OS minor-2 Ans:-

UNIT-III:-

Q1)

Safe state & unsafe state

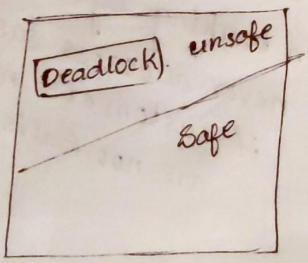
SAFE STATE:

- A state is said to be safe if system can allocate resources to each process (max) in some order.
- If there is a safe sequence then the system said to be in safe state. for processes $\{P_1, P_2, \ldots, P_n\}$
- Safe state is not a deadlock state and deadlock state is an unsafe state.

UNSAFE STATE:

- It may lead to deadlock, this OS cannot prevent process from requesting a resources.

fig:- Safe/unsafe States.



Avoidance => ensure that the system will never enter safe state.

Q2)

Banker's Algorithm :-

- Resource allocation graph (RAG) does not applicable for multiple instances of a resource
- due to this the banker's algorithm is used for multiple instances resource allocation.
- this name was given because it was used in Banking System.
  - In banking system never allocates cash if customer's needs are not satisfied.

this can be written as example for bankers algorithm.

- ~~take~~ the max number of instances are not more than no. of resources

Datastructures included

- 'n' no of processes
- 'm' no of resources

① Available :-

~~a vector length m indicates~~ the m - available no of resources.

if available [j] = [k] then
'k' instances of R_j are available.

② Max :-

m x n → determines the maximum demand of each processes.

if max [i] [j] equals k then
P_i request for k instance of resource type R_j.

③ Allocation :-

n x m → defines the no. of resources currently allocated to each process.

if allocation [i][j] equals to 'k' then process P_i currently allocated to Resource R_j.

④ Need :-

n x m - matrix defines remaining resource need of each process.

Need [i][j] ⇒ max [i] - allocation [i][j].

13) Deadlock detection.

If a system dosen't use any deadlock prevention or deadlock avoidance then deadlock may occur.

In this environment the system provide.

- an algorithm that examines weather deadlock occured.
- an algorithm to rescue from the deadlock.

Deadlock Detection Algorithm.

work and finish are the vectors of length mxn. respectively.

(1) Initialize work = available

for i=0, 1, 2.... n-1 if allocation≠0. then.

finish[i] = false.,

otherwise finish[i] = True.

(2) find Index 'i' such that both

a. finish[i] == false

b. Request$_i$ ≤ work

If no 'i' exists then go to step 4.

(3) work = work + allocation;

Finish[i] = True

Go to step (2).

(4) If finish[i] == false, for some i, $1 \leq 0 \leq a$

then system is deadlock state.

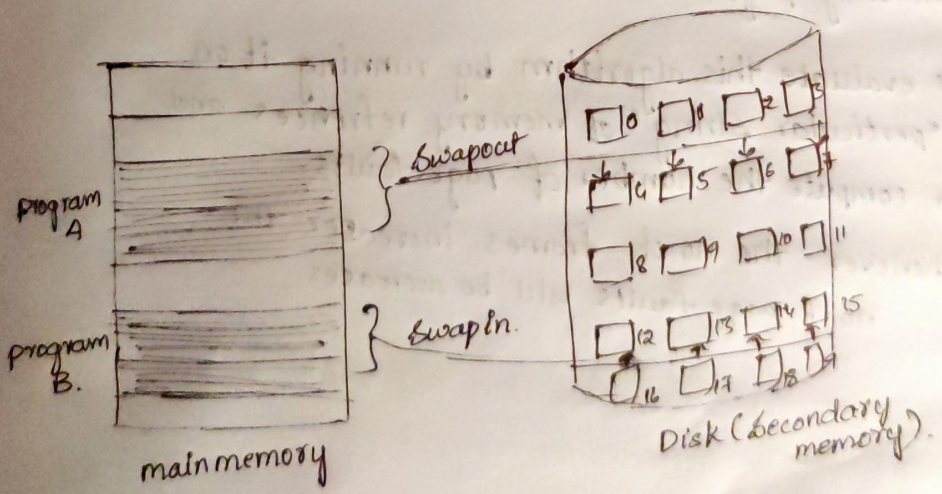
if finish[i] == True → Safe State.

UNIT - IV :-

Q4) Demand Paging.

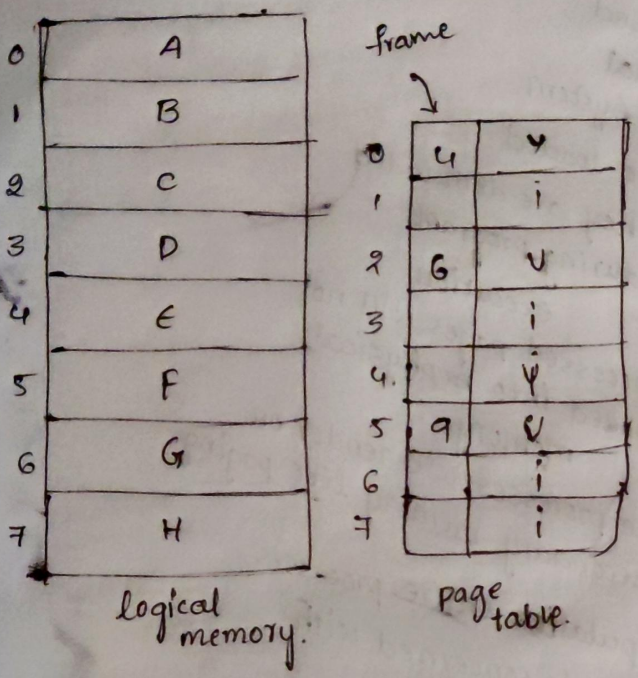
- commonly used in the virtual memory system.
- pages are loaded when they are demanded during program execution.
- not accessed pages will not loaded into physical memory.
- the processes will render on secondary memory like paging.
- swapping manipulates entire processes, whereas pager is concerned with individual pages.

fig:- demand paging (transfering ~~disk~~ paged memory to contiguous disk space).



EAT = (1-P) x ma + P x page fault time.

fig ②. page table when some pages are not in the main memory.



Q5) page replacement algorithms:-

→ This algorithm helps to decide which pages are swapped out from the main memory for every incomming page.

→ we evaluate this algorithm by running it on a "perticular string" of memory refrences and then compute the number of page faults.

→ whenever the no. of frames increases the no. of page faults will be decreases.

③ FIFO Replacement algorithm

→ It is one of the simplest page replacement algorithm.

→ associates with each page when the page is brought into the main memory.

→ In this type of algorithm the oldest page is replaced which is present in main memory for longest period of time.

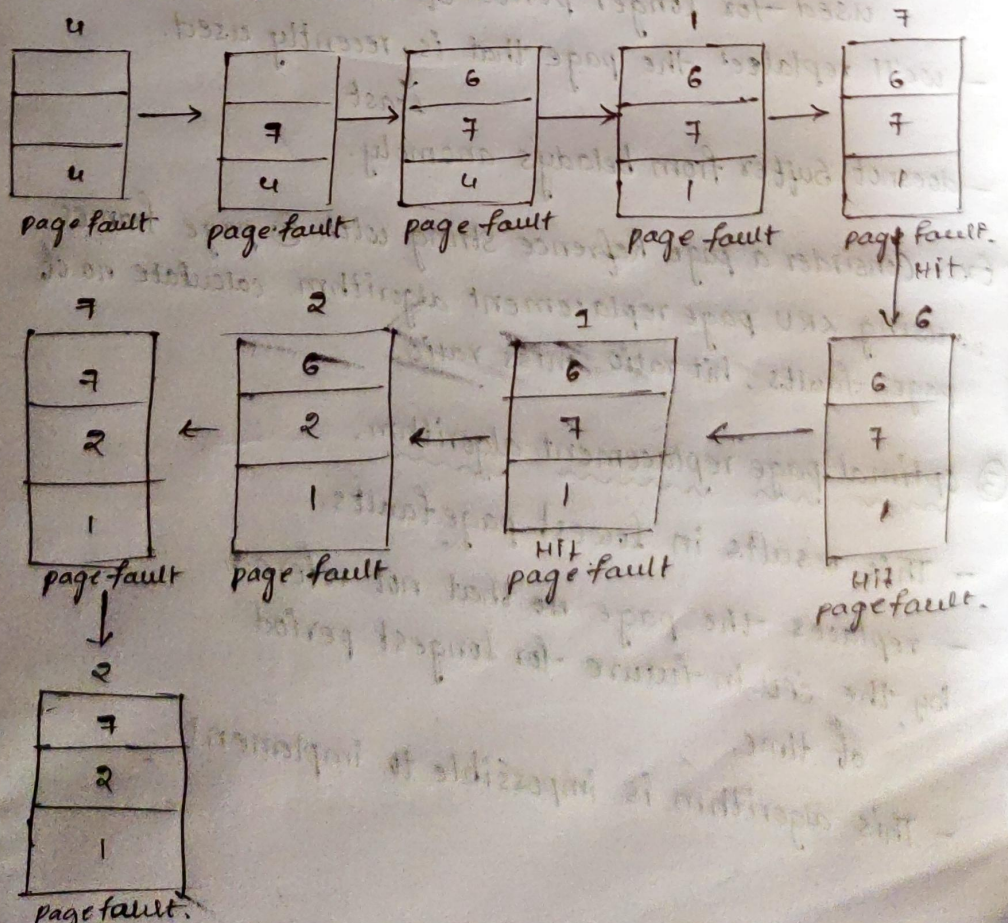
Ex:- consider a page refrence string with ③ page frames using FIFO page replacement algothm calculate, hit ratio, miss ratio.

Refrence string

4, 7, 6, 1, 7, 6, 1, 2, 7, 2.

Sol assume that all page frames empty main memory divided ③ slots.

4, 7, 6, 1, 7, 6, 1, 2, 7, 2.



total no.of page fault = 6.

total no.of page refrences = 10.

Calculate Hit Ratio.

Total no.of pages hits

= total no.of page ref - total no.of page fault.

= 10-6.

= 4.

Hit ratio → $\frac{\text{total no.of hits}}{\text{total no.of page ref.}}$

$= \frac{4}{10} = 0.4 = 40\%$.

② LRU page Replacement algorithm.:

- mainly works on the least recently used.
- In this the page refrenced by CPU have not been used for longer period of time.
- we'll replaced the page that is recently used.
  ↓ least.
- doesnot buffer from belady's anomoly.

Ex:- Consider a page Refrence string with ③ page frames using LRU page replacement algorithm. calculate no.of page faults, hit ratio miss ratio.

③ optimal page replacement algorithm.

- This results in lowest page faults.
- replaces the page that not refered by the CPU in future for longest period of time.
- This algorithm is impossible to implement.

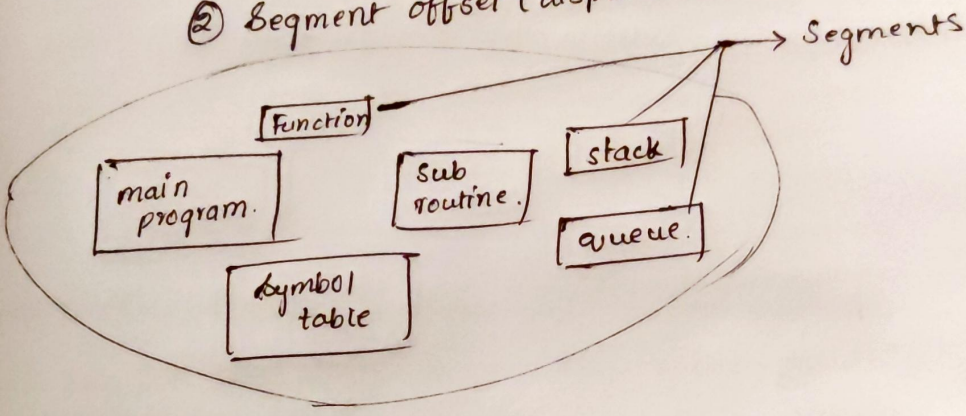
Q6) Segmentation:-

- memory management technique that ~~used to~~ supports the programmers memory view.
- each ~~big~~ logical address may have the collection of segments.
- each segment contain segment name and segment length.

Logical address is divided into 2 parts.

① Segment number (s)

② Segment offset (displacement) (d)



Segment Hardware:-

- Segment are represented by tables.
- entry in the segment table consists of ② types

1. Segment base:
   Hold the starting physical address.
2. Segment limits:
   Specifies the range length of the segment.