Q → 4(b) Explain in detail about the Threads and their management?

Ans → 4(b) A Thread is a light weight sub process. It is the basic unit of CPU utilization. It include Thread ID, Program Counter, Register set & stack. It share with other thread belonging to the same process its code section, data section and the other OS resources such as files. A traditional or heavy weight process has a single thread of control. If a process has multi-ple threads of a control it can do more than one task at the same time.

An application is typically implemented as seperate process with several threads of control A word processor may have a thread for displaying graphics, another thread for reading key strokes from the user and a third thread for performing spell checking in back ground and this is called Multithreading.

* Benefits of Multi threadings.

(a) Responsiveness ⇒ Multithreading and interactive application may allow the program to continue running even if part of it block or performing a lengthy operation. thereby increasing responsiveness to the user.

(b) Resource sharing ⇒ By default, thread share the memory and the resources of the process to which

they belong.
The benefits of code sharing is that it allow an application to have several different threads of activity are within the same address space.

(c) Economy ⇒ Allocating the memory and resources for the process creation is costly. Alternatively because thread share the resources of the process to which they belong, it is more economical to create and the context switch threads.

(d) Utilisation of Multiprocessor Architecture.
The benefits of multithreading canbe increase in the multiprocessor architecture, where each thread may be running in parallel on a different processor

* Types of Thread:
There are two types of thread.

(a) User Level Thread ⇒ User threads are supported above the Kernel and are implemented by thread library at User Kernel. The Library provide support for thread creation, scheduling and management with no support from Kernel. Because the kernel is unaware of User Level thread, all the thread creation and scheduling done in the User space without the need for kernel intervention. Therefore User Level thread are generally fast to create and manage they have drawbacks.

② Kernel Level Thread ⇒ These threads are supported strictly by OS. The Kernel perform thread creation, scheduling and management in the Kernel space Area.

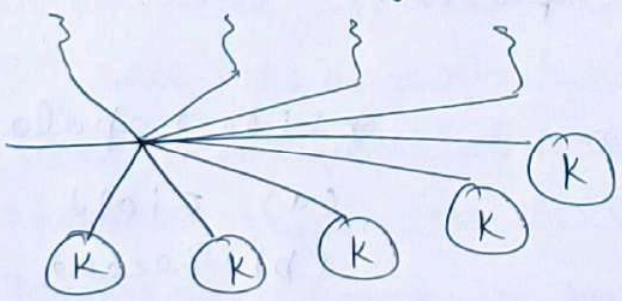
Because thread management is done by the OS. Kernel thread are generally slower to create and manage than the User level thread.

In multiprocessor environment, the Kernel can schedule threads on different processors.

Most OS includes, support Kernel threads windows 2000, windows NT.

→↑← Multi-threading Model ⇒ some OS provide a combined approach User Level threads and the Kernel level threads.

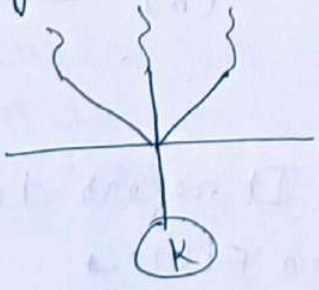
① Many to Many model



User Level Thread

Kernel Level Thread.

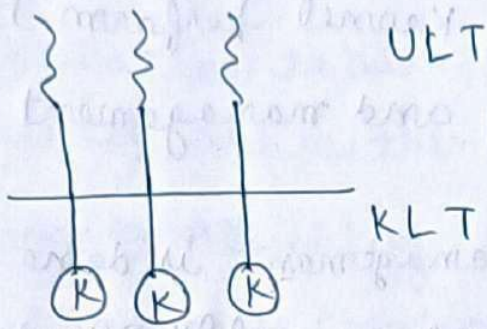
② Many to one model



ULT

KLT

③ One to One Model



question 5(a)

Explain about the concept of File Concept. Define in detail about the File organization & access mechanism?

Ans→ File Concept ⇒ A file is a named collection of data such as image, video, music, word file, presentation & document etc.

It has following properties such as:

(a) Long term existence.
(b) Sharable b/w process.
(c) Structure.

* file operation

(a) Create
(b) Delete
(c) Open
(d) Read
(e) write
(f) Truncate.

* file topologies.

(a) Field
(b) Record
(c) File
(d) Database.

* File organization ⇒ It refers to the way the data is stored in a File.
FO is very imp, because it determine the method of access, efficiency, flexibility and storage

device to use.

* Types of File organization: There are following types of File organization are in DBMS ever.

(a) Sequential File Organization ⇒ This is the most straight forward technique of file arrangement. Files are saved in this method in sequential order.

(b) Heap File Organization ⇒ It is most fundamental and basic type of organizational structure. It's based on data chunks. The record are inserted at the end of the file in the heap file organization.

(c) Hash File organization ⇒ The computation of the hash function on some of the fields of the records is used by Hash File organization.

(d) B+ File organization ⇒ The advanced way of an indexed sequential access mechanism is the B+ tree File organization.

(e) Indexed sequential Access method ⇒ It is an advanced sequential File organization method. Records are stored in the File using the primary key in this way.

(f) Cluster File organization ⇒ Cluster are created when two or more records are saved in the same file.

* File access mechanism ⇒ There are three ways to access a file in computer system

(a) Sequential access ⇒ It is the simplest access method. Information in the file is processed in order, one record after the other.

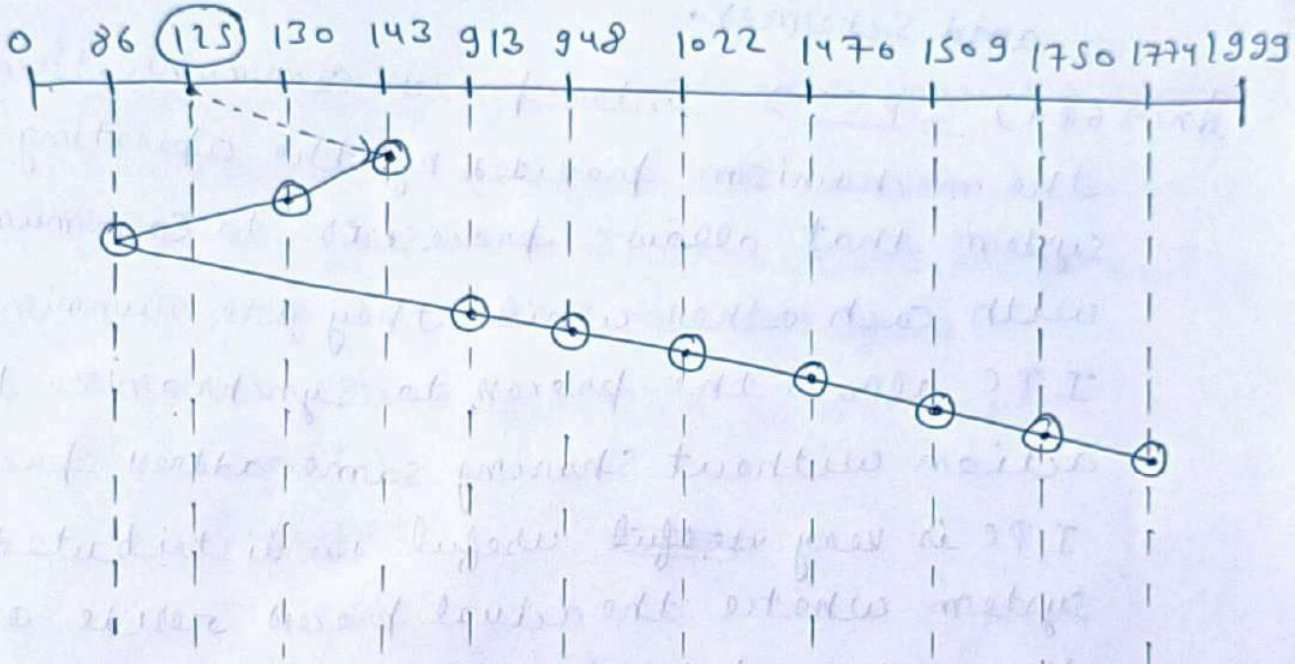
(b) Direct access ⇒ It is also known as relative access method. A filed length logical records that allows the program to read & write record rapidly.

(c) Index sequential method ⇒ It is the other method of accessing a file that is built on the top of the sequential access method.
These methods construct an index for the file.
* It is built on top of Sequential access.
* It Control the pointer by using index.

Q→5(b) A Hard disk having 2000 cylinders, numbered from 0 to 1999. the drive is currently serving the request at cylinder 143 and the previous request was at cylinder 125. The status of the queue is as follows 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130.
What is the total distance (in cylinders) that the disk arm moves to satisfy the entire pending request for each of the following disk Scheduling algorithms?
(i) SSTF
(ii) FCFS

Ans → 5(b) ⇒

(i) SSTF



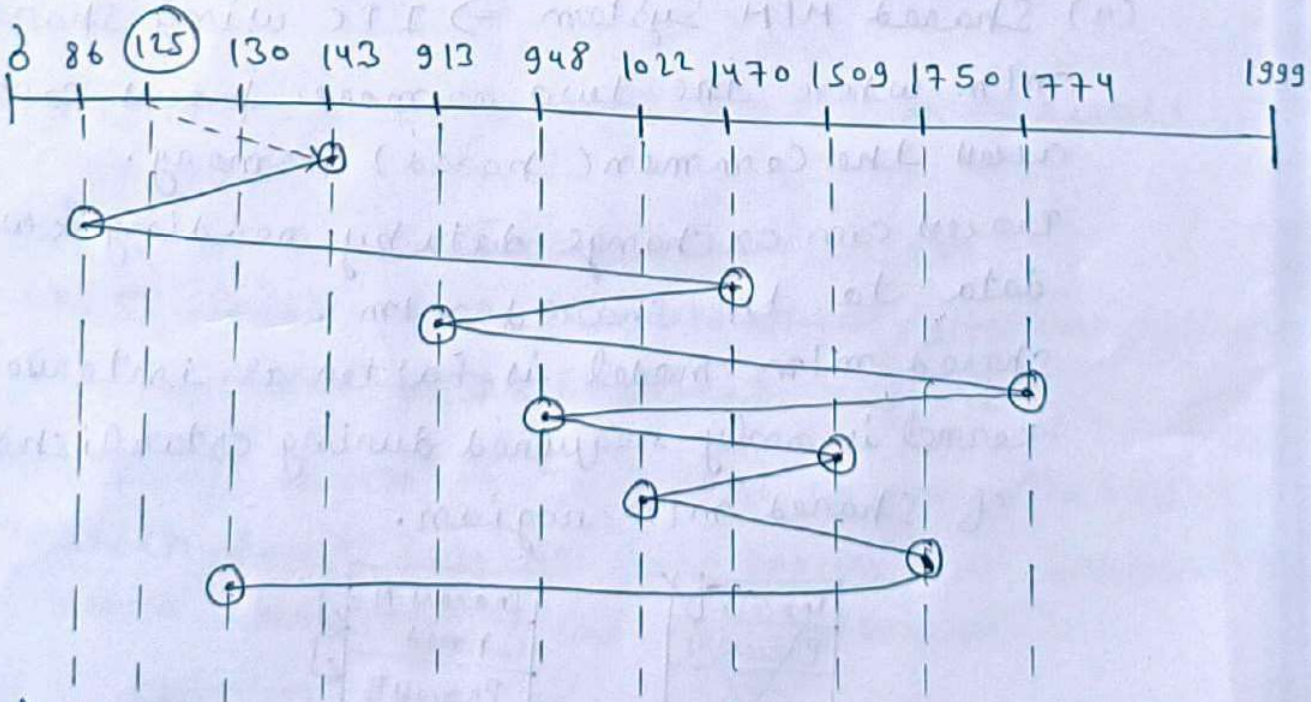
Total Distance = (143 − 86) + (1774 − 86)

= 57 + 1688

= **1745**

FCFS

(ii)



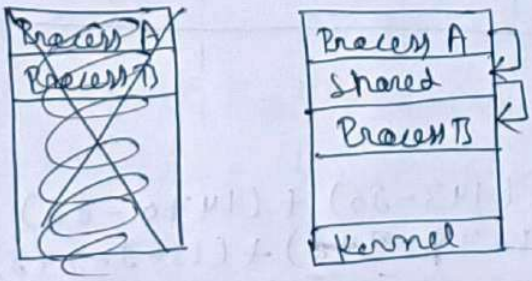
Total distance = (143−86) + (1470−86) + (1470−913) + (1774−913) + (1774−948) + (1509−948) + (1509−1022) + (1750−1022) + (1750−130) = **7081**

Q → 6(a) Explain in detail about the Inter Process communication model and Schemes.

Ans → 6(a) IPC ⇒ Inter process communication is the mechanism provided by the operating system that allows processes to communicate with each other while they are running. IPC allow the process to synchronize their action without sharing same address space. IPC is very ~~useful~~ useful in distributed system where the actual process reside on different computer & connected on a N/W.

* Model of IPC ⇒ There are mainly two fundamental model of IPC are :-

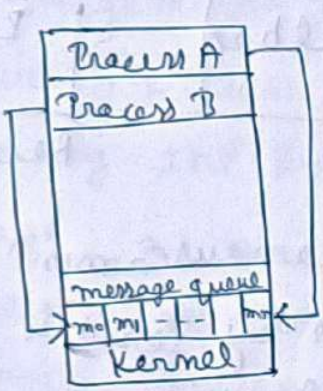
(a) Shared M/M System ⇒ IPC using shared m/m where the two or more process can access the common (shared) memory. Process can exchange data by reading & writing data to the shared m/m. Shared m/m model is faster as interrupt by Kernel is only required during establishment of shared m/m region.



Ex:- (a) Producer Consumer Problem.
(b) Reader writer Problem.

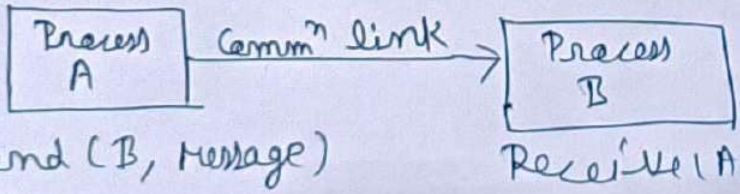
① Message Passing system ⇒ Message passing provides a mechanism to allow the process to communi-cate & synchronize their actions without sharing the same address space.

There are two operation of MPS are :-
(a) Send()
(b) Receive().

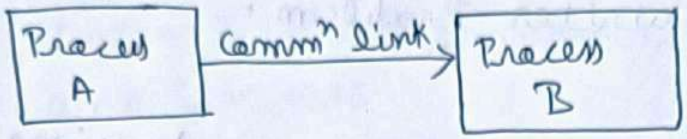


* Method to implement Send() or receive() operation
(a) Direct or Indirect Communication
(b) Synchronous or asynchronous communication.
(c) Automatic or Explicit buffering.

(a) Direct Comm^n ⇒ It is the communication which occur between two process the process only need a identity of each other process.



Symmetric Addressing.



Send (B, Message) Receive (id, Message)

(Asymmetric Addressing)

* Indirect communication ⇒ It is the Comm^n b/w two process through the mailbox, the process only need a unique id of each mailbox.

Send (M, Message) Receive (M, Message)

(b) Synchronous & Asynchronous Comm^n ⇒ Comm^n b/w the process take place by send & receive function.

(i) Blocking Send & Blocking Receive.
(ii) Non-Blocking Send & Blocking Receive.
(iii) Non-Blocking send & Non-Blocking Receive.

(c) Buffering ⇒ Buffer is used in direct & indirect Communication.

(i) Zero Capacity buffer.
(ii) Bounded Capacity buffer.
(iii) Unbounded Capacity buffer.

Q → 6(b) Explain in detail about the Monolithic and Microkernel systems?

Ans → 6(b) Monolithic Kernel ⇒ The monolithic Kernel manages the system's resource b/w the system application and the system H/w. Unlike the microkernel User and Kernel services are run in the same address space. It increases the Kernel size and also increases the size of OS. The monolithic Kernel offer CPU scheduling, device management, File management, memory management, process management and other OS service by the system calls.

* Advantage → (a) The monolithic Kernel runs quickly because of memory management, file management, process scheduling etc.

(b) All the components may interact directly with other and also with the Kernel.

* Disadvantage → (a) If the User needs to add a new service the User requires to modify the Complete OS.

(b) If any of the service fails, the entire system fails.

* Microkernel system ⇒ Microkernel is a type of kernel that permits the customization of the OS. It is privileged and provides low-level address space management as well as IPC. Furthermore, OS func like the VM manager, file system and CPU scheduler are built on top of the microkernel.

* Advantage ⇒ (a) These are modular, and several modules may be modified, reloaded, replaced without modifying the kernel.

(b) The architecture of microkernel is small and isolated but it may work better.

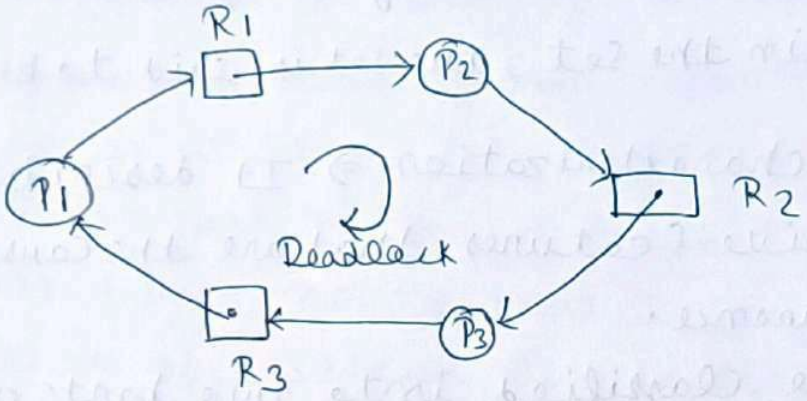
* Disadvantage → (a) A context switch is required in the microkernel when the drivers are run as processes.

(b) The microkernel system Performance might be Variable & Cause issues.

Q → 7(a) Explain in detail about Deadlock system model and Deadlock characterization.

Ans → 7(a) Deadlock ⇒ It is a situation where a group of processes are permanently blocked as a result each process having acquired a subset of the resources needed for its completion and waiting for release of remaining resource held by other process. So that it can make

impossible any of the process to be proceed Completely. This situation is called Deadlock Condition.



* System Model →

* For the purposes of deadlock discussion, a system can be modeled as a collection of limited resources that can be divided into different categories and allocated to a variety of processes, each with different requirements.
* Memory, printers, CPUs, open files, tape files, CD-ROMs and other resource are examples of resource categories.
* By definition, all resources within a category are equivalent and any of the resources within that category can equally satisfy a request from that category.
* Some categories may only have one resource.
* The Kernel keeps track of which resources are free and which are allocated, to which process they are allocated and a queue of processes waiting for this resource to become available

for all Kernel managed resources. Mutex can be used to control application managed resources.

* when every process in a set is waiting for a resource that is currently assigned to another process in the set, the set is said to be deadlocked.

* Deadlock characterization ⇒ It describe the distinctive features that are the cause of Deadlock occurence.

It can be classified into two parts are :-

(a) Deadlock Prerequisites ⇒

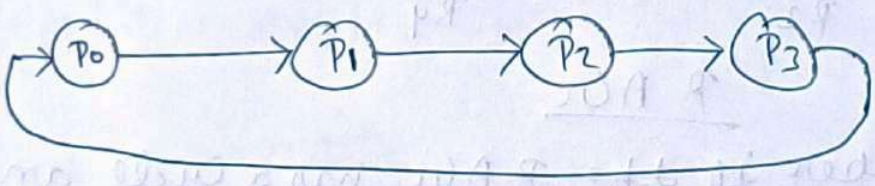
(i) Mutual exclusion ⇒ In a multiprogramming environment, there may be several processes requesting the same resource at a time. The mutual exclusion condition, allow only a single process to access the resource at a time. While the other processes requesting the same resource must wait and delay their execution until it has been released.

(ii) Hold & wait ⇒ The Hold and wait condition simply means that the process must be holding access to one resource and must be waiting to get hold of other resource that have been acquired by the other processes.

(iii) No preemption ⇒ A process acquiring a resource, can not be preempted in b/w, to release the acquired resource. Instead, the process must

Voluntarily release the resource it has acquired when the task of the process has been completed.

(iv) Circular wait ⇒ A condition is arise in which Po process is waiting for resource from P1, P1 process is waiting for resource from P2, P2 process is waiting for resource from P3 and so on. the last Pn process is waiting for resource from Po. So that It can create a condition of deadlock due to circular wait -



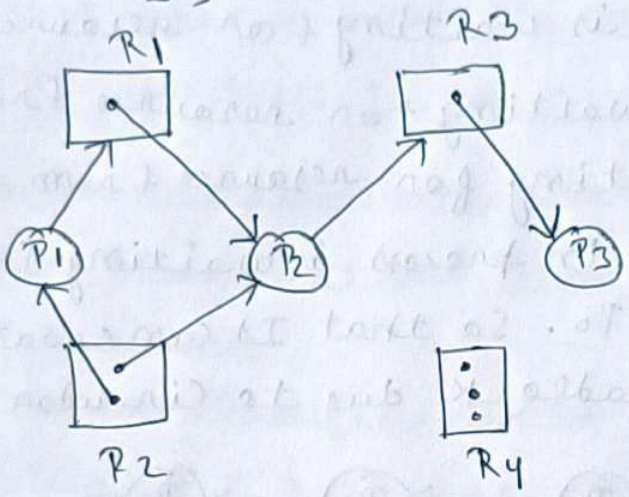
* System Resource Allocation ⇒ The system reallocation graph is a directed graph that briefs you about the deadlock more precisely. when a process requests for a resource it denoted by the request edge in the RAG In the graph resources are denoted by the Rectangle and process are denoted by Circle. If a resource has multiple instances then it is denoted by the dots inside the rectangle.

To understand deadlock we let us take an example. Consider we have following set of nodes and edges.

1. There are three active Processes P = { P1, P2, P3 }
2. There are four resources R = { R1, R2, R3, R4 }

3. The set of request edge and assignment edges we have

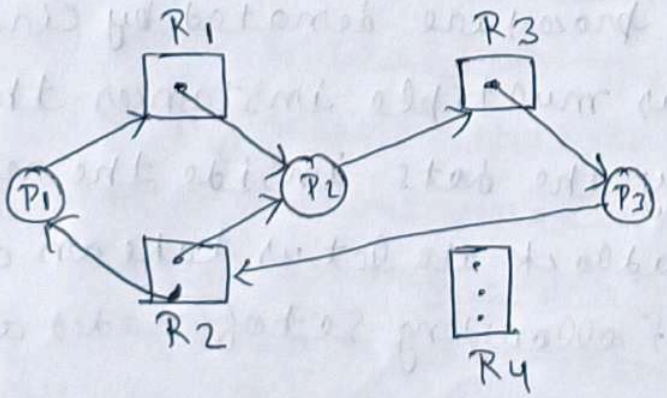
$E = \{ P_1 \to R_1, P_2 \to R_3, R_1 \to P_2, R_2 \to P_2, R_2 \to P_1, R_3 \to P_3 \}$



RAG

Remember if the RAG has a cycle and every resource has a single instance then it implies that a deadlock has occured.

* In this case, you will observe that there are two cycles in the RAG.
  - $P_1 \to R_1 \to P_2 \to R_3 \to P_3 \to R_2 \to P_1$
  - $P_2 \to R_3 \to P_3 \to R_2 \to P_2$



RAG $\subset$ Deadlock

0→7(b) Illustrate the following Page replacement algorithm

(i) FIFO

(ii) Optimal Page Replacement.

Use the refrence string 7,0,1,2,0,3,0,4,2,3, 0,3,2,1,2,0,1,7,0,1 for a m/m with three frames.

Ans→ 7(b) →(i) FIFO

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7*	7	7	2*	2*	2	2	4*	4	4	0*	0*	0*	0	0	0	0	7*	7	7
	0*	0	0	0	3*	3	3	2*	2	2	2	2	1*	1	1	1	1	0*	0
		1*	1	1	1	0*	0	0	3*	3	3	3	3	2*	2*	2*	2	2	1*
				H							H	H			H	H			

No. of Hits → 5 , Hit ratio = 5/20 = 0.25

No. of miss → 15 , Miss ratio = 15/20 = 0.75

(ii) Optimal Page Replacement,

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7
	0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0	0	0
		1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
				H		H		H	H		H	H		H	H	H		H	H

No. of hits = 11 , Hit ratio = 11/20 = 0.55

No. of miss = 9 , Miss ratio = 9/20 = 0.45