? What are the operations on it?

a) See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Short Answer Type Question No. 5.
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 13.
c) See Topic: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 7.

12. Write short notes on any *three* of the following:
a) DMA
b) Linked file allocation
c) Multilevel feedback queue scheduling
d) Time sharing OS or Batch processing OS
e) Free-space management technique (any *one*) of disk
f) Blocking and Non-blocking I/O

a) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(a).
b) See Topic: FILE SYSTEM, Long Answer Type Question No. 5(a).
c) See Topic: CPU SCHEDULING, Long Answer Type Question No. 4(b).
d) See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Long Answer Type Question No. 1(c).
e) See Topic: FILE SYSTEM, Long Answer Type Question No. 5(b).
f) See Topic: INPUT / OUTPUT MANAGEMENT, Long Answer Type Question No. 2.

QUESTION 2014 [6th-CS]

GROUP – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following questions:
i) Which of the following is false?
i) Segmentation suffers from external fragmentation
ii) Paging suffers from internal fragmentation
✓iii) Virtual memory is used only in multiuser system
iv) Segmented memory can be paged

ii) Thrashing
i) Reduces page I/O
ii) Decreases the degree of multiprogramming
✓iii) Implies excessive I/O
iv) Improves the system performance

iii) RAID configuration disks are used to provide
✓i) Fault tolerance ii) Nearest cylinder next
iii) High data intensity iv) Low data intensity

iv) Which of the following page replacement algorithms suffer from Balady's anomaly?
i) OPT ii) LRU ✓iii) FIFO iv) Both (i) & (ii)

v) Virtual memory concept is supported by
✓i) Demand paging ii) Simple segmentation
iii) Any dynamic memory allocation iv) Paged segmentation

vi) The short-term scheduler is responsible for:
i) Selecting the shortest jobs to enter the system
ii) Discarding job groups that should be removed from the system
✓iii) Selecting which process should be allocated to the CPU next
iv) Swapping jobs out of memory
v) None of these

vii) The average wait time for five processes PI-P5 with burst of 5, 19, 2, 16 and 7 milliseconds respectively, using SJF is:
i) 5 milliseconds ✓ii) 10.6 milliseconds
iii) 28.25 milliseconds iv) 9.8 milliseconds v) none of these

viii) Suppose that the operating system is running a non-preemptive scheduler and that process p is currently running. A context switch can occur:
✓i) When p terminates process or blocks
ii) When another process unblocks
iii) When another process enters
iv) When the time quantum is exhausted
v) When the priority of some other process exceeds the priority of p

ix) 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:
i) data consistency ✓ii) race condition iii) aging iv) Starvation

x) The segment of code in which the process may change common variables, update tables, write into files is known as
✓i) Program ii) critical section iii) non-critical section iv) Synchronizing

GROUP – B
(Short Answer Type Questions)

2. What is Direct Memory Access (DMA)? How is it performed? What are its benefits?
See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 8(a).

3. What is the cause of thrashing? How does the system detect trashing? Once it detects thrashing what can the system do to eliminate this problem?
Given memory partitions of 100 KB, 500 KB, 300 KB, and 600 KB (in order)
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 14.

4. How would each of the First Fit, Best Fit, Worst Fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order). Which algorithm makes the most efficient use of memory?
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 2.

5. Give details of how paging is implemented in hardware. What is a TLB and how is it implemented?
See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 4.

6. What is context switching? Why it considered to be an overhead? What are the differences between process and thread?
See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 3.

GROUP – C
(Long Answer Type Questions)

7. What resources are used when a thread created? How do they differ from those when a process is created? What is virtual memory? What is fragmentation? Explain different types of fragmentation. Which one may occur in paging system?
1st & 2nd part: See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 9.
3rd Part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 12(1st Part).
4th Part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 1(1st Part).
5th & 6th Part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 15.

8. a) What is swapping? What is its purpose?
b) Consider the following sequence of memory references generated by a single process 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 algorithms 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
c) Explain with diagram different File Allocation methods.
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?
a) & b) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 1.
c) See Topic: FILE SYSTEM, Short Answer Type Question No. 2.

9. a) Write down the four necessary conditions of deadlock.
b) What is banker's safety algorithm?
c)

Process	Current Allocation				Maximum Allocation				Available			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P1	0	0	1	2	0	0	1	2	2	1	0	0
P2	2	0	0	0	2	7	5	0				
P3	2	3	5	4	4	3	5	6				
P4	0	3	3	2	0	6	5	2				
P5												

Is this system currently in a safe state justifying your answer?
If a request from P3 arrives for (0, 1, 0, 0) can that request be safely granted immediately.
Differentiate between deadlock and starvation.

a) See Topic: DEADLOCK, Long Answer Type Question No. 1(a).
b) See Topic: DEADLOCK, Short Answer Type Question No. 2(2nd Part).
c) See Topic: DEADLOCK, Long Answer Type Question No. 2.

10. a) What is boot strapping? Distinguish between multiprogramming and multithreading OS? State the function of batch processing system.
b) "All unsafe states may not lead to deadlock" why or why not? Define the critical section and identify the requirements to be satisfied to solve the critical section problem.
c) Discuss dinning philosopher problem with the solution.

a) See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Short Answer Type Question No. 8.
b) 1st Part: See Topic: DEADLOCK, Short Answer Type Question No. 3.
2nd Part: See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 5(a) & (b).
c) See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 3(b).

11. a) What is semaphore? How is it used to overcome critical section problem?
b) How mutual exclusion, hold & wait and circular wait are different from each other? Explain with an example. Explain different types of thread.
c) How does the monitor give the solution for synchronization problem? What is binary semaphore?

a) See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 5(c).
b) 1st Part: See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 6(a).
2nd Part: See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 10.
c) See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 6(b).

QUESTION 2014 [5th-IT]

GROUP – A
(Multiple Choice Type Questions)

1. Answer all the questions:
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 the swap space reside?
a) RAM ✓b) DISK c) ROM d) on-chip cac

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 manageme

vi) TLB is a kind of
a) virtual memory b) interrupt ✓c) cache d) main memor

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 proce

ix) Context switching is
a) part of the spooling b) part of pooling
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. Mention the basic principle of Round Robin scheduling. Specify the impact of time quantum o its performance.

See Topic: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 1.

3. a) Describe process control block (PCB) in details.
b) What is a Process? Draw and explain the process state diagram?

a) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 4.
b) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 11.

4. a) What do you mean by preemptive and non-preemptive scheduling?
b) What are the different scheduling criteria?
See Topic: CPU SCHEDULING, Short Answer Type Question No. 4.

5. a) What is semaphore?
b) Write Peterson algorithm for two process critical section problem?
See Topic: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 8.

6. a) What is dispatch latency?
b) Explain RPC.
a) See Topic: CPU SCHEDULING, Short Answer Type Question No. 5.
b) See Topic: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 9.

GROUP – C
(Long Answer Type Questions)

7. What is paging? Explain the hardware for paging? How does paging differ from segmentation with respect to hardware? What is Virtual memory? How can segmentation be done with the concept of virtual memory? What is External fragmentation?
See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 11.

8. a) What is Deadlock? List four necessary conditions for the occurrence of deadlock.
b) Does presence of cycle in a resource allocation graph necessarily creates deadlock.
c) Consider the following snapshot of a system.

Processes	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

Answer the following questions using Banker's Algorithm:
(i) What is the Need Matrix?
(ii) Is the system in safe state? If yes what is the safe sequence?
(iii) If a request can be process P1 arrives for (0, 4, 2, 0) can the request be granted immediately.
a) & b) See Topic: DEADLOCK, Long Answer Type Question No. 4.
c) See Topic: DEADLOCK, Long Answer Type Question No. 1(b).

9.

Processes	Burst Time	Arrival Time
P1	10	0
P2	5	1
P3	2	3
P4	7	5
P5	9	7

Deduce the Average Turn Around Time and Average Waiting Time using Shortest Remaining Time First and Round Robin Scheduling.
See Topic: CPU SCHEDULING, Long Answer Type Question No. 2.

10. 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.
c) What is thrashing?
d) Explain Belady's anamoly.
a) & b) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 5.
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 3.

11. a) Briefly explain different free space management techniques.
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.
c) Explain compaction.
a) & b) See Topic: FILE SYSTEM, Long Answer Type Question No. 1.
c) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 8.

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: MEMORY MANAGEMENT, Short Answer Type Question No. 6.
b) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 1.
c) See Topic: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 1.

3. Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current head position of th 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. Mentio 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 t similar operation can act as semaphore or not.

See Topic: DEADLOCK, Short Answer Type Question No. 4.

5. How would each of the First Fit, Best Fit and Worst Fit algorithms place processes of 212K 417KB, 112KB and 426KB (in order). Which algorithm makes the most efficient use of memory?

See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 2.

6. What is deadlock? Describe the necessary and sufficient conditions for the occurrence deadlock.

See Topic: DEADLOCK, Short Answer Type Question No. 6.

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 Matri $(r_1 = 2, r_2 = 1, r_3 = 0, r_4 = 0)$.

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 a disadvantages of Round Robin scheduling.

a) See Topic: DEADLOCK, Long Answer Type Question No. 2.
b) See Topic: CPU 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) & d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 12(a) & (b).
b) & c) 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: PROCESS MANAGEMENT, Short Answer Type Question No. 3.
c) See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No: 2.
d) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 4.

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: PROCESS MANAGEMENT, Long Answer Type Question No. 1(d).
See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 9(a).
c) See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Long Answer Type Question No. 1(d).
d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(e).
e) See Topic: CPU SCHEDULING, Long Answer Type Question No. 4(a).

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: CPU SCHEDULING, Short Answer Type Question No. 7.

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(b).

b) See Topic: PROCESS MANAGEMENT, Long Answer Type Question No. 2(b).

4. What is Process Control Block? Discuss the structure of Process Control Block.

See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 4.

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 AND SYSTEM STRUCTURE, Short Answer Type Question No. 2.

b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 5.

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, Short Answer Type Question No. 8.

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. 16.

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 part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 2.

2nd part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 6.

3rd part: See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(d).

4th part: See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(a).

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. 17.

3rd & 4th part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7.

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: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 7.

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: CPU SCHEDULING, Long Answer Type Question No. 4(b).

b) See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Long Answer Type Question No. 1(c).

c) See Topic: FILE SYSTEM, Long Answer Type Question No. 5(c).

d) See Topic: PROCESS MANAGEMENT, Long Answer Type Question No. 2(e).

e) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(f).

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 AND SYSTEM STRUCTURE, Short Answer Type Question No. 9.

3. What are the difference between a trap and an interrupt? What is the use of each function?

See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Short Answer Type Question No. 10.

4. What is the purpose of the command interpreter? Why is it usually separate from the kernel?

See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Short Answer Type Question No. 11.

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: CPU SCHEDULING, Short Answer Type Question No. 8.

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. 8.

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: CPU SCHEDULING, Long Answer Type Question No. 3.

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: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 5.
b) See Topic: DEADLOCK, Short Answer Type Question No. 6.
c) See Topic: DEADLOCK, Long Answer Type Question No. 5.

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: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 10.
b), c) & d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 13.

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 SYSTEM, Long Answer Type Question No. 4.
b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 4.

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: PROCESS MANAGEMENT, Long Answer Type Question No. 1(f).
b) See Topic: PROCESS MANAGEMENT, Long Answer Type Question No. 1(g).
c) See Topic: FILE SYSTEM, Long Answer Type Question No. 5(d).
d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(c).
e) See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 9(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 the

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: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 4(a).

3. What is the main objective of Multiprogramming? Draw and describe process state transitions.

See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 12.

4. Categorize different types of attackers in the context of security.

See Topic: PROTECTION & SECURITY, Short Answer Type Question No. 2.

5. What is the problem of fragmentation and how can it be solved?

See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 1.

6. Why are page sizes always powers of 2? What is the difference between logical and physical addresses?

1st Part: See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 6.

2nd Part: See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Short Answer Type Question No. 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: CPU SCHEDULING, Long Answer Type Question No. 3.

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: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 5.

b) See Topic: DEADLOCK, Short Answer Type Question No. 6.

c) See Topic: DEADLOCK, Long Answer Type Question No. 5.

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: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 10.

b), c) & d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 13.

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 SYSTEM, Long Answer Type Question No. 4.

b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 4.

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: PROCESS MANAGEMENT, Long Answer Type Question No. 2(c).

b) See Topic: DISK MANAGEMENT, Long Answer Type Question No. 5(b).

c) See Topic: CPU SCHEDULING, Long Answer Type Question No. 4(c).

d) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 14(g).

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: PROCESS MANAGEMENT, Short Answer Type Question No. 13.

3. a) What do you mean by real time system?
b) Differentiate between soft and hard real time system.
See Topic: INTRODUCTION AND SYSTEM STRUCTURE, Short Answer Type Question No. 12.

4. a) What is Medium Term scheduler?
b) Describe the functions of short-term and long-term scheduler.
See Topic: CPU SCHEDULING, Short Answer Type Question No. 9.

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. 7.

6. a) Explain Race condition in context of process synchronization.
b) What are semaphore and mutex?
See Topic: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 11.

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: PROCESS MANAGEMENT, Long Answer Type Question No. 1.
b) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 3(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: PROCESS MANAGEMENT, Short Answer Type Question No. 1.
b) See Topic: CPU SCHEDULING, Short Answer Type Question No. 4(a).
c) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 14.
d) See Topic: CPU SCHEDULING, Short Answer Type Question No. 10.

9. Write a program using 'signal' to demonstrate a race condition.
See Topic: PROCESS SYNCHRONIZATION, Long Answer Type Question No. 8.

10. Write a program using 'fork' to demonstrate the mother-child relationship of processes.
See Topic: MISCELLANEOUS, Long Answer Type Question No. 1.

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) FIFIO 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. 7.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 3.

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: PROCESS SYNCHRONIZATION, Short Answer Type Question No. 6.

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. Calcula e individual turnaround time and average waiting time.
See Topic: CPU SCHEDULING, Short Answer Type Question No. 2.

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: PROCESS MANAGEMENT, 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: PROCESS MANAGEMENT, Short Answer Type Question No. 7.
b) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 7.

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 the 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. 6.

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 Blocking 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 hav 2 pages each and 3 processes have 1 page each. The system uses inverted page table. Find th 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 MANAGEMENT, Short Answer Type Question No. 1.
b) & c) See Topic: MEMORY MANAGEMENT, Long Answer Type Question No. 8.
d) See Topic: PROCESS MANAGEMENT, Short Answer Type Question No. 8.

10. a) Explain the working of Shortest Seek Time First (SSTF) disk scheduling policy. What are i advantages and disadvantages?
b) Suppose a disk drive has 300 cylinders, numbered 0 to 299. The current position of the disk ar is 90. The queue of pending requests, in FIFO order is 36, 79, 15, 120, 199, 270, 89, 17 Calculate the average movements for the following algorithms:
FCFS
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) See Topic: PROTECTION AND SECURITY, Short Answer Type Question No. 1.
b) & c) See Topic: INPUT/OUTPUT MANAGEMENT, Long Answer Type Question No. 1.
d) See Topic: MEMORY MANAGEMENT, Short Answer Type Question No. 11.
e) See Topic: DEADLOCK, Short Answer Type Question No. 1.