Memory management.

* Introduction :-

Memory management is the core operating system function that tracks every memory location, allocates and free RAM space for running programs, that handles address translation to ensure efficient multitasking & system performance.

Basic memory management:-

↳ Memory management is a process of managing computer's memory (RAM). The operating system allocates memory to programs when they start and releases the memory when they finish. It ensures effiecient and safe use of memory.

* Need of memory management:-

↳ It is required to allocate memory to running programs.

↳ It ensures use of RAM efficiently.

↳ support multitasking.

↳ It prevents memory conflicts between programs.

↳ It improves system performance.

* Types of memory management in operating system

Based on how memory is allocated, memory management is mainly classified into two types.

① contiguous memory management.

② Non-contiguous memory management.

(1) contiguous memory management:-

contiguous memory management is a memory allocation technique in which each process is stored in one continuous block of memory.

↳ Types of contiguous memory management.

i) single partition allocation.

ii) multiple partition allocation.

↳ fixed partitioning

↳ variable partitioning.

Advantages:-

(1) It is simple and easy to implement.
(2) Fast access of memory.
(3) suitable for small systems.

Disadvantages:-

(1) It causes internal & external fragmentation.
(2) Poor memory utilization.
(3) Large program may not fit into one continious block.

2] Non-contiguous memory management:-

Non-contiguous memory management is memory allocation technique in which a process divided into several parts, and each part is stored in different locations of memory.

↳ Types of non-contiguous memory management

i) paging
ii) segmentation.
iii) paged segmentation.

Advantages:-

① Better memory utilization.
② It reduces fragmentation.
③ It supports multitasking.
④ Large program can be executed efficiently.

Disadvantages:-

① It is more complex than contiguous memory management.
② Requires additional tables (page table or segment table).

* partioning:-

partitioning is memory management technique in which the main memory (RAM) is divided into different sections called partitions.
Each partition stores one process at a time.
It helps the operating system allocate memory efficiently to programs.

There are basically two types of partioning

1) fixed partioning (static partioning)
2) Variable partioning (Dynamic partioning)

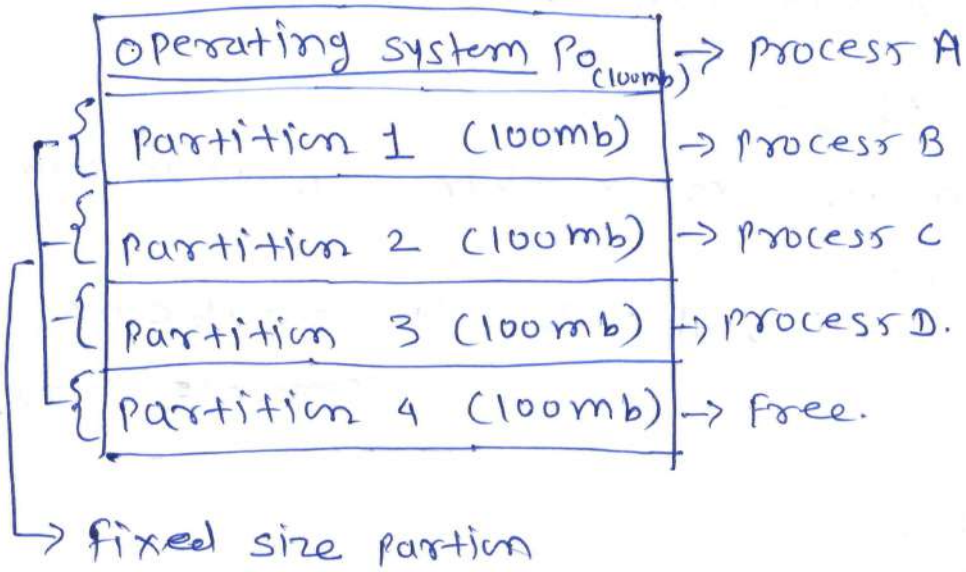
① Fixed partioning:-

↳ fixed partioning is memory management technique in which the main memory (RAM) is divided into a fixed number of partitions before the system starts.
↳ It is also called as static partioning.
↳ The size and number of these partions remain

constant during execution.

↳ Each partion can hold only one process at a time.

Diagram



* Working of fixed partioning :-

1. The operating system divides RAM into fixed size partions.
2. When process enters in the system, it is allocated to free partion.
3. If the process size is smaller than the partition size, the remaining memory is wasted.
4. If no partition is free, the process is placed in waiting queue, till partition is free.
5. When the program finishes, the partition becomes free and can be assigned to another program.

Advantages:-

(1) It is simple and easy to implement.
(2) Fast memory can be allotted.
(3) It is easy to manage.

④ Each process has its own partition.

⑤ It is suitable for small systems.

Disadvantages

① It causes internal fragmentation.

② Number of processess is limited by number of partitions.

③ Memory cannot be resized during execution.

④ Poor memory utilization.

⑤ Large process may not be loaded if they required large partition.

* Internal Fragmentation :-

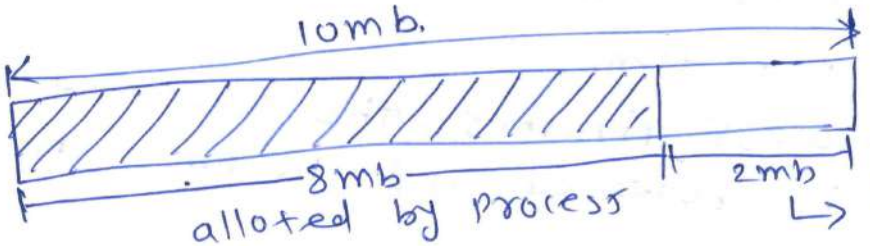
Definition :- Internal fragmentation occurs when a process occupies less memory than the size of its allocated partition, leaving unused space inside partition.

e.g.

Partition size is 10 mb

process size is 8 mb.

used space = 2 mb (wasted) // Internal fragmentation.



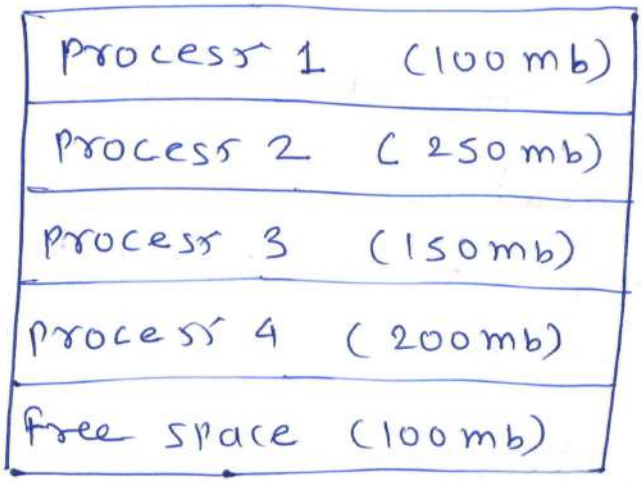
② Variable partitioning :-

Definition :- Variable partitioning is a memory management technique in which the operating system creates memory partition of diffent sizes according to memory requirement of each process.

A partition is created only when a process arrives & its size is just enough to accomodate that process.

Diagram:-

main memory



Each process receives only amount of memory it needs.

* Working of variable partitioning :-

1. The operating system checks the available free memory.
2. When a process requests memory, the OS creates a partition equal to process size.
3. The process is loaded into newly created partition.
4. When process finishes, the allocated partition is released and becomes free memory.

Advantages

1. Better memory utilization.
2. NO internal fragmentation.

③ memory is allocated according to process size.

④ supports processes of different sizes.

⑤ more flexible than fixed partitioning.

Disadvantages:-

① It causes external fragmentation.

② Memory allocation is more complex.

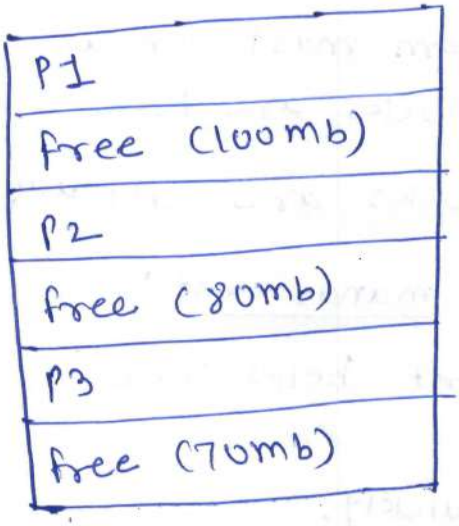
③ Requires memory management algorithms.

④ It may require compaction to combine free spaces.

* External fragmentation :-

External fragmentation occurs when free memory is divided into many small non-contiguous blocks.
Although the total free memory is enough, a large process cannot be allocated because the free memory is not available as one continous block.

e.g.



Total free memory = 250mb
A new process requires 200 mb.
The process cannot be loaded because there is no single continous free block of 200 mb.

This problem is called external fragmentation.

Compaction

Definition :-

Compaction is the process of combining all scattered free memory blocks into one large continious block.

It removes external fragmentation.

Before compaction

P1
free
P2
free
P3
free

After compaction.

P1
P2
P3
large free block.

[Now large processes can be allocated easily.]

* Free space management :-

Definition:-

Free space management is process of keeping track of free memory blocks so that the operating system can allocate memory to new process efficiently.

The operating system must know:
- which memory blocks are free
- which memory blocks are already allocated.

* Need of free space management :-

free space management helps the operating system to:

1) Allocate memory quickly.
2) Utilize memory efficiently.
3) Reduce memory wastage.

(4) Improve system performance.
(5) Keep track of available memory.

Techniques of free space management:-

There are two common techniques (types):

1. Bitmap Technique
2. Linked list Technique.

(1) Bitmap Technique:-

In the Bitmap technique, each memory block is represented by one bit in a bitmap.

0 → Free memory Block.
1 → Allocated memory Block.

The operating system checks bitmap to find free memory.

* How Bitmap works :-

suppose memory is divided into 8 blocks.

memory block	B1	B2	B3	B4	B5	B6	B7	B8
status	Used	Free	Used	Free	Free	Used	Used	Free

Bitmap representation

B1	B2	B3	B4	B5	B6	B7	B8
1	0	1	0	0	1	1	0

Where
1 means memory block is occupied
0 means memory block is free & available to use.

The operating system scans bitmap and allocates free blocks.

Advantages :-

1) It is very simple to understand.
2) It is easy to implement.
3) Requires less storage for tracking memory.
4) quickly identifies free memory blocks.

Disadvantages:-

1) Scanning bitmap may take time for large memories.
2) Finding consecutive free blocks can be slow.
3) Not efficient for very large memory systems.

2] Linked list techniques:-

Definition:-

In the linked list technique, all free memory blocks are connected using pointers to form a linked list.

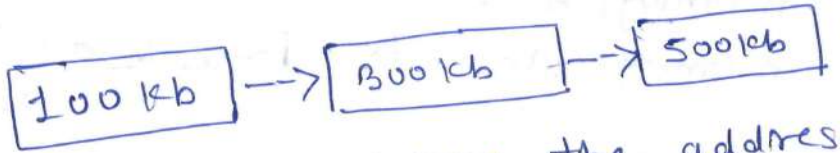
Each free block contains:

1) memory address.
2) pointer to next free block.

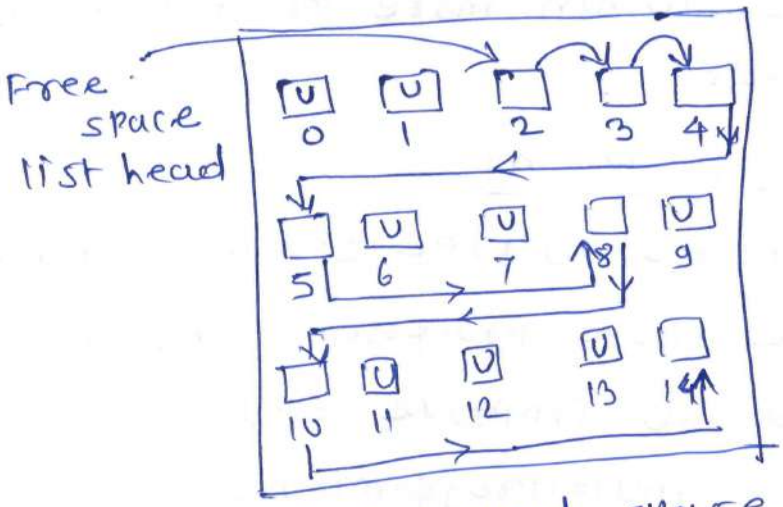
The operating system follows the pointer to locate available memory.

* How Linked List works:-

Free memory Blocks



Each block stores the address of next free block. Free blocks may be located in different locations in memory.



Where [U] indicates used space.

* Advantages :-

(1) If file is to be allocated a free block, the OS allocates first free block & move head pointer to next free block in the list.

(2) NO disk fragmentation as every block is utilized.

* Disadvantages :-

(1) It is not efficient to traverse list, it is required to reach each block to check and it requires extra time.

(2) pointers used here will also consume disk space, additional memory therefore required.

* Swapping :-

Swapping is a memory mgmt. technique in which the operating system temporarily transfers a process from the main memory (RAM) to the secondary storage (swap space on HDD) and later brings it back into RAM for execution.

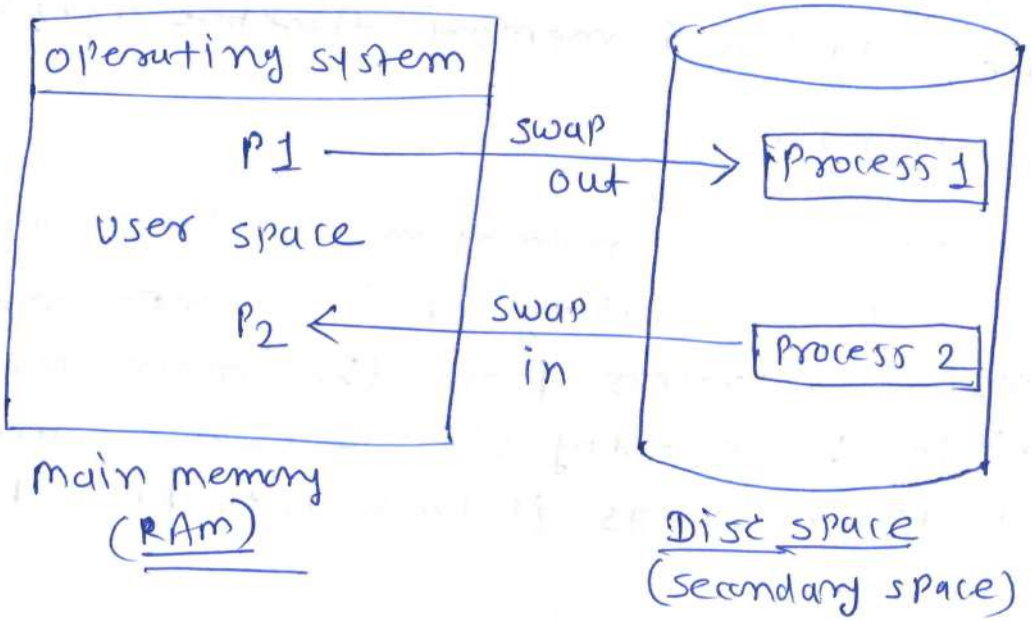
. It helps OS to run more process even when RAM is limited.

* Need of swapping :-

(1) It is required to free space in main memory
(2) To execute more processes than RAM capacity.
(3) It is used to improve CPU utilization.
(4) It supports multiprogramming.

* working of swapping process

↳ A process is loaded into RAM and starts execution.
↳ when RAM is full, the OS selects one process
↳ The selected process is moved to swap area on secondary storage (disc). This is known as (swap out).
↳ Another process is loaded into RAM. This process is known as swap In.
↳ when the swapped out process is needed it is brought back into RAM.



* Advantages of swapping :-

(1) Increases number of process that can run.
(2) makes better use of available RAM.
(3) Improves cpu utilization.
(4) supports multitasking.
(5) prevents memory shortage.

* Disadvantages of swapping :-

(1) Disk access is much slower than RAM.
(2) Frequent swapping decreases system performance.
(3) Increase process execution time.

* compaction :-

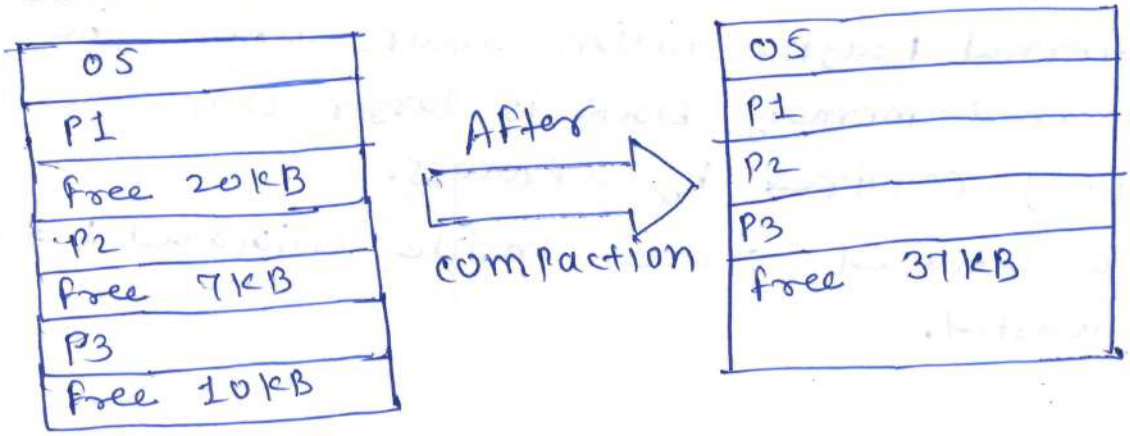
compaction is memory management technique used to remove external fragmentation by moving processes together in memory. It combines all free memory spaces into one large continous block.

Need of compaction :-

↳ When processes are loaded & removed from memory many small free spaces (holes) are created between them.

↳ These small holes may not be large enough to load a new process, even though the total free memory is sufficient.

↳ compaction solves this problem by combining all small holes into one large free space.



Advantages :-

1. Removes external fragmentation.
2. creates one large continous free memory block.
3. Improves memory utilization.
4. Allows larger processes to be loaded into memory.

Disadvantages :-

1. Takes extra CPU time.
2. move processes is time consuming.
3. execution may get interupted temporarily.
4. Not required in paging.

* Fragmentation :-

Fragmentation is memory management problem in which the available memory is divided into small unusable parts, resulting in inefficient use of memory.

* Types of fragmentation :-

- There are two types of ~~frag~~ fragmentation.
  1. Internal fragmentation
  2. External fragmentation.

1. Internal fragmentation :-

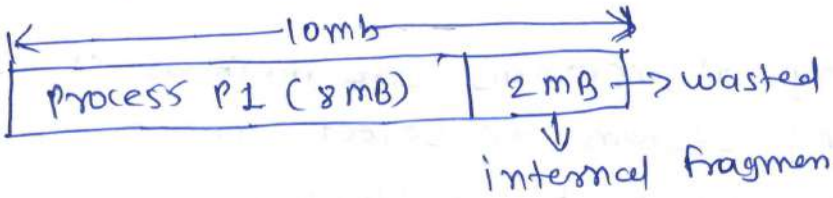
Internal fragmentation occurs when the allocated memory block is larger than the memory actually required by a process.
The unused space inside allocated block is wasted.

e.g. suppose memory is allocated in fixed size block of 10mb.

① process requires 8 MB.

② unused memory is 2 MB.

The unused memory inside allocated block is called internal fragmentation.



② External fragmentation :-

External fragmentation occurs when the free memory is divided into many small non-contiguous holes.

Although ~~to~~ total free memory is enough, there is no single continous block large enough for new process.

e.g. free memory blocks:

5MB, 8 MB, 7 MB

Total free space = 20 MB.

suppose, process requires 15MB, it cannot be loaded because no single continous block of 15MB is available.

Process A (20MB)
free (5 MB)
Process B (15MB)
free (8 MB)
process C (10MB)
free (7 MB)

Total free space = 20MB

To add 15 MB sized process NO continous 15MB block available.

such problem is known as external fragmentation.

* Partition algorithm :-

Partitioning algorithms are memory allocation techniques used by the operating system to decide which free memory partition should be allocated to a process.

OR

Partitioning algorithms are methods used by the operating system to select most suitable free memory block for a process.

Types of partitioning algorithm :-

1. First fit
2. Best fit
3. worst fit.

1] First fit algorithm :-

First fit is a memory allocation algorithm used in variable partition memory management. It allocates a process to the first free memory block that is large enough to hold the process.

* Working of first fit algorithm :-

1. start searching from beginning of free memory list.
2. compare the process size with the first free memory block.
3. If the block is large enough, allocate the process to that block.
4. If the block is too small, move to next free block.
5. stop searching after finding the first suitable block.

eg.

Process size P = 212KB

Memory blocks.

100KB free	500KB free	200KB free	300KB free	600KB free

Allocation process:-

Block 1 = 100KB - Not enough to hold 212KB process.

Block 2 = 500KB - Large enough to hold 212KB process.

Hence,

process is allocated to Block2, & Remaining 288KB will be next free block for next allocation process.

After allocation:

100KB free	Process P = (212KB) free space = 288KB	200KB free	300KB free	600KB free

* Advantages:-

1. It is very simple to implement.
2. Fast memory allocation.
3. Requires less searching time.
4. suitable when quick allocation is required.

* Disadvantages:-

1. May create external fragmentation.
2. Large free blocks may be split unnecessarily.
3. memory utilization decreases over time due to scattered free spaces.

(2) Best Fit Algorithm:-

Best fit is a memory allocation algorithm used in Variable partition memory management It allocates a process to the smallest available free memory block that is large enough to hold the process.

* Working of Best Fit algorithm:-

(1) The operating system checks all available free memory blocks.

(2) It compares size of process with each free block.

(3) It finds a block which is sufficient to hold the process

(4) Among these blocks, it selects the smallest suitable block in which process can be allocated effiently.

(5) The remaining memory stays available as a free block.

e.g. process P = 212KB.

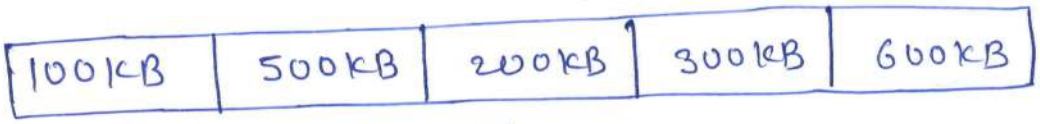
Now Best fit checks all available blocks.

Block	size	can process fit?
Block 1	100KB	NO ✗
Block 2	500KB	Yes ✓
Block 3	200KB	NO ✗
Block 4	300KB	Yes ✓ → Best fit
Block 5	600KB	Yes ✓

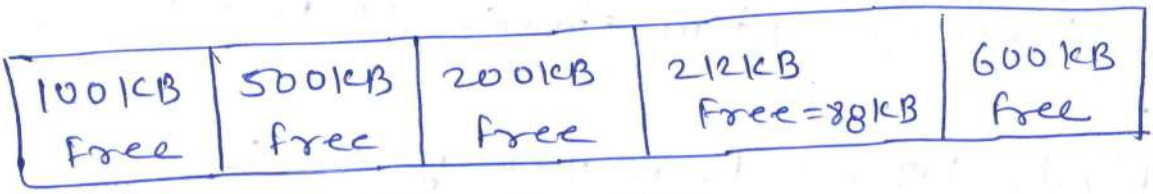
Therefore Block 4 of 300KB size is selected, coz it is the smallest block that can hold 212KB process.

Remaining memory. i.e. 300 − 212 = 88 KB. is free.

diagram



Before allocation.



After Best fit allocation.

* Advantages :-

↳ It selects smallest suitable memory block.

↳ It is best as it tries to reduce the amount of memory left over after each allocation.

↳ It can preserve larger free blocks for larger processes.

↳ It may provide effient memory utilization in some situations.

* Disadvantages :-

↳ It is slower than first fit because it generally needs to search the entire list of free blocks.

↳ It may create very small free holes.

↳ These small holes may become unusable.

↳ It can cause external fragmentation.

③ Worst Fit :-

Worst fit is a memory allocation ~~algorithm~~ algorithm used in variable partition memory management. It allocates process to the largest free memory blocks.

Working of worst fit algorithm :-

① The operating system checks all free memory blocks.
② It finds largest free memory block.
③ It allocates process to that block.
④ The remaining memory becomes a new free block.

e.g. Process size = 212KB

Worst fit checks all memory blocks.

↳ 100KB → Not selected
↳ 500KB → suitable to hold process
↳ 200KB → Not enough
↳ 300KB → suitable to hold process
↳ 600KB → Largest block - selected to hold process

Therefore the process is allocated to the 600KB block.

memory allocation diagram.

Before allocation.

100KB free	500KB free	200KB free	300KB free	600KB free

after allocation

100 KB free	500 KB free	200 KB free	300 KB free	Process = 212 KB free = 388 KB

600 - 212 = 388 KB

* Advantages:-

(1) It is easy to understand and implement.
(2) Leaves relatively large free block after allocation.
(3) Reduces creation of very small memory blocks.

* Dis advantages:-

(1) It is slower than first fit, as it requires more time to search all free memory blocks.
(2) Large memory blocks may wasted on small processes.
(3) can still cause external frugmentation.
(4) provides inefficient memory utilization than ~~first fit~~ & Best fit.

* Non-contiguous memory management Techniques:-

Non-contiguous memory management is a memory management technique in which the parts of process are stored at different locations in the main memory instead of one continous memory block.

The operating system keeps track of these memory locations and access them whenever required.

OR

Non-contiguous memory management is a technique in which a process is devided into parts and stored in different locations of memory.

* Need of non-contiguous memory management:-

① To make better use of memory available.

② To reduce external fragmentation.

③ To allow large programs to run even if continous memory is not available.

④ To improve memory utilization and system performance.

* Types of non-contiguous memory management

There are two main techniques:

① Paging

② Segmentation.

① Paging :-

Paging is non contiguous memory management ~~techniq~~ technique in which a process is divided into fix-sized blocks called pages and the main memory is devided into fixed size blocks called frames.

The pages of a process can be stored in different frames of main memory.

Simple definition :-

Paging is a memory management technique in which a process is divided into fixed size pages and these pages are stored in different frames of main memory.

* Need of Paging :-

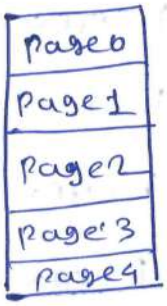
1. make better use of available memory.
2. Allow a process to be stored in different locations.
3. Avoid need of one large continous memory block.
4. Eliminate external frugmentation.
5. It support virtual memory.

* Basic concept of Paging :-

Paging devides memory into two parts:

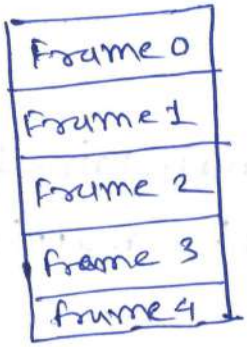
Logical memory :-

The process is divided into equal sized blocks called pages.



Physical memory :-

main memory (RAM) is divided into equal sized blocks called frames.



Important :- The size of page and size of frame are always the same.

* Working of paging mechanism :-

Suppor process has 4 pages

Page 0
Page 1
Page 2
Page 3

Suppose free frames are:

Frame 2
Frame 5
Frame 1
Frame 7

The operating system can store pages as follows:

Page	Frame
Page 0	Frame 2
Page 1	Frame 5
page 2	Frame 1
Page 3	Frame 7

No pages stored in continous locations.

To remember such different operating system maintains page table.

* Page Table :-

A page table is data structure maintained by operating system that stores the mapping between pages and frames.

e.g.

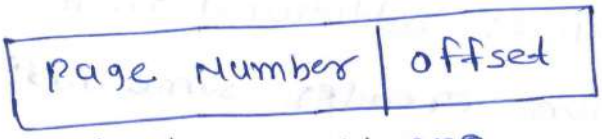
Page 1 → Frame 5

page number	frame number
0	2
1	5
2	1
3	7

As per above page table, when CPU wants to access page 1, the operating system knows that it is stored in frame 5.

* Address Translation :-

In paging logical address is divided into two parts:-



Logical address.

Page Number:-

It identifies which page contains the required data.

offset:-

It identifies exact location inside that page.

e.g.

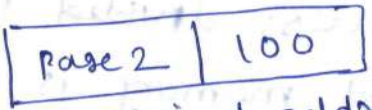
suppose

page number = 2

and offset = 100

& Page 2 is stored in frame 5

Then,



Logical address.

Page table entry is

Page 2 → Frame 5

Physical address

frame 5

& offset = 100

physical address.

The offset remains same. only page number is converted into related frame number.

* Advantages :-

① It eliminates external fragmentation.
② It allows non-contiguous memory allocation.
③ It improves memory utilization.
④ A process does not need one continous memory block.
⑤ It supports virtual memory.
⑥ Pages can be easily moved between RAM & secondary storage.

* Disadvantages :-

① Page table requires additional memory.
② Address translation creates some additional overhead.
③ Paging may cause internal fragmentation.
④ Managing pages and page tables increases system complexity.

Key points:-

↳ Paging is non-contiguous memory management technique.
↳ A process divided into fixed size pages.
↳ Physical memory is divided into fixed size frames.
↳ Page size = Frame size
↳ Pages can be stored in any available frames.
↳ The page table maps pages to frames.
↳ Paging eliminates external fragmentation.
↳ Paging may cause internal fragmentation.
↳ It is modern OS technique used in virtual memory.

* segmentation :-

segmentation is a non-contiguous memory management technique in which a process is divided into different logical parts called segments.

↳ Each segment represents a meaningful part of a program, such as code, data, stack or heap. The segments can be stored at different locations in main memory.

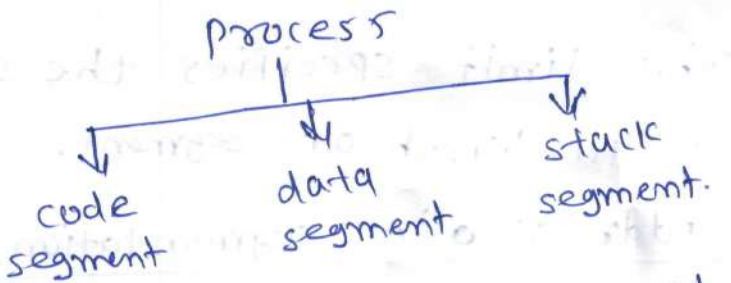
OR

segmentation is a memory management technique in which a program is divided into variable size logical segments and each segment is stored separately in memory.

* Need of segmentation :-

A program can be divided into different segments

e.g.



Unlike paging, the segments are not necessarily equal in size.

e.g. suppose program contains:

code	20 KB
Data	10 KB
stack	15 KB
Heap	25 KB

Each part is treated as separate segment.

The OS stores them at different locations in memory. and OS uses segment table to access them.

* segment Table :-

The operating system maintains a segment Table to keep information about each segment.

A segment table generally contains:

- Segment Number
- Base Address
- Limit

e.g.

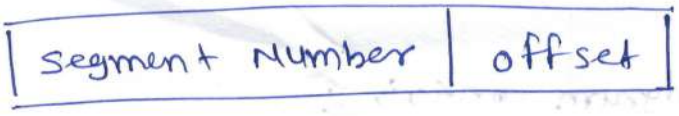
segment NO.	Base address	Limit
0	1000	20 kB
1	5000	10 kB
2	8000	15 kB
3	12000	25 kB

Base address :- The base address specifies the starting location of segment in physical memory.

Limit :- The limit specifies the size or maximum valid length of segment.

* Logical address in segmentation :-

In segmentation logical address contains:



segment NO. — It identifies the required segment.

offset — It identifies the exact location within that segment.

Address Translation :-

Suppose CPU generates

segment No. = 2

offset = 500

The OS checks segment table.

segment 2

Base address = 8000

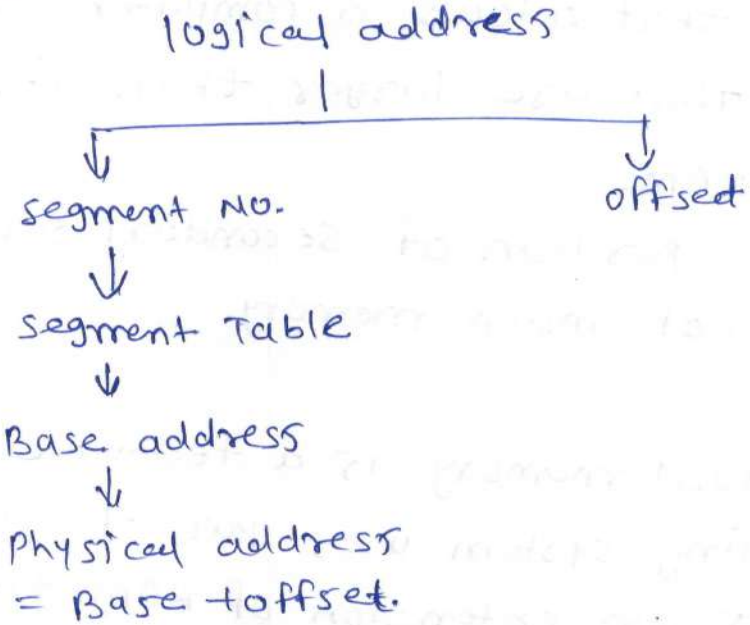
Limit - 15KB

The physical address calculated as:

Physical address = Base address + offset

∴ physical address = 8000 + 500 = 8500

diagram



* Advantages :-

(1) Logical organization :- The program is divided into its logical components such as code, data & stack.

(2) Easy protection :- different access permission is given to different segments.

(3) Easy sharing :- A segment can be shared between multiple processes.

④ Non-contiguous Allocation :-

Different segments can be stored at different locations in physical memory.

* Disadvantages

① It can cause external fragmentation.
② memory allocation is more complex.
③ segment table requires additional memory
④ compaction may be required to combine scattered free spaces.
⑤ Address translation adds overhead.

* virtual memory :-

virtual memory is a memory management technique that allows a computer to execute programs that are larger than the available physical RAM.

↳ It uses portion of secondary storage as an extention of main memory.

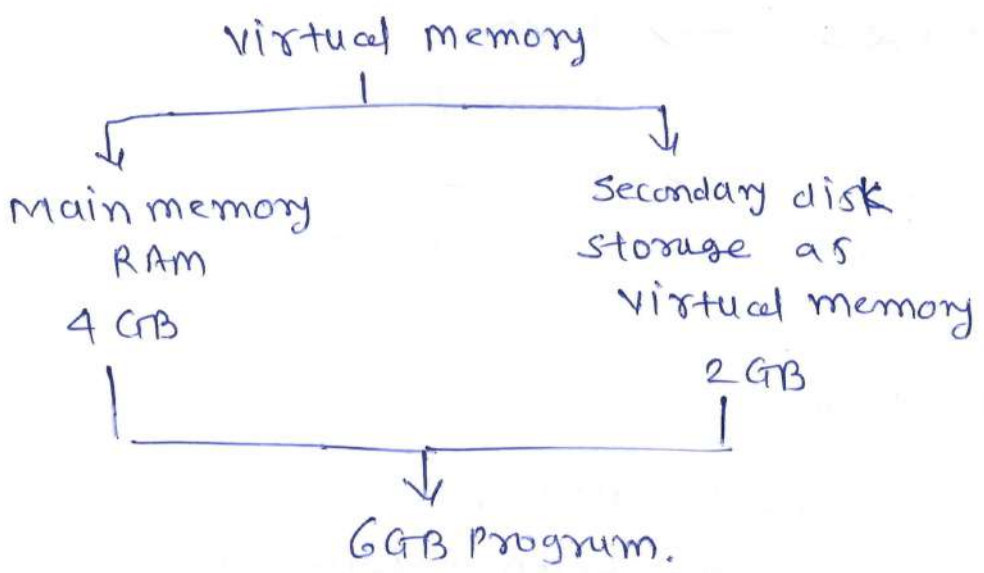
Defn —

virtual memory is a technique in which the operating system uses part of secondary storage as an extension of RAM to run large programs or programs at a same time.

* Need of virtual memory :-

↳ suppose a computer has 4GB RAM but a program requires 6GB memory.

↳ with virtual memory, the operating system can keep the currently required parts of the program in RAM and keep other parts on secondary storage



Thus, the program of 6GB memory requirement can execute even though RAM is smaller than the program's total memory requirement.

* Working of virtual memory :-

Virtual memory generally works using paging

(1) A program is loaded for execution.
(2) only the required part of program is loaded into RAM.
(3) other parts remain on secondary storage.
(4) When the CPU needs a part that is not currently in RAM, the OS brings that part into RAM.
(5) If RAM is full, the OS may move another page to secondary storage.
(6) The required page is then loaded into RAM.

* virtual memory and paging :-

Paging is commonly used to implement virtual memory.
The program is devided into pages and physical RAM is devided into Frames.

e.g. page 0 → frame 3
page 1 → frame 7
page 2 → frame 1

Some pages may remain on disk until they are required.

* Advantages :-

(1) Runs large programs:
programs larger than physical RAM can be executed

(2) Better multiprogramming: more processes can be kept active because only required portions need to be in RAM.

(3) Better memory utilization :- only required pages or segments need to occupy RAM.

(4) supports modern applications :- Large applications can run even when physical RAM is limited.

* Disadvantages

(1) Accessing secondary storage is much slower than accessing RAM.

(2) Frequent page fault reduce performance.

(3) Requires additional memory management hardware and software.

(4) It uses disk space.

(5) Excessive page swapping can cause thrashing.

* Thrashing — Thrashing occurs when OS spends most of its time moving pages between RAM and secondary storage instead of executing processes.

* Demand Paging :-

Demand Paging is a virtual memory technique in which a page is loaded into main memory (RAM) only when it is actually required by the CPU.

↳ The pages that are not currently required remain in secondary storage.

Definition –

Demand Paging is a technique in which pages are loaded into RAM only when they are needed for execution.

* Need of demand paging :-

Demand Paging is used to:

↳ save main memory (RAM)
↳ Run large programs.
↳ Run more processes at the same time.
↳ Reduce unnecessary loading of Pages.
↳ Improve memory utilization.

* Working of demand paging :-

① CPU requests a page

The CPU generates a logical address and requests a particular page.

② Check Page table

The OS checks the page table to determine whether the required page is present in RAM.

③ Page is Present

If the page is already in RAM:
The CPU access the page & execution continues.

④ Page is not present -

If the page is not present in RAM: A page fault occurs.

⑤ Find the page on disk -

The OS locates the required page in secondary storage.

⑥ Load the page into RAM -

The required page is loaded into an available frame.

⑦ Update the page table -

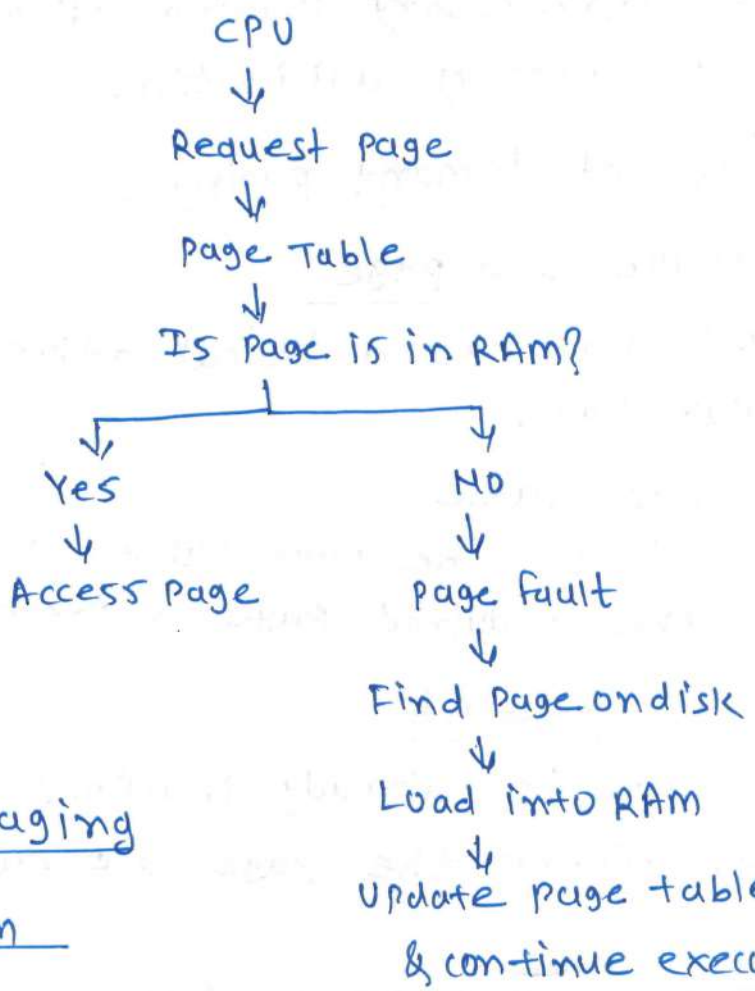
The page table is updated with the new frame information.

⑧ continue execution -

The CPU restarts the interrupted instruction & continue execution.

Demand paging

Diagram



* Page fault :-

A page fault occurs when the CPU tries to access a page that is not currently present in main memory (RAM).

e.g.

Suppose RAM has:

Page 0
Page 2
Page 4

The CPU request for Page 3, as page 3 is not present in RAM page fault occurs.

The operating system brings page 3 from disk into RAM.

* Advantages of demand paging –

1. Saves memory – only required pages are loaded into RAM.
2. Supports large programs – A program can be larger than available physical RAM.
3. Better multiprogramming – more processes kept active because only required pages occupy RAM.
4. Efficient memory utilization – unused pages do not occupy RAM unnecessarily.

* Disadvantages of demand paging –

1. Page fault overhead – A page fault requires disk access, which is much slower than RAM access.
2. Slower execution – Frequent page faults can reduce system performance.

③ Requires extra hardware and software- The system needs mechanisms such as page table and address translation.

④ Thrashing - If the page fault occur very frequently, the system may spend most of its time transfering pages between RAM and disk. This condition is called thrashing.

*Page Replacement algorithms:-

A page replacement algorithm is a technique used by the operating system to decide which page should be removed from main memory (RAM) when a new page is required and no free frame is available.

Definition:-
A page replacement algorithm selects a page from RAM to remove so that the required page can be loaded into memory.

*Need of page replacement:
Page replacement is required during demand paging.

e.g. suppose,
↳ RAM has 3 free frames.
↳ All 3 frames are already occupied.
↳ The CPU requests new page.
↳ The required page is not present in RAM.
↳ Page fault occurs, but there is no free frame.

So OS: ① select a page to remove
② Remove that page from RAM.
③ Load required page into free frame.
④ Update page table.
⑤ continue program execution.

* Main Page Replacement Algorithms:

1. FIFO - First In First out algorithm.
2. LRU - Least Recently Used algorithm.
3. Optimal page replacement algorithm.

* FIFO (First In First out) algorithm:-

Definition:-

FIFO is Page replacement algorithm in which the page that entered in main memory first is removed first when a new page needs to be loaded & all memory frames are occupied.

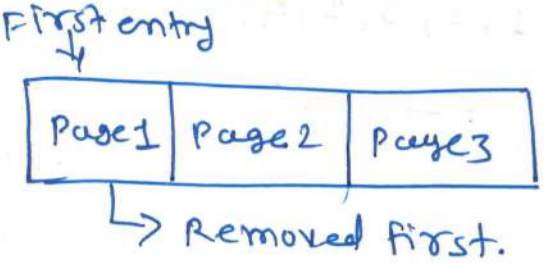
simple Definition

FIFO replace the oldest page in memory with the newly required page.

* working of FIFO:-

⇒ FIFO works like queue.

↳ The page that enters in memory first will be removed first.



e.g. suppose there are 3 frames in main memory.

Referencing string: 1, 2, 3, 4.

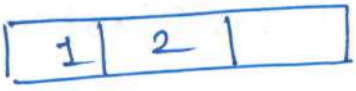
Step1 : page 1

Page Fault

Step 2 : Page 2

Page 2 is not in memory

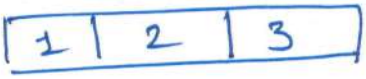
page fault.



Step 3 : Page 3

Page 3 is not in memory

Page fault



Now all frames are ~~Free~~ full.

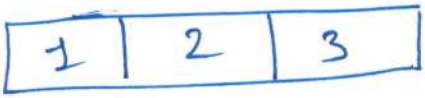
Step 4 : Page 4

Page 4 is required, but no free frame is available

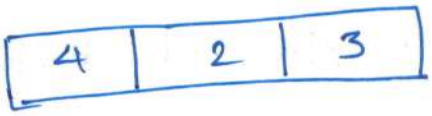
FIFO removes page 1 because page 1 ented in memory first.

So

before



After



So page 1 is replaced by page 4.

e.g. 2

Consider No. of frames = 3.

Reference string = 1, 2, 3, 1, 4.

Reference string	F1	F2	F3	Result
1	1	–	–	page fault
2	1	2	–	page fault
3	1	2	3	page fault
1	1	2	3	Hit
~~4~~	~~4~~	~~2~~	~~3~~	
4	4	2	3	page fault

Explaination

When page 4 is requested:

↳ page 1 entered memory first.

↳ so, FIFO removes page 1.

↳ page 4 is loaded in its place.

* Advantages :-

① FIFO is very easy to understand.

② It can be implemented using a simple queue.

③ It has low overhead as it does not required tracking of pager frequently.

④ It is simple to understand for basic systems.

* Disadvantages :-

① It may remove page which is frequenty used by OS.

② FIFO may produce more page faults than other algorithms.

③ FIFO may suffer from Belady's anomaly, where increasing number of frames can sometimes increase the number of page faults.

* LRU (Least Recently Used) algorithm:-

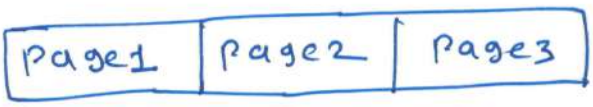
↳ LRU (Least Recently Used) is a page replacement algorithm in which the page that has not been used for the longest period of time, is removed from main memory when a new page is required and all frames are occupied.

simply

LRU replaces the page that has been unused for longest time.

* Working of LRU :-

suppose there are 3 frames:



Operating system used pages in following order,

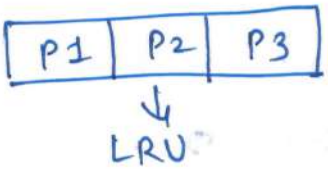
Page3 → Page1 → Page2.

So, Page 2 is least recently used page.

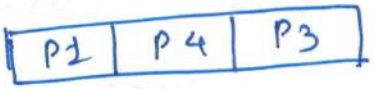
If a new Page4 is required:

It will be replaced as follow:

Before replacement



After Replacement



So, Page2 is replaced by Page4

e.g. consider: No. of frames 3

Reference string = 1, 2, 3, 1, 4.

Reference string	F1	F2	F3	Result
1	1	–	–	Page fault
2	1	2	–	Page fault
3	1	2	3	Page fault
1	1	2	3	Hit
4	1	4	3	Page fault, Page2 Replaced

Total Page Fault – 4

Hit = 1

* Advantages :-

1. It gives better performance than FIFO.
2. It user information about recent page usage.
3. It does not have Belady's anomaly.
4. It has fewer page faultr than FIFO.

* Disadvantages :-

1. It is more difficult to implement than FIFO.
2. It requires tracking the recent usage of pages.
3. Additional hardware or software support may be required.
4. LRU can have additional overhead.

* Optimal page Replacement algorithm :-

Optimal page replacement algorithm is a page replacement techniaue in which the operating system replaces the page that will not be used for the longest period of time in the future.

simply,

Optimal page Replacement replaces the page whose next use is not asked by OS for long time in future.

* Working of optimal page Replacement algorithm

suppose there are 3 frames.

Page1	Page 2	Page 3

Now new page4 is required.

suppose the future reference sequence is:

Page 2 → Page 4 → Page 1 → Page 3.

In future:

Page 1 → used after page 4.

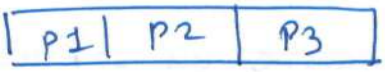
page 2 → used immidiately.

Page 3 → used much later.

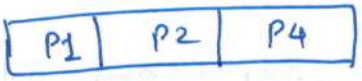
Therefore, page3 will be needed long after in future.

So, optimal replaces page 3.

Before



after



P3 is replaced.

e.g. consider: Number of frames :- 3

Reference string - 1, 2, 3, 4, 2, 1, 5, 2, 3

Reference	F1	F2	F3	Result
1	1	–	–	Fault
2	1	2	–	Fault
3	1	2	3	Fault.
4	1	2	4	Fault 3→Replaced
2	1	2	4	Hit
1	1	2	4	Hit
5	1	2	5	Fault 4 – Replaced.
2	1	2	5	Hit
3	3	2	5	Fault 1 – replaced.

Why page 3 is replaced?

↳ When page 4 is requested.

P1	P2	P3

current status.

In future

P2 → P1 → P5 → P2 → P3.

So P3, is replaced because its use in far in future.

* Important rule :-

When all frames are full:

↳ If a page will never be used again.

↳ Replace that page immediately.

↳ If all pages will be used again, replace page whose next use is fur in the future.

* Advantage -

① For given reference string and fixed no. of frames, it produces minimum possible no. of page faults.

② It provides best performance against which other page replacement algorithms can be compared.

③ It does not suffer from Belady's anomaly.

* Disdvantages -

① Future is unknown :- The OS system can't know exactly which pages to process will request in future.

② Not practically implementable -

As future page reference are unknown, it cannot be practically implemented in real OS.

③ Mainly used for comparison –

It is mainly used as a benchmark to compare algorithms such as FIFO and LRU.