Chapter: Memory Management

External Fragmentation

- When there is enough total memory space to satisfy a request, but the available space is not contiguous.
- Solution
  - Compaction – shuffle the memory contents so as to place all free memory together in one large block.

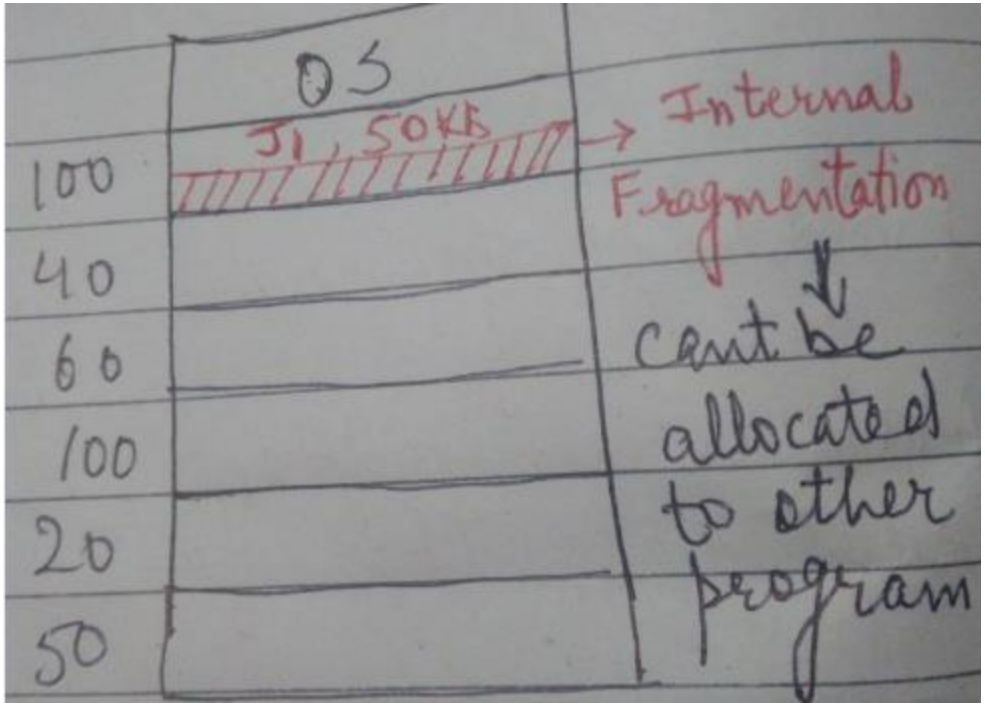
Partition Allocation

One method of allocating contiguous memory is to **divide** all available **memory into equal sized partitions**, and to assign each process to their own partition.

Partition Allocation Algorithms:

a) First Fit:

- Checks all partitions serially
- When partition with size = or > encounters, it is allocated for storage.



Partition Allocation Algorithms

b) Best Fit: This approach will check all the free partitions and will allocate that free partition to a process which leads to minimum internal fragmentation.

Disadvantage:

1) Complexity is more
2) Overhead to check all the partitions to find best suitable space

c) Worst Fit: Allocate the largest memory hole

Fragmentation

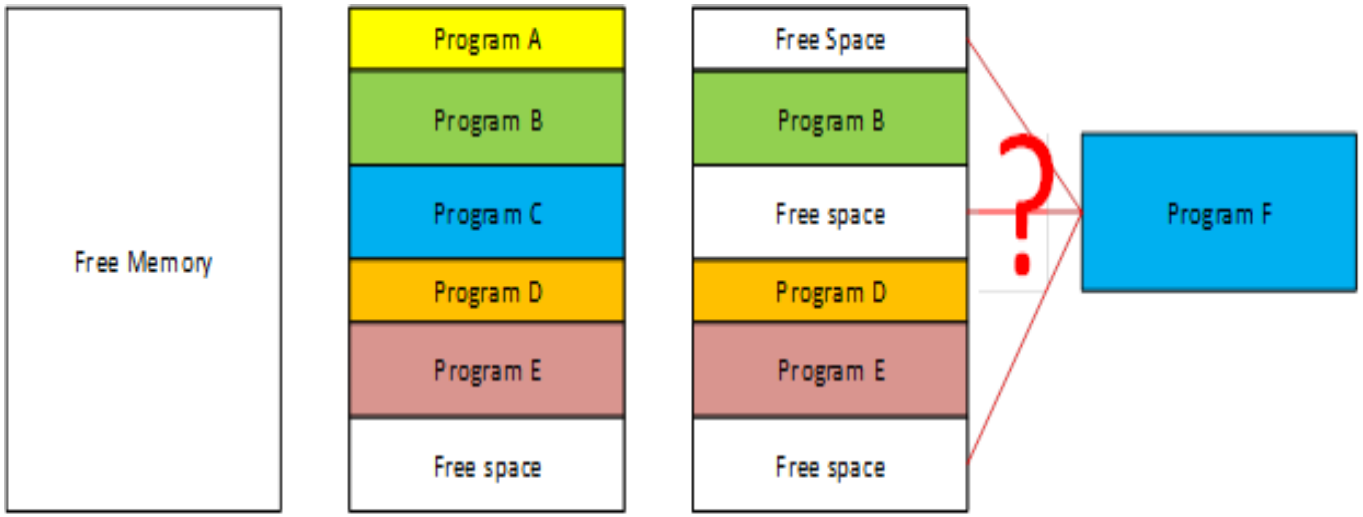
1. Internal Fragmentation: memory is allocated in blocks of a **fixed size**, whereas the actual memory needed will rarely be that exact size.

Partition of greater size is allocated to process of small size, rest of the space of that partition is wasted.

Fragmentation

External Fragmentation: means that the available memory is broken up into lots of little pieces, none of which is big enough to satisfy the next memory requirement, although the sum total could.

- Due to non utilization of space, even space is empty



Fragmentation

- **For Fixed sized partitions**

There is:

- Internal Fragmentation
- External Fragmentation

- **For Variable sized partitions**

There is:

- External Fragmentation

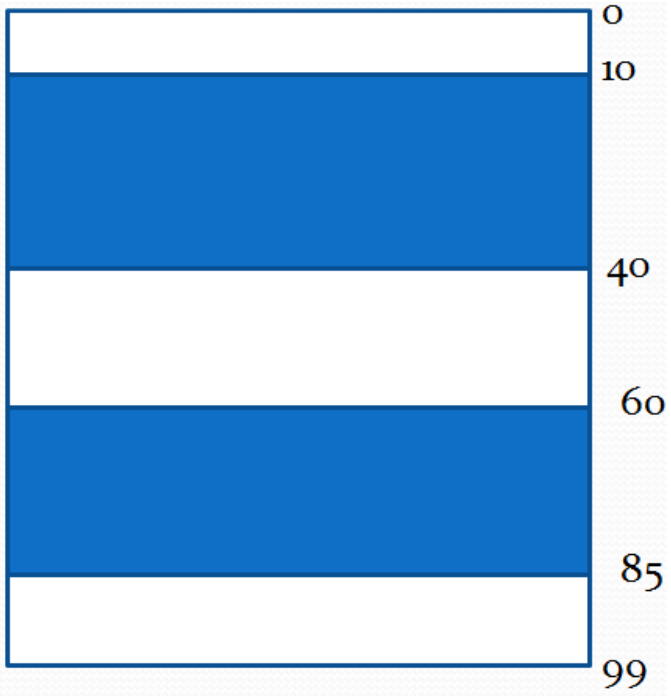
Partition Allocation Table

Contains information of space/ memory used by memory management module for allocation of memory to a process

a) No. of partitions in main memory
b) Size of each partition
c) Starting location of partition
d) Status of partition (allocated or free)

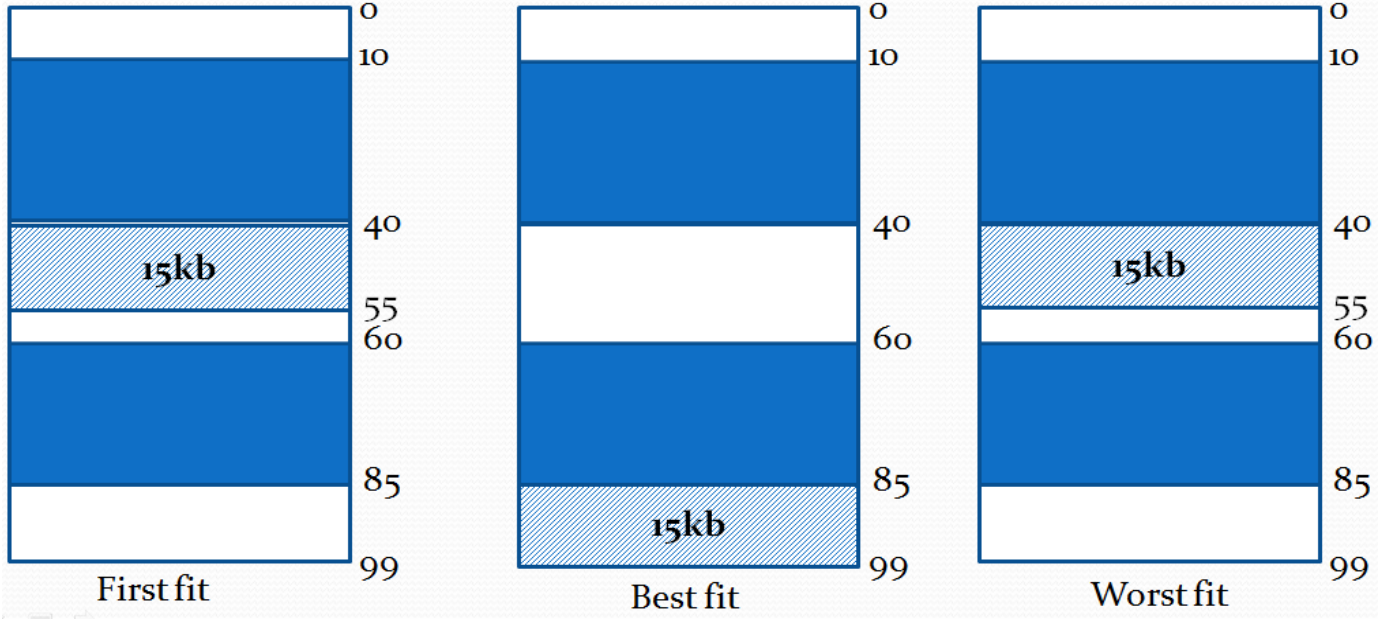
Memory Allocation - Example

Given 3 free memory partitions of 10KB, 20 KB and 15KB (in order) How would each of the first-fit, best- fit and worst fir algorithms place processes of 15KB, 10KB, 20KB and 5KB (in order)



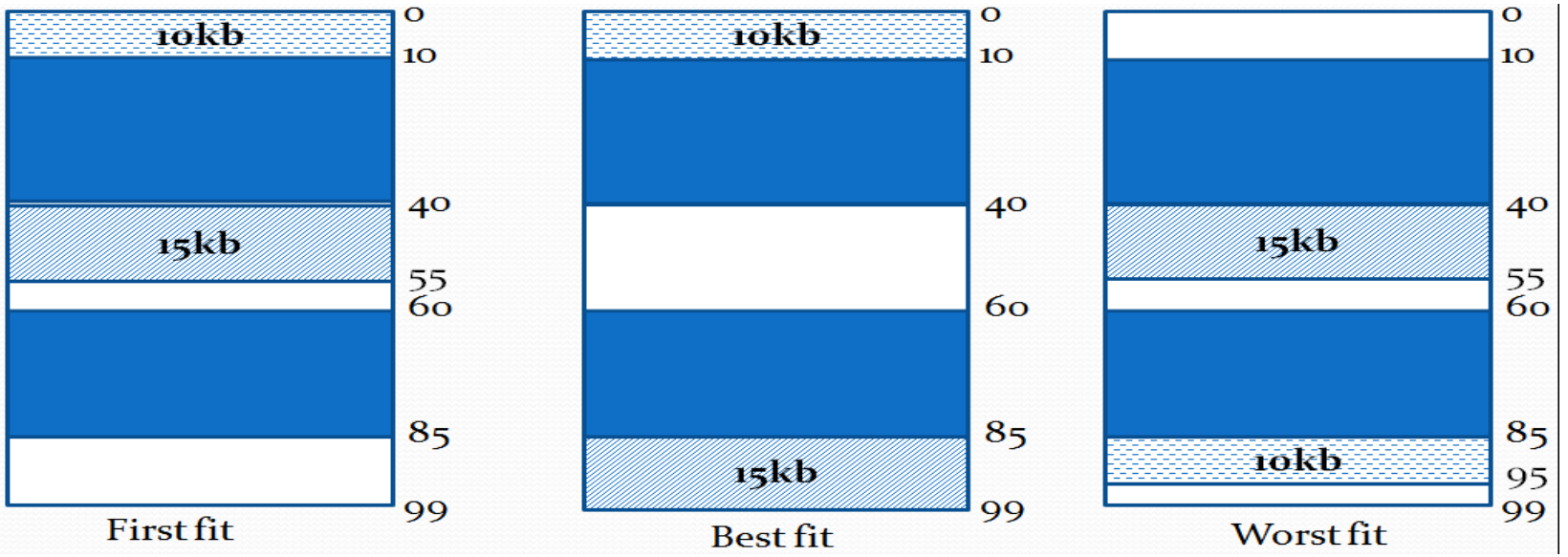
Memory Allocation - Example

For 15KB



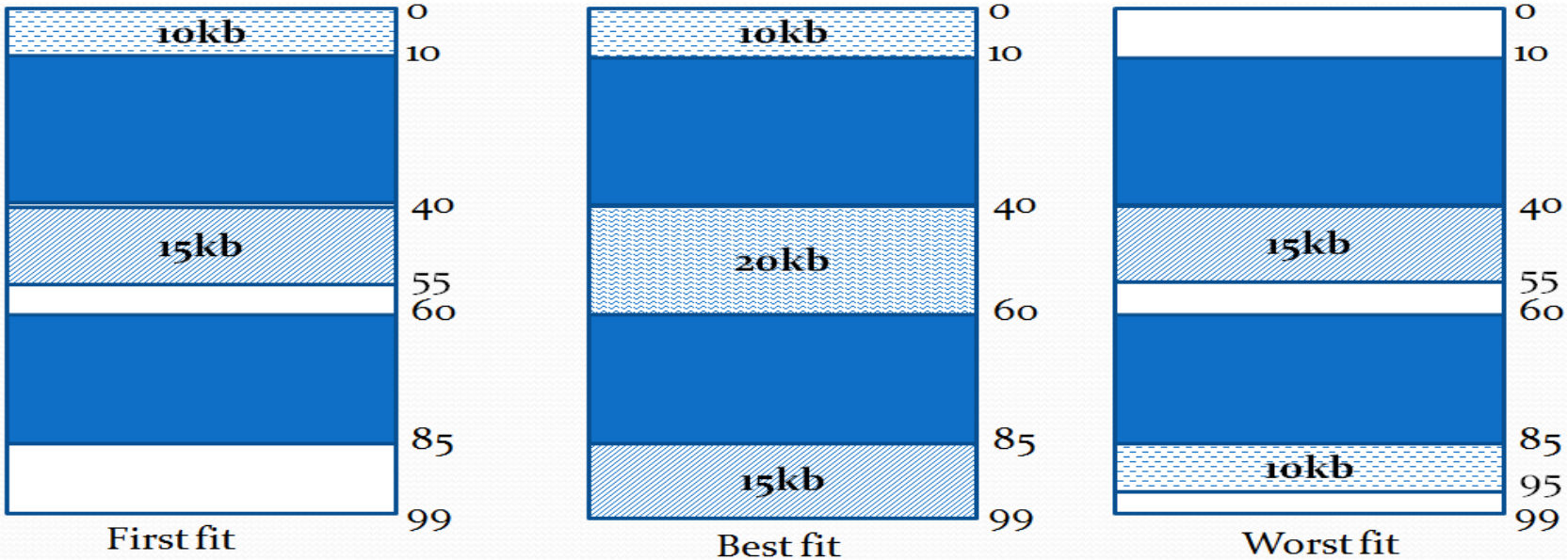
Memory Allocation - Example

For 10KB



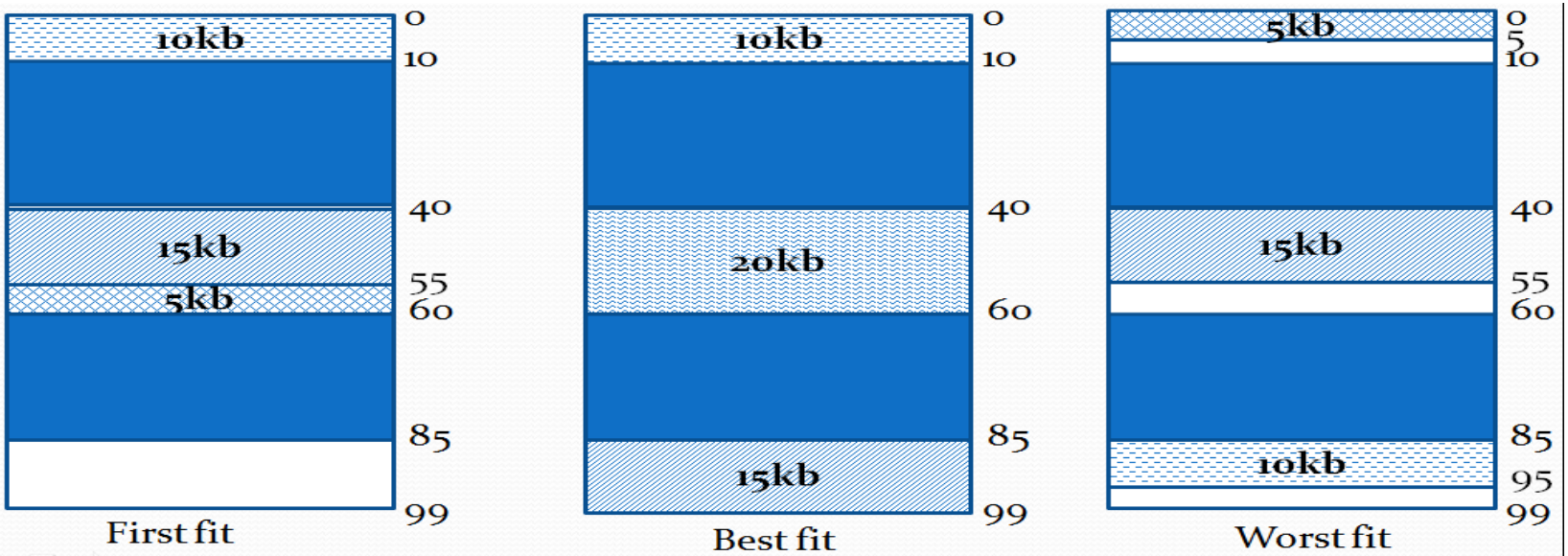
Memory Allocation - Example

For 20KB



Memory Allocation - Example

For 5KB



Memory Allocation - Example

Given four free memory partitions of 150 KB, 200 KB , 100 KB and 50 KB(in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 150 KB, 100 KB, 200 KB and 50 KB (in order)? Which algorithm performs the best?

Overlay

- The main problem in Fixed partitioning is the size of a process has to be limited by the maximum size of the partition.
- In order to solve this problem, solution used is called as Overlays.
- The concept of **overlays** is that whenever a process is running it will not use the complete program at the same time, it will use only some part of it.
- whatever part you required, you load it an once the part is done, then you just unload it, means just pull it back and get the new part you required and run it.

Overlay

- “The process of **transferring a block** of program code or other data into internal memory, replacing what is already stored”.

- overlay is a technique to run a program that is bigger than the size of the physical memory by keeping only those instructions and data that are needed at any given time.Divide the program into modules in such a way that not all modules need to be in the memory at the same time.

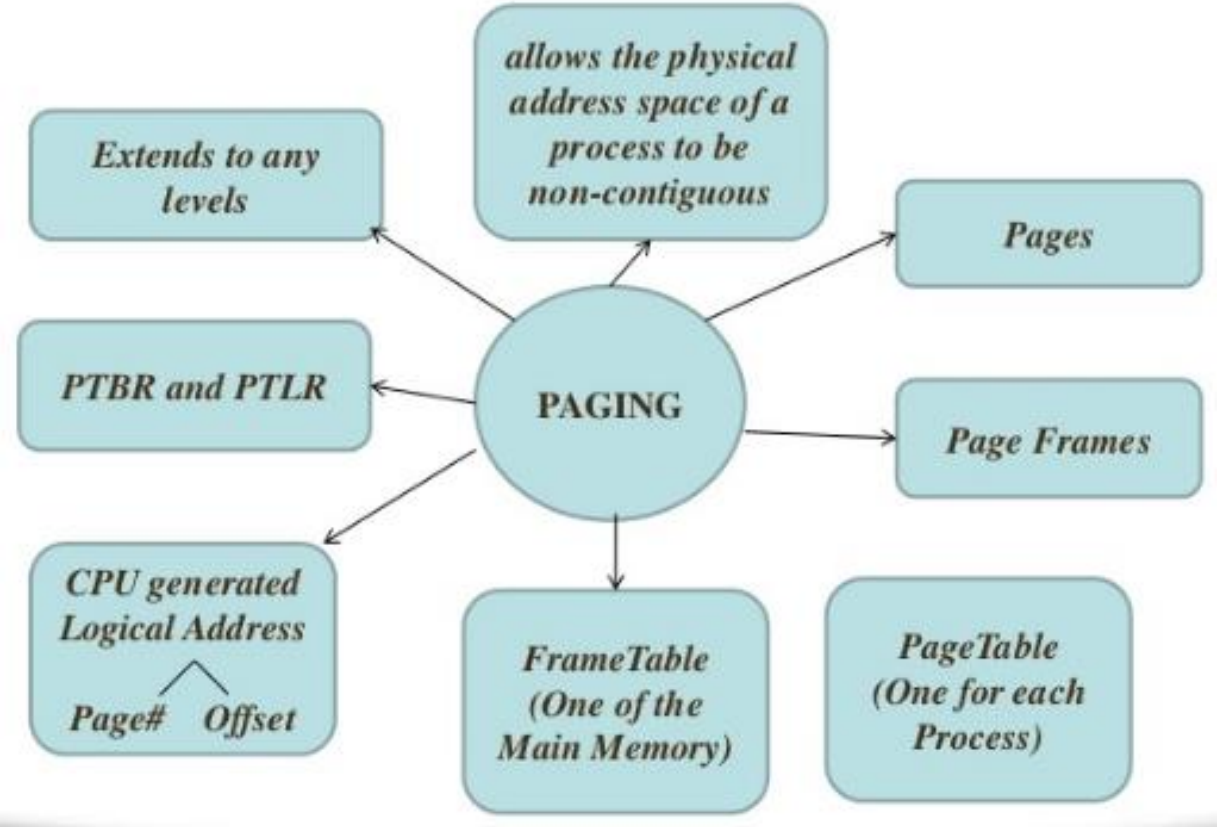
Overlay

- Keep only those instructions and data into the memory that are needed at some time.
- Overlay of instructions or data is needed when process is larger than amount of memory allocated to it.
- It is implemented by user, No special support is needed from the OS.
- Complex to implement.

Paging / Paged Memory Management

– Paging is a memory management scheme that allows processe's physical memory to be discontinuous, and eliminates problems with fragmentation by allocating memory in equal sized blocks known as ***pages***.

Paging


allows the physical address space of a process to be non-contiguous
Extends to any levels
Pages
PTBR and PTLR
PAGING
Page Frames
CPU generated Logical Address
Page#
Offset
FrameTable (One of the Main Memory)
PageTable (One for each Process)

Paging / Paged Memory Management

- Every process is divided in to number of pages
- Memory is divided into partitions whose size is same as page size : **frames**
- Each frame has frame no.
- Page size is same as frame size
- Put any page in any free frame
- Paging allows non-contiguous memory allocation
- Page numbers, frame numbers and frame sizes are determined by the machine architecture
- **Paging leads to Internal Fragmentation**

Pages size and Frame size is always in powers of two

Paging / Paged Memory Management

- CPU generates logical address and put the pages into random frames
- **Mapping** is required to map which page is stored in which page number
- From frame no. physical address could be found

Paging / Paged Memory Management

- Physical address space of a process in non- contiguous
- **Implementation:**
  - **Frames: Fixed sized blocks of the physical memory**
  - **Pages: Fixed sized slots of the logical memory**
- When a process is to be executed, its pages are loaded into any available memory frames.
- **Page Table: is a data structure.** Used to translate logical address to physical address
- CPU generated logical addresses to fetch the instructions.

Address Translation

– Address generated by CPU is divided into:

a) Page Number (p): used as an **index into page table,** which contains base address of each page in the physical memory.

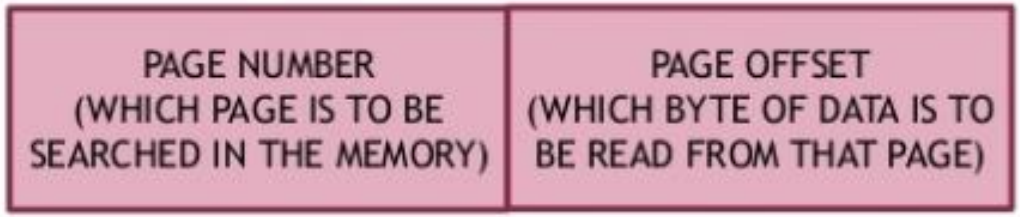
b) Page Offset (d): combined with base address to **define physical memory address** that is sent to the memory unit.

Actual address of any byte in page or frame (Position of instruction in page or frame)

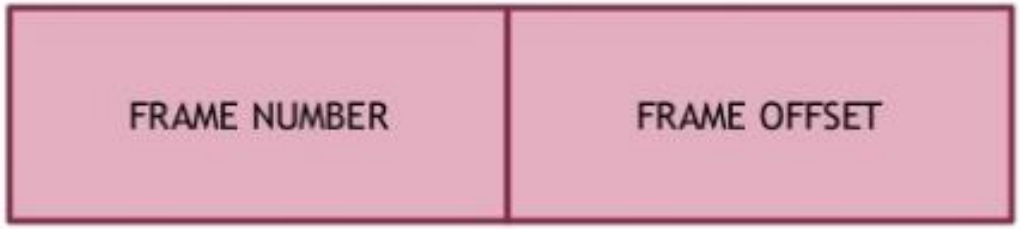
- Address of physical memory, where page resides.
- The number of bits in the offset determines the maximum size of each page, and should correspond to the system frame size.

Address Translation

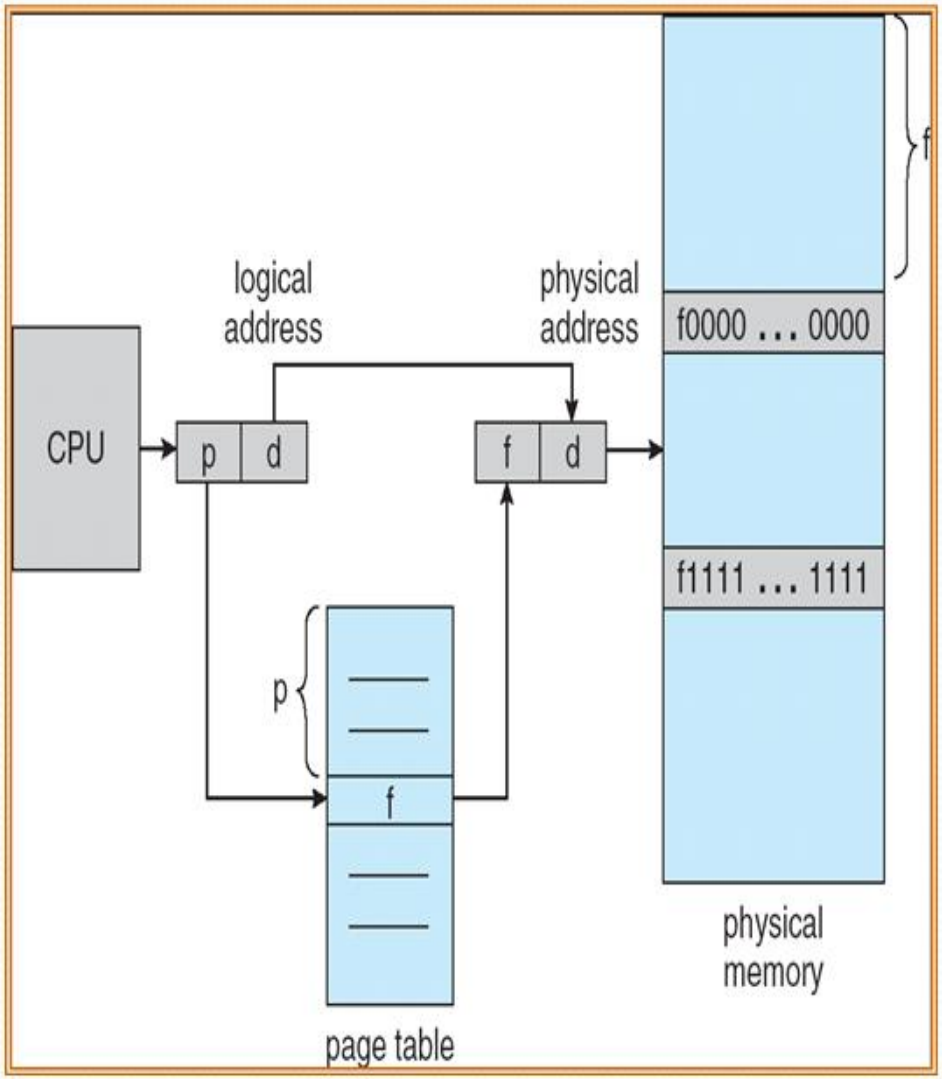
LOGICAL ADDRESS

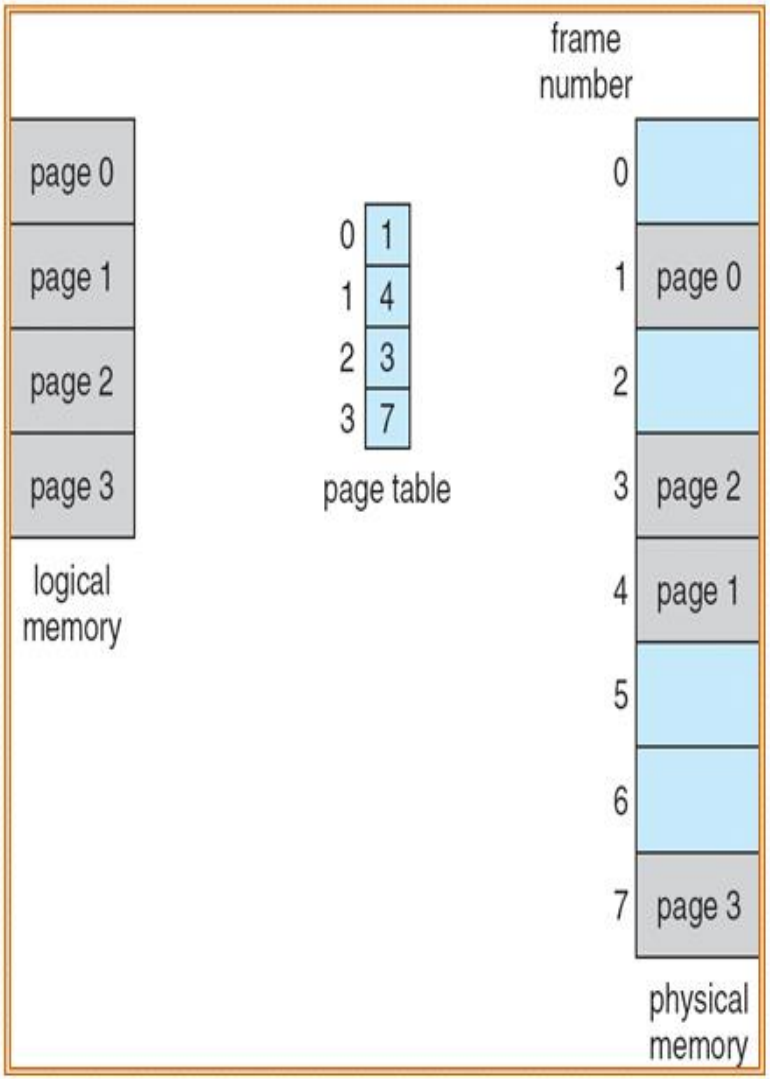


PHYSICAL ADDRESS



Address Translation





Translation of Address to memory

How many address combinations can be generated from bits:

1 bit → 2^1=2 i.e 0 or 1

2 bit → 2^2=4 i.e 00,01,10,11 ……..so on

n bits → 2^n combination of address locations can be generated.

If we have n bit address, and system is byte addressable (every location/partition is of 1 Byte) it will support memory of:

2^n x 1Byte

Translation of Address to memory

Translate n bit address to memory

Q1. Assume memory is byte addressable and address is of size 14bit. Compute size of memory.

Ans

Using 14 bits how many address combinations we can generate? 2^14.

2^4 x 2^10 =16 K *1 Byte

= 16KB