Unit - 4

Memory Management

Page No.
Date

Memory management is the process of controlling and coordinating computer memory, assigning portions called blocks to various running programs to optimize overall system performance.

Memory management is mainly concerned with the allocation of main memory to requesting process. No process can ever run, before a certain amount of memory is allocated to it.

Five requirements of memory management are:

1. Relocation:

Relocation is to find a way to map virtual address into physical address.

When a process is loaded in main memory, since there are several instructions insides the process so here address of these different instructions inside the process are relocatable address, which are convated into actual address by loader in software approach.

2. Sharing:

Any protection mechanism must have the flexibility to allow several processes to access the same portion of main memory.

3. Protection

Memory protection is to prevent a process from accessing memory that has not been allocated to it. Every process should protect against unwanted interference by other processes.

4. Logical Organization

Main memory is organized as linear or one-dimensional space that consists of sequence of bytes or words.

Page No.
Date

Secondary memory at its physical level is similarly organized. Most of the programs are organized into modules.

5. Physical Organization

The computer memory is organized as main memory and secondary memory. The main memory provides fast access. However the secondary memory is slower but can used for long term storage with large capacity.

Address binding

Address binding is the process of mapping the program's logical or virtual addresses to corresponding physical or main memory addresses.

Address binding is part of computer memory management and it is performed by the operating system on behalf of the applications that need access to memory.

Instructions and data to memory addresses can be done in following ways:

i) Compile time:

When it is known at compile time where the process will reside, compile time binding is used to generate the absolute code.

ii) Load time:

When it is not known at compile time where the process will reside in memory, then the compiler generates relocatable code.

iii) Execution time:

If the process can be moved during its execution from one memory segment to another, then binding must be delayed to be done at run time.

Page No.
Date

Dynamic loading

Dynamic loading refers to mapping an executable or library into a process's memory after is has started. In dynamic loading, a routine of program is not loaded until it is called by the program.

Dynamic loading makes better memory space utilization and unused routines are never loaded. It is useful when large amounts of code are needed to handle infrequently occuring cases.

Dynamic Linking

Linking is the process of collecting and combining various modules of code and data into a executable file that can be loaded into memory and executed. OS can link system level libraries to a program.

When it combines the libraries at load time, the linking is called static linking and when this linking is done at the time of execution, it is called as dynamic linking.

Overlays:

Overlays is used to keep in memory only those instructions and data that are needed at any given time. When other instructions are needed, they are loaded into space that was occupied previously by instructions that are no longer needed.

Swapping:

Swapping makes it possible for the total physical address space of all processes to exceed the real physical memory of the system, thus increasing the degree of multiprogramming in a system.

Logical V/S Physical Address Space

Logical address space is set of all logical address generated by program. Physical address space is a set of all physical addresses corresponding to these logical addresses.

Logical address is generated by CPU. Physical address is the address of main memory and it is loaded into the main memory address register.

Q. Consider a logical address space of eight pages of 1024 words, each mapped onto a physical memory of 32 frames then:

a) How many bits are in logical address?

b) How many bits are in physical address?

Sol. The logical address is split into two parts:

the page address and then offset.

Page address = $8 = 2^3$ = 3 address bits.

offset = $1024 = 2^{10}$ = 10 address bits.

Total = 13 address bits.

The physical address is split into two parts:

the frame address and then offset.

frame address = $32 = 2^5$ = 5 address bits

offset = $1024 = 2^{10}$ = 10 address bits

Total = 15 address bits.

Holes in Memory partitioning

The OS keeps a table indicating which parts of memory are available and which are occupied. Initially, all memory is available for user processes and is considered one large block of available memory known as hole. When a part of memory is occupied by a process and left rest of the memory also known as hole. A hole can be created when a process is completed and leaves the memory.

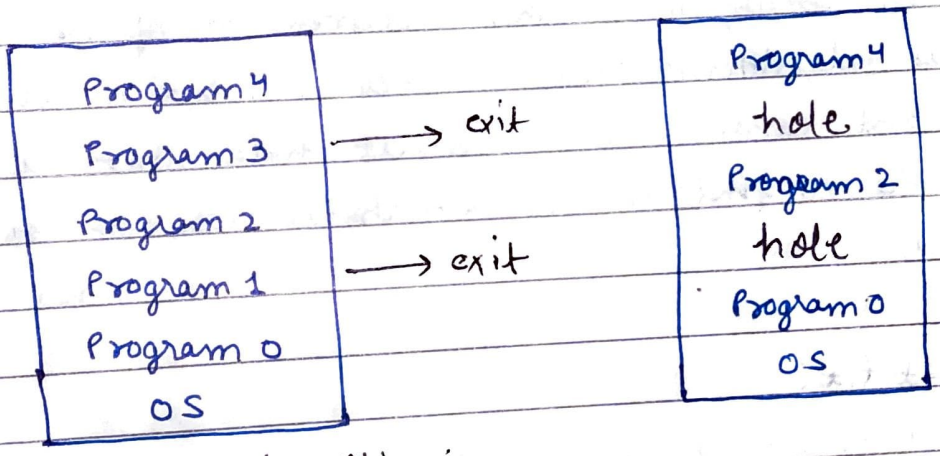
There are two Management techniques: Contiguous and Non-Contiguous.

In contiguous technique, executing process must be loaded entirely in main-memory. Contiguous Technique can be divided into:

1. fixed / Static partitioning
2. Variable/ dynamic partitioning

1) fixed / Static partitioning:

Partition main memory into a set of non-overlapping memory regions called partitions. fixed partitions can be of equal or unequal sizes.



2) Variable / Dynamic partitioning

Initially RAM is empty and partitions are made during the run-time according to process's need instead of partitioning during system configure. The size of partitions will be equal to incoming processes. The partition size varies according to the need of the process so that the internal fragmentation can be avoided to ensure efficient utilisation of RAM.

Page No.
Date

Dynamic Memory Allocations technique

first fit, best fit and worst fit are the the most common strategies used to select a free hole from the set of available holes.

1. first fit:

In the first fit approach is to allocate the first free partition or hole large enough which can accomodate the process. It finishes after finding the first suitable free partition.

2. Best fit:

The best fit deals with allocating the smallest free partition which meets the requirement of the requesting process. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

3. Worst fit:

In worst fit approach is to locate largest available free portion so that the portion left will be big enough to be useful. It is the reverse of best fit.

Internal Fragmentation

It somehow relates to fixed size partitioning. The system allocates memory to various programs and processes by dividing them into small blocks as required by the program. However, more memory is allocated sometimes than is needed by the process, which eventually results in excess memory going waste or left unused.

Page No.
Date

external fragmentation

The main memory forms holes between portions of allocated memory that are too small to hold any process. It's downside of storage allocation algorithms, when contiguous blocks of unused space cannot serve a new request because the spaces are too small for large memory application needs.

In simple terms, the non-contiguous blocks create holes in the memory resulting in unused storage that are outside the allocated regions, meaning it cannot be used along with the main memory for larger memory tasks.

They end up being isolated and cannot be totally eliminated from the memory space. This is called external fragmentation. It can be removed by compaction which shuffles contents of the memory to place all free memory together.

Q. Given memory partitions of 100K, 500K, 200K, 300K and 600K (in order). How would each of the first fit, best fit and worst fit algorithms place processes of 212K, 417K, 112K and 426K ? which algorithm makes the most efficient use of memory?

Sol

Process	Memory
P_1	212K
P_2	417K
P_3	112K
P_4	426K

first fit:

Memory partition	Process Number	Memory Requested	Status	Internal fragmentation
100K	-	-	free	-
500K	P_1	212K	Busy	288K
200K	P_3	112K	Busy	88K
300K	-	-	free	-
600K	P_2	417K	Busy	183K

Total available: 1700K Total used: 741K

Page No.
Date

Best fit:

Memory Partition	Process Number	Memory Requested	Status	Internal fragmentation
100 K	–	–	free	–
500 K	P_2	417 K	Busy	83 K
200 K	P_3	112 K	Busy	88 K
300 K	P_1	212 K	Busy	88 K
600 K	P_4	426 K	Busy	174 K

Total available: 1700 K Total used: 1167 K.

Worst fit:

Memory partition	Process Number	memory Requested	Status	Internal Fragmentation
100 K	–	–	free	–
500 K	P_2	417 K	Busy	83 K
200 K	–	–	free	–
300 K	P_3	112 K	Busy	188 K
600 K	P_1	212 K	Busy	388 K

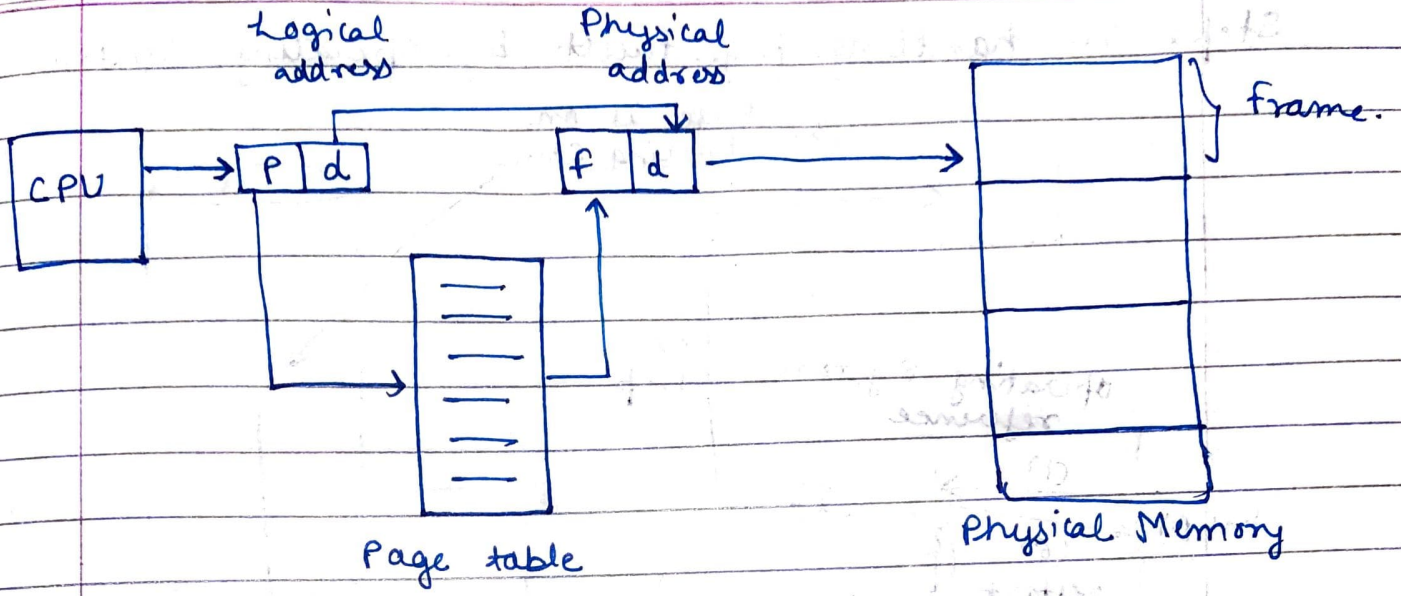
Total available: 1700 K Total used: 741 K.

Best fit algorithm makes most efficient use of memory, because in this, all the processes are allocated memory partition with minimum internal fragmentation.

Paging

Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non-contiguous. Paging avoids external fragmentation and the need for compaction.

- Logical address: An address generated by the CPU.
- Logical address space: The set of all logical addresses generated by a program.
- Physical address: An address actually available on memory unit.
- Physical address space: The set of all physical addresses corresponding to the logical addresses.



The mapping from virtual to physical address is done by memory management unit which is a hardware device and this mapping is known as paging technique.

- The physical address space is conceptually divided into a number of fixed-size blocks called frames.
- The logical address space is also splitted into fixed-size blocks called pages.

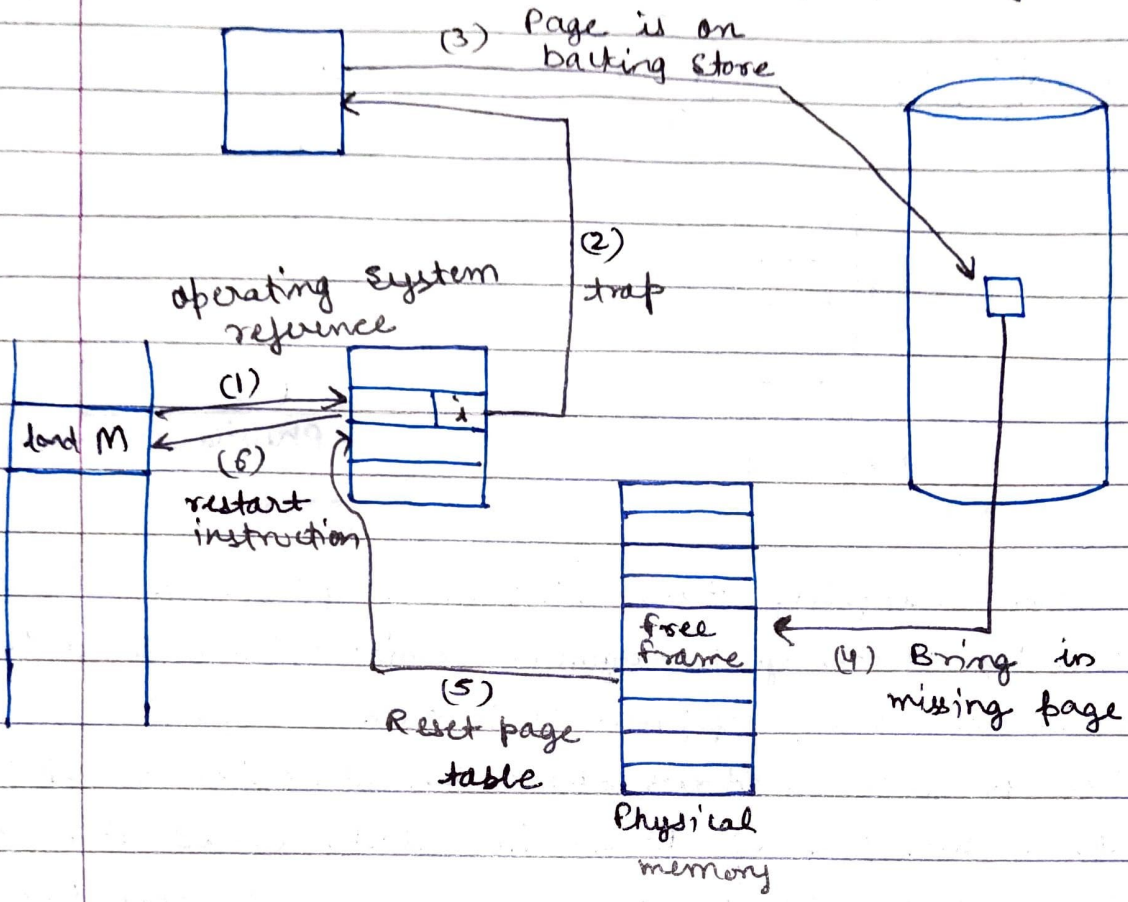
Page size = frame size.

Page fault or Page error:

An interrupt that occurs when a program requests data that is not currently in real memory. The interrupt triggers the OS to fetch the data from a virtual memory and load it into RAM. An invalid page fault or page error occurs when the OS cannot find the data in virtual memory.

A page fault occurs when an access to a page that has not been brought into main memory takes place.

Steps in handling page fault by operating system



Segmentation

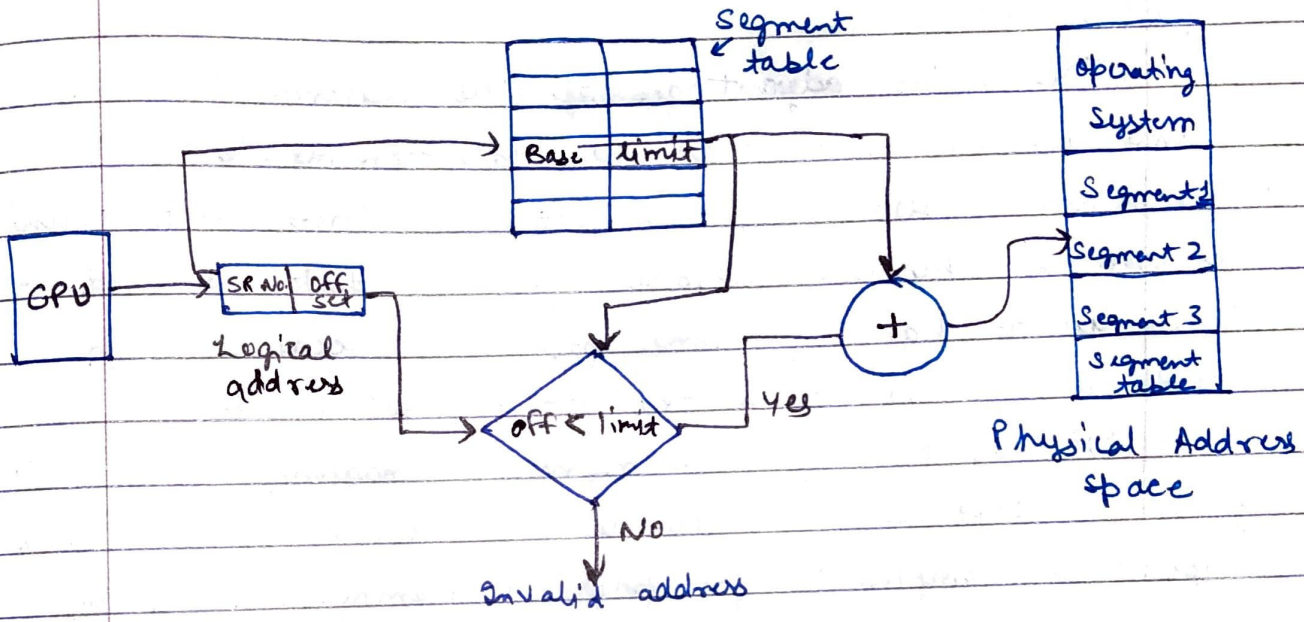
Segmentation is a memory-management scheme that supports the programmer view of memory. A logical address space is a collection of segments. Each segment has a name and a length.

The address specify both the segment name and the offset within the segment. The programmer therefore specifies each address by two quantities : a segment name and an offset.

For simplicity of implementation, segments are numbered and are referred to by segment number. Segment number indexes a process segment table. Each entry in segment table has a segment base and segment limit. Segment base contains the physical address where segment resides in memory, whereas segment limit specifies the length of segment.

Page No.
Date

Segmentation is a memory management technique in which, the memory is divided into variable size parts. Each part is known as segment which can be allocated to a process. The details about each segment are stored in a table called segment table.



Segmentation	Paging
1) Program is divided into variable size segments.	1) Program is divided into fixed size pages.
2) User is responsible for dividing program into segments.	2) OS is responsible for dividing pages.
3) It is slower	3) It is faster.
4) It is visible to users.	4) It is not visible to user.
5) It eliminates internal fragmentation	5) It can't eliminates internal fragmentation.
6) It can't eliminates external fragmentation	6) It can eliminates external fragmentation.
7) OS maintains a list of free holes in memory.	7) OS maintains a list of free frame.

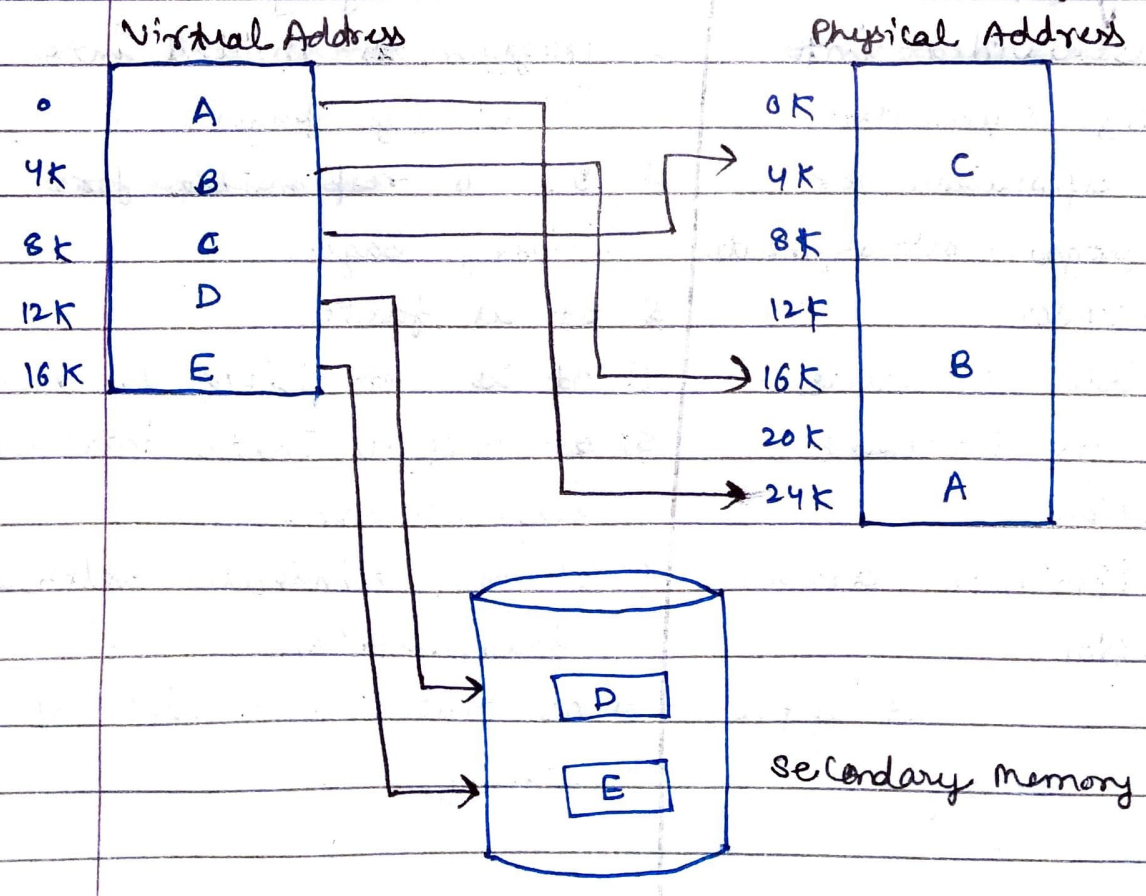
Page No.
Date

Virtual Memory

A computer can address more memory than the amount physically installed on the system. This extra memory is actually called virtual memory and it is a section of hard disk that's set up to emulate the computer's RAM.

The main advantage of this scheme is that programs can be larger than physical memory. Virtual memory serves two purposes. first, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection because each virtual address is translated to a physical address.

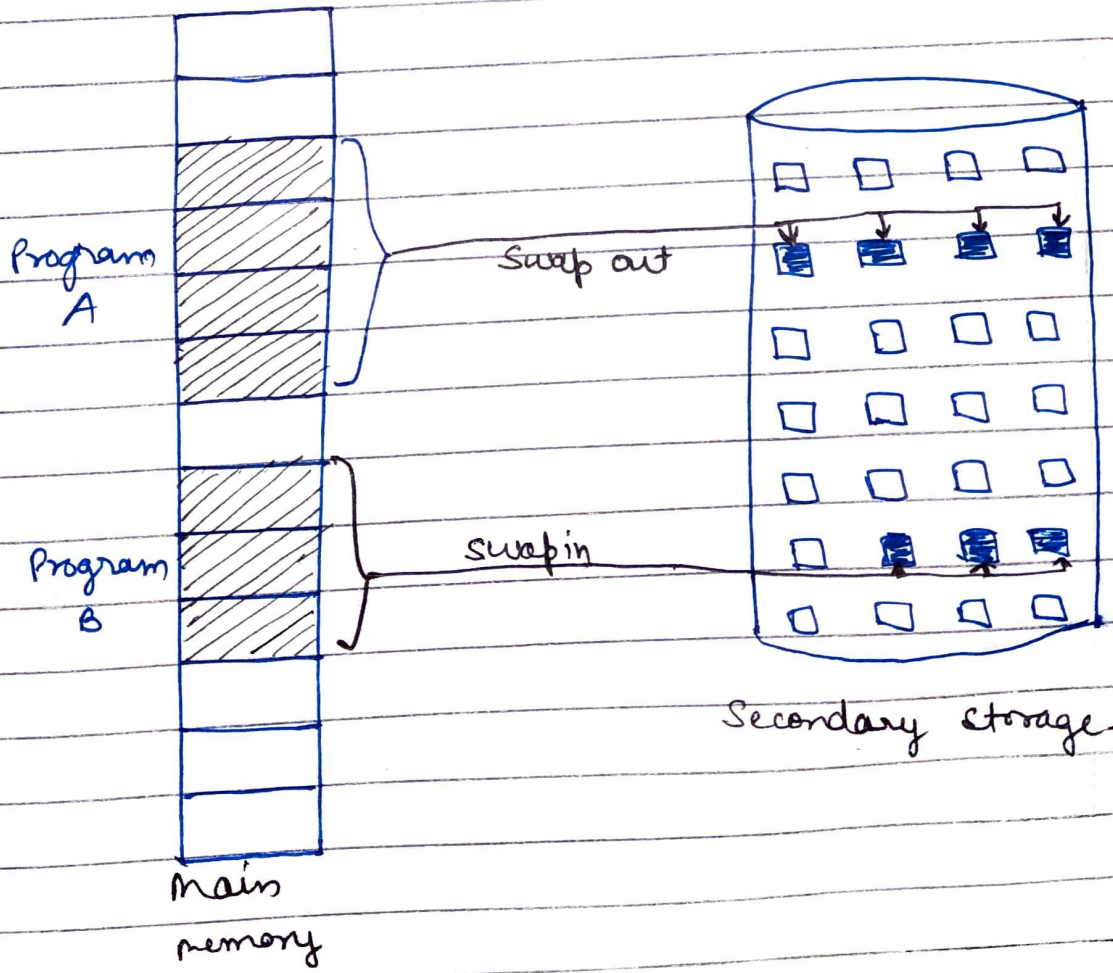
Virtual memory is a memory management capability of an OS that uses hardware and software to allow a computer to compensate for physical memory shortage by temporarily transferring data from RAM to disk storage.



Demand Paging

The process of loading the page into memory on demand whenever page fault occurs is known as demand paging.

Process: If CPU tries to refer a page that is currently not available in the main memory, it generates an interrupt indicating memory access fault. The OS puts interrupted process in a blocking state. for the execution to proceed the OS must brings the required page into the memory. The OS will search for the required page in the logical address space. The required page will be brought from logical address space to physical address space. The page table will updated automatically. The signal will be sent to the CPU to continue the program execution and it will place the process back into ready state.



Page Replacement Algorithm

(1) First in first Out (FIFO):

A FIFO replacement algorithm associates with each page, the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen. we can create a FIFO queue to hold all pages in memory. we replace the page at the head of the queue. when a page is brought into memory, we insert it at the tail of the queue.

(2) Optimal Replacement Algorithm:

The optimal policy selects for replacement that page for which the time to the next reference is the longest. An optimal page replacement algorithm has the lowest page fault rate of all algorithms. This algorithm is impossible to implement because it would require the operating system to have perfect knowledge of future events.

(3) LRU page replacement:

If we use the recent past as an approximation of the near future, then we can replace the page that has not been used for the longest period of time.

This approach is the least recently used (LRU) algo.

Q. Consider the following reference string
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
i) FIFO
ii) LRU page replacement algorithm?
Assuming three and four frames in each case and frames are initially empty.

Sol. i) FIFO

3 frame

1	2	3	4	(2)	1	5	6	2	1
1	1	1	4	4	4	4	6	6	6
	2	2	2	2	1	1	1	2	2
		3	3	3	3	5	5	5	1

(2)	3	7	6	(3)	2	1	(2)	3	6
6	3	3	3	3	2	2	2	2	6
2	2	7	7	7	7	1	1	1	1
1	1	1	6	6	6	6	6	3	3

Page fault = 16

4 frame.

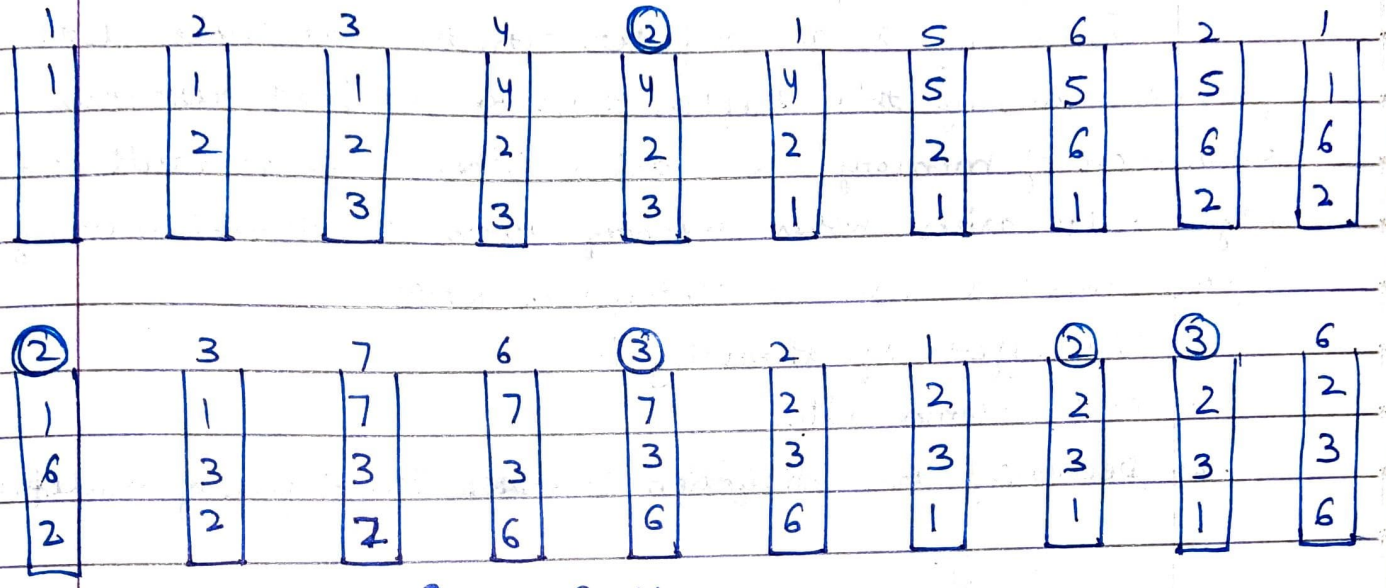
1	2	3	4	(2)	(1)	5	6	2	1
1	1	1	1	1	1	5	5	5	5
	2	2	2	2	2	2	6	6	6
		3	3	3	3	3	3	2	2
			4	4	4	4	4	4	1

(2)	3	7	6	(3)	2	1	(2)	3	(6)
5	3	3	3	3	3	1	1	1	1
6	6	7	7	7	7	7	7	3	3
2	2	2	6	6	6	6	6	6	6
1	1	1	1	1	2	2	2	2	2

Page faults = 14

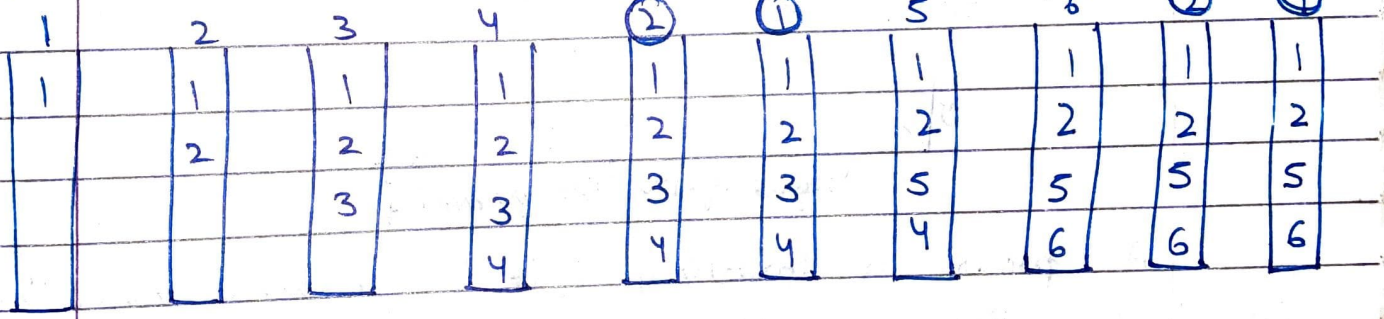
ii) LRU page replacement algorithm.

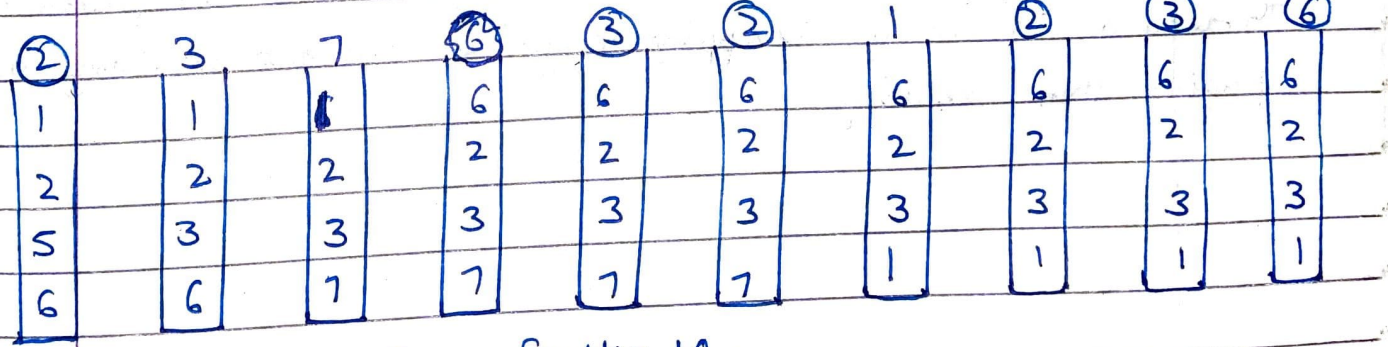
3 frame:



Page faults = 15.

4 frame:





Page fault = 10.

Page No.
Date

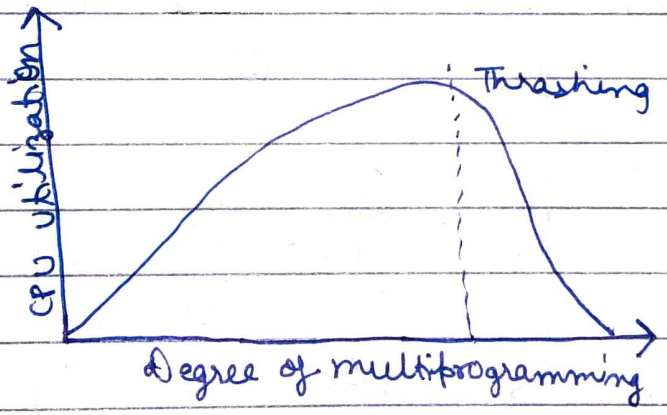
Thrashing:

Thrashing occurs when a system spends more time in processing page faults rather than executing instructions.

Thrashing is a computer activity that makes little progress. Usually this happen due to limited resources or exhaustion of memory. It arises when a page fault occurs. Page fault arises when memory access of virtual memory space does not map to content of RAM.

The effect of thrashing is:

1. CPU becomes idle.
2. Decreasing the utilization increases the degree of multiprogramming



A process is thrashing if it is spending more time in paging than execution. Thrashing is discrete phenomenon. Thrashing is busy swapping pages in and out or doing more paging then executing.