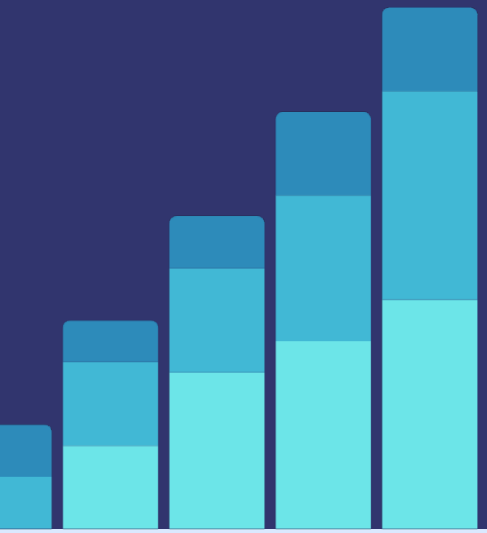
Memory Management

Chapter 3



What is Memory Management

It ensures that blocks of memory space are properly managed and allocated so the operating system (OS), applications and other running processes have the memory they need to carry out their operations.

What do you mean by memory management?

Memory is the important part of the computer that is used to store the data. Its management is critical to the computer system because the amount of main memory available in a computer system is very limited.

At any time, many processes are competing for it. Moreover, to increase performance, several processes are executed simultaneously.

For this, we must keep several processes in the main memory, so it is even more important to manage them effectively.

Role of Memory management

Memory manager is used to keep track of the status of memory locations, whether it is free or allocated. It addresses primary memory by providing abstractions so that software perceives a large memory is allocated to it.

Memory manager permits computers with a small amount of main memory to execute programs larger than the size or amount of available memory. It does this by moving information back and forth between primary memory and secondary memory by using the concept of swapping

Memory Management (Hardware)

In hardware, memory management involves components that physically store data, such as RAM (random access memory) chips, memory caches, and flash-based SSDs(solid-state drives).

Memory Management (OS)

In the OS, memory management involves the allocation (and constant reallocation) of specific memory blocks to individual programs as user demands change.

Memory Management (Application)

Allocation

When the program requests a block of memory, the memory manager must allocate that block out of the larger blocks it has received from the operating system. The part of the memory manager that does this is known as the allocator.

Recycling

When memory blocks have been allocated, but the data they contain is no longer required by the program, then the blocks can be recycled for reuse. There are two approaches to recycling memory: either the programmer must decide when memory can be reused (known as manual memory management); or the memory manager must be able to work it out (known as automatic memory management).

Binding of Instructions and Data to Memory

- **Compile time**: If memory location known a priori, **absolute code** can be generated; must recompile code if starting location changes
- **Load time**: Must generate **relocatable code** if memory location is not known at compile time
- **Execution time**: Binding delayed until run time if the process can be moved during its execution from one memory segment to another

Process Address Space

S.N	Memory Addresses & Description
1	Symbolic addresses The addresses used in a source code. The variable names, constants, and instruction labels are the basic elements of the symbolic address space.
2	Relative addresses At the time of compilation, a compiler converts symbolic addresses into relative addresses.
3	Physical addresses The loader generates these addresses at the time when a program is loaded into main memory.

Logical Address

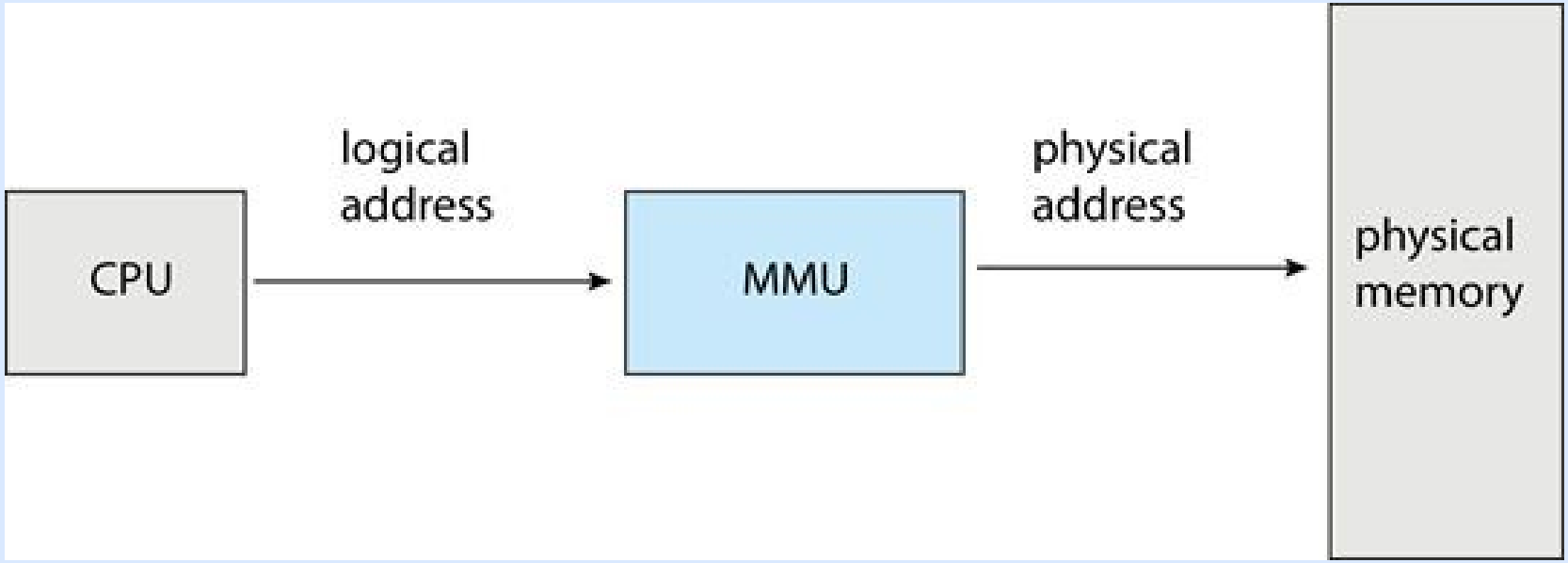
It is a virtual address generated by the CPU while a program is running. It is referred to as a virtual address because it does not exist physically. Using this address, the CPU access the actual address or physical address inside the memory, and data is fetched from there.

Physical Address

Physical Address is the actual address of the data inside the memory. The logical address is a virtual address and the program needs physical memory for its execution. The user never deals with the Physical Address. The user program generates the logical address and is mapped to the physical address by the Memory Management Unit(MMU).

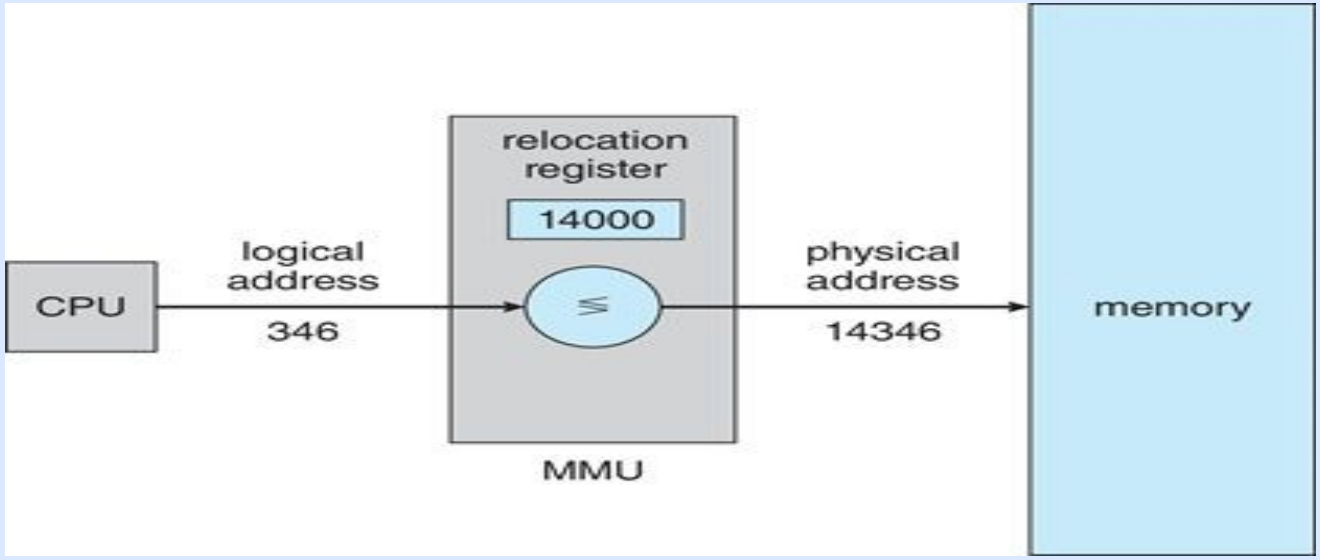
Memory-Management Unit (MMU)

Hardware device that at run time maps virtual to physical address

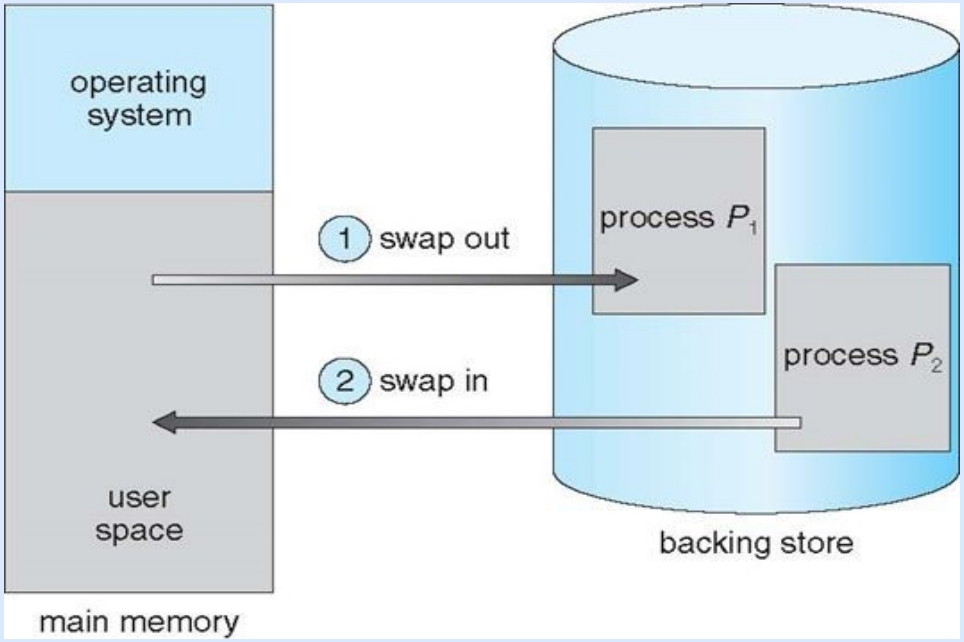


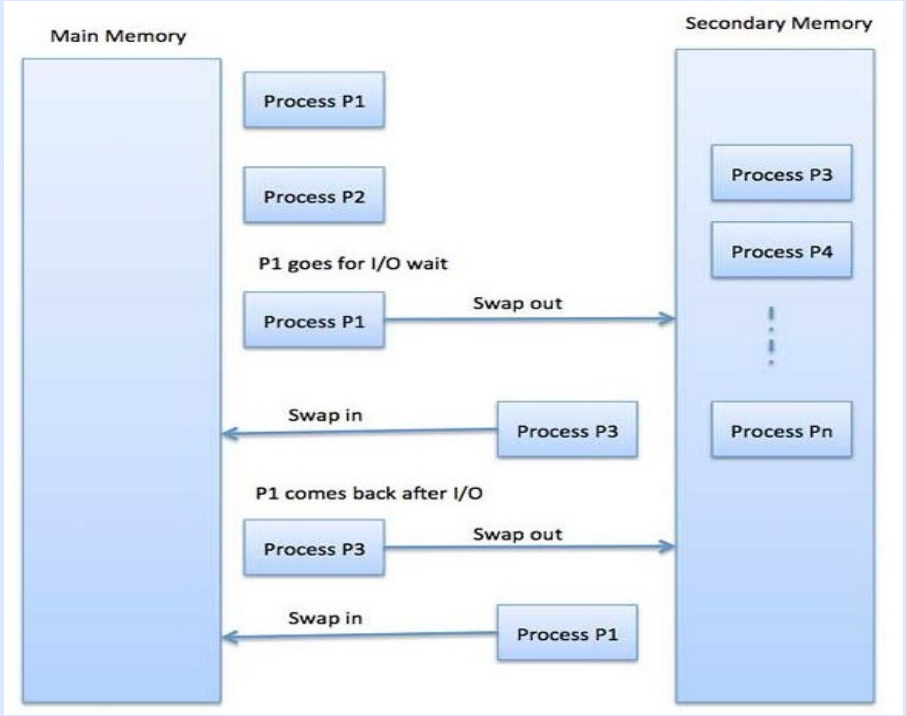
Memory-Management Unit (MMU)

Hardware device that at run time maps virtual to physical address



Swapping


operating system
1 swap out
2 swap in
process P_1
process P_2
user space
backing store
main memory


Main Memory
Secondary Memory
Process P1
Process P2
Process P3
Process P4
P1 goes for I/O wait
Process P1
Swap out
Swap in
Process P3
Process Pn
P1 comes back after I/O
Process P3
Swap out
Swap in
Process P1

Fragmentation

Fragmentation refers to the process or state where something is broken into smaller, disconnected parts. Fragmentation is of two types:

External fragmentation

Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous, so it cannot be used.

Internal fragmentation

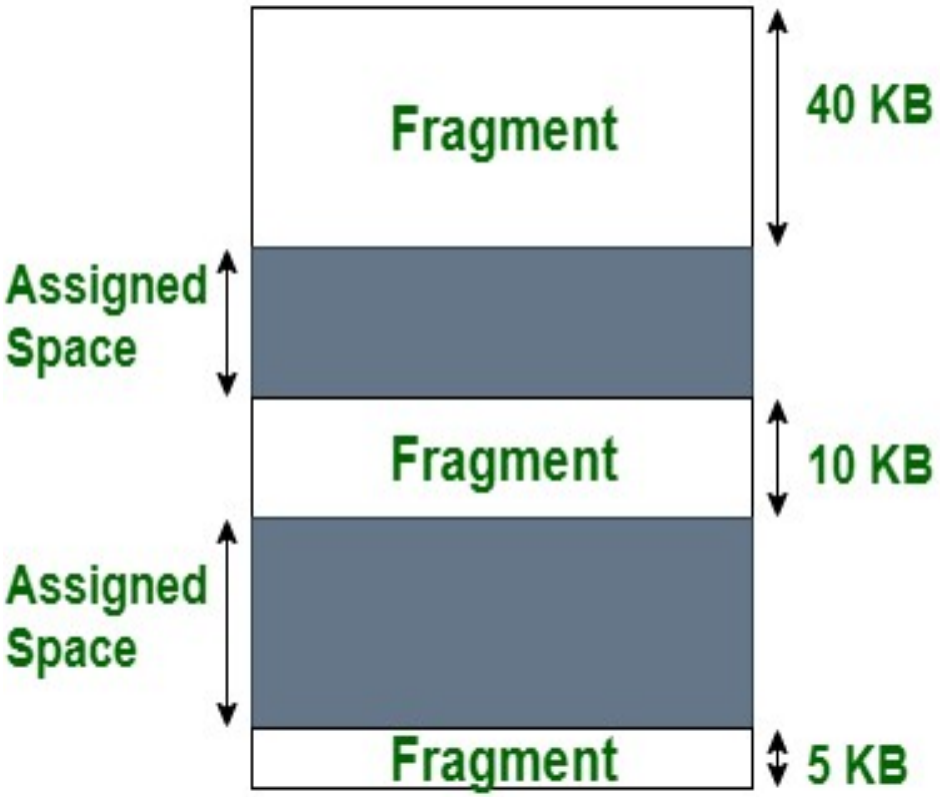
Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another process.

Example:

Imagine you have a memory space of 10 KB, but it is divided into the following non-contiguous free blocks:

Block 1: 2 KB free
Block 2: 1 KB free
Block 3: 2 KB free
Block 4: 1 KB free
Block 5: 2 KB free
Block 6: 2 KB free

Now, if a process requests 5 KB of memory, the system cannot satisfy this request, even though there are a total of 10 KB of free memory available. This is because the free memory is fragmented into smaller chunks that cannot be combined to form a single block large enough for the request.



Process 07
needs 50KB
memory space

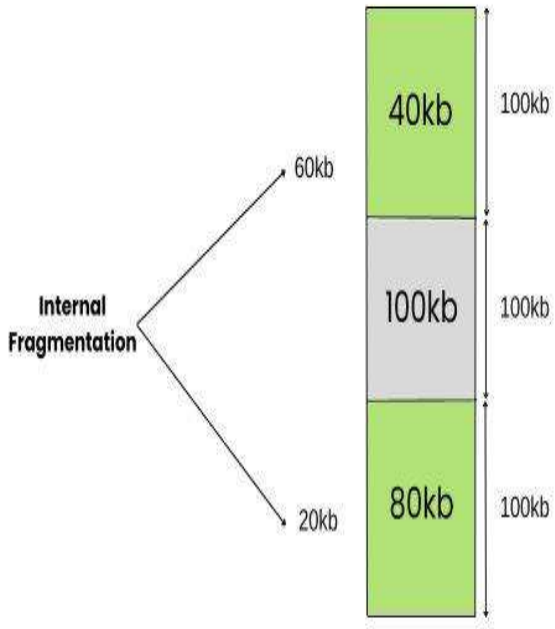
Example :
Let say a computer allocates memory in fixed-size blocks of 4 KB each. Now consider the following situation:

Process 1 requests 3 KB of memory.

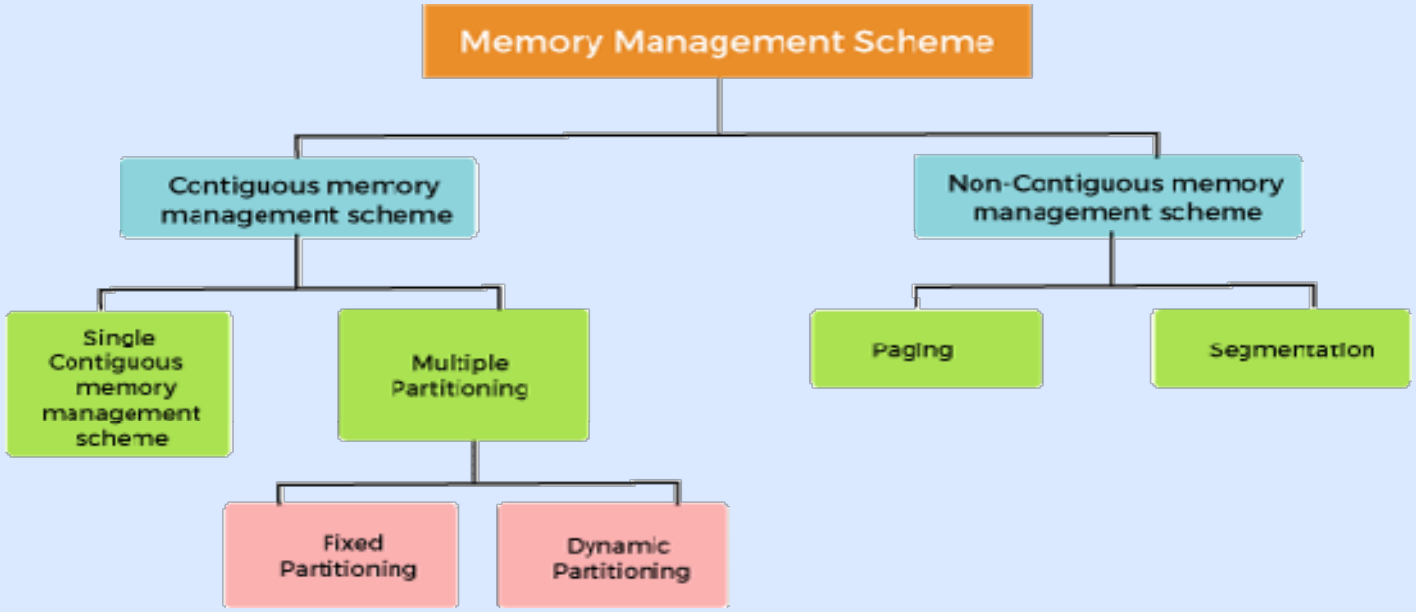
The system allocates a 4 KB block (since the minimum size is 4 KB).

As a result, 1 KB of memory within the allocated block remains unused.

This 1 KB is considered internal fragmentation because it's part of the allocated block but isn't used by the process.



Memory Management Techniques:



Classification of memory management schemes

Disadvantages of Single contiguous memory management schemes:

- Wastage of memory space due to unused memory as the process is unlikely to use all the available memory space.
- The CPU remains idle, waiting for the disk to load the binary image into the main memory.
- It can not be executed if the program is too large to fit the entire available main memory space.

It does not support multiprogramming, i.e., it cannot handle multiple programs simultaneously.

Advantages of Single contiguous memory management schemes

- Simple to implement.
- Easy to manage and design.
- In a Single contiguous memory management scheme, once a process is loaded, it is given full processor's time, and no other processor will interrupt it.

Multiple Partitioning:

The single Contiguous memory management scheme is inefficient as it limits computers to execute only one program at a time resulting in wastage in memory space and CPU time.
The problem of inefficient CPU use can be overcome using multiprogramming that allows more than one program to run concurrently.

Multiple Partitioning:

- Fixed Partitioning
- Dynamic Partitioning

Fixed Partitioning

The main memory is divided into several fixed-sized partitions in a fixed partition memory management scheme or static partitioning. These partitions can be of the same size or different sizes.
Each partition can hold a single process. The number of partitions determines the degree of multiprogramming, i.e., the maximum number of processes in memory.

Fixed Partitioning

Advantages of Fixed Partitioning memory management schemes:

Simple to implement.
Easy to manage and design.

Disadvantages of Fixed Partitioning memory management schemes:

This scheme suffers from internal fragmentation.
The number of partitions is specified at the time of system generation.

Dynamic Partitioning

The dynamic partitioning was designed to overcome the problems of a fixed partitioning scheme. In a dynamic partitioning scheme, each process occupies only as much memory as they require when loaded for processing.

Dynamic Partitioning

Advantages of Dynamic Partitioning memory management schemes:

Simple to implement.
Easy to manage and design.

Disadvantages of Dynamic Partitioning memory management schemes:

This scheme also suffers from internal fragmentation.
The number of partitions is specified at the time of system segmentation.

Non-Contiguous memory management schemes:

In a Non-Contiguous memory management scheme, the program is divided into different blocks and loaded at different portions of the memory that need not necessarily be adjacent to one another.

What is paging?

Paging is a technique that eliminates the requirements of contiguous allocation of main memory. In this, the main memory is divided into fixed-size blocks of physical memory called frames. The size of a frame should be kept the same as that of a page to maximize the main memory and avoid external fragmentation.

Advantages and Disadvantages of Paging

Here is a list of advantages and disadvantages of paging −

- Paging reduces external fragmentation, but still suffer from internal fragmentation.
- Paging is simple to implement and assumed as an efficient memory management technique.
- Due to equal size of the pages and frames, swapping becomes very easy.
- Page table requires extra memory space, so may not be good for a system having small RAM.

Segmentation

Segmentation is a memory management technique in which each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions. Each segment is a different logical address space of the program.

THANK YOU

It 412
PLATFORM TECHNOLOGIES