Memory Management

M/y mngt is the functionality of an OS which handles or manages primary m/y and moves process back and forth b/w main m/y and disk during execution. M/y mngt keep track of each and every m/y location.

It include,

* Checks how much m/y is to be allocated to a process.
* Decide which process will get m/y at what time.
* It tracks every free m/y.
* Allocate m/y.
* Deallocate m/y.
* Keep track of status of each every m/y location.

Logical Address Space

Address generated by CPU is called logical address. The set of all logical address generated by a pgm is called logical address space.

Physical Address Space

Set of all physical address corresponding to the logical address is called phy address space.

Single contingous m/y allocation

It is a simple m/y allocation scheme, which require no special h/w features. It is usually associated with minicomputer. In this scheme there is no multiprogramming. Entire ~~a~~ available m/y is allocated to a single job. Here main m/y is devided into 3.

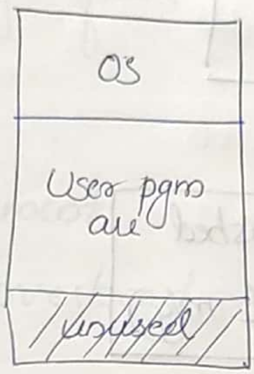


fig:) Single contingous allocation.

One portion is permanantly allocated to the OS. All the remaining m/y is available for a single job that has to be processed

advantage

* Simple allocation
* Entire scheme require less m/y.
* Easy to implement and use.

disadvantage

* m/y is not fully utilised.
* Processor is not fully utilised.
* User pgm is being limited to size available in main m/y

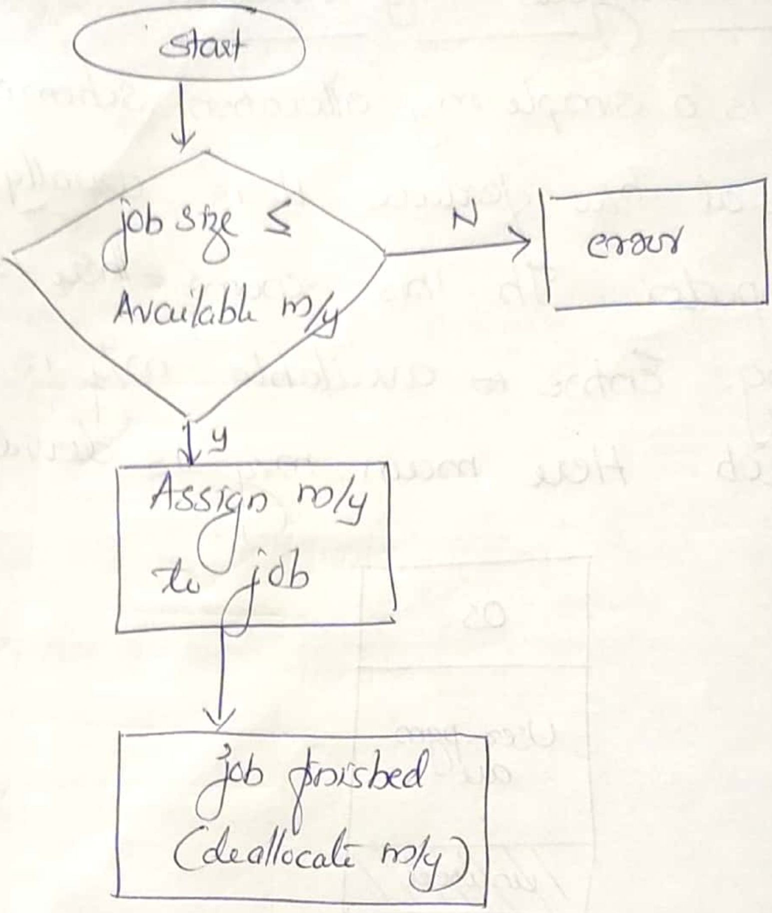


fig 2) Single Contiguous allocation.

Partioned allocation

Partioned allocation is devided into 2.

1. fixed (or static) partitioning
2. variable (or dynamic) "

1. Fixed Partitioning

This is the oldest and simplest technique used to put more than one process in MM. Here the partitions are fixed but size of each partition may or may not be the same.

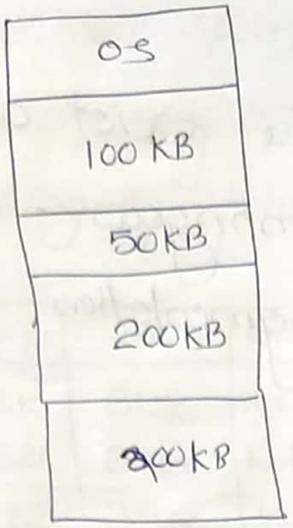


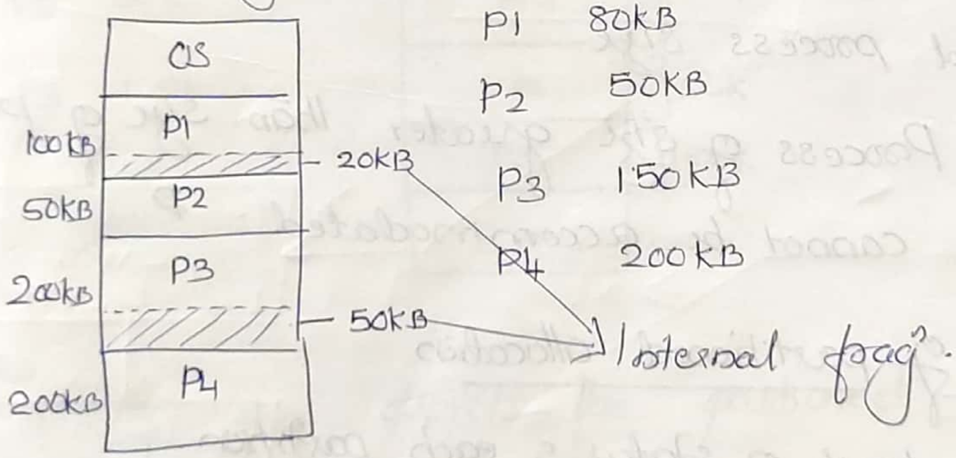
fig 3) fixed size partitioning

advantage

* Easy to implement
* Little os overhead

Disadvantage

* Internal fragmentation

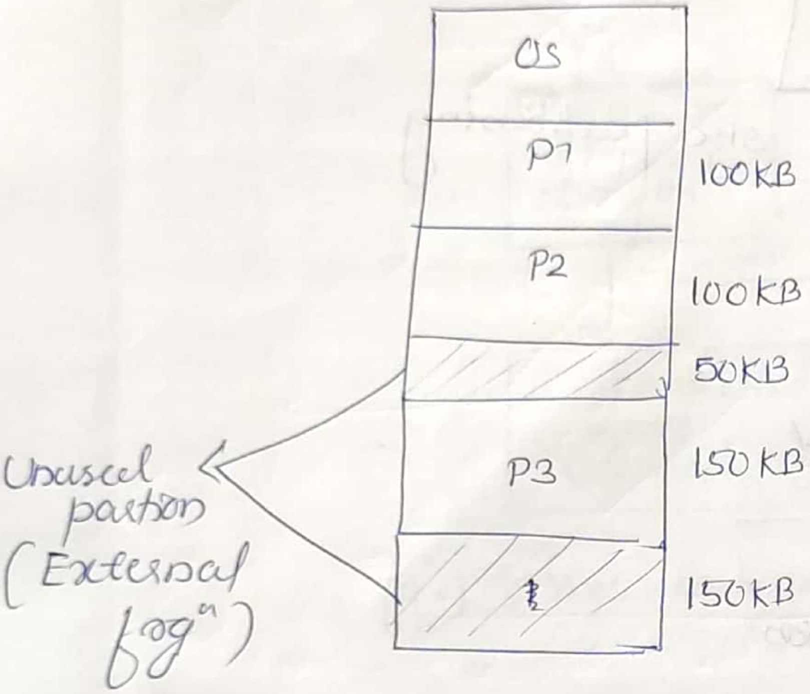


Each pgm requires a particular partion. Then it leaves a small m/y in that partion that are unused. These unused space is called internal fragmentation.

Scanned with CamScanner

* External fragmentation

Even though there exist a lot of free space that are not contingous. Such condition is called external fragmentation



* Limited process size

Process of size greater than size of partition in mm cannot be accommodated.

Functions of partitioned allocation

1. keeping track of status of each partition.
2. Determining who gets mly.
3. Allocation – An available partition of sufficient size is assigned.
4. Deallocation – When the job terminates, the partition is indecated "not in use" and is available.

a. Static partition specification

Here the m/y is devided into partitions before the processing of jobs.

eg:

	Size	Locⁿ	Status
1	8K	312K	In use
2	32K	320K	Not in use
3	32K	352K	In use

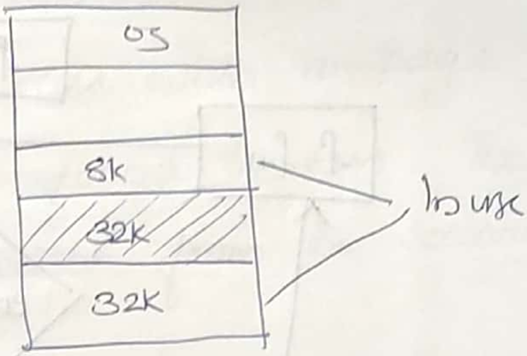


fig static partition specification.

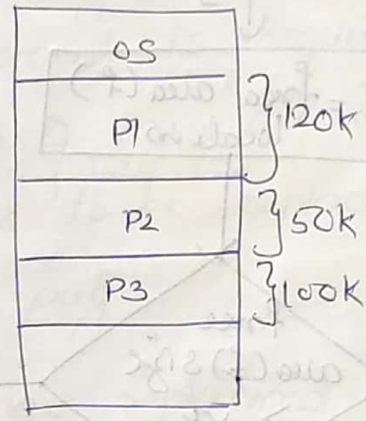
2. Dynamic partition specification

Partitions are created during job processing so as to mach partition size to job size.

P1 120K

P2 50K

P3 100K



following fig depicts the partitioned allocation algorithm.

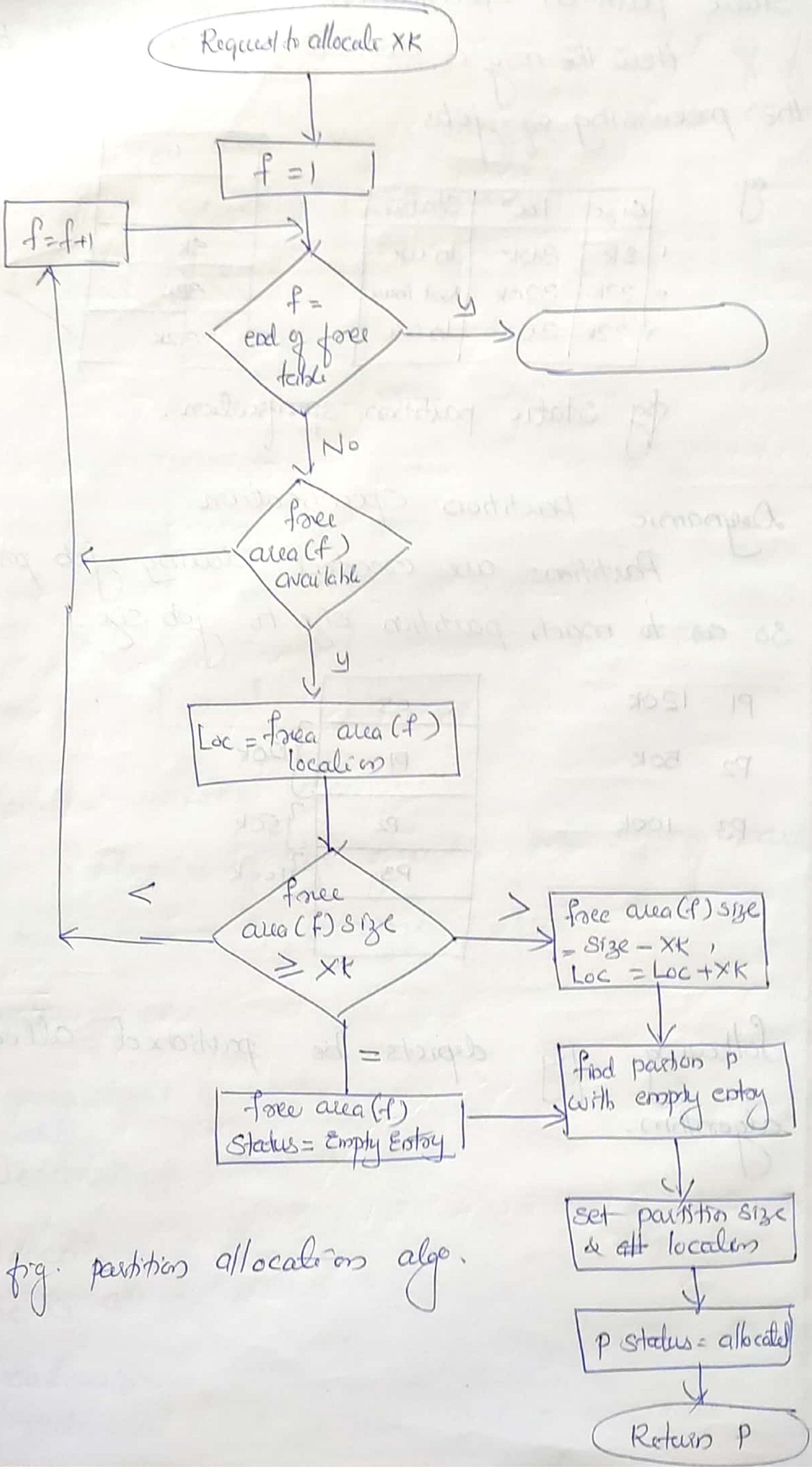


fig. partition allocation algo.

Hit Ratio

$$\text{Hit Ratio} = \frac{\text{Total no. of page hit}}{\text{Total hit + miss}}$$

Page fault or page miss occurs when a page referenced by the CPU is not found in MM. The required page has to be brought from the secondary m/y into the MM.

Page Replacement

prev. q.

It is a process of swapping out an existing page from the frame of a MM and replacing it with the required page.

Page replacement is required when, All the frames of MM are already occupaid. Thus a page has to be replaced to create a room for the required page.

A good page replacement algo is one that minimises the no. of page fault.

Page Replacement Algorithms

Page replacement algorithms are the techniques using which an OS decides which m/y pages to swap out and which page should be allocated. Paging happens whenever a page fault occurs and a ~~free~~ new page ~~cannot~~ will be swaped in to the m/y.

Scanned with CamScanner

When a page that was selected for replacement and was paged out, is referenced again it has to read in from disk and this requires for i/o completion. This process determines the quality of page replacement algorithm

Reference String :-

The string of mry references is called reference string. They are generated artificially or by tracing a given s/m and recording the address of each mry reference.

Different page replacement algorithms are,

1. FIFO First In First Out algo.
2. LRU Least Recently Used algo.
3. Optimal Page replacement algo.

FIFO

This is the simplest page replacement algorithm. The OS keeps track of all pages in the mry in a queue, the oldest page will be replaced first.

Reference String : 1 2 3 4 1 2 5 1 2 3 4 5

a. Using 4 frames

	5 →	1 →	2 →	3 →	4 →	5 →
1	5	5	5	5	4	4
2	2	1	1	1	1	5
3	3	3	2	2	2	2
4	4	4	4	3	3	3
1111	1	1	1	1	1	1

Total page fault = 10 ie 4+1+1+1+1+1+1

Initially all slots are empty. So when 1, 2, 3, 4 came they are allocated the empty slots. Thus 4 page faults.

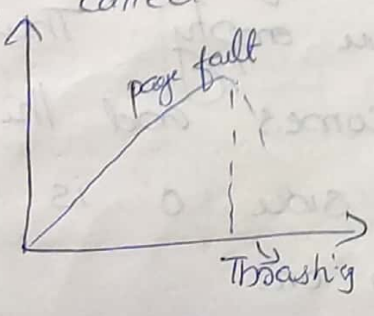
Again 1, 2 comes it is already in the m/y. So it when 5 came, it is not available in m/y so it replaces the oldest page slot ie 1 will be replaced with 5 and so on.

Belady's Anomaly

It is possible to have more page faults when increasing the number of page frames while using FIFO page replacement algorithm.

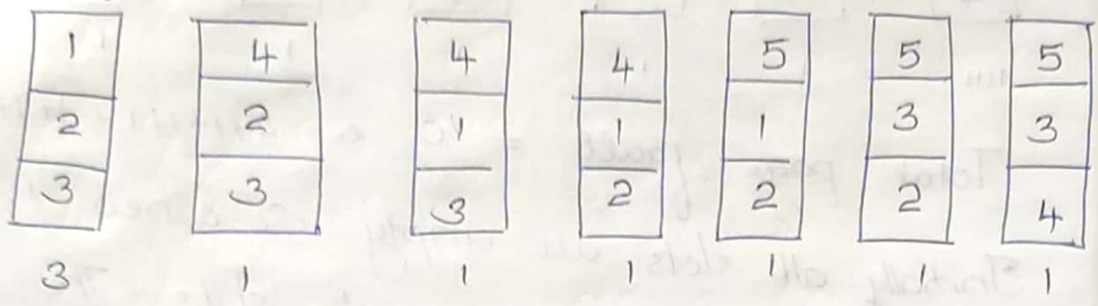
Thrashing

If the no. of page fault are so high, so that the CPU remains busy in just reading the page from the secondary m/y than execution. This is called thrashing



Ref. string : 1 2 3 4 1 2 5 1 2 3 4 5

b) Using 3 frames

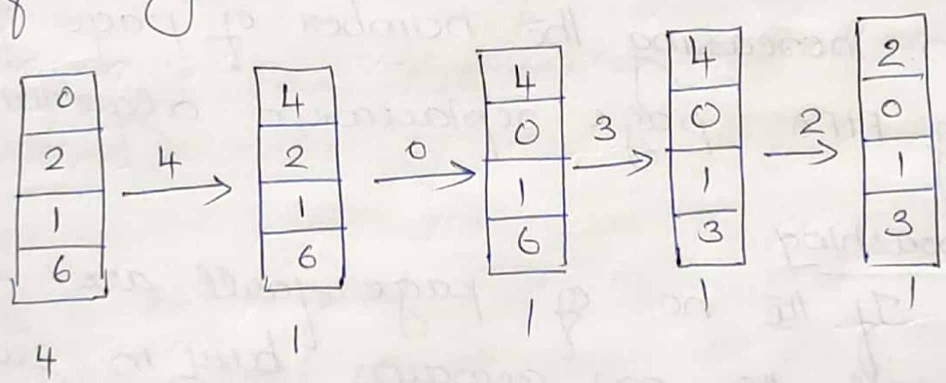


Total page fault = 3+1+1+1+1+1+1
= 9

LRU (Least Recently Used) page replacement algo.

This algo. replaces the page which has not been referred for a long time. This algo. is very easy to implement.

Ref. string : 0 2 1 6 4 0 1 0 3 1 2 1



Total page fault = 8

Here 0, 2, 1, 6 are allocated first time bcz at the very beginning all slots are empty. Thus 4 page fault occured. Then 4 comes and the page 0 will be replaced with 4 since '0' is not used for a long time. and so on

Optimal page replacement algo.

Here the pages are replaced ~~with~~ which would not be used for longest duration of time in the future. It has the lowest page fault rate of all algorithm.

Ref. String : 0216 40 10 3121

0	0	3
2	2	2
1	1	1
~~0~~ 6	4	4
4	1	1

Total page fault = 4+1+1 = 6.

This algo. is practically impossible to implement bcz pages will not be used in future for the longest time cannot be predicted.

Previous Year Question

Q what is multi processor S/m

A multi processor S/m have multiple processor working in parallel that share the computer clock, m/y, bus, peripheral device etc.

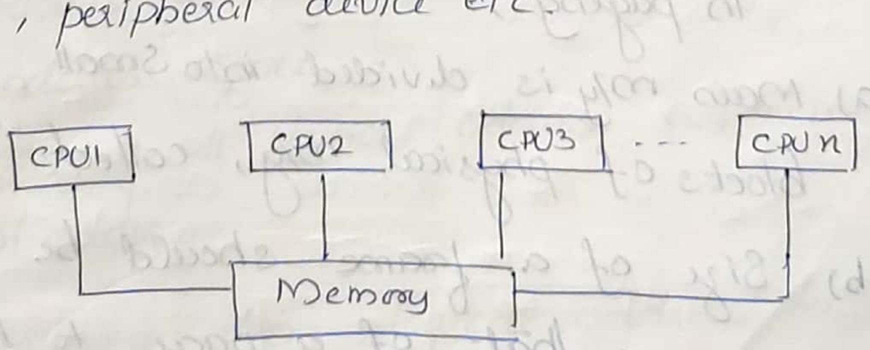


fig) multi processing architecture

Scanned with CamScanner

Advantage

1. More Reliable

   If one processor fails, the S/m will not halt. Other processors are on working condition.

2. The no. of process execculed per unit time increases thus the S/m will be very efficient.

3. multiprocessor S/m are cheaper than single processor S/m in the long run bcz they share the data storage.

Disadvantage

1. Complicated OS is required.
2. Large main m/y is essential for proper working.

Paging

Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main m/y in the form of pages.

In paging,

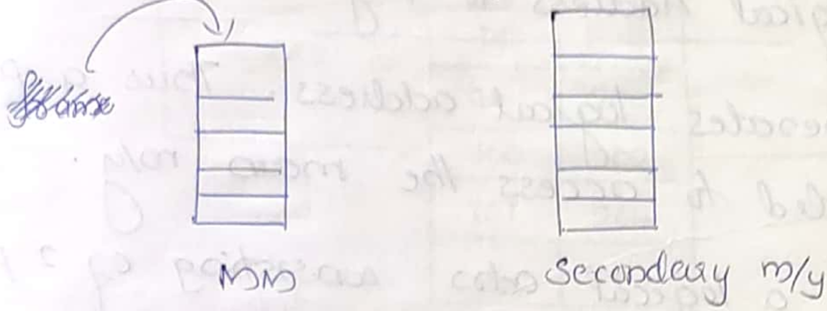
a) Main m/y is divided into small fixed sized blocks of physical m/y, called "frames"

b) Size of a frame should be kept the same as that of a page to have max utilization, and to avoid external fragmentation.

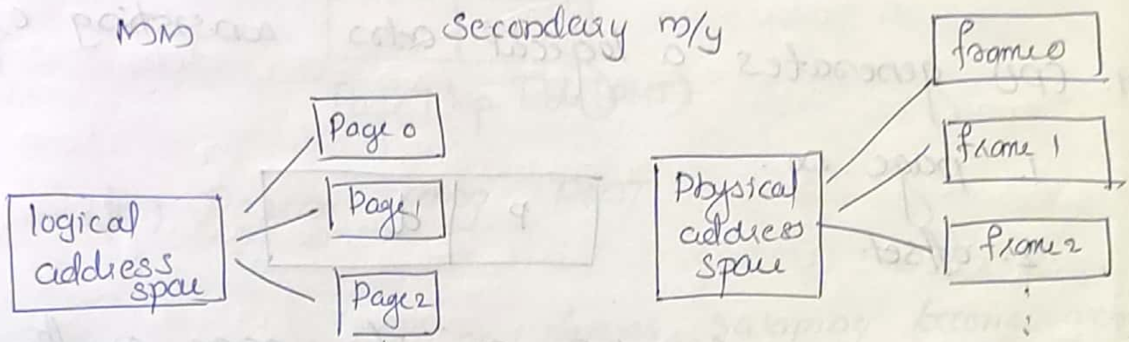
Paging is a fixed sized partitioning scheme.

In paging, secondary m/y and main m/y are divided into equal fixed sized partition.

* The partition of secondary m/y is called "pages"
* The partition of main m/y is called "frames"



ie



* logical address or Virtual adrs
  Address generated by CPU
* logical adrs space
  The set of all logical address generated by the program.
* physical address
  An address actually available on m/y unit
* physical address space
  The set of all physical address corresponding to the logical address.

The mapping from logical address to physical adrs is done by MMU (m/y mngt unit) which is a h/w device and this mapping is known as paging technique.

LA	→	MMU	→	PA
logical address				physical address

fig 2) Mapping

Translating Logical Address to Physical address

CPU generates logical address. Thus a phy. address is needed to access the main m/y.

Step 1: CPU generates a logical adrs consisting of 2 parts.

1. page no.
2. offset.

p	d

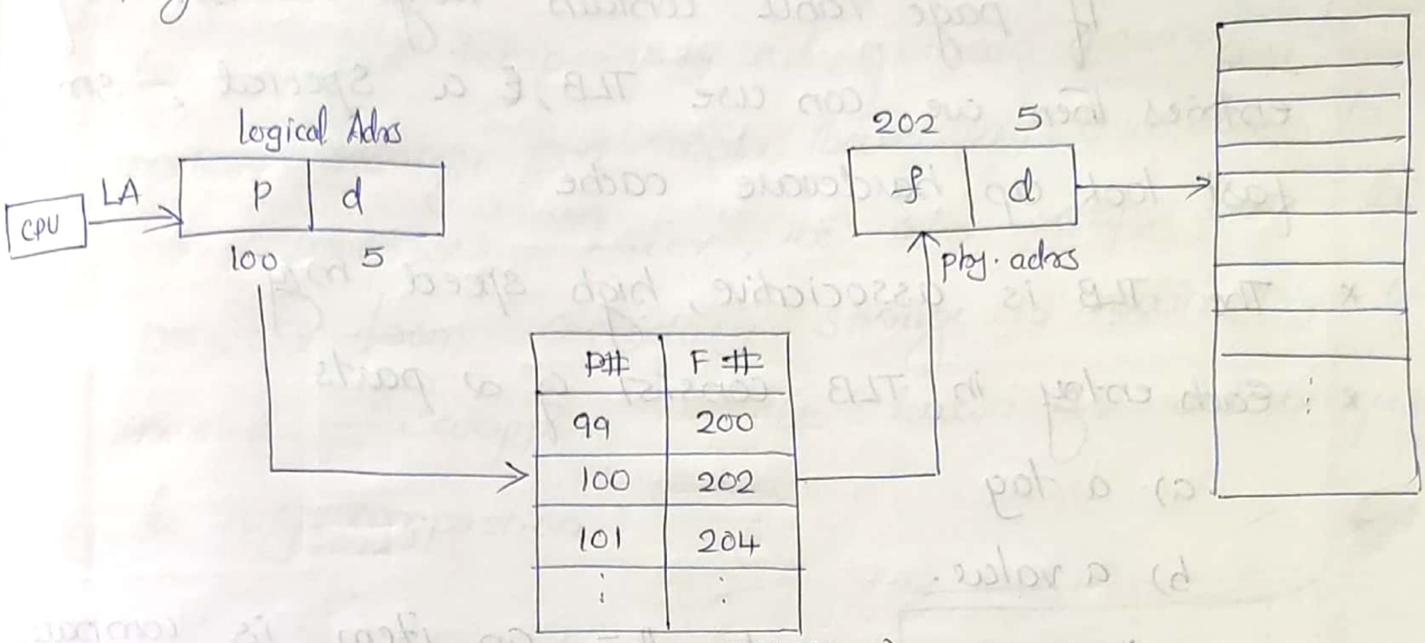
Page no - specifies the specific page of the process from which CPU wants to read the data. 'Page offset' specifies the specific word on the page that CPU wants to read.

Step 2: 'p' matches the entries in PMT (page map table) if a match found, corresponding frame number 'f' will be returned.

Step 3: The frame number 'f' will be added to the offset 'd' in the logical address.

frame no.	offset
f	d

frame no. specifies the specific frame where the required page is stored. offset i.e. d specifies the specific word that has to be read from that page.



Page Map Table (PMT)

fig) Paging using PMT

Advantage

* Due to equal size of pages and frames, swaping become very easy.
* It allows to store parts of a single process in a non-contingous fashion.
* It solves the problem of external fragmentation.
* Reduces overhead thus simple to implement.

Disadvantage

* Suffers from internal fragmentation.
* There is an overhead of maintaing a page table for each process.
* The time taken to fetch the instruction increases since now two m/y access are required.
* PMT require extra m/y space so, not good for S/m having small RAM.

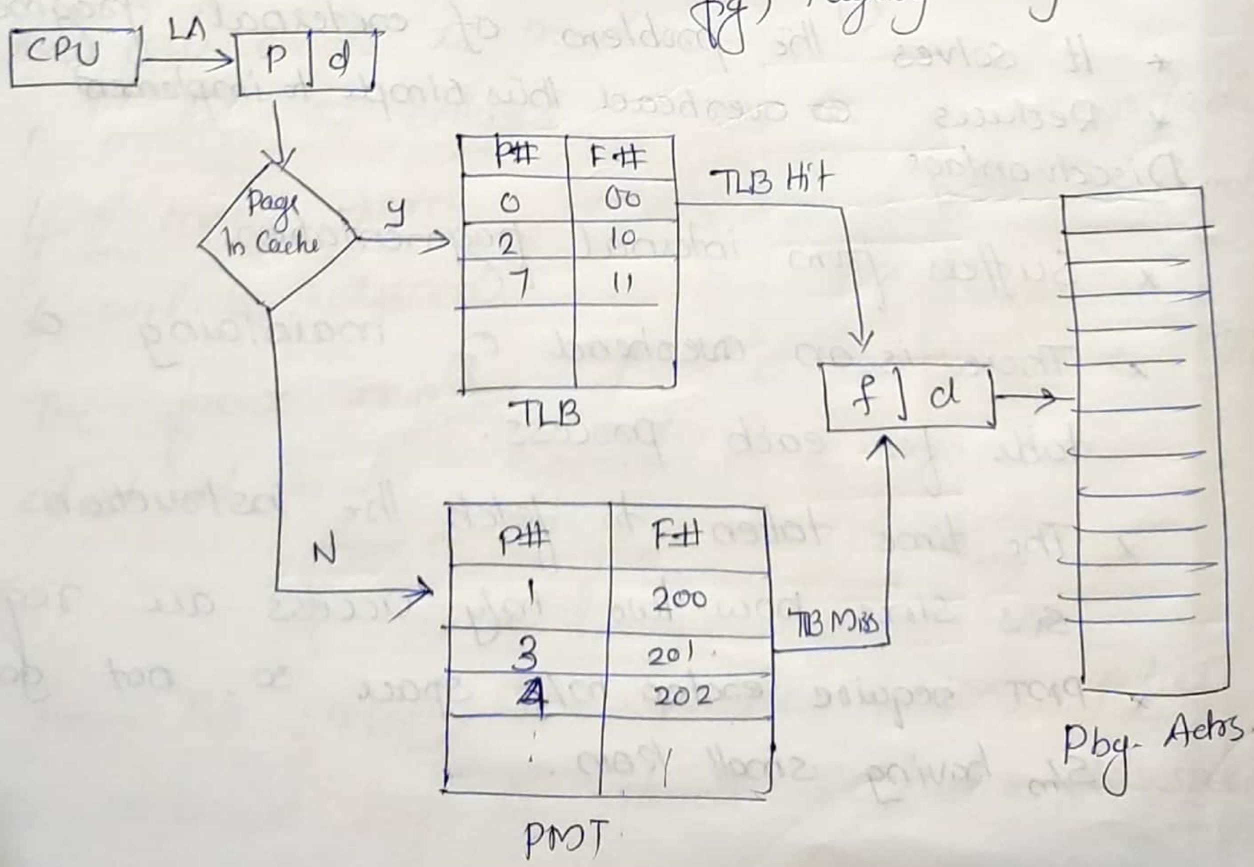
Translation Look Aside Buffer (TLB)

If page table contain large number of entries then we can use TLB, & a special, small fast look up hardware cache.

* The TLB is associative, high speed m/y
x Each entry in TLB consist of 2 parts
  a) a tag
  b) a value.

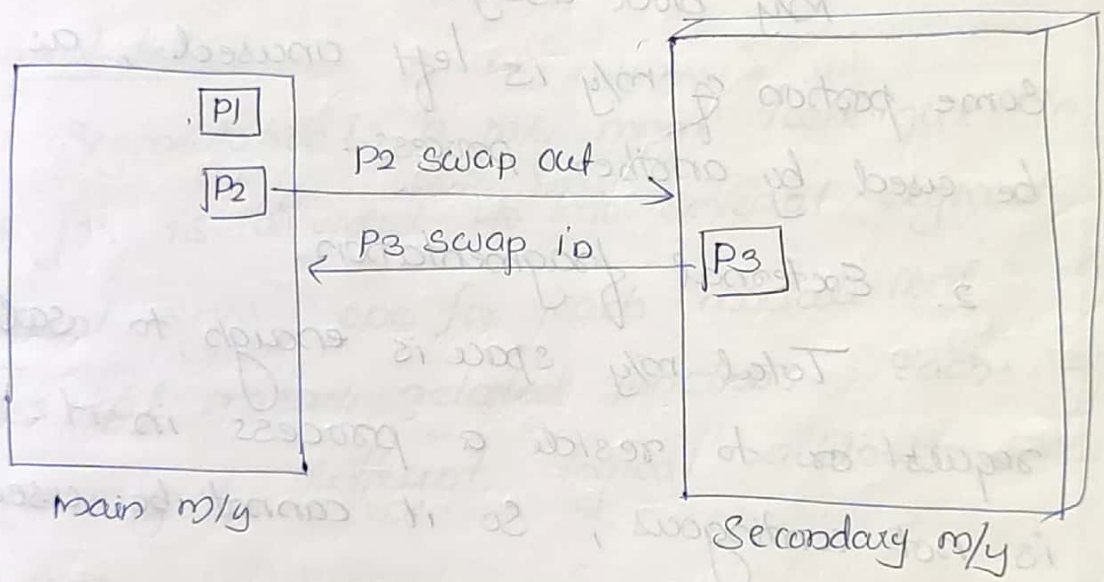
x When this m/y is used, then an item is compared with all tags simultaneously. If the item is found, the corresponding value is returned. It is called "TLB Hit". If no such entry found, it is called "TLB Miss".

fig) Paging Using TLB



Swapping

Swapping is a mechanism in which a process can be swapped temporarily out of main memory to secondary storage and make that m/y available to other processes. Later, the s/m swaps back the process from secondary storage to main m/y if needed. "Swapping is also known as a technique for m/y compaction."



The total time taken by swaping process include, the time it takes to move the entire process to a secondary disk and then to copy the process back to m/y and the time the process takes to regain the main m/y.

Roll Out :- If a higher priority process is to be executed, then lower priority process is swapped out into disk.

Roll In :- Higher priority process are swapped into the main m/y.

Fragmentation

As processes are loaded and removed from m/y, the free m/y space is broken into little pieces. It happens after sometimes that process cannot be allocated to m/y blocks considering their small size and m/y block remain unused. This problem is known as fragmentation. 2 types.

1. Internal fragmentation

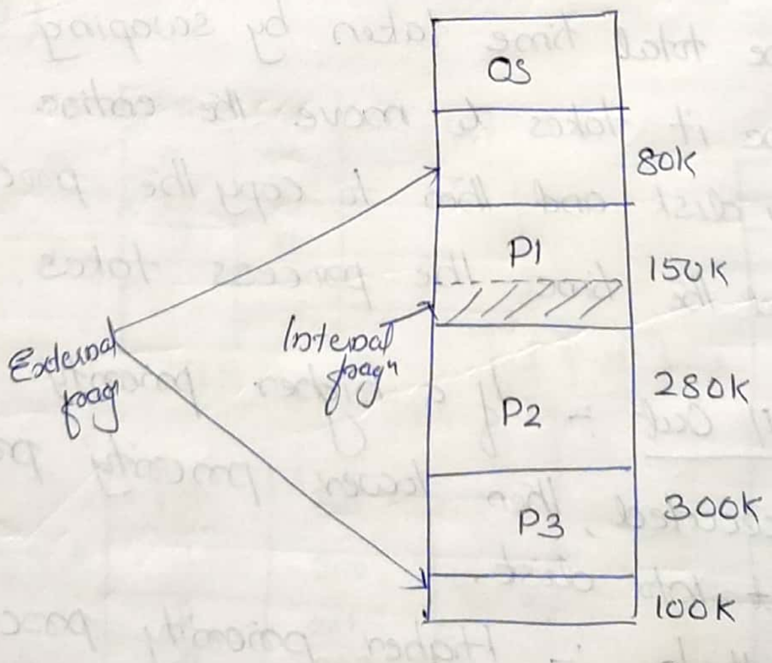
   M/y block assigned to process is bigger, some portion of m/y is left unused, as it cannot be used by another process.

2. External fragmentation.

   Total m/y space is enough to satisfy a request or to reside a process in it. But it is not contingous, so it cannot be used.

eg:-

P1	130K
P2	280K
P3	300K
P4	180K



P4 cannot be allocated.

Here 100k and 80k is available thus a total of $100+80 =$ 180k. But they are not contingous. Thus even if the space is available it cannot be used for allocation is called external fragmentation.

30k is available in P1 block is called internal fragmentation.

Solution to fragmentation problem is "Compaction" ie putting all free spaces together.

Segmentation

Segmentation is a m/y mngt technique in which each job is divided ~~to~~ into several segment of different sizes, one for each module that contain pieces that perform related functions. Each segment is actually a different logical address space of the pgm.

When a process is to be executed, its corresponding segmentation are loaded into non-contingous m/y. A table that stores the information about all such segments are called "Segment table".

Segment table maps 2dimensional Logical adrs into one dimensional phy. adrs. Each table entry has,

1. Base Address :- Starting adrs of segment.
2. Limit :- Specifies length of the segment.

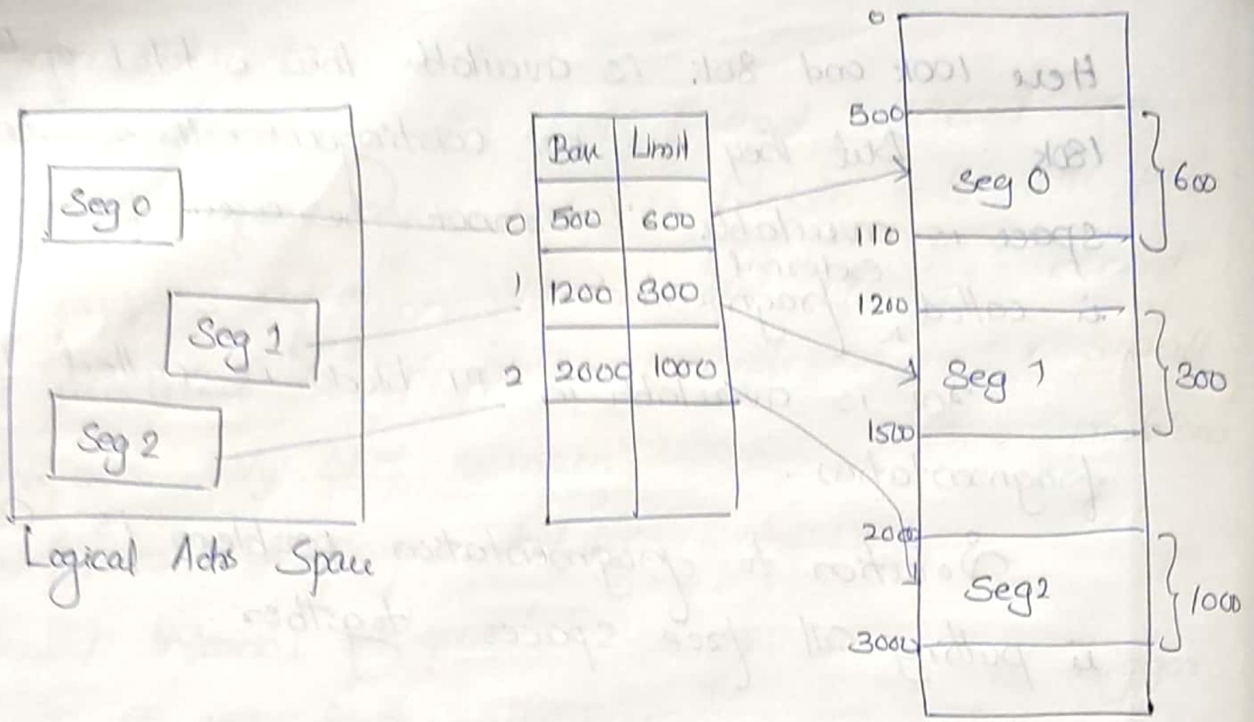


fig: Logical view of segmentation.

CPU generates a logical adds which contain 2 parts.

1. segment no.
2. offset.

The segment no. is mapped to segment table. The limit of segment is compared with offset. If the offset is less than its limit, it is valid otherwise an error occured.

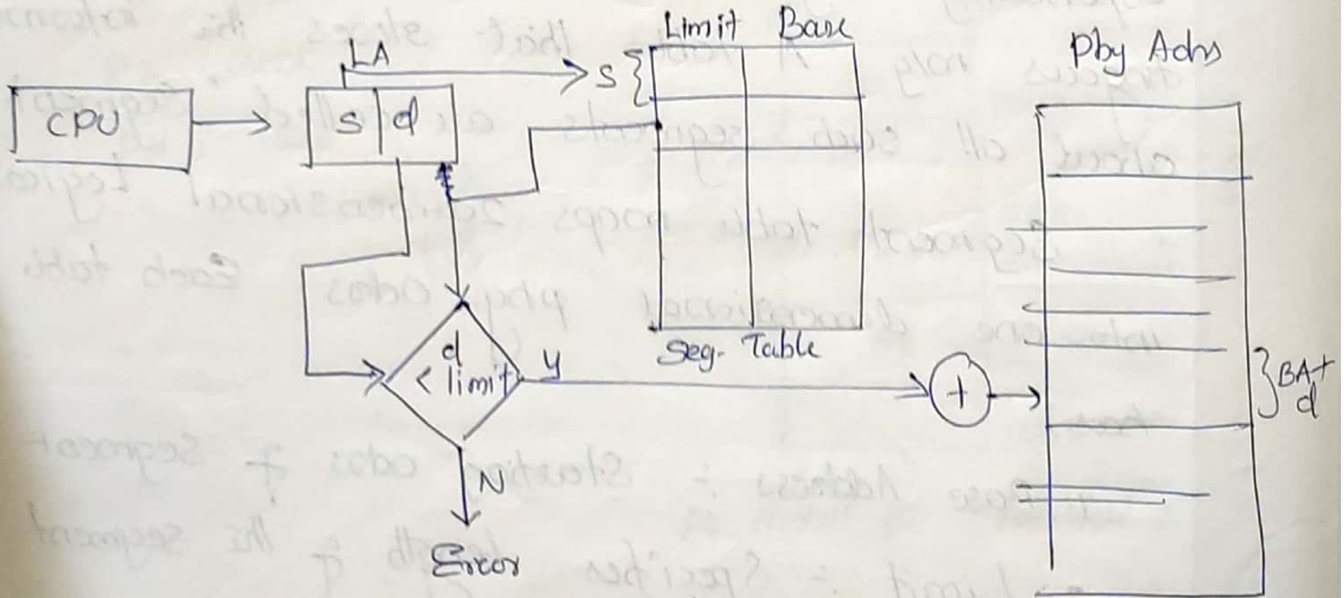


fig) Translation of 2D LA to 1D phy. adds.

Advantage

- No Internal fragmentation
- Segmentation table require less space than PMT in paging.
- Less overhead

Disadvantage

- Cause external fragmentation
- Difficult to allocate contingous m/y to variable sized partition
- Costly m/y management algo.

Virtual Memory (VM)

VM is a storage allocation scheme in which secondary m/y can be addressed as though it were part of main m/y.

It is a storage mechanism that provide user an illusion of having a very big main m/y. This is done by making a part of secondary as the main m/y. It is implemented using both h/w and s/w. It maps m/y addresses used by a pgm, called virtual addresses into physical addresses in compr m/y.

1- All m/y references within a process are logical addresses that are dynamically translated into phy. address at run time. ie process can be swapped in and out of main m/y such that it occupy diff. places in main m/y at diff times during the course of execution.

2. A process may be broken into no. of pieces and these pieces need not be continously located in the main m/y during execution. The address translation and use of page or segment table permit this.

If above 2 conditions are satisfied, then it is not necessary that all pages or segments are present in the main m/y during execution. So required pages need to be loaded into m/y whenever required. VM is implemented using "Demand paging or Demand Segmentation".

Advantage

1. Degree of multiprogramming increased.
2. CPU utilization will be high.
3. Demand paging is a popular method of VM mngt.
4. Run large application with less RAM.

Disadvantage

1. S/m becomes slower bcz of swapping.
2. Takes more time in switching b/w application.
3. User will have the lesser hard disk space.

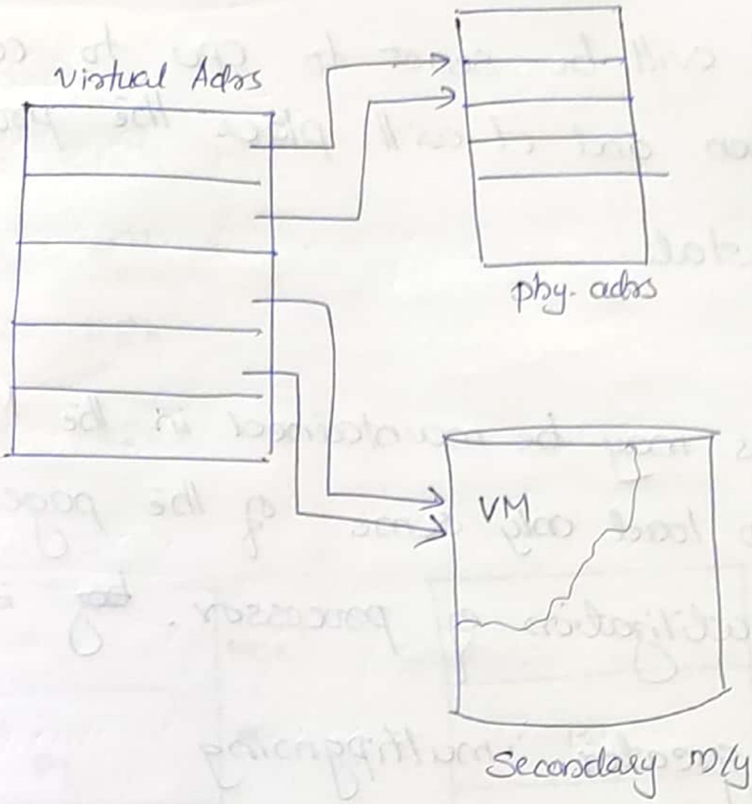


fig) logical repres

<u>Demand Paging</u>

The process of loading the page into m/y on demand (whenever page fault occurs) is known as demand paging.

1. If CPU try to refer a page ie currently not available in the MM, it generates an interrupt indicating a page fault.
2. The OS block the interrupted process from execution.
3. Bring the required page into the m/y. Thus OS will search for the required page in the logical address space and brought it to phy. adrs.
4. The page table will be updated automatically.

5. The signal will be sent to CPU to continue the pgm excecution and it will place the process back into ready state.

advantage

1. More process may be maintained in the MM bcz we are going to load only some of the pages. ~~eg~~

2. Efficient utilization of processor. ~~bcz it is more~~

3. It allows greater multipgming

Partition Allocation Schemes

M/y is divided into different blocks or partition. Each process is allocated acc. to requirement. ~~Bo~~ various partition allocation schemes are,

1. First Fit
2. Best Fit
3. Worst Fit

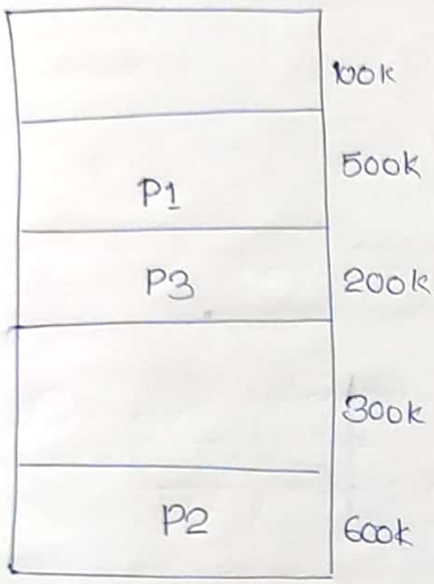
1. First fit :- Allocate at first free large enough partition.

2. Best Fit :-
Allocate at smallest partition among the free partition.

3. Worst Fit :-
Allocate at largest possible free space.

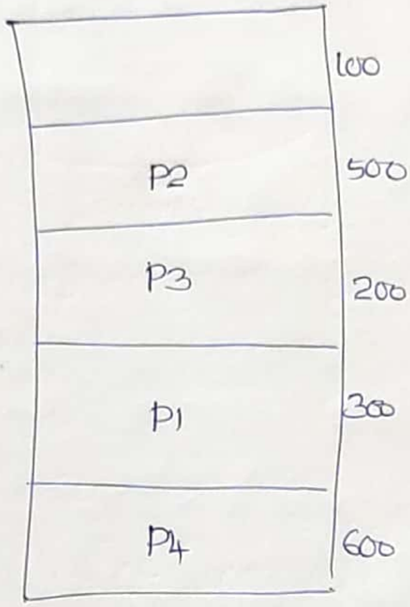
eg: Consider

P1	212k
P2	417k
P3	112k
P4	426k

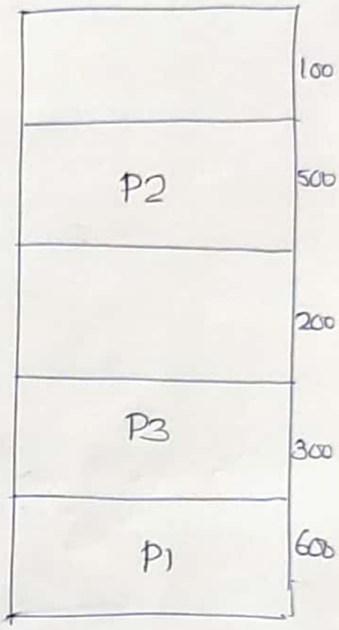


First Fit

P4 Cannot allocate



Best Fit



Worst Fit

P4 cannot allocate

Scanned with CamScanner