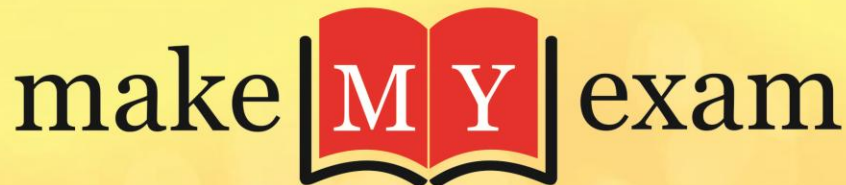


Operating System

IBPS SO-IT Officer Exam 2017



India's trusted Educational Blog

Subscribe to our
You Tube Channel



IBT make MY exam
Topic: Mathematics
UNIT DIGITS
AMIT PANDEY
How to find unit digit (very useful trick) Number System : By Amit...

IBT make MY exam
Topic: Mathematics
% Convert in P by Q form
AMIT PANDEY
How to convert decimal expansion in rational form P/Q BY Amit

IBT make MY exam
TOPIC Vice President of India
• Qualification
• Election and Power
VIPAN GOYAL
Vice President of India (Qualification, Election & Powers) by Vipin...

IBT make MY exam
Parts of Speech
PLURAL
By SAJJID AMAN KHAN
Parts of Speech : PLURAL By Sajid Aman Khan (General English).....

make MY exam
REMAINDER THEOREM - I
AMIT PANDEY
Number System REMAINDER THEOREM(Part-1) By Amit Pandey

IBT make MY exam
Parts of Speech
NOUN
By SAJJID AMAN KHAN
Parts of Speech : NOUN By Sajid Aman Khan (General English)....

IBT make MY exam
10 - 16 April 2017
CURRENT AFFAIRS BULLETIN
Neelima Singh
Current Affairs Bulletin of the week 10 April To 16 April 2017

IBT make MY exam
TOPIC President of India Part-I
• Qualification
• Election
VIPAN GOYAL
President of India (Qualification & Election) By Vipin Goyal

Are you ready for SBI PO EXAM ?
SBI PO- PRE EXAM TRAINING
DAY-1 (2 HOURS)
REASONING ABILITY
Online 8 days
SBI PO - 8 Days online PRE EXAM TRAINING

SBI PO- PRE EXAM TRAINING
DAY-2 (2 HOURS)
REASONING ABILITY
Online 8 days
SBI PO online PET : Day 2 Reasoning Ability

SBI PO- PRE EXAM TRAINING
DAY-3 (2 HOURS)
QUANTITATIVE APTITUDE
Online 8 days
SBI PO Online PET: Day 3 Maths - Percentage, Profit & Loss & Alli...

SBI PO- PRE EXAM TRAINING
DAY-7
GENERAL ENGLISH
Reading Comprehension
Sentence Rearrangement
Cloze Test
Online 8 Days
SBI PO online PET : Day 7 GENERAL ENGLISH - CLOZE TEST , S. REAR...

Video lectures for all subjects

Current Affairs bulletin

Exam Notification & Analysis

Pre Exam Training

G.K Sessions

Debates & discussions

Success Stories

Live Interview Sessions

Group discussions

Operating System:-

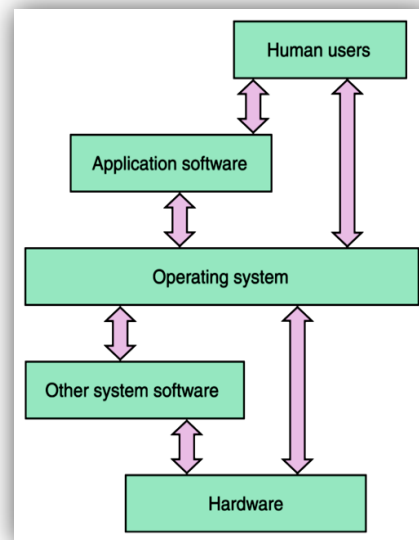
An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Following are some of important functions of an operating System.

1. Memory Management
2. Processor Management
3. Device Management
4. File Management
5. Security
6. Control over system performance
7. Job accounting
8. Error detecting aids
9. Coordination between other software and users

Following are some of the important activities that Operating System does-

- ❖ **Security** - By means of password and similar other techniques, preventing unauthorized access to programs and data.
- ❖ **Control over system performance** -- Recording delays between request for a service and response from the system.
- ❖ **Job accounting** -- Keeping track of time and resources used by various jobs and users.
- ❖ **Error detecting aids** -- Production of dumps, traces, error messages and other debugging and error detecting aids.
- ❖ **Coordination between other software and users** -- Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.



Types of operating System:-

1. Real Time operating System:-

Real time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. Real time processing is always on line whereas on line system need not be real time. The time taken by the system to respond to an input and display of required updated information is termed as response time. So in this method response time is very less as compared to the online processing. Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. Real-time operating system has well-defined, fixed time constraints otherwise system will fail. For example Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, and home-appliance controllers, Air traffic control system etc. There are two types of real-time operating systems.

- a) **Hard real-time systems:-** Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems secondary storage is limited or missing with data stored in ROM. In these systems virtual memory is almost never found.
 - b) **Soft real-time systems:-** Soft real time systems are less restrictive. Critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, Multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers etc.
- ### 2. Distributed operating System:-

Distributed systems use multiple central processors to serve multiple real time application and multiple users. Data processing jobs are distributed among the processors accordingly to which one can perform each job most efficiently. The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These

are referred as loosely coupled systems or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers and so on.

3. Time-sharing operating systems:-

Time sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing. The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, objective is to maximize processor use, whereas in Time-Sharing Systems objective is to minimize response time. Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response. Operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are following:-

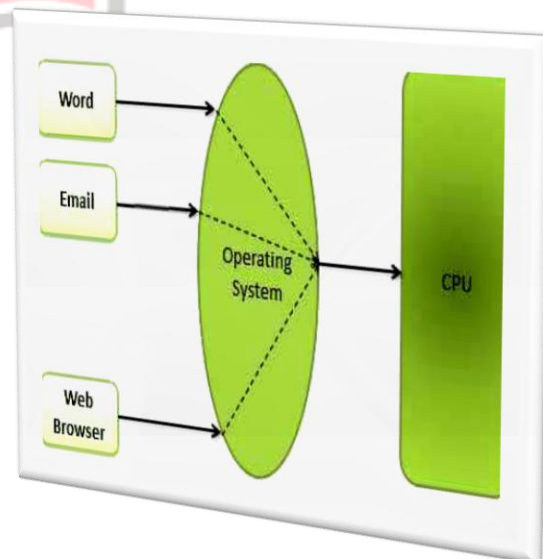
- ❖ Provide advantage of quick response.
- ❖ Avoids duplication of software.
- ❖ Reduces CPU idle time.
- ❖ Disadvantages of Timesharing operating systems are following.
- ❖ Problem of reliability.
- ❖ Question of security and integrity of user programs and data.
- ❖ Problem of data communication.

4. Batch operating system:-

The users of batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. Thus, the programmers left their programs with the operator. The operator then sorts programs into batches with similar requirements.

5. Multitasking :-

Multitasking refers to term where multiple jobs are executed by the CPU simultaneously by switching between them. Switches occur so frequently that the users may interact with each program while it is running. Operating system does the following activities related to multitasking. The user gives instructions to the operating system or to a program directly, and receives an immediate response. Operating System handles multitasking in the way that it can handle multiple operations / executes multiple programs at a time. Multitasking Operating Systems are also known as Time-sharing systems. These Operating Systems were developed to provide interactive use of a computer system at a reasonable cost. A time-shared operating system uses concept of CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared CPU. Each user has at least one separate program in memory.



6. Multiprogramming:-

When two or more programs are residing in memory at the same time, then sharing the processor is referred to the multiprogramming. Multiprogramming assumes a single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. Following figure shows the memory layout for a multiprogramming system. Operating system does the following activities related to multiprogramming.

- ❖ The operating system keeps several jobs in memory at a time.
- ❖ This set of jobs is a subset of the jobs kept in the job pool.
- ❖ The operating system picks and begins to execute one of the job in the memory.
- ❖ Multiprogramming operating system monitors the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle unless there are no jobs

Advantages

- ❖ High and efficient CPU utilization.
- ❖ User feels that many programs are allotted CPU almost simultaneously.

Disadvantages

- ❖ CPU scheduling is required.
- ❖ To accommodate many jobs in memory, memory management is required.

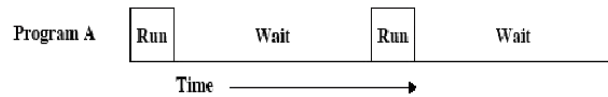
Spooling :-

Spooling is an acronym for simultaneous peripheral operations on line. Spooling refers to putting data of various I/O jobs in a buffer. This buffer is a special area in memory or hard disk which is accessible to I/O devices. Operating system does the following activates related to distributed environment. OS handles I/O device data spooling as devices have different data access rates. OS maintains the spooling buffer which provides a waiting station where data can rest while the slower device catches up. OS maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashion. It becomes possible to have the computer read data from a tape, write data to disk and to write out to a tape printer while it is doing its computing task.

Advantages:-

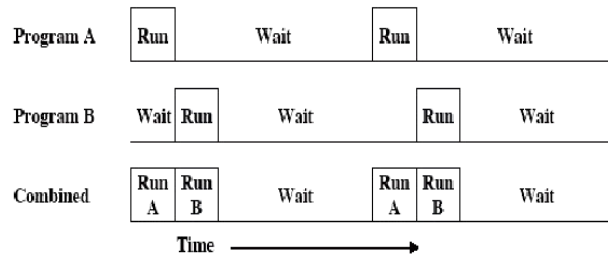
The spooling operation uses a disk as a very large buffer. Spooling is capable of overlapping I/O operation for one job with processor operations for another job.

• Uniprogramming



Processor must wait for I/O instruction to complete before preceding

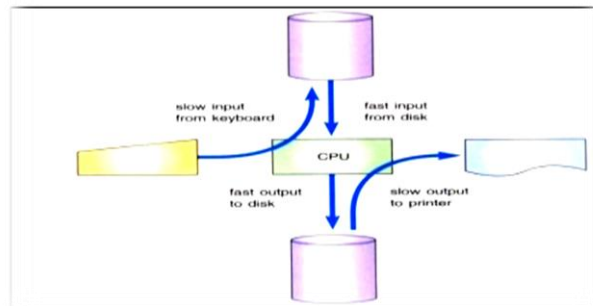
• Multiprogramming with two programs



When one job needs to wait for I/O, the processor can switch to the other job

SPOOLING

- A larger buffer from hard disk
- Buffer store the data through I/O operation
 - Because I/O operation is slow and CPU operation is fast



Process-:

A process is a program in execution. The execution of a process must progress in a sequential fashion. Definition of process is following. A process is defined as an entity which represents the basic unit of work to be implemented in the system.

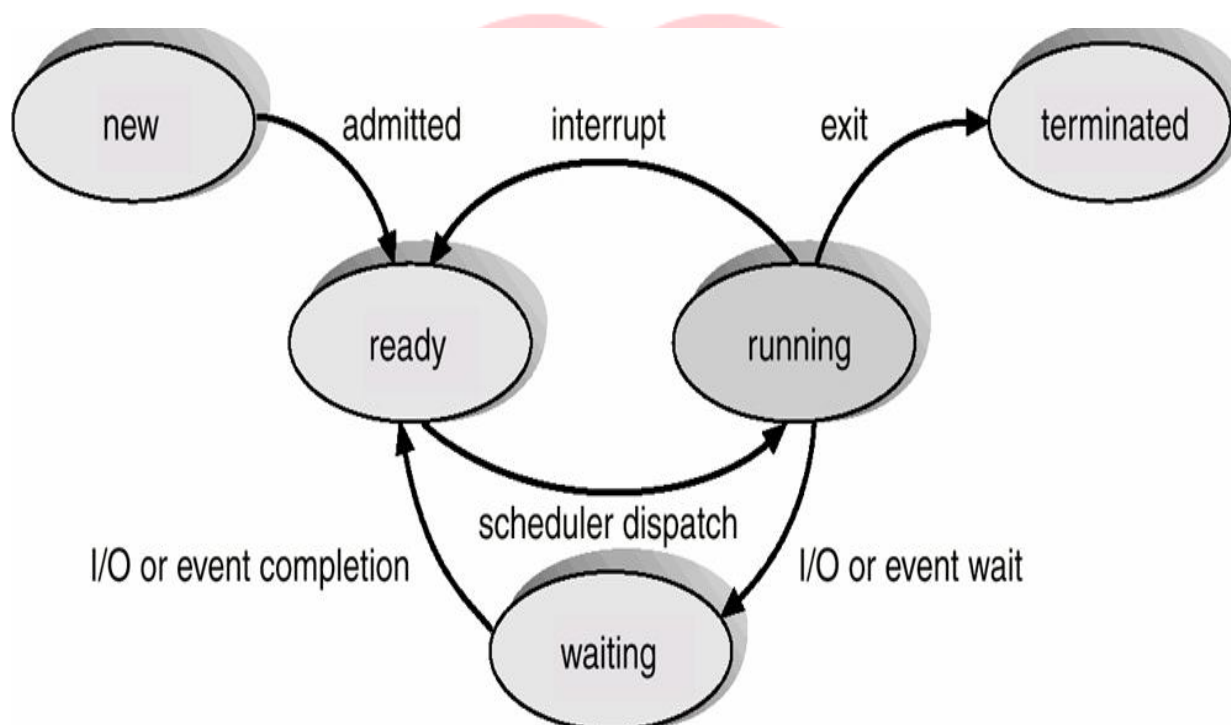
Program-:

A program by itself is not a process. It is a static entity made up of program statement while process is a dynamic entity. Program contains the instructions to be executed by processor. A program takes a space at single place in main memory and continues to stay there. A program does not perform any action by itself.

Process States-:

As a process executes, it changes state. The state of a process is defined as the current activity of the process. Process can have one of the following five states at a time.

S.N.	State & Description
1	New: The process is being created.
2	Ready: The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run.
3	Running: Process instructions are being executed (i.e. The process that is currently being executed).
4	Waiting: The process is waiting for some event to occur (such as the completion of an I/O operation).
5	Terminated: The process has finished execution.



Process Control Block (PCB) -: Each process is represented in the operating system by a process control block (PCB) also called a task control block. PCB is the data structure used by the operating system. Operating system groups all information that needs about particular process. PCB contains many pieces of information associated with a specific process which are described below.

S.N.	Information & Description
1	Pointer: Pointer points to another process control block. Pointer is used for maintaining the scheduling list.
2	Process State : Process state may be new, ready, running, waiting and so on.
3	Program Counter: Program Counter indicates the address of the next instruction to be executed for this process.
4	CPU registers: CPU registers include general purpose register, stack pointers, index registers and accumulators' etc. number of register and type of register totally depends upon the computer architecture.
5	Memory management information: This information may include the value of base and limit registers, the page tables, or the segment tables depending on the memory system used by the operating system. This information is useful for deallocating the memory when the process terminates.
6	Accounting information: This information includes the amount of CPU and real time used, time limits, job or process numbers, account numbers etc.

Process control block includes CPU scheduling, I/O resource management, file management information etc. The PCB serves as the repository for any information which can vary from process to process. Loader/linker sets flags and registers when a process is created. If that process get suspended, the contents of the registers are saved on a stack and the pointer to the particular stack frame is stored in the PCB. By this technique, the hardware state can be restored so that the process can be scheduled to run again.

Schedulers-:

Schedulers are special system software's which handles process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run. Schedulers are of three types-

- ❖ Long Term Scheduler
- ❖ Short Term Scheduler
- ❖ Medium Term Scheduler

Long Term Scheduler :-

It is also called job scheduler. Long term scheduler determines which programs are admitted to the system for processing. Job scheduler selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling. The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system. On some systems, the long term scheduler may not be available or minimal. Time-sharing operating systems have no long term scheduler. When process changes the state from new to ready, then there is use of long term scheduler.

Short Term Scheduler-:

It is also called CPU scheduler. Main objective is increasing system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects process among the processes that are ready to execute and allocates CPU to one of them. Short term scheduler also known as dispatcher, execute most frequently and makes the fine grained decision of which process to execute next. Short term scheduler is faster than long term scheduler.

Medium Term Scheduler-:

Medium term scheduling is part of the swapping. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium term scheduler is in-charge of handling the swapped out-processes. Running process may become suspended if it makes an I/O request. Suspended

processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other process, the suspended process is moved to the secondary storage. This process is called swapping, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix. Comparison between Scheduler:-

S.N.	Long Term Scheduler	Short Term Scheduler	Medium Term Scheduler
1	It is a job scheduler	It is a CPU scheduler	It is a process swapping scheduler.
2	Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between both short and long term scheduler.
3	It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming	It reduces the degree of multiprogramming.
4	It is almost absent or minimal in time sharing system	It is also minimal in time sharing system	It is a part of Time sharing systems.
5	It selects processes from pool and loads them into memory for execution	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

CPU Scheduling: CPU scheduling is a process which allows one process to use the CPU while the execution of another process is on hold (in waiting state) due to unavailability of any resource like I/O etc, thereby making full use of CPU. The aim of CPU scheduling is to make the system efficient, fast and fair. Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out by the short-term scheduler (or CPU scheduler). The scheduler selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them.

Types of CPU Scheduling:

(i) Non-Preemptive Scheduling

(ii) Preemptive Scheduling

Non-Preemptive Scheduling: Under non-preemptive scheduling, once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state. This scheduling method is used by the Microsoft Windows 3.1 and by the Apple Macintosh operating systems. It is the only method that can be used on certain hardware platforms, because It does not require the special hardware (for example: a timer) needed for preemptive scheduling.

Preemptive Scheduling: In this type of Scheduling, the tasks are usually assigned with priorities. At times it is necessary to run a certain task that has a higher priority before another task although it is running. Therefore, the running task is interrupted for some time and resumed later when the priority task has finished its execution.

Scheduling Algorithms : We'll discuss four major scheduling algorithms here which are following :

1. First Come First Serve(FCFS) Scheduling
2. Shortest-Job-First(SJF) Scheduling
3. Priority Scheduling
4. Round Robin(RR) Scheduling
5. Multilevel Queue Scheduling
6. Multilevel Feedback Queue Scheduling

1. First Come First Serve(FCFS) Scheduling : Points to remember-

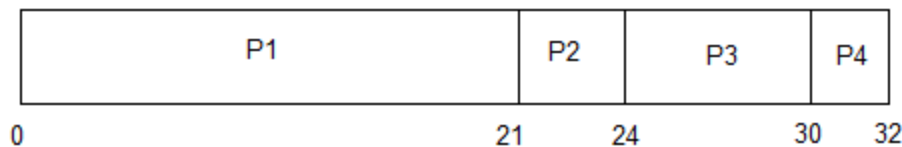
- ❖ Jobs are executed on first come, first serve basis.
- ❖ Easy to understand and implement.
- ❖ Poor in performance as average wait time is high.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2



m

The average waiting time will be = $(0 + 21 + 24 + 30) / 4 = \underline{18.75 \text{ ms}}$



This is the GANTT chart for the above processes

2. Shortest-Job-First(SJF) Scheduling : Points to remember-

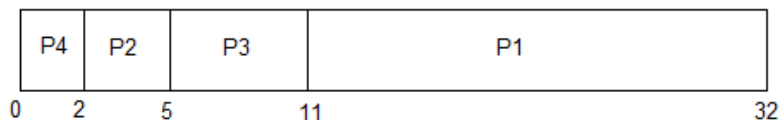
- ❖ Best approach to minimize waiting time.
- ❖ Actual time taken by the process is already known to processor.
- ❖ Impossible to implement.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2



In Shortest Job First Scheduling, the shortest Process is executed first.

Hence the GANTT chart will be following :

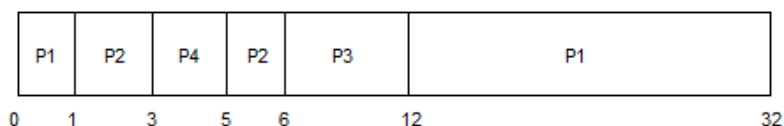


Now, the average waiting time will be = $(0 + 2 + 5 + 11)/4 = 4.5$ ms

In Preemptive Shortest Job First Scheduling, jobs are put into ready queue as they arrive, but as a process with short burst time arrives, the existing process is preempted.

PROCESS	BURST TIME	ARRIVAL TIME
P1	21	0
P2	3	1
P3	6	2
P4	2	3

The GANTT chart for Preemptive Shortest Job First Scheduling will be,



The average waiting time will be, $((5-3) + (6-2) + (12-1))/4 = 4.25$ ms

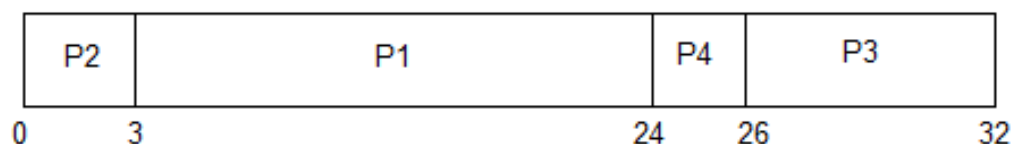
The average waiting time for preemptive shortest job first scheduling is less than both, non-preemptive SJF scheduling and FCFS scheduling.

3. Priority Scheduling : Points to remember-

- ❖ Priority is assigned for each process.
- ❖ Process with highest priority is executed first and so on.
- ❖ Processes with same priority are executed in FCFS manner.
- ❖ Priority can be decided based on memory requirements, time requirements or any other resource requirement.

PROCESS	BURST TIME	PRIORITY
P1	21	2
P2	3	1
P3	6	4
P4	2	3

The GANTT chart for following processes based on Priority scheduling will be,

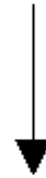


The average waiting time will be, $(0 + 3 + 24 + 26) / 4 = \underline{13.25 \text{ ms}}$

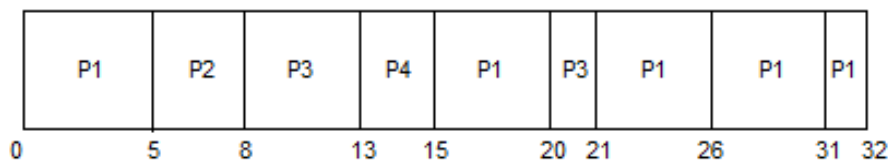
4. Round Robin(RR) Scheduling : Points to remember-

- ❖ A fixed time is allotted to each process, called quantum, for execution.
- ❖ Once a process is executed for given time period that process is preempted and other process executes for given time period.
- ❖ Context switching is used to save states of preempted processes.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2



The GANTT chart for round robin scheduling will be,



The average waiting time will be, 11 ms.

5. **Multilevel Queue Scheduling:** Another class of scheduling algorithms has been created for situations in which processes are easily classified into different groups.

For example: A common division is made between foreground (or interactive) processes and background (or batch) processes. These two types of processes have different response-time requirements, and so might have different scheduling needs. In addition, foreground processes may have priority over background processes.

A multi-level queue scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority, or process type. Each queue has its own scheduling algorithm.

For example: separate queues might be used for foreground and background processes. The foreground queue might be scheduled by Round Robin algorithm, while the background queue is scheduled by an FCFS algorithm.

In addition, there must be scheduling among the queues, which is commonly implemented as fixed-priority preemptive scheduling. For example: The foreground queue may have absolute priority over the background queue.

6. **Multilevel Feedback Queue Scheduling:** In a multilevel queue-scheduling algorithm, processes are permanently assigned to a queue on entry to the system. Processes do not move between queues. This setup has the advantage of low scheduling overhead, but the disadvantage of being inflexible. Multilevel feedback queue scheduling, however, allows a process 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 lower-priority queue may be moved to a higher-priority queue. This form of aging prevents starvation.

Scheduling Criteria:

There are many different criteria's to check when considering the "best" scheduling algorithm:

❖ **CPU utilization:**

To make out the best use of CPU and not to waste any CPU cycle, CPU would be working most of the time (Ideally 100% of the time). Considering a real system, CPU usage should range from 40% (lightly loaded) to 90% (heavily loaded.)

❖ **Throughput:**

It is the total number of processes completed per unit time or rather say total amount of work done in a unit of time. This may range from 10/second to 1/hour depending on the specific processes.

❖ **Turnaround time:**

It is the amount of time taken to execute a particular process, i.e. The interval from time of submission of the process to the time of completion of the process(Wall clock time).

❖ **Waiting time:**

The sum of the periods spent waiting in the ready queue amount of time a process has been waiting in the ready queue to acquire get control on the CPU.

❖ **Load average:**

It is the average number of processes residing in the ready queue waiting for their turn to get into the CPU.

❖ **Response time:**

Amount of time it takes from when a request was submitted until the first response is produced. Remember, it is the time till the first response and not the completion of process execution(final response)

Thread-:

A thread is a flow of execution through the process code, with its own program counter, system registers and stack. A thread is also called a light weight process. Threads provide a way to improve application performance through parallelism. Threads represent a software approach to improving performance of operating system by reducing the overhead thread is equivalent to a classical process. Each thread belongs to exactly one process and no thread can exist outside a process. Each thread represents a separate flow of control. Threads have been successfully used in implementing network servers and web server. They also provide a suitable foundation for parallel execution of applications on shared memory multiprocessors. Following figure shows the working of the single and multithreaded processes.

Difference between Process and Thread

S.N.	Process	Thread
1	Process is heavy weight or resource intensive.	Thread is light weight taking lesser resources than a process.
2	Process switching needs interaction with operating system.	Thread switching does not need to interact with operating system.
3	In multiple processing environments each process executes the same code but has its own memory and file resources.	All threads can share same set of open files, child processes.
4	If one process is blocked then no other process can execute until the first process is unblocked.	While one thread is blocked and waiting, second thread in the same task can run.
5	Multiple processes without using threads use more resources.	Multiple threaded processes use fewer resources.
6	In multiple processes each process operates independently of the others.	One thread can read, write or change another thread's data.

Difference between User Level & Kernel Level Thread:-

S.N.	User Level Threads	Kernel Level Thread
1	User level threads are faster to create and manage.	Kernel level threads are slower to create and manage.
2	Implementation is by a thread library at the user level.	Operating system supports creation of Kernel threads.
3	User level thread is generic and can run on any operating system.	Kernel level thread is specific to the operating system.
4	Multi-threaded application cannot take advantage of multiprocessing.	Kernel routines themselves can be multithreaded.

Swapping- :

Swapping is a mechanism in which a process can be swapped temporarily out of main memory to a backing store, and then brought back into memory for continued execution. Backing store is a usually a hard disk drive or any other secondary storage which fast in access and large enough to accommodate copies of all memory images for all users. It must be capable of providing direct access to these memory images. Major time consuming part of swapping is transfer time. Total transfer time is directly proportional to the amount of memory swapped. Let us assume that the user process is of size 100KB and the backing store is a standard hard disk with transfer rate of 1 MB per second. The actual transfer of the 100K process to or from memory will take $100\text{KB} / 1000\text{KB per second}$

= 1/10 second

= 100 milliseconds

Fragmentation- :

As processes are loaded and removed from memory, the free memory space is broken into little pieces. It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused. This problem is known as Fragmentation.

Paging-:

External fragmentation is avoided by using paging technique. Paging is a technique in which physical memory is broken into blocks of the same size called pages (size is power of 2, between 512 bytes and 8192 bytes). When a process is to be executed, its corresponding pages are loaded into any available memory frames. Logical address space of a process can be non-contiguous and a process is allocated physical memory whenever the free memory frame is available. Operating system keeps track of all free frames. Operating system needs n free frames to run a program of size n pages. Address generated by CPU is divided into **Page number (p)** - page number is used as an index into a page table which contains base address of each page in physical memory.

Page offset [D] - page offset is combined with base address to define the physical memory address.

Segmentation-:

Segmentation is a technique to break memory into logical pieces where each piece represents a group of related information. For example, data segments or code segment for each process, data segment for operating system and so on. Segmentation can be implemented using or without using paging. Unlike paging, segments are having varying sizes and thus eliminates internal fragmentation. External fragmentation still exists but to lesser extent.

Deadlocks-:

Deadlocks are a set of blocked processes each holding a resource and waiting to acquire a resource held by another process.

How to avoid Deadlocks-:

Deadlocks can be avoided by avoiding at least one of the four conditions, because all these four conditions are required simultaneously to cause deadlock.

1. Mutual Exclusion-:

Resources shared such as read-only files do not lead to deadlocks but resources, such as printers and tape drives, require exclusive access by a single process.

2. Hold and Wait-:

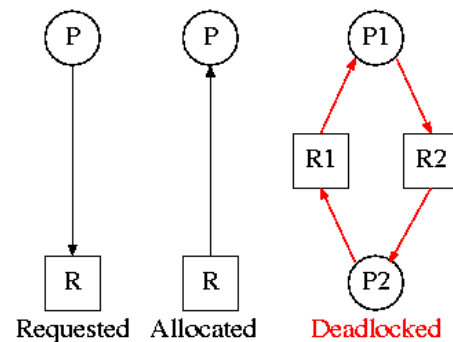
In this condition processes must be prevented from holding one or more resources while simultaneously waiting for one or more others.

3. No Preemption-:

Preemption of process resource allocations can avoid the condition of deadlocks, where ever possible.

4. Circular Wait-:

Circular wait can be avoided if we number all resources, and require that processes request resources only in strictly increasing (or decreasing) order.

**Handling Deadlock-:**

The above points focus on preventing deadlocks. But what to do once a deadlock has occurred. Following three strategies can be used to remove deadlock after its occurrence.

1. Preemption-:

We can take a resource from one process and give it to other. This will resolve the deadlock situation, but sometimes it does cause problems.

2. Rollback-:

In situations where deadlock is a real possibility, the system can periodically make a record of the state of each process and when deadlock occurs, roll everything back to the last checkpoint, and restart, but allocating resources differently so that deadlock does not occur.

3. Kill one or more processes-: This is the simplest way, but it works.

India's Best Selling Test Series for

IBPS SO Pre & Main Exam

IBPS SO Law Officer	IBPS SO I.T. Officer	IBPS SO Marketing Officer	IBPS SO Agri. Officer
20 Tests	20 Tests	20 Tests	20 Tests
<ul style="list-style-type: none">• 10 Pre + 10 Main Tests• Complete review & analysis• Bilingual English & Hindi	<ul style="list-style-type: none">• 10 Pre + 10 Main Tests• Complete review & analysis• Bilingual English & Hindi	<ul style="list-style-type: none">• 10 Pre + 10 Main Tests• Complete review & analysis• Bilingual English & Hindi	<ul style="list-style-type: none">• 10 Pre + 10 Main Tests• Complete review & analysis• Bilingual English & Hindi
₹ 299/-	₹ 299/-	₹ 299/-	₹ 299/-

www.ibtindia.com/testseries