Q.2 Explain various page replacement using example.

Ans → 0 Various page replacement Algorithm :—

(i) **Optimal Page Replacement Algorithms** ⇒ This algo. replaces page that will not be used for longest period of time.

Eg- 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 no. of frames = 3

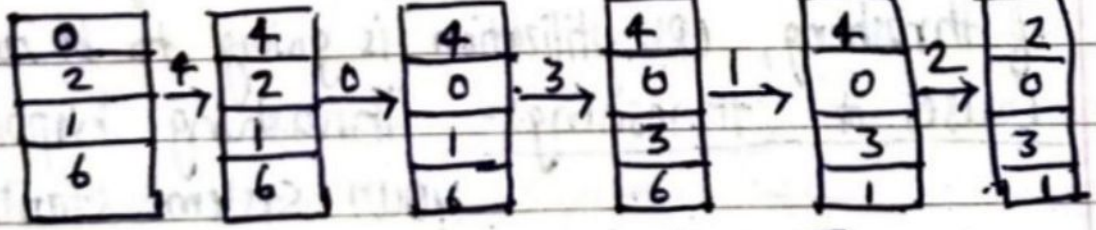
7	7	7	2	2	2	2	2	7	
	0	0	0	0	4	0	0	0	
		1	1	3	3	3	1	1	

No. of page faults = 9

(ii) **FIFO Page Replacement Algorithm** ⇒ The oldest page, which has spent the longest time in memory is chosen & replaced. A page is insested at REAR end of queue & is replaced at FRONT of queue.

Eg- String :- 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Misses :- x x x x x x x x x



(iii) **LRU (Least Recently Used)** ⇒ This algo. replaces the pages that has not been used for the longest period of time. It is based on observation that pages that have not been used for long time will probably remain unused for longest time and are to be replaced.

BANKER's Algorithm :—

Q. 5 processes P_0 through P_4 : 3 resource types
A (10 instances), B (5 instances) and C (7 instances)

	Allocation	Max	Available	Need
	A B C	A B C	A B C	
P_0	0 1 0	7 5 3	~~3 3 2~~	7 4 3
P_1	2 0 0	3 2 2	~~5 3 2~~	1 2 2
P_2	3 0 2	9 0 2	~~7 4 3~~	6 0 0
P_3	2 1 1	2 2 2	~~7 4 5~~	0 1 1
P_4	0 0 2	4 3 3	~~7 5 5~~ 10 5 7	4 3 1

(Need = Max - Allocation)

Safe sequence $\langle P_1\ P_3\ P_4\ P_0\ P_2 \rangle$

DATE : / /
PAGE NO. :

Banker's Algorithm :— ~~Safety Algorithm~~

① work = available = 3 3 2

② $Need_i \leq work \Rightarrow work = work + allocation$

Finish [F F F F F] (0 1 2 3 4)

P_2 : 7 4 3 ≤ 3 3 2 X
P_1 : 1 2 2 ≤ 3 3 2 ✓ $W = W + alloc. = 332 + 200 = 532$
P_2 : 6 0 0 ≤ 5 3 2 X
P_3 : 0 1 1 ≤ 5 3 2 ✓ $W = W + alloc. = 532 + 211 = 743$
P_4 : 4 3 1 ≤ 7 4 3 ✓ $W = W + alloc. = 743 + 002 = 745$
P_0 : 7 4 3 ≤ 7 4 5 ✓ $W = W + alloc = 745 + 010 = 755$
P_2 : 6 0 0 ≤ 7 5 5 ✓ $W = W + alloc. = 755 + 302 = 1057$ (A B C)

Finish [T T T T T] (0 1 2 3 4)

Safe Sequence = $\langle P_1\ P_3\ P_4\ P_0\ P_2 \rangle$

Resource Request Algorithm :—

If P_1 requests (1, 0, 2) determine if it can be granted immediately

$P_1 \longrightarrow R(1\,0\,2)$

$Need(P_1) = Max - alloc = 322 - 200 = 122$

① Request ≤ need ⇒ 1 0 2 ≤ 1 2 2 ✓ go to ②
② Request ≤ Available ⇒ 1 0 2 ≤ 3 3 2 ✓ go to ③
③ available = available - request
= 3 3 2 - 1 0 2 = 2 3 0
Allocation = Allo. + req. = 2 0 0 + 1 0 2 = 3 0 2
Need = Need - Request = 1 2 2 - 1 0 2 = 0 2 0

typical 10,000 rpm.

On EIDE & SCSI drives the disk controller is part of drive itself. It controls the drive's servomotors & translates the fluctuating voltages from head into digital data for CPU.

Q.5 Given queue: 95, 180, 34, 119, 11, 123, 62, 64 with R/W head initially at track 50 & trail track being at 199 to calculate by SSTF & Scan & look algorithm.

Ans -

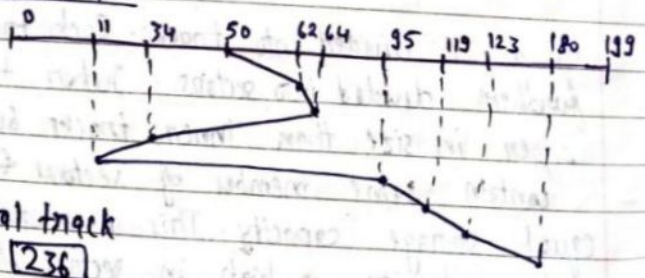
-: 95, 180, 34, 119, 11, 123, 62, 64

R/W on track = 50

tail track = 199

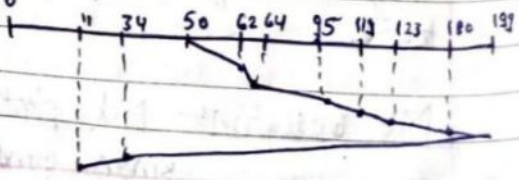
Er Sahil Ka Gyan

For SSTF:-



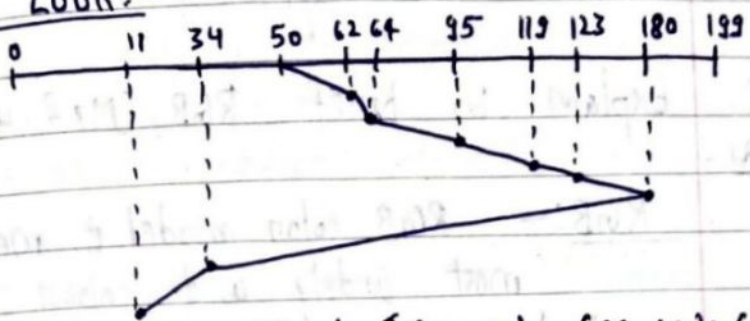
Total track = 236

For SCAN:-



Total track = 360

For LOOK:-

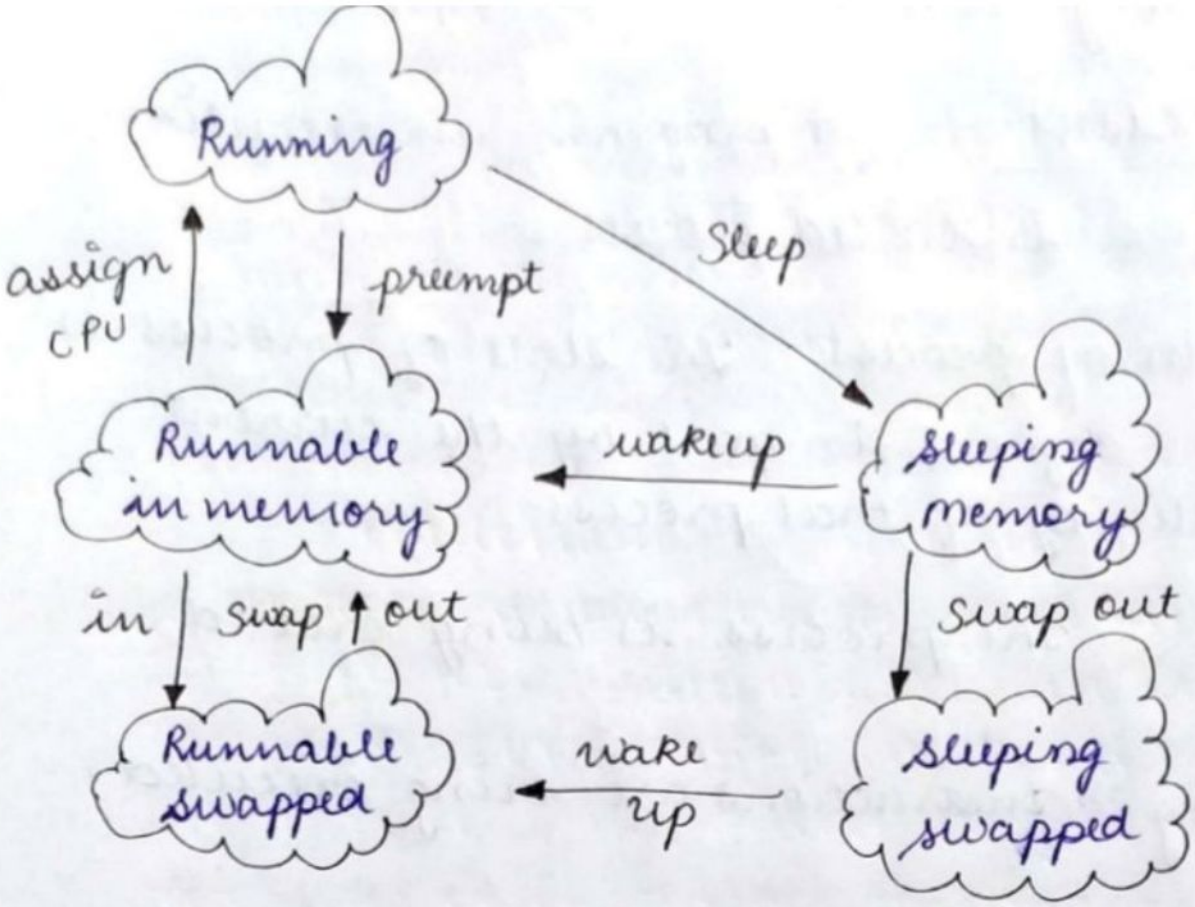


Total head Movement = {(62-50) + (64-62) + (95 + (119-95) + (123-119) + (180 + (180-34) + (34-11)}

= (12 + 2 + 31 + 24 + 4 + 57 + 146 + 23)

= 299

Er Sahil Ka Gyan



Process State Transition Diagram

* Process Control Block

Each process is represented in the operating system by a process control block (PCB) — also called a Task Control Block. It contains many pieces of information associated with a specific process, including these:-

1. Process state
2. Program counter
3. CPU Registers
4. CPU-Scheduling information
5. Memory Management information

}
}

The above code can be summarised as:-

- while() is used to produce data, again and again, if it wishes to produce, again and again.
- produce() function is called to produce data by the producer.
- wait(E) will reduce the value of semaphore variable "E" by one i.e when the producer produces something then there is a decrease in the value of the empty space in the buffer. If the buffer is full i.e the value of the semaphore variable "E" is "0", then the program will stop its execution and production will be done
- wait(S) is used to set the variables "S" to "0" so that no other process can enter into the critical section.
- append() is used to append the newly produced data in the buffer
- signal(S) is used to set the semaphore variable "S" to "0" so that other processes can come into the critical section now because the

by programming languages to achieve mutual exclusion between processes.

for ex:- JAVA synchronized methods. JAVA provides wait() and notify() construct.

1) It is a conditional variables and procedures combined together in a special kind of module or a package.

2) The processes running outside the monitor can't access the internal variable of the monitor but can procedure of the ~~no mirror~~ monitor.

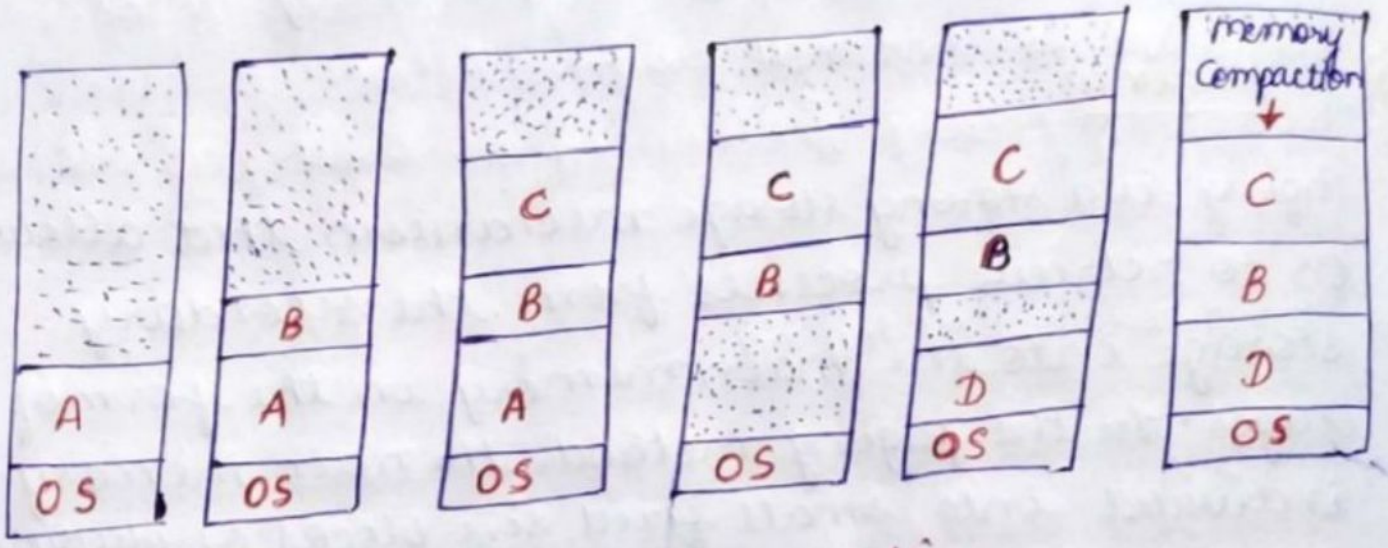
3) Only one process at a time can execute code inside monitors.

Syntax :

```
Monitor Demo // Name of Monitor
{
variables;
conditional variable;
procedure p1 {....}
procedure p2 {....}
}
```

↳ Internal fragmentation :- It occurs when the process does not fully utilize the memory allocated to it.

The solution to the problem of external fragmentation is called Memory compaction.



Memory compaction

* Advantages and Disadvantages of memory compaction.

The main disadvantage of contiguous memory allocation is memory waste and inflexibility. As the memory is allocated to a file or a process keeping in mind that it will grow during the run. But until a process or a file grows many blocks allocated to it remains unutilized. And they even they cannot be allocated to the other process leading to wastage of memory.

In case, the process or the file grows beyond the expectation i.e beyond the allocated memory block, then it will abort the message "No disk space"

* file :- A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disk.

In general, a file is a sequence of bits, bytes, lines, or records whose meaning is defined by the file creator and user.

* file type :- A file type refers to the ability of the operating system to distinguish different types of files such as text files source files and binary files etc. Many operating systems supports many types of files. operating systems like MS-DOS and UNIX the following type of files -

Ordinary files

- These are the files that contains user information.
- These may have text databases or executable program.
- The user can apply various operations on such files like add, modify, delete or even remove the entire file

Directory files

- These files contains list of files names and other information related to these files.

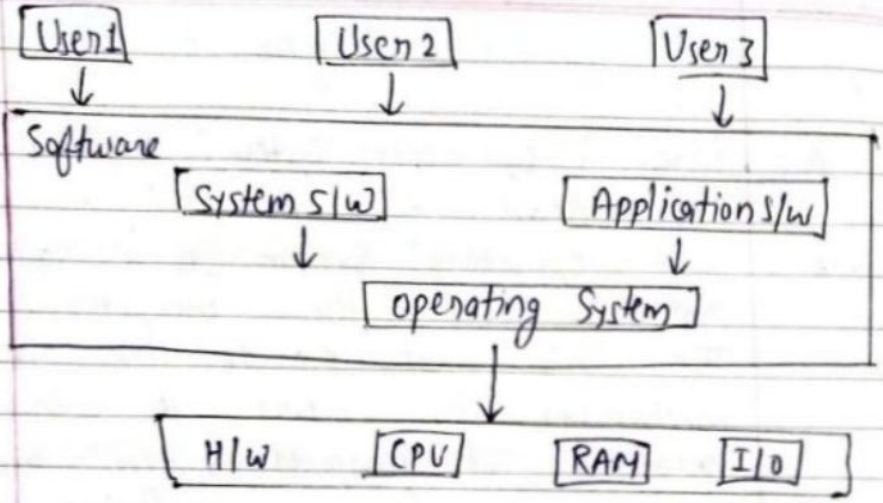


fig:- Architecture of OS.

Q.2 Explain the following:-

(i) Process & Program:-
Process contains a set of instruction designed to complete a specific task.
Program can be described as an instance of program running on a computer or as entity that can be assigned to and executed on a processon.

(ii) Threads:- A thread is a path of execution within a process. Thread is a single sequence stream with in a process. Threads have some properties as of process so they are called as light weighted process. They are executed one after another. but gives the illusion as if they are executing in parallel.

(iii) System Call:- A system call is a programmatic way in which computer program requests a service from kernel of os it is executed on. It is way for program to interact with OS.

Q3 Explain various services of an OS

Ans:- OS provides services to both user & to the programs.

1. It provides programs an environment to execute.
2. It provide users the services to execute the programs in convenient manner.

Few common services provided by an OS

1. Program Execution
2. Ilo operations
3. File System Manipulation

Eg - Java thread, POSIX	Eg - Window solaris

(ii)

	Processes	Thread
(a)	Process means any program is in execution.	Thread means segment of process.
(b)	Process takes more time to terminate.	Thread takes less time to terminate.
(c)	It takes more time for creation.	It takes less time for creation
(d)	Process is less efficient in term of comm^n process.	Thread is more efficient in term of comm^n process.
(e)	Consumes more resources.	Consumes less resources.
(f)	Process is isolated.	Thread shares memory.
(g)	Process is called heavy weight process.	Thread is called light weighted process.

Purchase the Notes

100₹

Per semester
(All subjects)

Notes (Hand written) ✅
Most Questions ✅

All Branches

Min 100%
amount will go
into charity ⭐

For specific
Subject - 50₹

UPI ID -
sahilkagyan337@ybl

Er Sahil ka Gyan

🔥 Steps for getting NOTES and Most Questions -

👉Do payment using UPI ID -

sahilkagyan337@ybl

👉Take screenshot of transaction and send me on Email -

ersahildrive@gmail.com

Then finally access all Notes and most questions 🔥

Scan & Pay Using PhonePe App

SAHIL KHAN