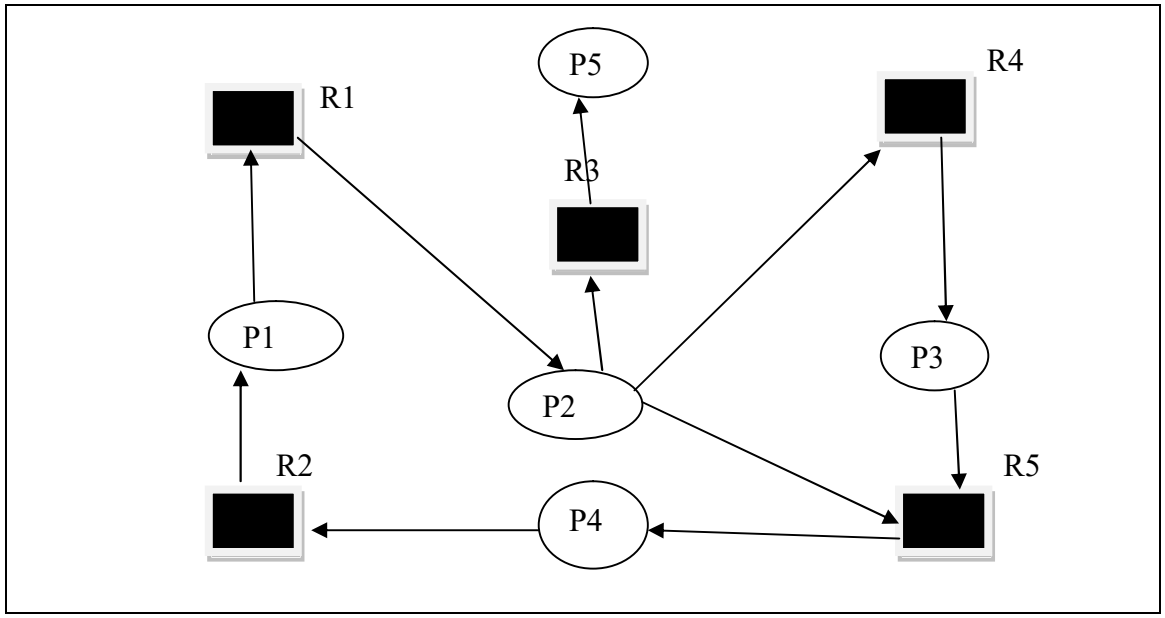
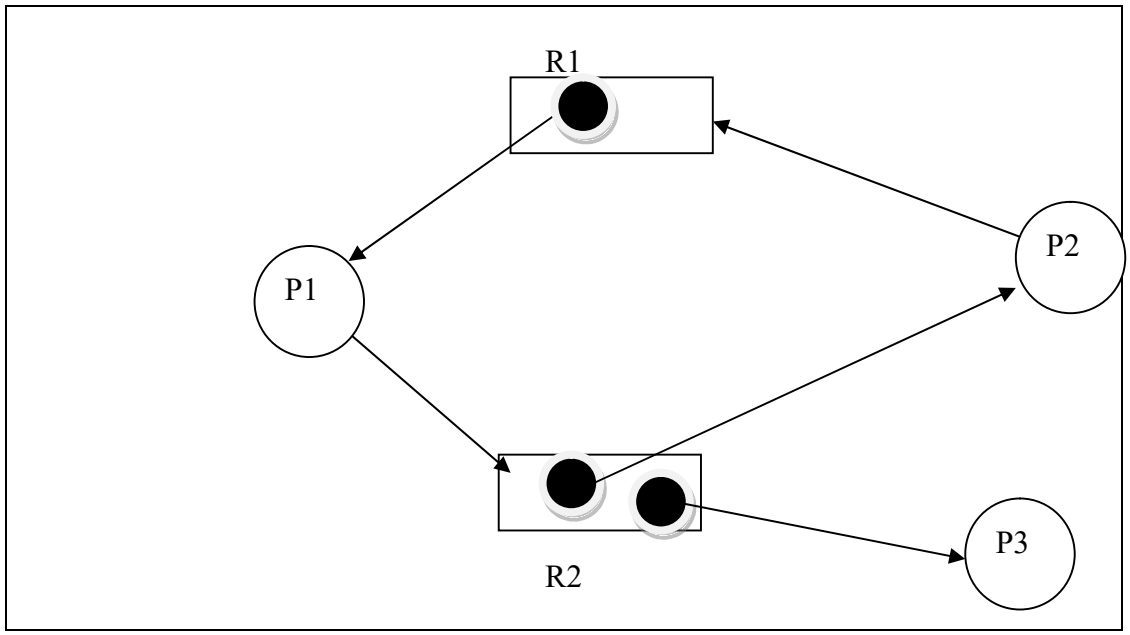
Name: Maskur Al Shal Sabil **ID: IT18021**

1. (a) What is deadlock ? Write down the deadlock detection. 3

(b) Consider the following resource allocation graph and find the wait graph. 4



(c) Consider the resource allocation graph in the figure- 7



Find if the system is in a deadlock state otherwise find a safe sequence .

2. (a) What is paging ? Write down the advantage and disadvantages of paging. 4

(b) Differentiate between contiguous memory allocation and non contiguous scheduling . 4

(c) Define preemption ?A system is having 10 user processes each requiring 3 units of resource R. What is the minimum number of units of R such that no deadlock will occur ? 6

3. (a) What is Belady's Anomaly ? Which algorithm suffer from it and why ? 4
Also write the cause behind Belady's Anomaly.

(b) Explain Virtual memory and when it's needed . 4

(c) An operating system uses the banker's algorithm for deadlock avoidance when managing the allocation of three resource types X, Y and Z to three processes P0, P1 and P2. The table given below presents the current system state. Here, the Allocation matrix shows the current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.

	Allocation			**Max**		
	X	**Y**	**Z**	**X**	**Y**	**Z**
P0	**0**	**0**	**1**	**8**	**4**	**3**
P1	**3**	2	0	**6**	**2**	**0**
P2	**2**	**1**	**1**	**3**	**3**	**3**

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available.

The system is currently in safe state. Consider the following independent requests for additional resources in the current state-

REQ1: P0 requests 0 units of X, 0 units of Y and 2 units of Z

REQ2: P1 requests 2 units of X, 0 units of Y and 0 units of Z

Determine weather REQ1 be permitted or not. 6

4. (a) What is swapping? Write down its benefit. 4

(b) Write down the difference between Demand paging and segmentation . 4

(c) Consider a single level paging scheme. The virtual address space is 16 GB and page table entry size is 4 bytes. What is the minimum page size possible such that the entire page table fits well in one page? 6

5. (a) What is Buffering ? 2

(b) Explain the multistep processing of a user program with a neat block diagram . 6

(c) Consider a reference string: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2. the number of frames in the memory is 3. Find out the number of page faults respective to:

1. Optimal Page Replacement Algorithm
2. FIFO Page Replacement Algorithm
3. LRU Page Replacement Algorithm 6

6. (a) What is SAN and how does it differ from NAS. 4

(b) How many ways file can be accessed? 4

(c) As a ICT student obviously it's important to do programming . Here we give you a task that you should write down a program to find the number of page faults . you can choose any programming language. 6

7. (a) What is File system? 3

(b)Write down the advantages and disadvantages of various disk algorithm. 5

(c) Explain the following DML command with syntax : 6

DELETE,DROP,TRUNCATE

8. (a) What is Swapping? 2

(b) What if a system runs out of swap space? 5

(c) Explain the structure of Directory in Operating system with diagram. 7

1. (a) What is deadlock ? Write down the deadlock detection.

Ans:

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

Deadlock Detection :

Using Resource Allocation Graph, it can be easily detected whether system is in a **Deadlock** state or not.

The rules are-

Rule-01:

In a Resource Allocation Graph where all the resources are single instance.

- If a cycle is being formed, then system is in a deadlock state.
- If no cycle is being formed, then system is not in a deadlock state.

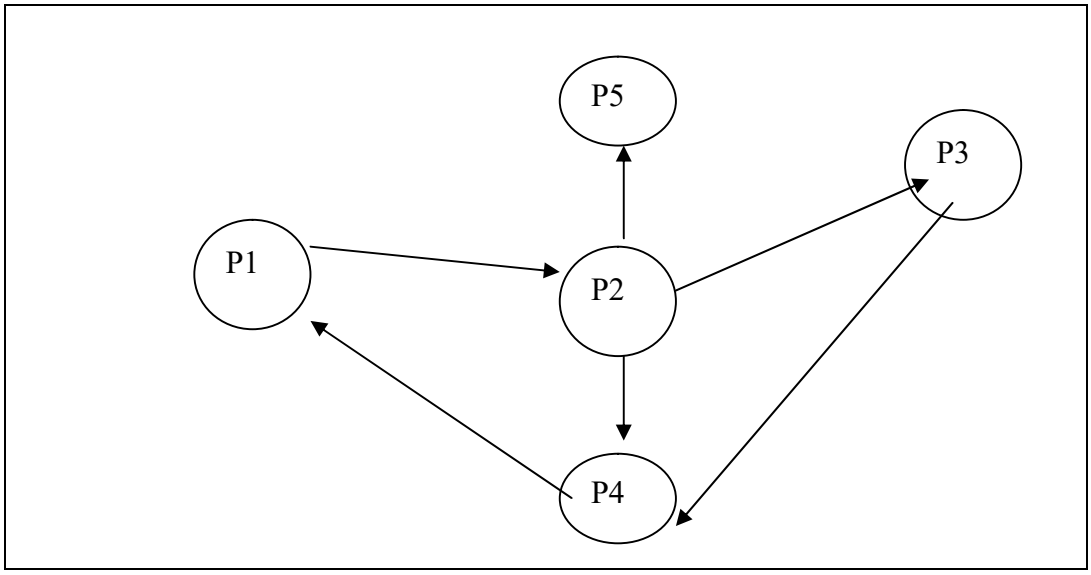
Rule-02:

In a resource allocation graph where al the resources are not single instance,

- If a cycle is being formed, then system may be in a deadlock state.
- If no cycle is being formed, then system is not in a deadlock state.
- Presence of a cycle is a necessary but not a sufficient condition for the occurrence of deadlock.

(b) Consider the given resource allocation graph and find the wait graph.

The wait graph of this resource allocation graph are given below :



(c) Consider the resource allocation graph in the figure- Find if the system is in a deadlock state otherwise find a safe sequence .

The given resource allocation graph is multi instance with a cycle contained in it.
So, the system may or may not be in a deadlock state.

Using the given resource allocation graph, we have-

	Allocation		Need	
	R1	**R2**	**R1**	**R2**
Process P1	1	0	0	1
Process P2	0	1	1	0
Process P3	0	1	0	0

Available = [R1 R2] = [0 0]

Step-01:

Since process P3 does not need any resource, so it executes.
After execution, process P3 release its resources.

Then,

Available

= [0 0] + [0 1]

= [0 1]

Step-02:

With the instances available currently, only the requirement of the process P1 can be satisfied.

So, process P1 is allocated the requested resources.

It completes its execution and then free up the instances of resources held by it.

Then-

Available

= [0 1] + [1 0]

= [1 1]

Step-03:

With the instances available currently, the requirement of the process P2 can be satisfied.

So, process P2 is allocated the requested resources.

It completes its execution and then free up the instances of resources held by it.

Then-

Available

= [1 1] + [0 1]

= [1 2]

Thus,

There exists a safe sequence P3, P1, P2 in which all the processes can be executed.

So, the system is in a safe state.

2. (a) What is paging ? Write down the advantage and disadvantages of paging.

Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages. In the Paging method, the main memory is divided into small fixed-size blocks of physical memory, which is called frames. The size of a frame should be kept the same as that of a page to have maximum utilization of the main memory and to avoid external fragmentation.

Advantages of Paging:

The advantage of paging method are given below::

- Easy to use memory management algorithm
- No need for external Fragmentation
- Swapping is easy between equal-sized pages and page frames.

Disadvantages of Paging

Here, are drawback/ cons of Paging:

- May cause Internal fragmentation
- Complex memory management algorithm
- Page tables consume additonal memory.
- Multi-level paging may lead to memory reference overhead.

(b) Differentiate between contiguous memory allocation and non contiguous scheduling . 4

Basis the comparison	Contiguous memory allocation	Noncontiguous memory allocation
Basic	Allocates consecutive blocks of memory to a process.	Allocates separate blocks of memory to a process.
Overheads	Contiguous memory allocation does not have the overhead of address translation while execution of a process.	Noncontiguous memory allocation has overhead of address translation while execution of a process.
Execution rate	A process executes fatser in contiguous memory allocation	A process executes quite slower comparatively in noncontiguous memory allocation.
Solution	The memory space must be divided into the fixed-sized partition and each partition is allocated to a single process only.	Divide the process into several blocks and place them in different parts of the memory according to the availability of memory space available.
Table	A table is maintained by operating system which maintains the list of available and occupied partition in the memory space	A table has to be maintained for each process that carries the base addresses of each block which has been acquired by a process in memory.

(c) Define preemption ?A system is having 10 user processes each requiring 3 units of resource R. What is the minimum number of units of R such that no deadlock will occur ?

Preemption as used with respect to **operating systems** means the ability of the **operating system** to **preempt** (that is, stop or pause) a currently scheduled task in favor of a higher priority task. The resource being scheduled may be the processor or I/O, among others.

In worst case,

The number of units that each process holds = One less than its maximum demand

So,

Process P1 holds 2 units of resource R

Process P2 holds 2 units of resource R

Process P3 holds 2 units of resource R and so on.

Process P10 holds 2 units of resource R

Thus,

Maximum number of units of resource R that ensures deadlock = 10 x 2 = 20

Minimum number of units of resource R that ensures no deadlock = 20 + 1 = 21

3. (a) What is Belady's Anomaly ? Which algorithm suffer from it and why ? Also write the cause behind Belady's Anomaly.

Belady's Anomaly is the phenomenon of increasing the number of page faults on increasing the number of frames in main memory.

Following page replacement algorithms suffer from Belady's Anomaly-

> FIFO Page Replacement Algorithm

>Random Page Replacement Algorithm

>Second Chance Algor

Reason Behind Belady's Anomaly-

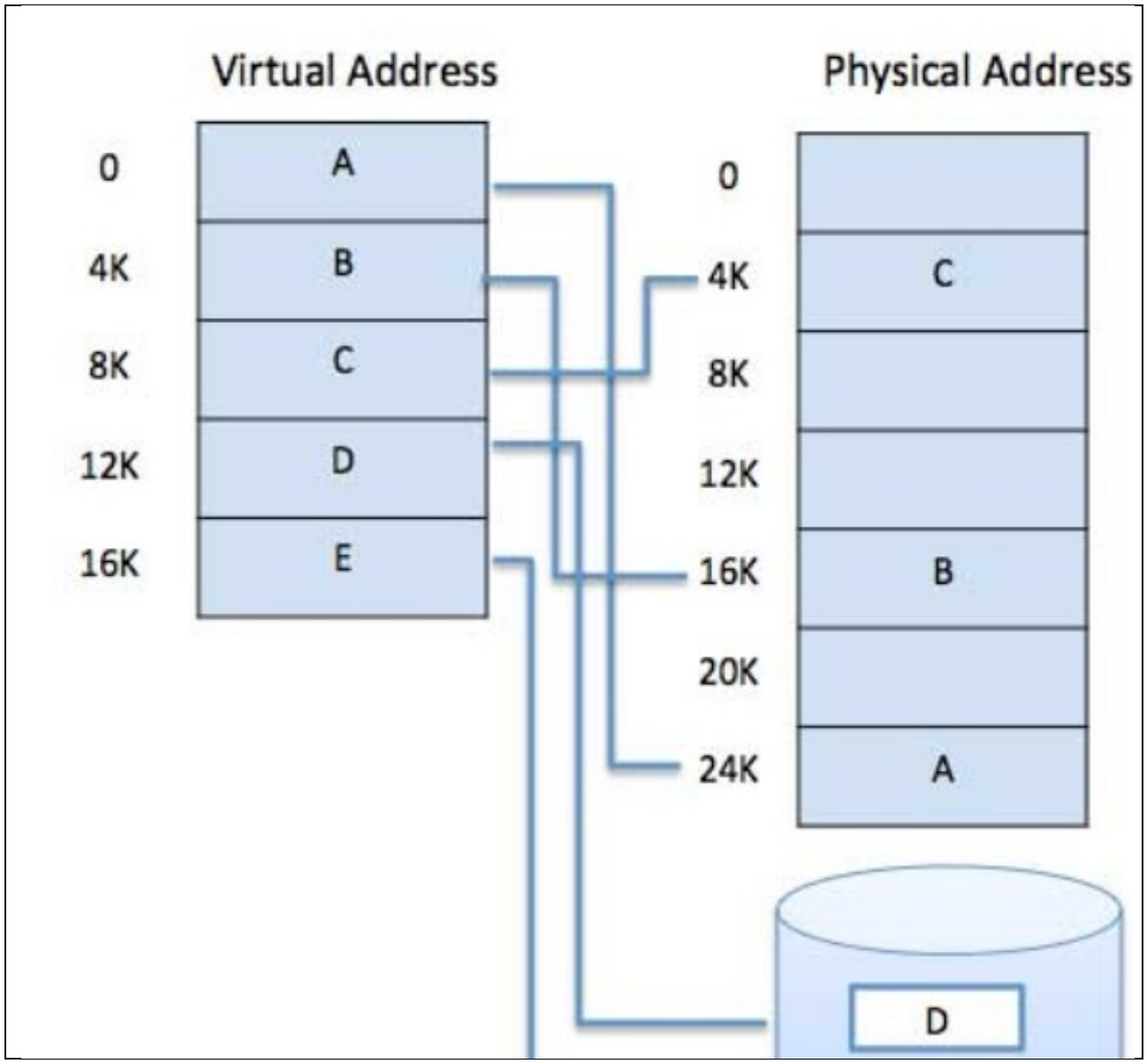
- An algorithm suffers from Belady's Anomaly if and only if it does not follow stack property.
- Algorithms that follow stack property are called as **stack based algorithms**.
  - Stack based algorithms do not suffer from Belady's Anomaly.
  - This is because these algorithms assign priority to a page for replacement that is independent of the number of frames in the main memory.

(b) Explain Virtual memory and when it's needed .

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called **virtual memory** and it is a section of a hard disk that's set up to emulate the computer's RAM.

The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

Modern microprocessors intended for general-purpose use, a memory management unit, or MMU, is built into the hardware. The MMU's job is to translate virtual addresses into physical addresses. A basic example is given below –



Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system. Demand segmentation can also be used to provide virtual memory.

Following are the situations, when entire program is not required to be loaded fully in main memory and need virtual memory

- User written error handling routines are used only when an error occurred in the data or computation.
- Certain options and features of a program may be used rarely.
- Many tables are assigned a fixed amount of address space even though only a small amount of the table is actually used.

- The ability to execute a program that is only partially in memory would counter many benefits.
- Less number of I/O would be needed to load or swap each user program into memory.
- A program would no longer be constrained by the amount of physical memory that is available.
- Each user program could take less physical memory, more programs could be run the same time, with a corresponding increase in CPU utilization and throughput.

(c) According to question,

$$\text{Available} = [\ X\ Y\ Z\ ] = [\ 3\ 2\ 2\ ]$$

Now,

$$\text{Need} = \text{Max} - \text{Allocation}$$

So, from the question we have,

	Allocation			**Max**			**Need**		
	X	**Y**	**Z**	**X**	**Y**	**Z**	**X**	**Y**	**Z**
P0	0	0	1	8	4	3	8	4	2
P1	3	2	0	6	2	0	3	0	0
P2	2	1	1	3	3	3	1	2	2

Currently, the system is in safe state.

Checking Whether REQ1 Can Be Entertained-

Need of P0 = [0 0 2]
Available = [3 2 2]
Clearly,

With the instances available currently, the requirement of REQ1 can be satisfied.
So, banker's algorithm assumes that the request REQ1 is entertained.
It then modifies its data structures as-

	Allocation			Max			Need		
	X	**Y**	**Z**	**X**	**Y**	**Z**	**X**	**Y**	**Z**
P0	0	0	**3**	8	4	3	8	4	**0**
P1	3	2	0	6	2	0	3	0	0
P2	2	1	1	3	3	3	1	2	2

Available

= [3 2 2] – [0 0 2]

= [3 2 0]

- Now, it follows the safety algorithm to check whether this resulting state is a safe state or not.
- If it is a safe state, then REQ1 can be permitted otherwise not.

- With the instances available currently, only the requirement of the process P1 can be satisfied.
- So, process P1 is allocated the requested resources.
- It completes its execution and then free up the instances of resources held by it.

Then-

Available

= [3 2 0] + [3 2 0]

= [6 4 0]

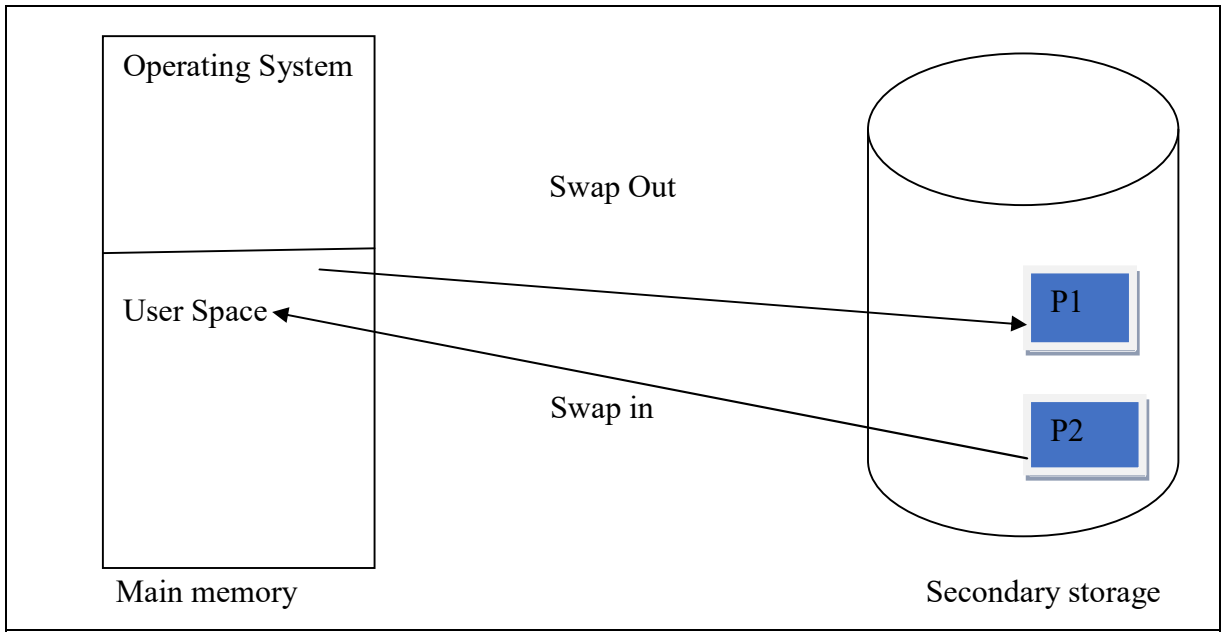
Now,

- It is not possible to entertain any process.
- The system has entered the deadlock state which is an unsafe state.

Thus, REQ1 will not be permitted.

4. (a) What is swapping? Write down its benefit. 4

Swapping is a mechanism in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk) and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory.



Benefits of Swapping :

Here, are major benefits/pros of swapping:

- It offers a higher degree of multiprogramming.
- Allows dynamic relocation. For example, if address binding at execution time is being used, then processes can be swap in different locations. Else in case of compile and load time bindings, processes should be moved to the same location.
- It helps to get better utilization of memory.
- Minimum wastage of CPU time on completion so it can easily be applied to a priority-based scheduling method to improve its performance.

(b) Write down the difference between Demand paging and segmentation .

Following are the important differences between Demand Paging and Segmentation −

Sr. No.	Key	Demand Paging	Segmentation
1	Definition	Paging is a memory management technique in which process address space is broken into blocks of the same size called pages.	On other hand segmentation is a memory management technique in which each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions.
2	Block Size	As mentioned above in Pagination process address space is broken into fixed sized blocks which are called as pages. So block size is fixed in case of Pagination.	On other hand in Segmentation process address space is broken in varying sized blocks which are called as sections. So block size is not fixed in case of Segmentation.
3	Block size dependency	In Pagination the size of blocks is dependent on system memory and gets assigned accordingly.	In Segmentation the size is not dependent on system memory and is all up to user's choice that of what size blocks are needed.
4	Performance	In context of performance Pagination is faster as compared to Segmentation.	Segmentation is slower in speed as compared to Pagination.
5	Data Load	In case of Pagination pages get loaded in main memory at runtime when user demands it.	In case of Segmentation all the sections get loaded at the time of compilation.
6	Data Record	In case of Pagination Page map table in demand paging manages record of	In case of Segmentation, Segment map table in segmentation demonstrates every segment address in the

Sr. No.	Key	Demand Paging	Segmentation
		pages in memory.	memory.

(c) Consider a single level paging scheme. The virtual address space is 16 GB and page table entry size is 4 bytes. What is the minimum page size possible such that the entire page table fits well in one page?

For page table, to fit well in one page, we must have-

Page table size <= Page size

Let page size = B bytes.

Number of Pages of Process-

Number of pages the process is divided
= Process size / Page size
= 16 GB / B bytes
= 2^{34} / B

Page Table Size-

Page table size
= Number of entries in the page table x Page table entry size
= Number of pages the process is divided x Page table entry size
= (2^{34} / B) x 4 bytes
= (2^{36} / B) bytes

Now,

According to the above condition, we must have-

$(2^{36} / B)$ bytes <= B bytes

$B^2 >= 2^{36}$

$B >= 2^{18}$

Thus, minimum page size possible = 2^{18} bytes or 256 KB.

5. (a) What is buffering ?

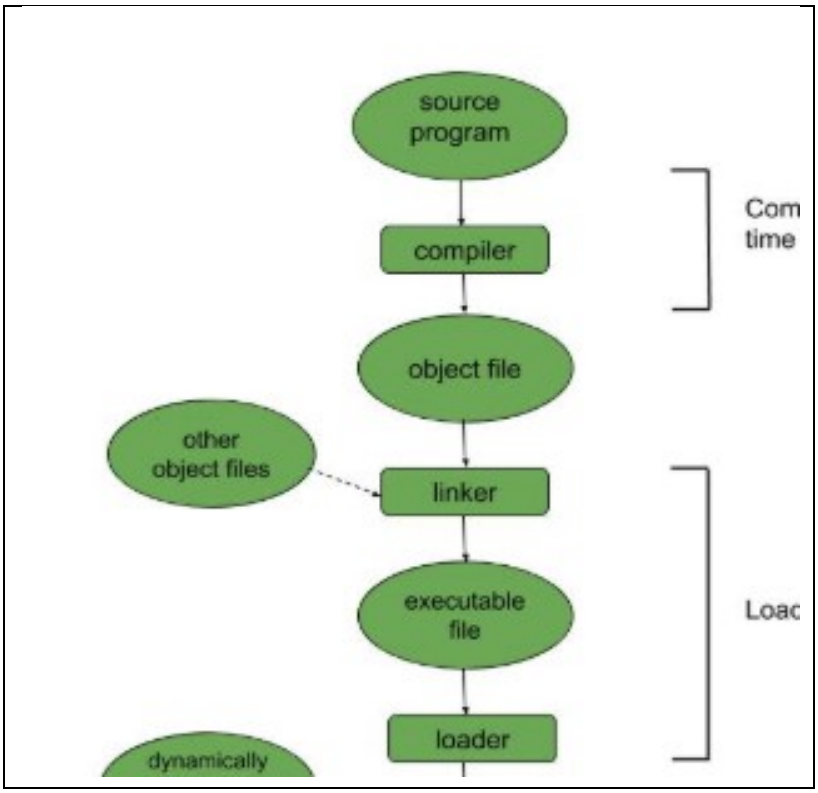
A buffer contains data that is stored for a short amount of time, typically in the computer's memory (RAM). The purpose of a buffer is to hold data right before it is used. For example, when you download an audio or video file from the Internet, it may load the first 20% of it into a buffer and then begin to play. While the clip plays back, the computer continually downloads the rest of the clip and stores it in the buffer. Because the clip is being played from the buffer, not directly from the Internet, there is less of a chance that the audio or video will stall or skip when there is network congestion.

(b) Explain the multistep processing of a user program with a neat block diagram.

Ans:

Program is a sequence of instructions written by the user that instruct the computer to perform the task of solving some problem. The program resides on the disk as a binary executable file. In order to run the program, it must be brought from the disk into the main memory . During the execution, it accesses data and instructions from the memory and once the execution is completed, the process is terminated and the memory is reclaimed for use by another process.

The addresses in the source code are generally symbolic (like the variable count). The compiler binds these addresses to relocatable addresses and these are bonded by the linker or loader to the absolute addresses (Binding – mapping from one address space to another). The binding of instructions and data to the memory addresses can be done at any of the following steps of program execution:



Compile Time:
If we initially know where the process is residing in the memory at the time of compilation, then absolute code can be generated. If the starting location changes, then the code has to be recompiled.

Load Time:
If the memory address where the process resides is not known at the compile time, then the compiler must generate relocatable code (does not have a static memory address for running). If the starting address changes, then the program must be reloaded to incorporate this value.

Execution Time:
If the process can be moved from one segment to another during its execution time, then binding must be delayed till execution time. Special hardware is required for this type of binding

(c) Consider a reference string: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2. the number of frames in the memory is 3. Find out the number of page faults respective to:

4. Optimal Page Replacement Algorithm
5. FIFO Page Replacement Algorithm
6. LRU Page Replacement Algorithm

Optimal Page Replacement Algorithm

Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	2	2	2
Frame 2		7	7	7	7	7	7	7	7	7
Frame 1	4	4	4	1	1	1	1	1	1	1
Miss/Hit	Miss	Miss	Miss	Miss	Hit	Hit	Hit	Miss	Hit	Hit

Number of Page Faults in Optimal Page Replacement Algorithm = 5

LRU Page Replacement Algorithm

Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	6	7	7
Frame 2		7	7	7	7	7	7	2	2	2
Frame 1	4	4	4	1	1	1	1	1	1	1
Miss/Hit	Miss	Miss	Miss	Miss	Hit	Hit	Hit	Miss	Miss	Hit

Number of Page Faults in LRU = 6

FIFO Page Replacement Algorithm

Request	4	7	6	1	7	6	1	2	7	2
Frame 3			6	6	6	6	6	6	7	7
Frame 2		7	7	7	7	7	7	2	2	2
Frame 1	4	4	4	1	1	1	1	1	1	1
Miss/Hit	Miss	Miss	Miss	Miss	Hit	Hit	Hit	Miss	Miss	Hit

Number of Page Faults in FIFO = 6

6 (a) What is SAN and how does it differ from NAS.

A storage area network (SAN) is a dedicated high-speed network or subnetwork that interconnects and presents shared pools of storage devices to multiple servers. A SAN moves storage resources off the common user network and reorganizes them into an independent, high-performance network. This enables each server to access shared storage as if it were a drive directly attached to the server. When a host wants to access a storage device on the SAN, it sends out a block-based access request for the storage device.

The way SAN differ from NAS is given below:

SAN and network-attached storage (NAS) are both network-based storage solutions. A SAN typically uses Fibre Channel connectivity, while NAS typically ties into to the network through a standard Ethernet connection. A SAN stores data at the block level, while NAS accesses data as files. To a client OS, a SAN typically appears as a disk and exists as its own separate network of storage devices, while NAS appears as a file server.

SAN is associated with structured workloads such as databases, while NAS is generally associated with unstructured data such as video and medical images. "Most organizations have both NAS and SAN deployed in some capacity, and often the decision is based on the workload or application," Sinclair says.

(b) How many ways file can be accessed?

The way that files are accessed and read into memory is determined by Access methods. Usually a single access method is supported by systems while there are OS's that support multiple access methods.

1. Sequential Access

- Data is accessed one record right after another is an order.
- Read command cause a pointer to be moved ahead by one.
- Write command allocate space for the record and move the pointer to the new End Of File.
- Such a method is reasonable for tape.

2. Direct Access

- This method is useful for disks.
- The file is viewed as a numbered sequence of blocks or records.

- There are no restrictions on which blocks are read/written, it can be dobe in any order.
- User now says "read n" rather than "read next".
- "n" is a number relative to the beginning of file, not relative to an absolute physical disk location.

3. Indexed Sequential Access

- It is built on top of Sequential access.
- It uses an Index to control the pointer while accessing files

(c) As a ICT student obviously it's important to do programming . Here we give you a task that you should write down a program to find the number of page faults . you can choose any programming language.

```
#include <bits/stdc++.h>

using namespace std;

void pageFault(int frame_size, int* ref, int len)
{
	// To dynamically allocate an array,
	// it represents the page frames.
	int* arr = new int[frame_size];

	// To initialize the array
	for (int i = 0; i < frame_size; i++) {
		arr[i] = -1;
	}

	// To count page faults
	int cnt = 0;
	int start = 0;
	int flag;
	int elm;

	for (int i = 0; i < len; i++) {
		elm = ref[i];
		// Linear search to find if the page exists
		flag = 0;
		for (int j = 0; j < frame_size; j++) {
```

```
                                        if (elm == arr[j]) {
                                                flag = 1;
                                                break;
                                        }
                                }
        // If the page doesn't exist it is inserted,
        // count is incremented
        if (flag == 0) {
                if (start < frame_size) {
                        arr[start] = elm;
                        start++;
                }
                else if (start == frame_size) {
                        arr[0] = elm;
                        start = 1;
                }
                cnt++;
        }
}
cout << "When the number of frames are: " << frame_size << ", ";
cout << "the number of page faults is : " << cnt << endl;
}

int main()
{
// Reference array
int ref[] = { 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 };
int len = sizeof(ref) / sizeof(ref[0]);
// The frame size
int frame_size = 3;

pageFault(frame_size, ref, len);

// Increase value of frame size
frame_size = 4;

// The page fault increases
// even after increasing the
// the number of frames.
// This is Belady's Anomaly
pageFault(frame_size, ref, len);
}
```

Output:

When the number of frames are: 3, the number of page faults is : 9
When the number of frames are: 4, the number of page faults is : 10

7. (a) What is file system ?

A file is a collection of correlated information which is recorded on secondary or non-volatile storage like magnetic disks, optical disks, and tapes. It is a method of data collection that is used as a medium for giving input and receiving output from that program.

In general, a file is a sequence of bits, bytes, or records whose meaning is defined by the file creator and user. Every File has a logical location where they are located for storage and retrieval.

Commonly used term used in File System are given below :

Field:

This element stores a single value, which can be static or variable length.

DATABASE:

Collection of related data is called a database. Relationships among elements of data are explicit.

FILES:

Files is the collection of similar record which is treated as a single entity.

RECORD:

A Record type is a complex data type that allows the programmer to create a new data type with the desired column structure. Its groups one or more columns to form a new data type. These columns will have their own names and data type.

(b) Write down the advantages and disadvantages of various disk algorithm.

Advantages and Disadvantages of various Disk scheduling algorithms

1. First Come First Serve (FCFS) :
Advantages –

1. First Come First Serve algorithm has a very simple logic, it executes the process requests one by one in the sequence they arrive.
2. Thus, First Come First Serve is very simple and easy to understand and implement.
3. In FCFS eventually, each and every process gets a chance to execute, so no starvation occur.

Disadvantages –

1. This scheduling algorithm is nonpreemptive, which means the process can't be stopped in middle of execution and will run it's full course.
2. FCFS being a nonpreemptive scheduling algorithm, the short processes which are at the back of the queue have to wait for the long process at the front to finish
3. The throughput of FCFS is not very efficient.
4. FCFS is implemented on small systems only where input-output efficiency is not of utmost importance.

2. Shortest Seek Time First (SSTF) :

Advantages –

1. The total seek time is reduced compared to First Come First Serve.
2. SSTF improves and increases throughput.
3. Less average waiting time and response time in SSTF.

Disadvantages –

1. In SSTF there is an overhead of finding out the closest request.
2. Starvation may occur for requests far from head.
3. In SSTF high variance is present in response time and waiting time.
4. Frequent switching of the Head's direction slows the algorithm.

3. SCAN :

Advantages –

1. Scan scheduling algorithm is simple and easy to understand and implement.
2. Starvation is avoided in SCAN algorithm.
3. Low variance Occurs in waiting time and response time.

Disadvantages –

1. Long waiting time occurs for the cylinders which are just visited by the head.
2. In SCAN the head moves till the end of the disk despite the absence of requests to be serviced.

4. C-SCAN :

Advantages –

1. C-SCAN Algorithm is the successor and the improved version of the SCAN scheduling Algorithm.
2. The Head move from one end to the other of the disk while serving all the requests in between.
3. The waiting time for the cylinders which were just visited by the head is reduced in C-SCAN compared to the SCAN Algorithm.
4. Uniform waiting time is provided.
5. Better response time is provided.

Disadvantages –

1. More seek movements are caused in C-SCAN compared to SCAN Algorithm.

2. In C-SCAN even if there are no requests left to be serviced the Head will still travel to the end of the disk unlike SCAN algorithm.

5. LOOK :

Advantages –

1. If there are no requests left to be services the Head will not move to the end of the disk unlike SCAN algorithm.
2. Better performance is provided compared to SCAN Algorithm.
3. Starvation is avoided in LOOK scheduling algorithm.
4. Low variance is provided in waiting time and response time.

Disadvantages –

1. Overhead of finding the end requests is present.
2. Cylinders which are just visited by Head have to wait for long time.

(c) Explain the following DML command with syntax :

DELETE,DROP,TRUNCATE

1. DELETE :

Basically, it is a Data Manipulation Language Command (DML) . It is use to delete the one or more tuples of a table. With the help of "DELETE" command we can either delete all the rows in one go or can delete row one by one. i.e., we can use it as per the requirement or the condition using Where clause. It is comparatively slower than TRUNCATE cmd.

- **SYNTAX –**
  If we want to delete all the rows of the table:
  ```
  DELETE from ;
  ```
- **SYNTAX –**
  If we want to delete the row of the table as per the condition then we use WHERE clause,
  ```
  DELETE from  WHERE  ;
  ```

Note –

Here we can use the "ROLLBACK" command to restore the tuple.

2. DROP :

It is a Data Definition Language Command (DDL). It is use to drop the whole table. With the help of "DROP" command we can drop (delete) the whole structure in one go i.e. it removes the named elements of the schema. By using this command the existence of the whole table is finished or say lost.

- **SYNTAX –**
  If we want to drop the table:
  ```
  DROP table ;
  ```

3. TRUNCATE :

It is also a Data Definition Language Command (DDL). It is use to delete all the rows of a relation (table) in one go. With the help of "TRUNCATE" command we can't delete the single row as here WHERE clause is not used. By using this command the existence of all the rows of the table is lost. It is comparatively faster than delete command as it deletes all the rows fastly.

- **SYNTAX –**
  If we want to use truncate :
  ```
  TRUNCATE  ;
  ```

8. (a) What is Swapping?

Swapping is a mechanism in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk) and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory.

(b) What if a system runs out of swap space?

When the operating system is out of RAM and has no swap, it discards clean pages. It cannot discard dirty pages because it would have to write them somewhere first. This causes thrashing and poor performance if there is insufficient RAM to hold the working set. That's one of the main reasons you really want swap -- so the operating system can make a better decision about what pages to evict.

With no swap, the system will run out of virtual memory (strictly speaking, RAM+swap) as soon as it has no more clean pages to evict. Then it will have to kill processes.

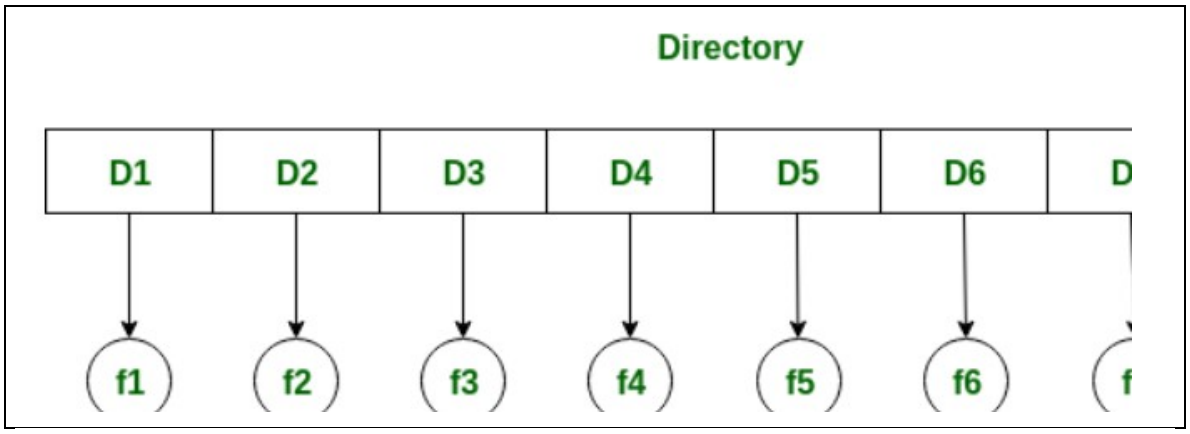
Running out of RAM is completely normal. It's just a negative spin on *using* RAM. Not running out of RAM could equally well be described as "wasting RAM". Once all RAM is in use, the operating system makes intelligent decisions about what to keep in RAM and what not to. Without any swap, it has fewer choices.

With or without swap, when evicting pages isn't sufficient, the operating system will start by refusing to permit operations that require memory (such as `mmap` and `fork`) to succeed. However, sometimes that's not enough and processes have to be killed.

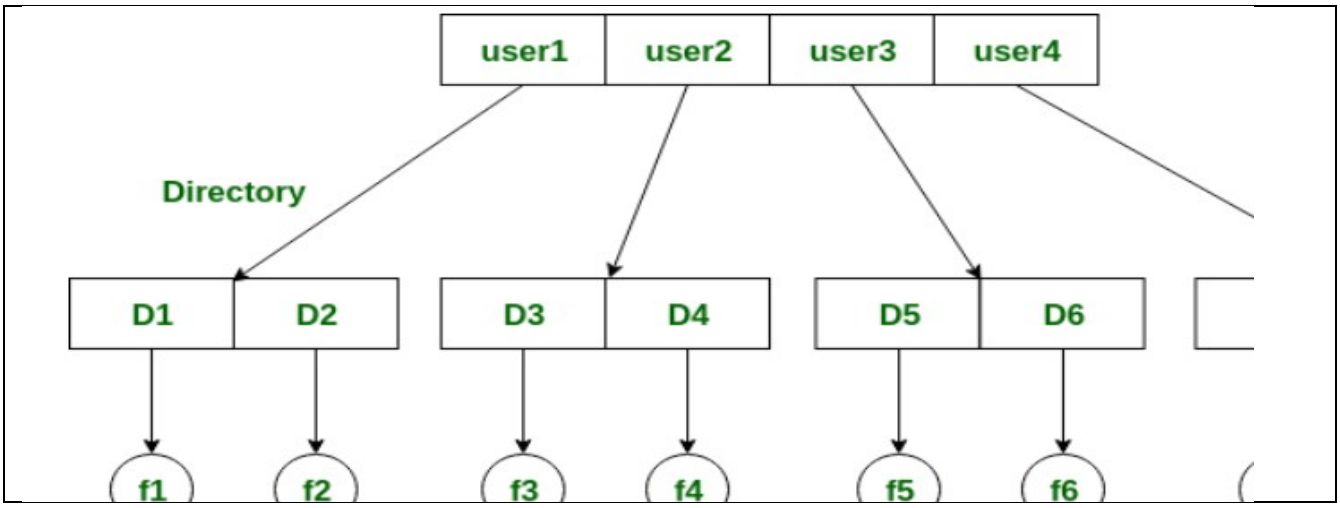
(c) Explain the structure of Directory in Operating system with diagram.

A **directory** is a container that is used to contain folders and file. It organizes files and folders into a hierarchical manner.
There are several logical structures of a directory, these are given below.

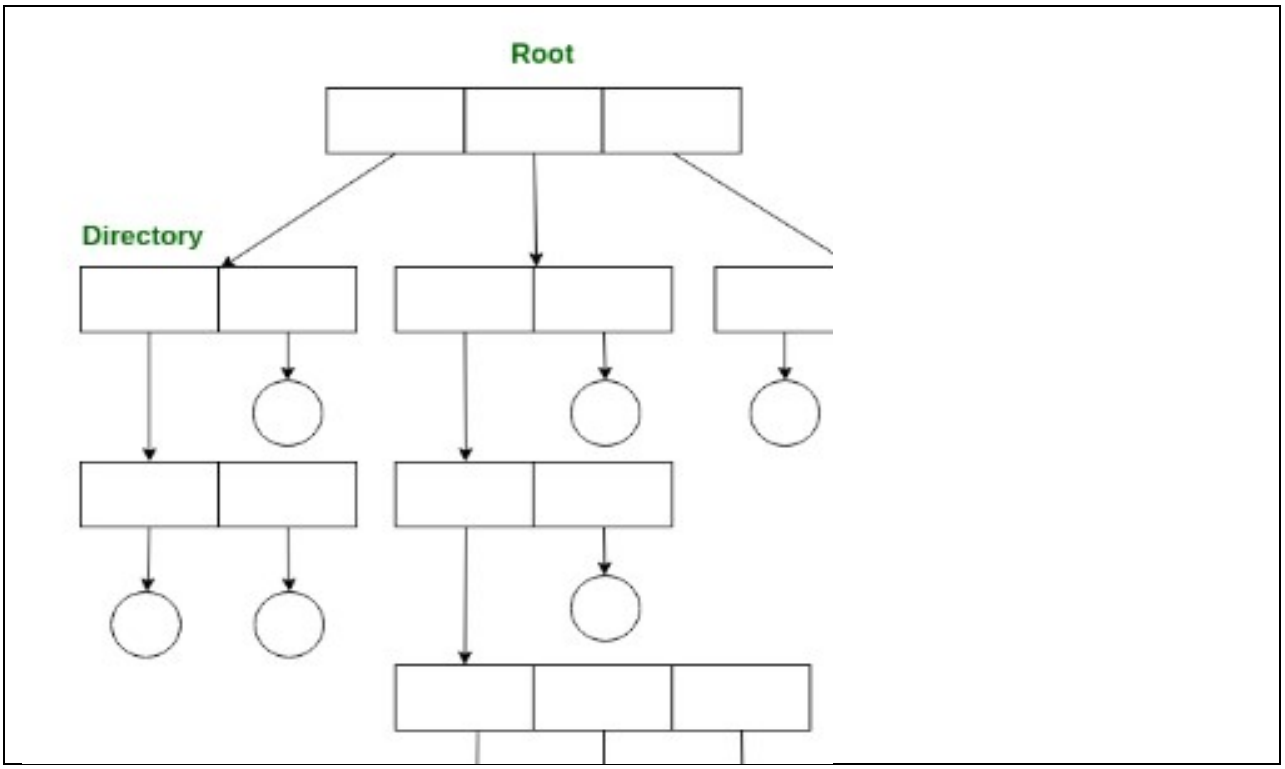
Single-level directory –
Single level directory is simplest directory structure. In it all files are contained in same directory which make it easy to support and understand.
A single level directory has a significant limitation, however, when the number of files increases or when the system has more than one user. Since all the files are in the same directory, they must have the unique name . if two users call their dataset test, then the unique name rule violated.



Two-level directory –
As we have seen, a single level directory often leads to confusion of files names among different users. the solution to this problem is to create a separate directory for each user.
In the two-level directory structure, each user has there own *user files directory (UFD)*. The UFDs has similar structures, but each lists only the files of a single user.
system's *master file directory (MFD)* is searches whenever a new user id=s logged in. The MFD is indexed by username or account number, and each entry points to the UFD for that user.



Tree-structured directory –
Once we have seen a two-level directory as a tree of height 2, the natural generalization is to extend the directory structure to a tree of arbitrary height.
This generalization allows the user to create there own subdirectories and to organize on their files accordingly.



A tree structure is the most common directory structure. The tree has a root directory, and every file in the system have a unique path.

Acyclic graph directory –
An acyclic graph is a graph with no cycle and allows to share subdirectories and files. The same file or subdirectories may be in two different directories. It is a natural generalization of the tree-structured directory.
It is used in the situation like when two programmers are working on a joint project and they need to access files. The associated files are stored in a subdirectory, separating them from other projects and files of other programmers, since they are working on a joint project so they want the subdirectories to be into their own directories. The common subdirectories should be shared. So here we use Acyclic directories.

It is the point to note that shared file is not the same as copy file . If any programmer makes some changes in the subdirectory it will reflect in both subdirectories.