1) What is scheduler ? Explain various types of scheduler

Process scheduling is the activity of process manager that handles the removal of the running process from CPU and the selection of another process on the basis of particular strategy.

There are three types of process scheduler—

1) Long term scheduler or Job scheduler-

Long term scheduler brings new process into ready state. It connect degree of of multi-programming i.e. number of process present in ready state at any point of time. It is imp. to know that long term scheduler chooses process wisely between both input/output & CPU bound process. I/O bounds tasks are which use much of their time in input & output operation while CPU bound process spend their time in CPU.

2) Short term scheduler or CPU scheduler —

It is responsible for selecting one process from ready state for scheduling it on the running state. Short term scheduler only select those process to schedule it doesn't load the process on running. Here all the scheduling algorithms are used. The CPU scheduler is responsible for ensuring that their is no starvation

Dispatcher is responsible for loading the process selected by short term scheduler. Context switching is done by dispatcher.

Anvay

A dispatcher does -

i) switching context

ii) switching to user mode

iii) Jumping to the proper location in the newly loaded program.

3) Medium term scheduler.

It is responsible for suspending and resuming the process. It mainly does swapping. Swapping is may be necessary to improve the process mix or because a change in memory requirement has overcommi - tted available memory, requiring memory to free up. It reduces the degree of multi-programming.

Answer.

2) What is Round-robin scheduling algorithm?

The round robin scheduling algorithm in OS is one of the CPU scheduling algorithm in which every process gets a fixed amount of time quantum to execute process.

In this algorithm every process gets executed in a cyclic way which means that process that have their burst time remaining after the expiration of the time quantum are sent back to the ready queue and wait for their next turn to complete execution unit until its terminates. This process is done in FIFO order which suggest that process are executed on a first come-first serve basis

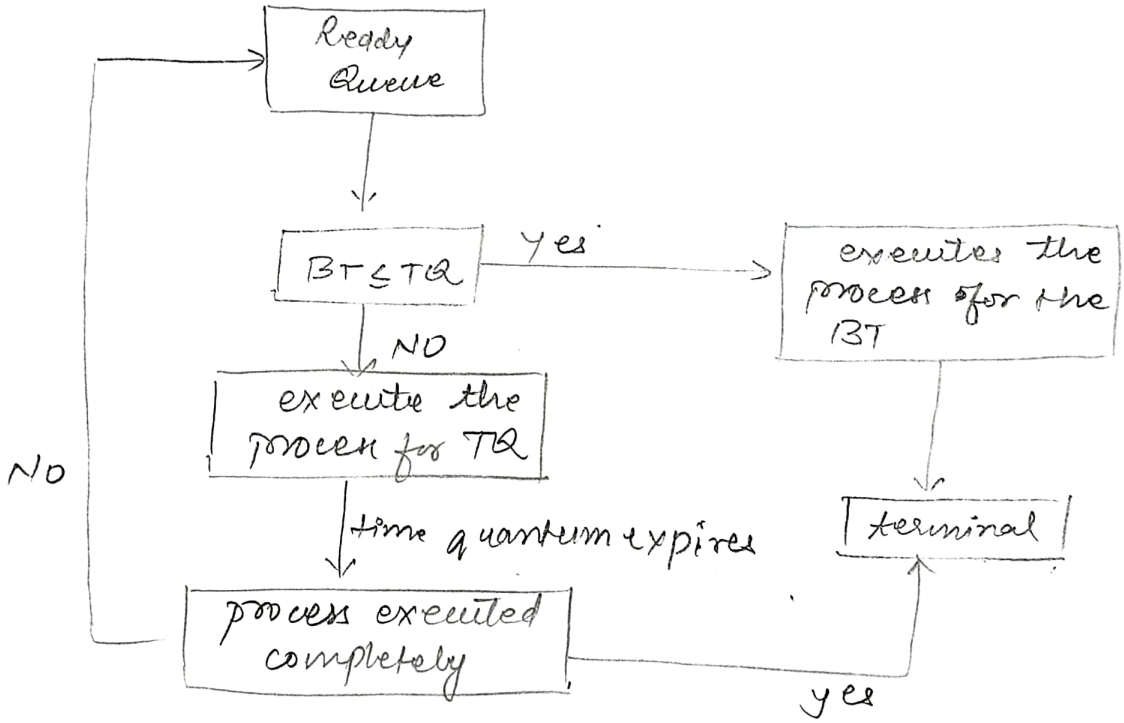
How does round-robin scheduling algorithm works-

1) all the process are added to ready queue
2) The of burst time of of every process is compared to time quantum of CPU.
3) If burst time of process is less than or equal to time quantum in the round robin scheduling algorithm, the process is executed to its burst time
4) If the process is greater then time quantum then it'll be processed until its time quantum reached.
5) When the time quantum expires, it checks if the process is executed completely or not.

Answer

6) on completion of process. process terminates. Else, it goes back again in the ready queue.

flow diagram of Ready Queue.



Characteristic of Round Robin scheduling in OS.

- Round Robin scheduling in OS is clock driven.
- It's a preemptive type of CPU scheduling algorithm.
- It's a time sharing technique.
- Round robin scheduling algorithm is widely used in traditional OS.

Example

consider six process with their burst time and arrival time given.

quantum time = 4

Process id	arrival time	Burst time
P_1	0	7
P_2	1	6
P_3	2	3
P_4	3	1
P_5	4	5
P_6	6	4

Ready Queue -
at first process P1 will be executed for time 4 unit since initially there are no process so it'll be only process in the ready queue.

Ready Queue →

P_1
5

burst time

along with the execution of P_1 four more process P_2 P_3, P_4, P_5 arrive in the ready queue and P_1 will be added back to the remaining 1 unit

Reday Queue →

P_2	P_3	P_4	P_5	P_1
6	3	1	5	1

During the execution of P_2, P_6 has arrived in the ready queue, since P_2 is not completed yet P_2 will also be added back to ready queue

Ready queue

P_3	P_4	P_5	P_1	P_6	P_2
3	1	5	1	4	2

Similarly P_3 and P_4 have been completed but P_5 has remaining burst time of 1 unit. Hence it'll be added back in the ready queue.

Amar

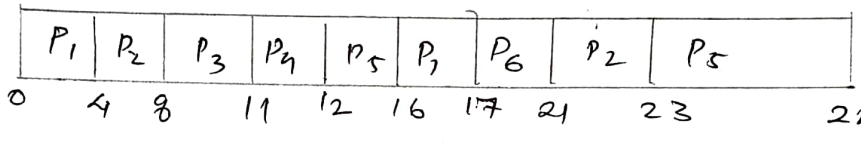
Ready Queue -

P_1	P_6	P_2	P_5
1	4	2	1

The next process P_6 & P_2 will be executed only P_5 will be left with 1 unit of burst time.

P_5
1

Gantt Chart -



(*) Multilevel Queue feedback Scheduling.

Multilevel feedback queue scheduling algorithm is like multilevel queue scheduling but in this process can move between the queue & thus more efficient than multilevel queue scheduling.

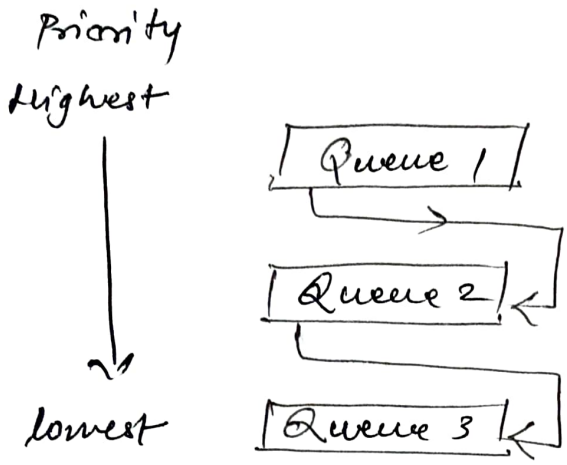
Characteristic of multilevel feedback queue scheduling -

1) process are permanently assigned to a queue on entry to a system and process are not allowed to move between queue.

2) As process are permanently assigned to a the queue this set-up has advantage of low scheduling overhead.

Multilevel queue feedback allow process to move between queues, this keep analyzing the behaviour (time of execution) and process according to which it changes it's priority.

Ans -



Implementation of Multilevel feedback Queue Scheduling

- When a process start executing the operating system can insert it into any of the above three queue depending upon its priority. for example if it's some background process, then the operating system would not like to be given higher priority queue such a queue 1 or 2. It'll directly assign to lower queues.
- In queue 1 process executes for 4 unit and if its complete in this 4 unit or it gives CPU or I/O operation in these 4 unit then the priority of that process won't change and if it again comes to the ready queue it'll start execution from Queue1.
- If a process doesnot complete in 4 unit of time in Queue 1. then priority get reduced and it shifted to lower priority queue.
- In general if process doesn't complete in given time quantum than its get shifted to lower priority queue
- A process in lower priority queue can only get executed when Higher priority queues are empty.
- A it may get interrupted if higher priority process ~~queue~~ come in ready queue

Anuj

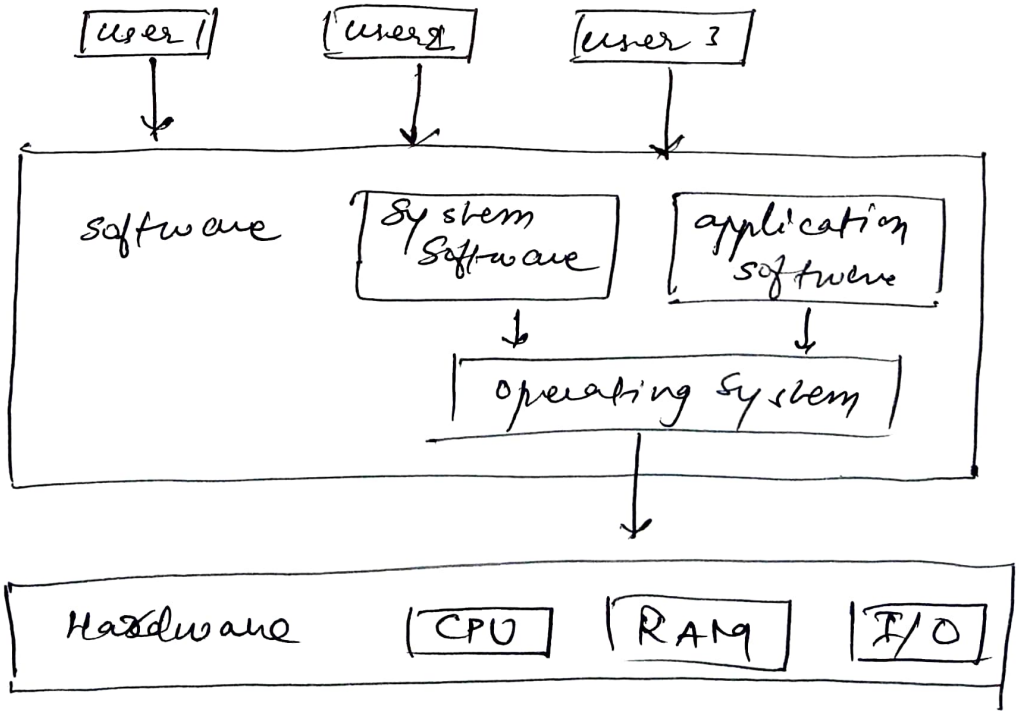
3) What is operating system ? Explain monolithic & micro kernel.

Operating System —

An operating system is an interface between a computer & user. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input & output controlling disk & printers.

An operating system is a software that enables application to interact with a computers hardware. The software that contain core component of operating system is known as kernel.

Generic architecture design of operating system is.



Ans

Monolithic kernel –

This is a type of operating system architecture in which the entire operating system work in the kernel space.

The monolithic kernel differ from other operating sys. structure like micro kithic kernel as this's provide the virtual interface alone over the computer computer hardware which makes it more useful.

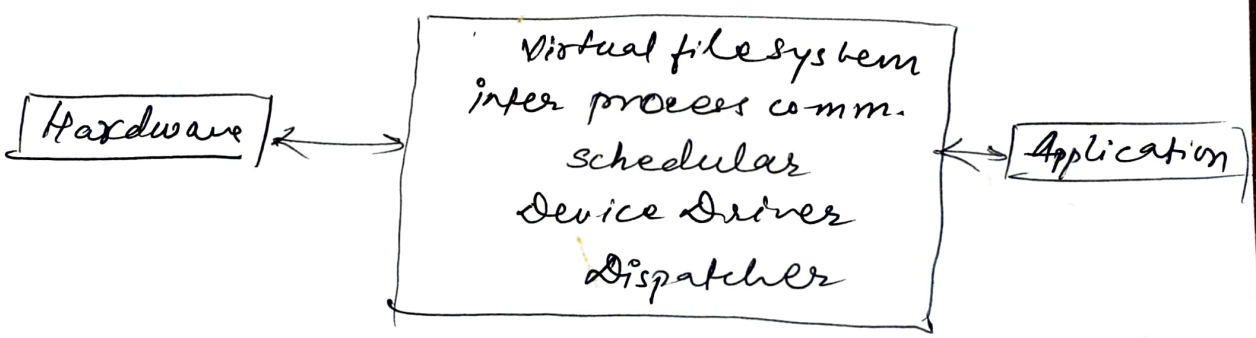
The operating system is written as a collection of procedures that are linked together into a single large executable program. Each process in the system is free to call any other process. This ensures efficiency. No chance of information hiding.

Example - MS DOS & Linux.

Basic Structure – 1) Main function - invokes requested services procedure

2) Service Procedure - carry out system calls
3) utility function - help service procedure to perform certain task.

Architecture –

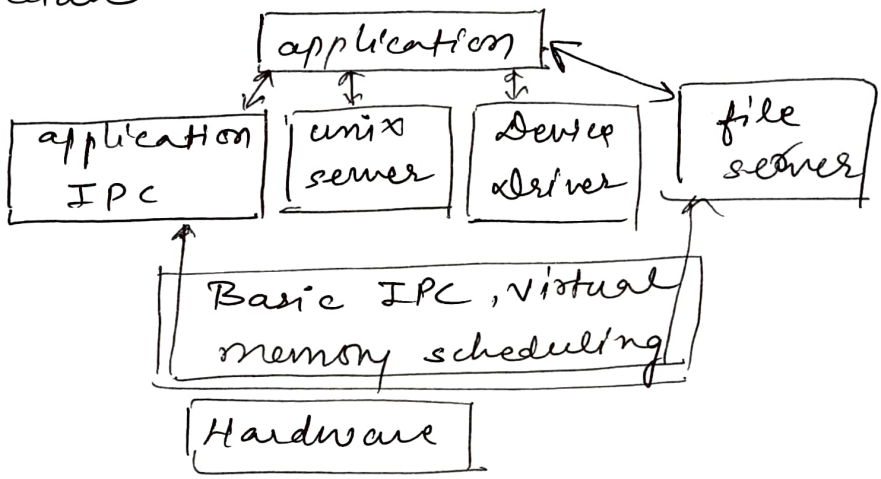


Aman

Micro kernel —

A micro kernel is one of the classification of kernels. In micro kernel the user service and kernel service are implemented in different space. The user service are kept in user address space and kernel service are kept in kernel address space.
Thus it reduce the size of kernel and as well as size of operating system.

architecture —



The operating system remains unaffected since user service & kernel services are kept isolated so if any user fail it does not affect kernel service.
Therefore micro kernel are easily extendible.

Advantage —

1) The architecture is small & isolated therefore it functions well.

2) Expansion of system is easier

~~Eclipse~~ Eclipse IDE is good example of micro kernel.

Ans

4) Difference between windows & linux operating system

Linux	Windows
1) Monolithic kernels are used in Linux operating systems	1) Micro kernels are used in windows operating system
2) Linux is more efficient	2) Windows is less efficient
3) Provides more security	3) Windows is less secure
4) Linux is widely used for hacking purpose based system	4) windows doesn't provide much efficiency in hacking
5) user accounts - 1) Regular 2) Root 3) Service account	5) user account - i) administrator ii) Standard iii) Child iv) Guest

Anwar

5) Explain Process . Explain process state with diagram

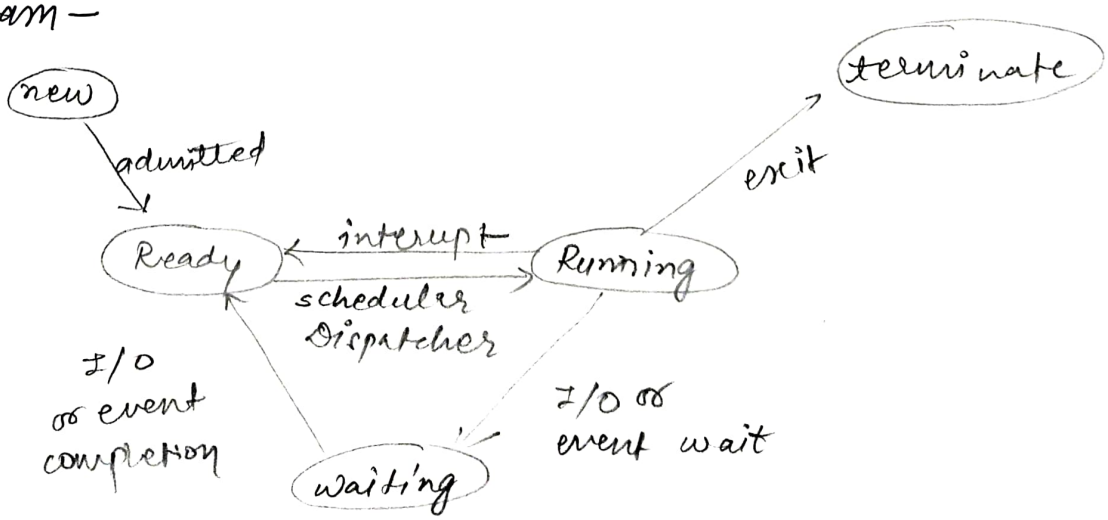
Process is a program in which execution and it is more than a program code called as text section and this concept works under all the operating system because all the task performed by the operating system needs a process to perform the task.

The process executes when it changes the state

Each process may be in the any of these state.

1) New - the process is being created
2) Running - in this state the instruction are being executed.
3) Waiting - the process is in waiting state until an event occurs like I/O operation completion or recieving a signal.
4) Ready - the process is ready to being assigned to a processor.
5) terminated - the process has ~~fixed~~ execution finished

Diagram -



Answer

6) What is critical section? Explain Bankers Algorithm.

Critical Section — when more than one process access the same code segment that segment is known as critical section. The critical section contains shared variables or resource which are needed to be synchronised to maintain the consistency of data var. In concurrent programming with, if one thread tries to change the value of shared data at the same time at another thread tries to read the value, the result is unpredictable.

Entry Section

critical section

Exit Section

Remainder Section

psuedo code —

```
do{
    flag = 1;
    while (flag) ; //entry section
            //critical section
    if (!flag)
            //remainder section
} while (true);
```

A thread must aquire a lock prior to executing a critical section. The lock can be acquired by only one thread. there are various ways to implement lock in the above psuedo code.

Bankers Algorithm in operating system.

The bankers algorithm is a deadlock avoidance algorithm that tests for safety by simulating the allocation of predetermined maximum possible amounts of all resource then makes an "s-state" (safe state) to check if their any possible activity before deciding where allocation should be allowed to continue.

Why bankers algorithm named so?
Bankers Algorithm is named so because it is used in banking system to check whether loan can be sanctioned to a person or not. The bank would never allocate its money in such a way that it can no longer satisfy the need of all customer the bank would try to be in safe state

Let 'n' be the number of process in system and 'm' be the number of resource

Available :-
- it is a 1-D array of size 'm' indicating the number of available resource of each type.
- Available[j] = k means there are 'k' instance of resource type Rj.

Max:
- It is a 2-D array of size 'n*m' that defines the max. demand of each process in a system.
- Max[i,j] = k means process Pi may request at most 'k' instance of resource type Rj.

Aman

Allocation :

- It is a 2-D array of size 'n * m' that defines the number of resource of each type currently allocated to each process.
- Allocation[i, j] = k means process P_i is currently allocated 'k' instance of resource type R_j.

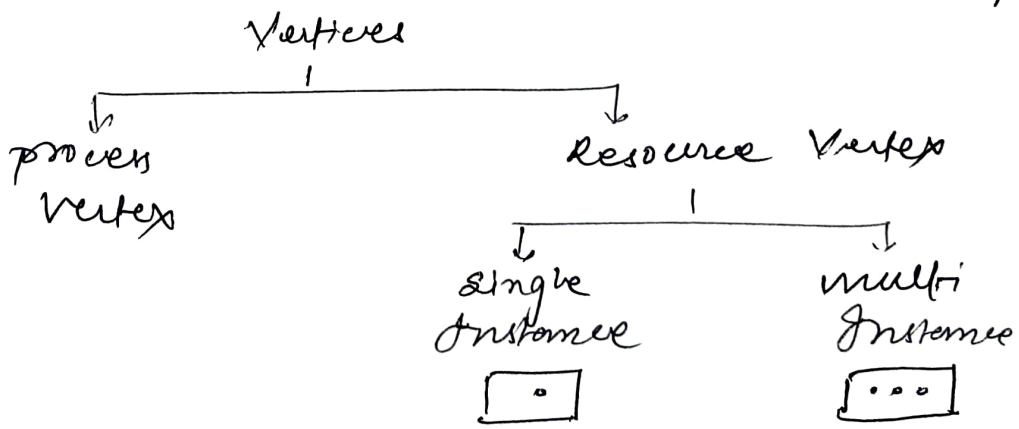
Need :-

- It is a 2-D array of size 'n * m' that indicates the remaining resource need by each process
- need[i, j] = k means P_i is currently need 'K' instance of resource type R_j.
- need[i, j] = max[i, j] - Allocation[i, j]

7) Explain Resource Allocation Graph.

Resource allocation graph is used to so that we can see How many resources are available, How many are allocated everything can be represented in terms of Diagram One advantage of having diagram is sometimes it possible to see deadlock directly using RAG.

1. Process vertex - every process will be represented as a process vertex. Generally, the process will be represented with a circle
2. Resource vertex - Every resource will be Represented as a resource vertex. It's also two type-
   i) single Instance type - It represent as a box inside the box - there will be one dot. So the number of dot represent How many instance it holds.
   ii) multi - instance type - same as single instance but represented by many dots

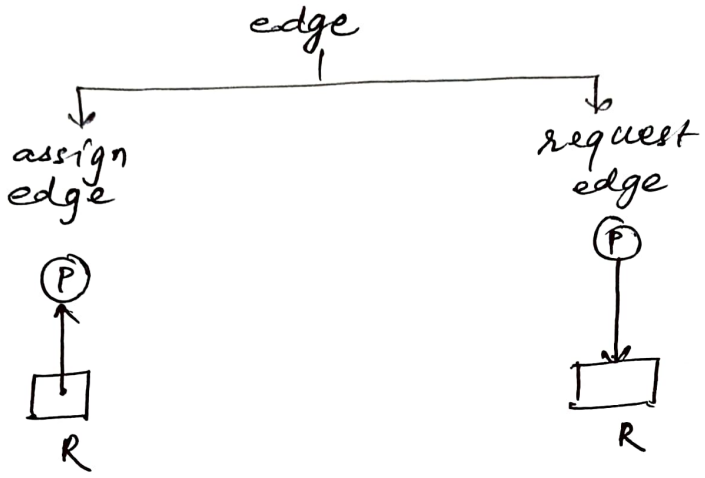


There are two type of edges also -

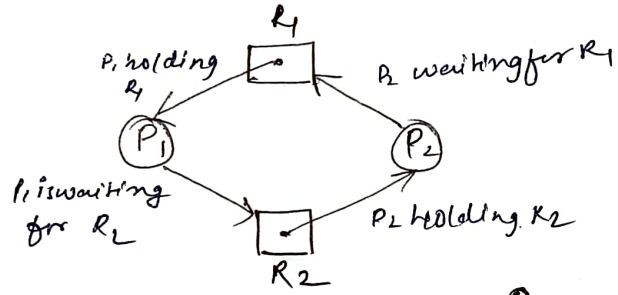
1. assign edge. If you already assign a resource to a process then it is called assign edge.

Aman.

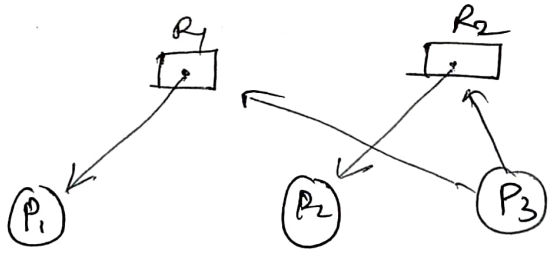
2) Request edge – it means the future process might want some resource to complete execution that is called request edge.



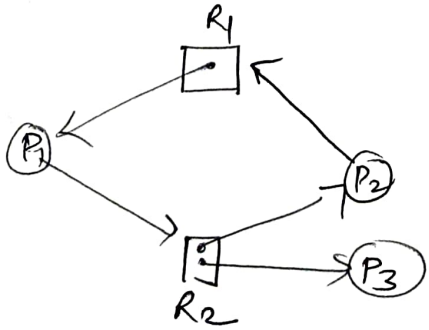
example 1. Single instance RAG.



Single instance with deadlock



Single instance without deadlock



Anam

from the above example it is not possible to say whether RAG is in safe state or not.

constructing allocation & request matrix.

Process	Allocation R_1	Allocation R_2	Request R_1	Request R_2
P_1	1	0	0	1
P_2	0	1	1	0
P_3	0	1	0	0

Allocation matrix:

- for constructing the allocatim matrix, just go to the resource and see to which process it is allocated.
- R_1 is allocated to P_1, therefore write 1 in allocation matrix and similarly, R_2 is allocated to P_2 as well as P_3 and for remaining just write 0.

Request Matrix –

- In order to find out the request matrix, you have to go to the process and see outgoing edges.
- P_1 is requesting resource of R_2. so write 1 in the ~~next~~ matrix similarly P_2 is requesting for R_1 and other write 0.

Checking deadlock –

	R_1	R_2
Available	0	0
P_3	0	1
new Available	0	1
P_1	1	0
new Available	1	1
P_2	0	1
Available	1	2

As P_3 does not require any extra resource to complete the execution and after completion of P_3 release its own resource

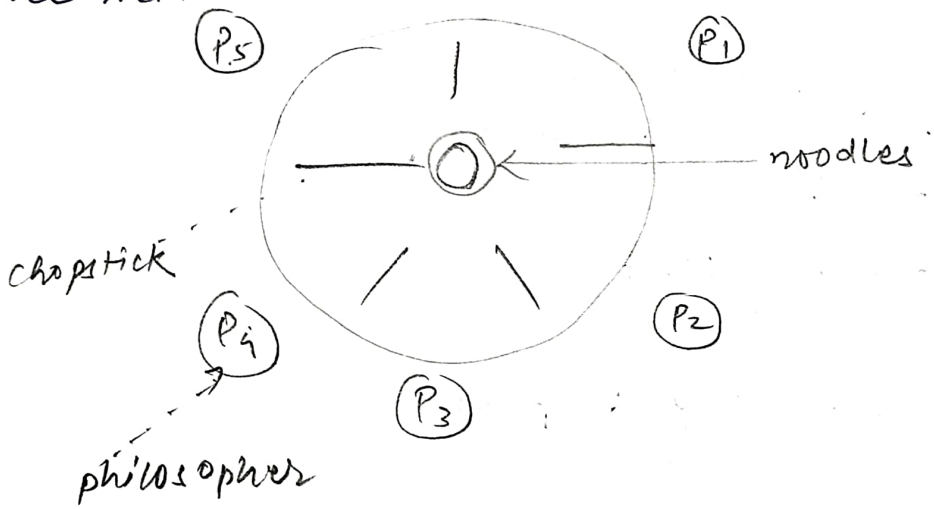
Now using new available resource we can satisfy requirement of P_1 & it'll also release its resources.

Now we can easily satisfy P_2

∴ there is no deadlock.

8) Classical problem of synchronisation

1) Dining philosopher- The dining phillospher problem state that k philosophers seated around a circular table with one chopstic between each pair of philosopher. A philosopher may eat if he can pick two chopstick adjacent to him. One chopstick may be picked up by any one of its adjacent follower but not both this problem involves allocation of limited resource to a group of proces in a deadlock free and starvation free manner.



2) Bounded Buffer- Bounded Buffer problem is also called producer-consumer problem. This problem is generalised in terms of producer-consumer problem. Solution to this problem is, creating two counting semaphore, "full" & "empty" to keep track of current number of full & empty buffer respectiv-ely. Producers produce a product and consumer consumes a product. but both use one of the container each time.

3) Sleeping Barber- Barber shop with one barber, one barber chair and N chairs to wait in When no customer the barber goes to sleep in Barber chair and must be woken up when customer come in. When Barber is cutting hair new customer take empty seat to wait or leave

9) What is file? Explain various file allocation method.

files are used for all inputs and outputs of information in the operating system, to standardize access to both hardware and software.

Input occur when the content of file is modified or written to. Output occurs when the content of one file is read or transferred to another file.

Types of file —

- Recognised by system are either regular, directory or special.
- Administering file - there are many ways to work with files on a system. usually you create a text file from a text editor.
- file and directory link
- DOS files
- Command summary for files.

File allocation method — this defines how files are allocated in disk

i) contiguous allocation
ii) linked allocation
iii) indexed allocation

the main idea behind this method is to provide :—

i) efficient disk space utilisation.
ii) fast access to file blocks.

1) Contagious allocation —

Each file occupies a contagious set of block on the disk for example if a file require n number of block & is given block b as starting then block assigned to file is : b, b+1, b+2 . . —

Anuay

The directory entry of a file with contagious allocation contains -

1) address of the starting block.
2) length of allocated portion.

Advantage :-
i) Both sequential & direct access are supported by this.
ii) Extremely fast since number of seek are minimal

2) Linked list Allocation. —
Each file is a linked list of disk block which need not to be contagious. The disk block can be scattered anywhere.
The directory entry contains a pointer to the starting and ending file block. Each block contains a pointer to the next block occupied by the file

Advantage =
i) this is very flexible in terms of file size, file size can be increased.
ii) Does not suffer from external fragmentation

3) Indexed allocation -
A special block known as index block contain the pointer to all block occupied by a file. Each file has its own index. the ith entry in the index block contain the disk address of the ith file block.

Advantage —
- support direct access to the block occupied by the file and therefore provide fast access to the file block.
- overcomes the problem of external fragmentation

Anuj

10) What are the function of operating System

i) Security – The operating system uses password protection to protect user data and similar other techniques. Also prevent not ~~authoris~~ authorized access to program.

2) control over system performance –

Monitors overall performance of system. Record response time between service request and system response to having a complete view of the system health.

3) Job accounting –

operating system keeps track of time & resource used by various task and user, this information can be used to track resource usage for a particular user.

4) Error detecting aids –

the operating system constantly monitors the system to detect error and avoid malfunctioning of a computer system.

5) coordination between software & user.

operating system also co-ordinate and assign, interpreters compilers, assemblers and other software to various user of computer system.

6) Memory management.

operating system manages primary memory or main memory. OS decides for which process get to access in memory & for how long memory will be assigned.

7) Process Management is also done by OS, it decides which process will have access to processor and how much processing time each process has.

Answer