JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Lecture-27 By Vivek Dubey

COMPUTER SCIENCE AND ENGINEERING

CS403PC

COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CS403PC

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Scheme of Operating System

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

S. No.	Course Code	Course Title	L	T	P	Credits
3	CS403PC	Operating Systems	3	0	0	3
6	CS406PC	Operating Systems Lab	0	0	3	1.5

Lecture-27 By Vivek Dubey

UNIT-4a
Memory Management and Virtual Memory

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Logical versus Physical Address Space**
- **Swapping Memory,**
- **Contiguous Allocation**,
- Paging,
- Segmentation,
  - Segmentation with Paging,
  - Demand Paging,
  - Page Replacement,
  - Page Replacement Algorithms.

Lecture-27 By Vivek Dubey

Concept Regarding
Memory Management and Virtual Memory

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- A memory -> *just like a human brain*.
- Memory **is used to store** -> *data and information*.
- Computer Memory -> *the storage spaces in the Computer*, where data **is to be processed** and instructions **are required to process** -> *the data*.
- Just as processes **share the CPU**,
  processes **also share** -> *memory.*

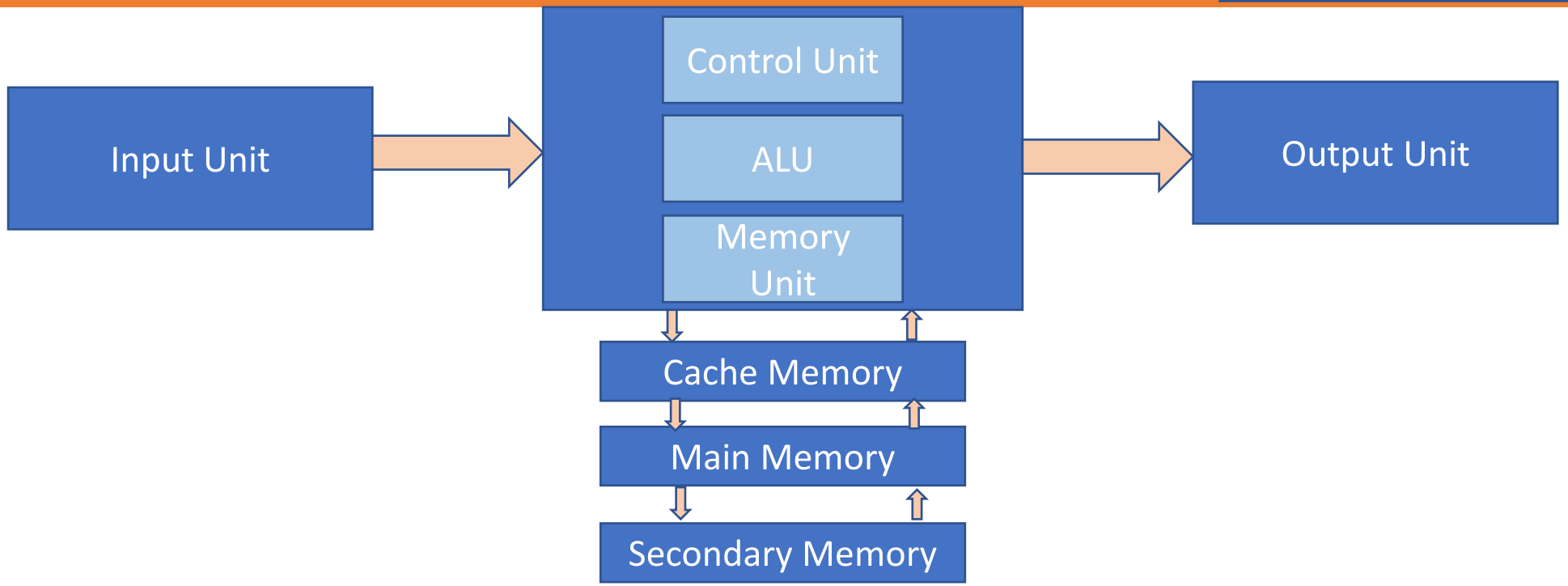
Lecture-27 By Vivek Dubey

Block Diagram of Computer System

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

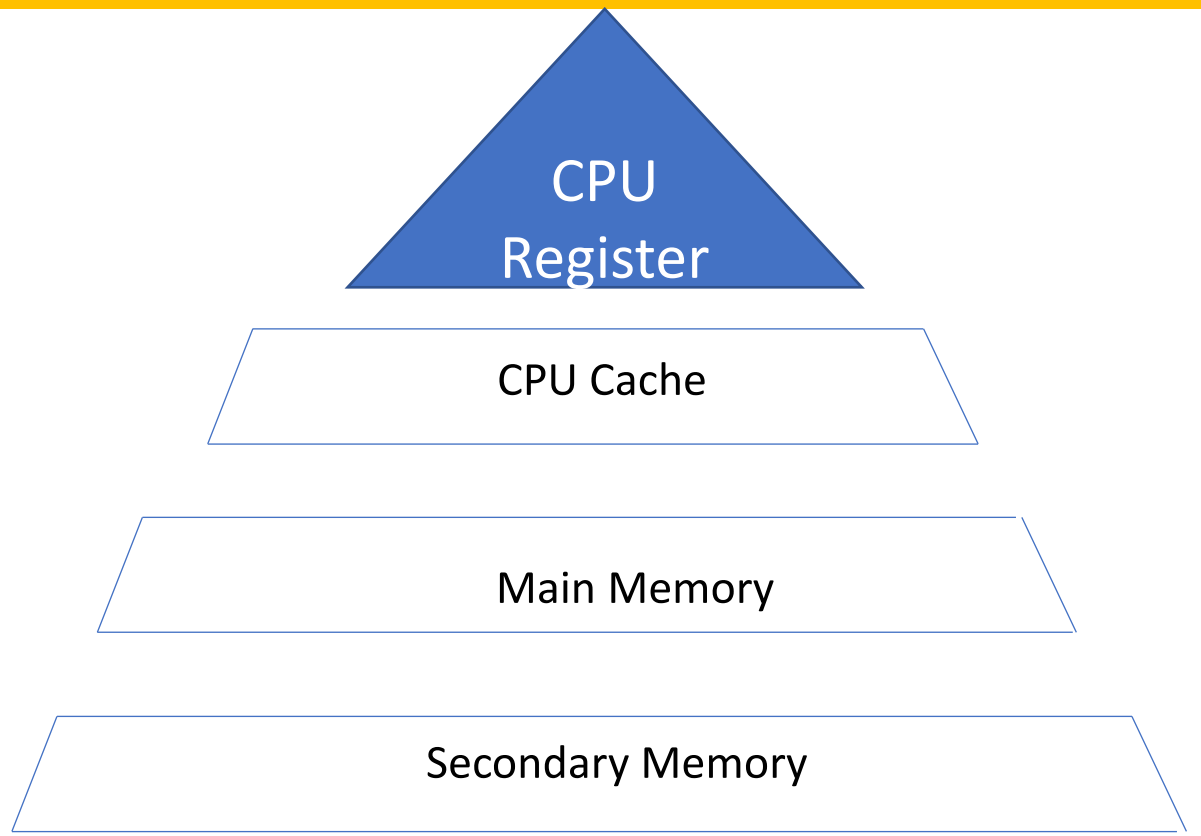
- Very first Understand Main Memory and Secondary Memory through block diagram of Computer System:

Lecture-27 By Vivek Dubey



What is Memory Hierarchy and explain High-Low Speed, Cost, Storage

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Lecture-27 By Vivek Dubey

What is Memory Management?

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Memory Management** (MM) -> process
  - **of controlling and coordinating** -> *computer memory* and
  - **optimize** *the overall performance -> of the system*.
- MM -> **most important function** -> *of an operating system* that -> *manages primary memory*.
- MM **helps** -> *processes to move back and forward* **between** *the main memory* **and** *secondary memory*.
- MM **helps** -> **OS to keep track** *of every memory location*, irrespective of whether -> **allocated** *to some process* or **remains** *free*.

Lecture-27 By Vivek Dubey

What are the Need / Importance / Functions of *Memory Management in OS*?

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Following are reasons for using memory management:

- ✓**Allocate and de-allocate** -> *memory* (before and after process execution).
- ✓**Keep track** -> *of used memory space by processes*.
- ✓**Minimize** -> *fragmentation issues.*
- ✓**Proper utilization** -> *of main memory*.
- ✓**Maintain data integrity** -> *while executing of process*.

Lecture-27 By Vivek Dubey

Memory Address

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Like** -> *we have the address -> of our house,* so that **anyone** can reach -> *out to us.*
- In the same way, **we store** -> *the data in the memory* at different locations with addresses **so that we can access** -> data *again whenever required in the future.*
- **Two types of addresses** *are used* for memory in the OS i.e.
  - the **physical address** and
  - the **logical address**

Lecture-27 By Vivek Dubey

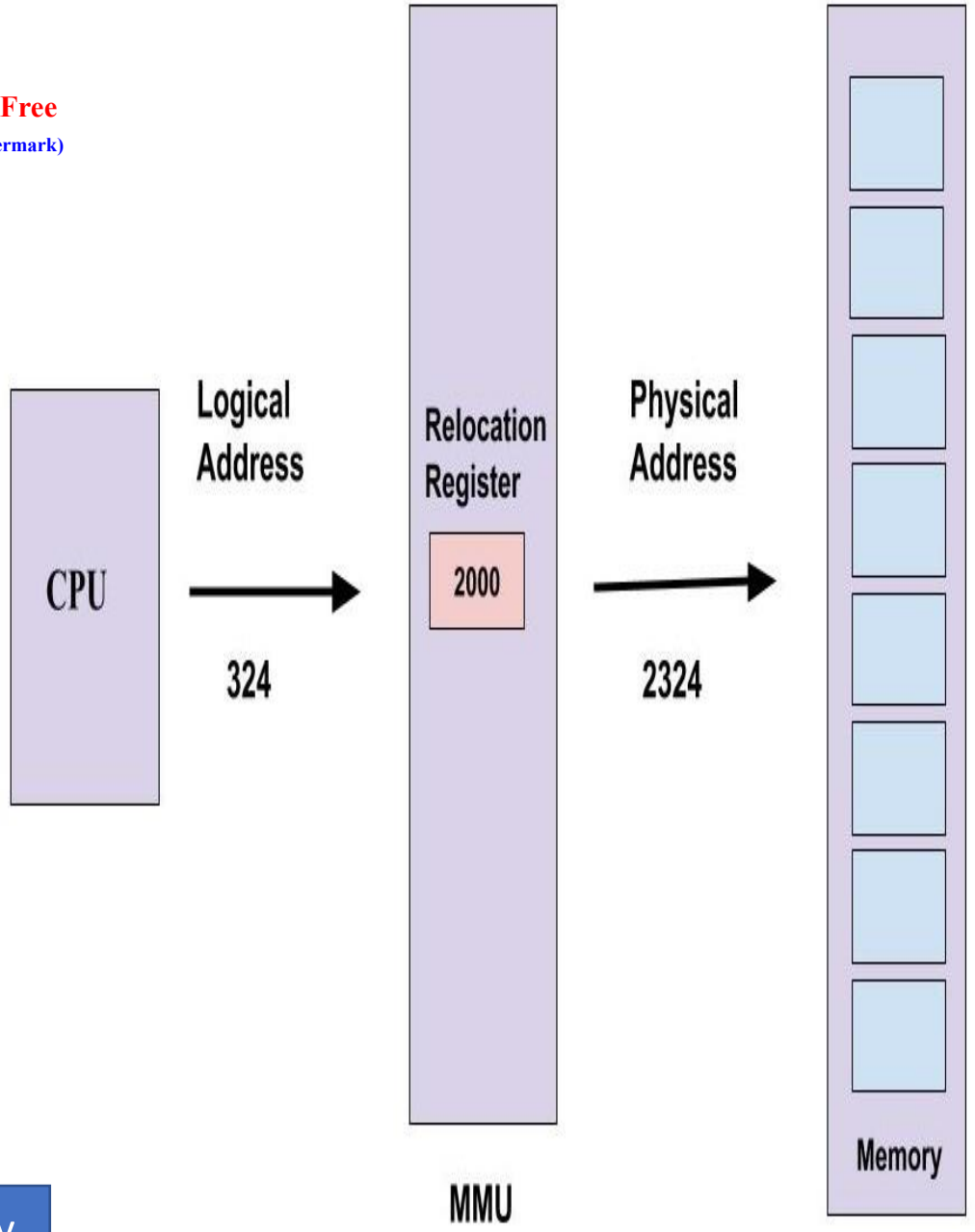
Physical Address (PA)

- Refers -> *to a location in the memory* and allows -> *access to data in the main memory*.
- PA is not -> directly accessible to the user program, **a logical address needs to be mapped** -> to it to make the address accessible.
- This mapping is done
  - -> *by the **MMU*** (Memory Management Unit) -> a hardware component -> responsible *for translating a logical address to a physical address*.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Logical Address (LA) or Virtual Address (VA)

- An address -> **generated by the CPU** *during program execution.* A logical address doesn't exist physically.
- A reference -> *to access the physical address*. A logical address usually ranges from zero to maximum (max).
- **The user program** -> *that generates the logical address* **assumes** -> that **the process runs on locations** *between 0 to the max.*
- This **logical address**(generated by CPU) combines with the **base address** generated by the MMU to form the **physical address**.



Lecture-27 By Vivek Dubey

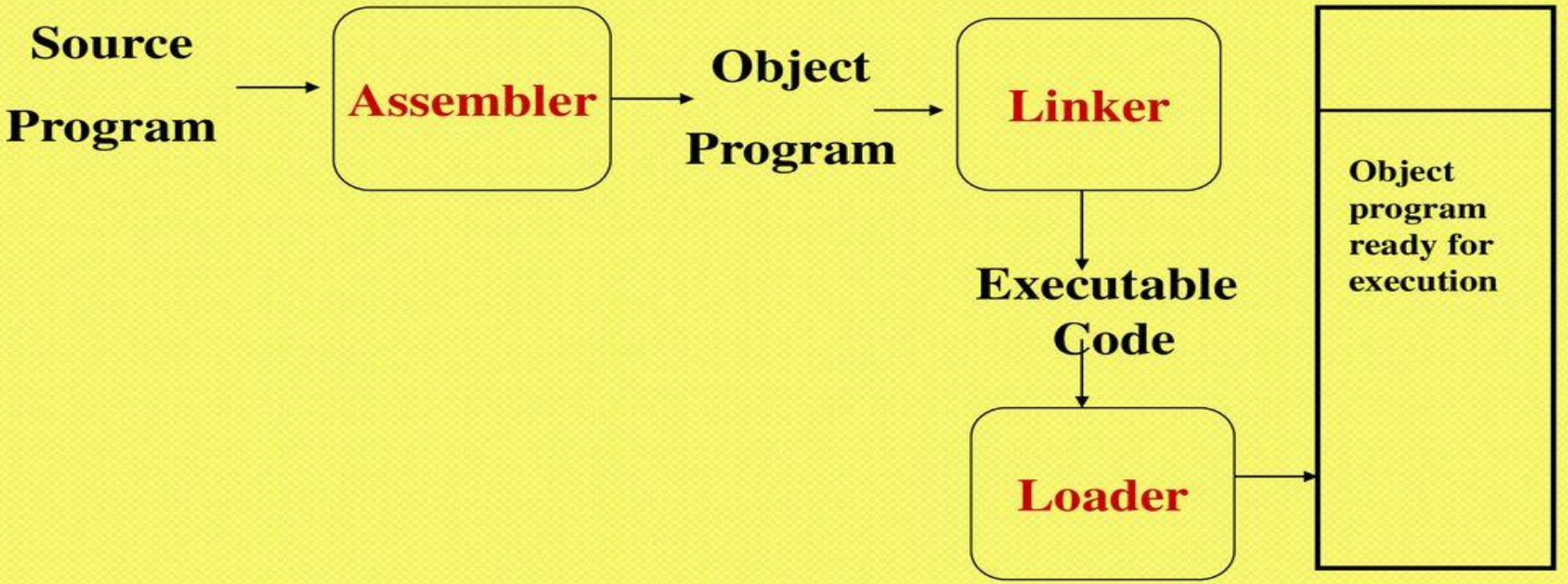
Differentiate Between Logical Address V Physical Address

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Paramenter	Logical Address	Physical Address
Basic	generated by CPU	location in a memory unit
Address Space	Logical Address Space is set of all logical addresses generated by CPU in reference to a program.	Physical Address is set of all physical addresses mapped to the corresponding logical addresses.
Visibility	User can view the logical address of a program.	User can never view physical address of program.
Generation	generated by the CPU	Computed by MMU
Access	The user can use the logical address to access the physical address.	The user can indirectly access physical address but not directly.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

ROLE OF LOADER AND LINKER



Lecture-27 By Vivek Dubey

Static and Dynamic Loading

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

To load a process into the main memory is done by a loader.

There are two different types of loading :

- **Static loading**:- *loading the entire program into a fixed address*. It requires more memory space.
- **Dynamic loading**:- The entire program and all data of a process *must be in physical memory for the process to execute.*

- ***To gain proper memory utilization**, dynamic loading is used. **In dynamic loading**, a routine is not loaded until it is called. **All routines are residing on disk** in a relocatable load format. **One of the advantages of dynamic loading is that** unused routine is never loaded. **This loading is useful when** a large amount of code is needed to handle it efficiently.*

Lecture-27 By Vivek Dubey

Static and Dynamic linking

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **To perform a linking task**, a linker is used. A linker is a program *that takes one or more object files* generated by a compiler **and combines them** *into a single executable file.*
- **Static linking:**
  - *In static linking*, **the linker combines** a*ll necessary program modules into a single executable program*. So, **there is** *no runtime dependency. (Some operating systems support only static linking, in which system language libraries are treated like any other object module.)*
- **Dynamic linking:**
  - **The basic concept of dynamic linking is** *similar to dynamic loading*. In dynamic linking, **"Stub" is included** *for each appropriate library routine reference.* **A stub is** *a small piece of code.* **When the stub is executed**, it checks whether the needed routine is *already in memory or not*. **If not available**, then the program loads *the routine into memory*.

Lecture-27 By Vivek Dubey

Fragmentation

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **As processes are loaded and removed from memory**, *the free memory space is broken into little pieces.*
- **It happens after sometimes** *that processes cannot be allocated to memory blocks* considering their small size and memory blocks remains unused.
- **This problem** is known as Fragmentation.
- **Fragmentation** is of two types:
  - **External fragmentation**
  - **Internal fragmentation**

Lecture-27 By Vivek Dubey

Types of Fragmentation and Descriptions

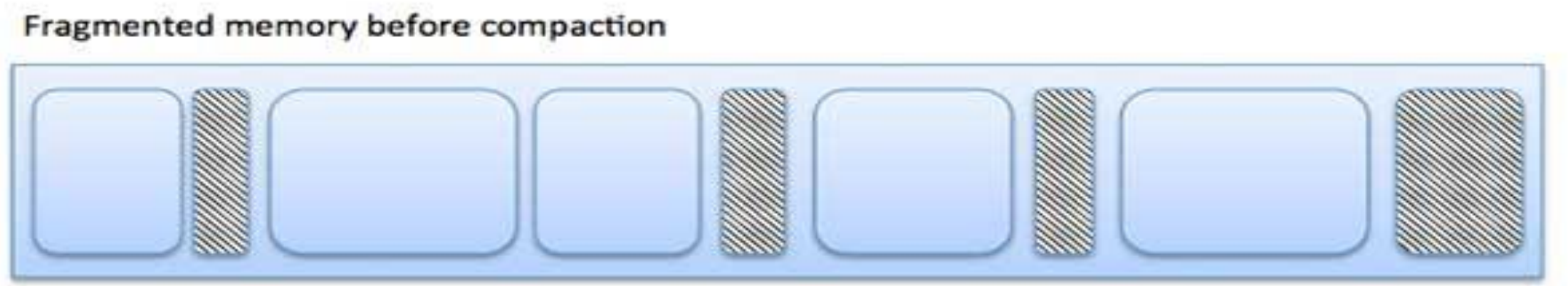
Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove this Watermark)

- **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.

Lecture-27 By Vivek Dubey

How fragmentation can cause waste of memory and a compaction technique can be used to create more free memory out of fragmented memory..

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

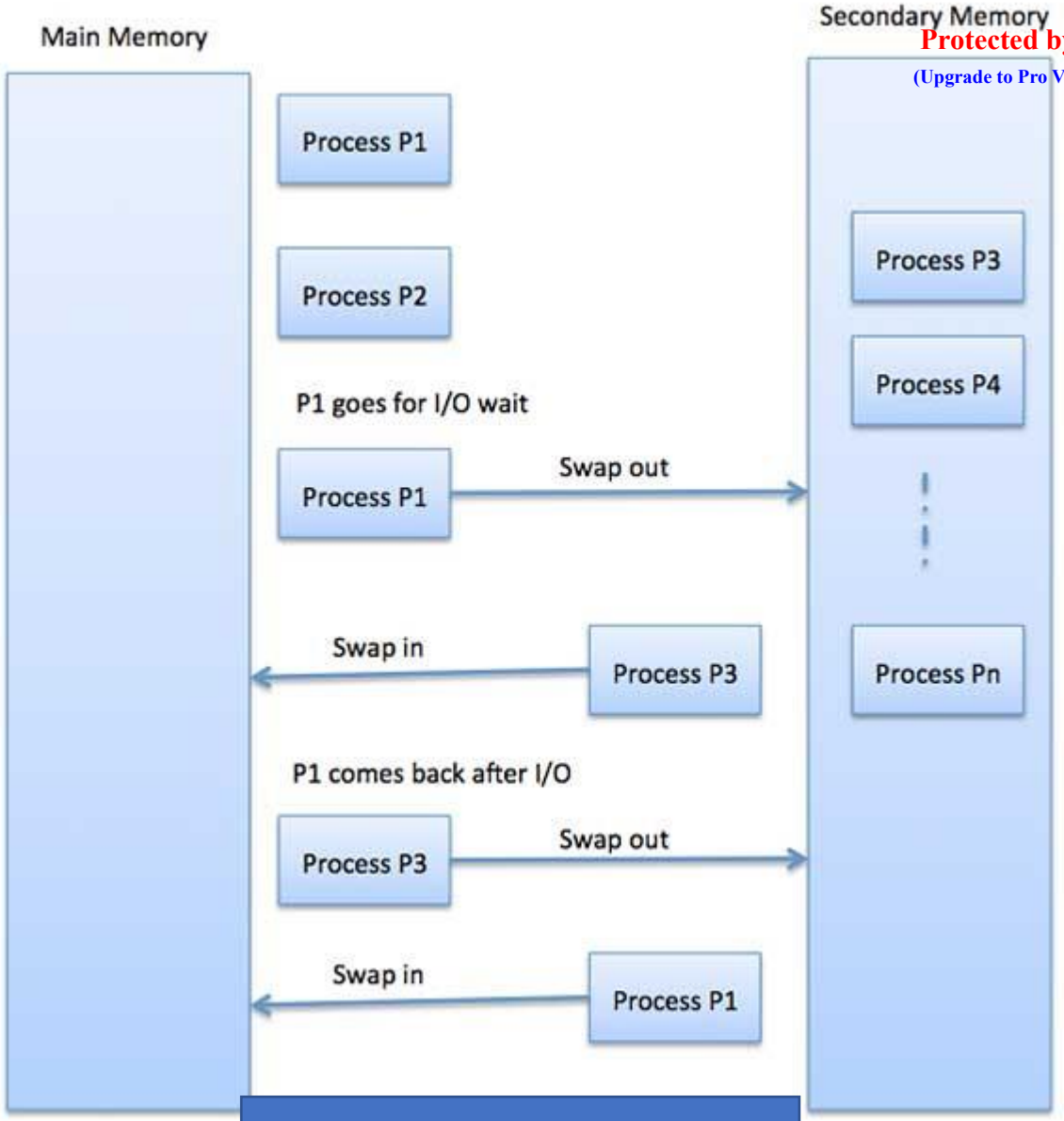


Swapping Memory in OS

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Swapping is a mechanism in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk) and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to
- Though performance is usually affected by swapping process but it helps in running multiple and big processes in parallel and that's the reason **Swapping is also known as a technique for memory compaction**.

Lecture-27 By Vivek Dubey



Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

The total time taken by swapping process includes the time it takes to move the entire process to a secondary disk and then to copy the process back to memory, as well as the time the process takes to regain main memory.

Let us assume that the user process is of size 2048KB and on a standard hard disk where swapping will take place has a data transfer rate around 1 MB per second. The actual transfer of the 1000K process to or from memory will take

2048KB / 1024KB per second
= 2 seconds
= 2000 milliseconds

Now considering in and out time, it will take complete 4000 milliseconds plus other overhead where the process competes to regain main memory.

Partition Allocation Methods in Memory Management

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

In the operating system, the following are four common memory management techniques.

- **Single contiguous allocation**
- **Partitioned allocation**
- **Paged memory management**
- **Segmented memory management**

Lecture-27 By Vivek Dubey

Partition Allocation Methods ..Contd..

Protected by PDF Anti-Copy Free
(Upgrade to Pro-Version to Remove the Watermark)

- **Single contiguous allocation:** Simplest allocation method used by MS-DOS. All memory (except some reserved for OS) is available to a process.
- **Partitioned allocation:** Memory is divided into different blocks or partitions. Each process is allocated according to the requirement.
- **Paged memory management:** Memory is divided into fixed-sized units called page frames, used in a virtual memory environment.
- **Segmented memory management:** Memory is divided into different segments (a segment is a logical grouping of the process' data or code).In this management, allocated memory doesn't have to be contiguous.

Memory Placement Algorithm in **Partition Allocation**

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

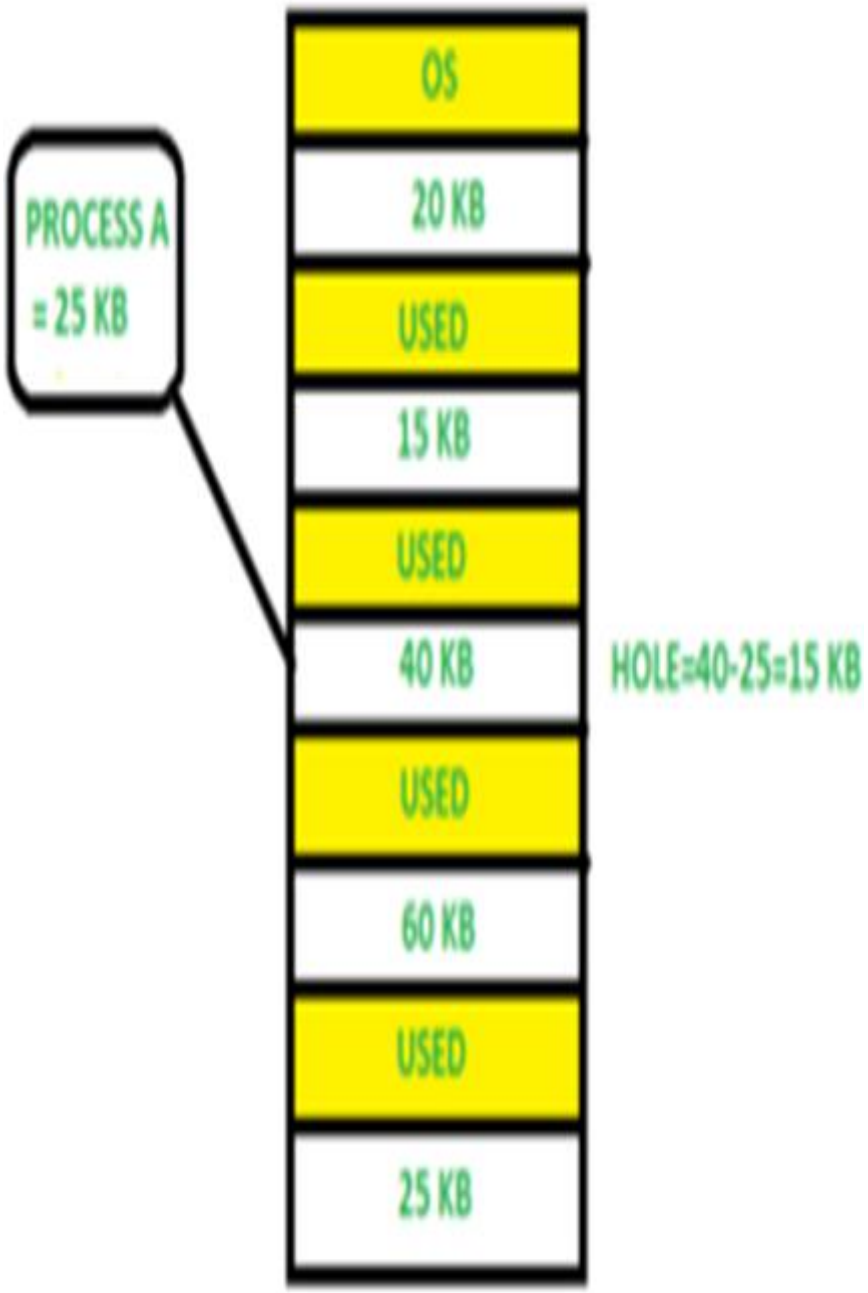
- In **Partition Allocation**, when there is more than one partition freely available to accommodate a process's request, a partition must be selected.
- To choose a particular partition, a partition allocation method is needed. A partition allocation method is considered better if it avoids internal fragmentation.
- There are different Placement Algorithm:
  - A. First Fit
  - B. Best Fit
  - C. Worst Fit
  - D. Next Fit

Lecture-27 By Vivek Dubey

First-Fit

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **In the first fit,** the partition is allocated which is *the first sufficient block from the top of Main Memory.*
- **It scans memory** from the beginning and **chooses** the first available block that is large enough.
- Thus, **it allocates** *the first hole that is large enough.*

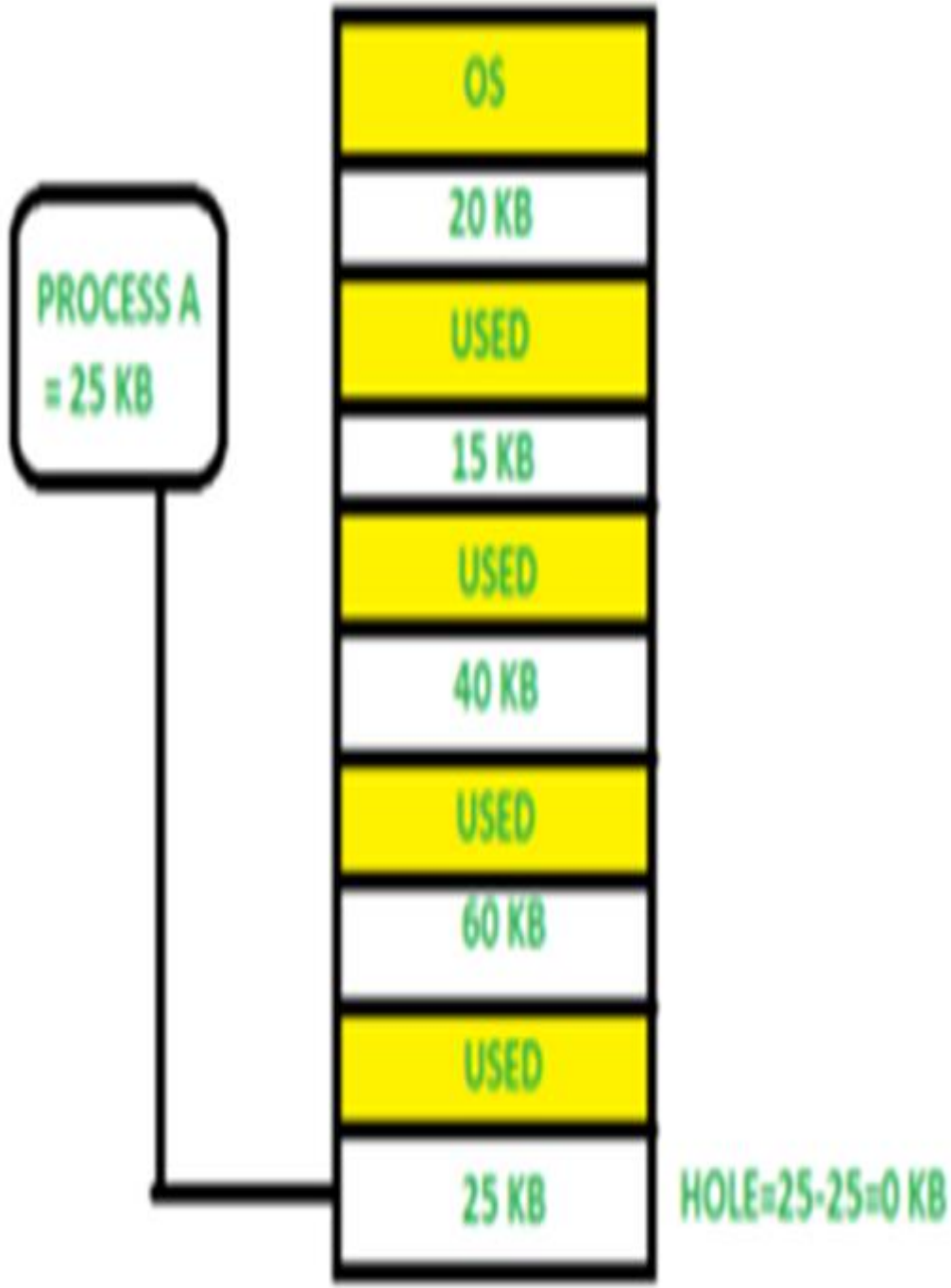


Lecture-27 By Vivek Dubey

Best-Fit

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

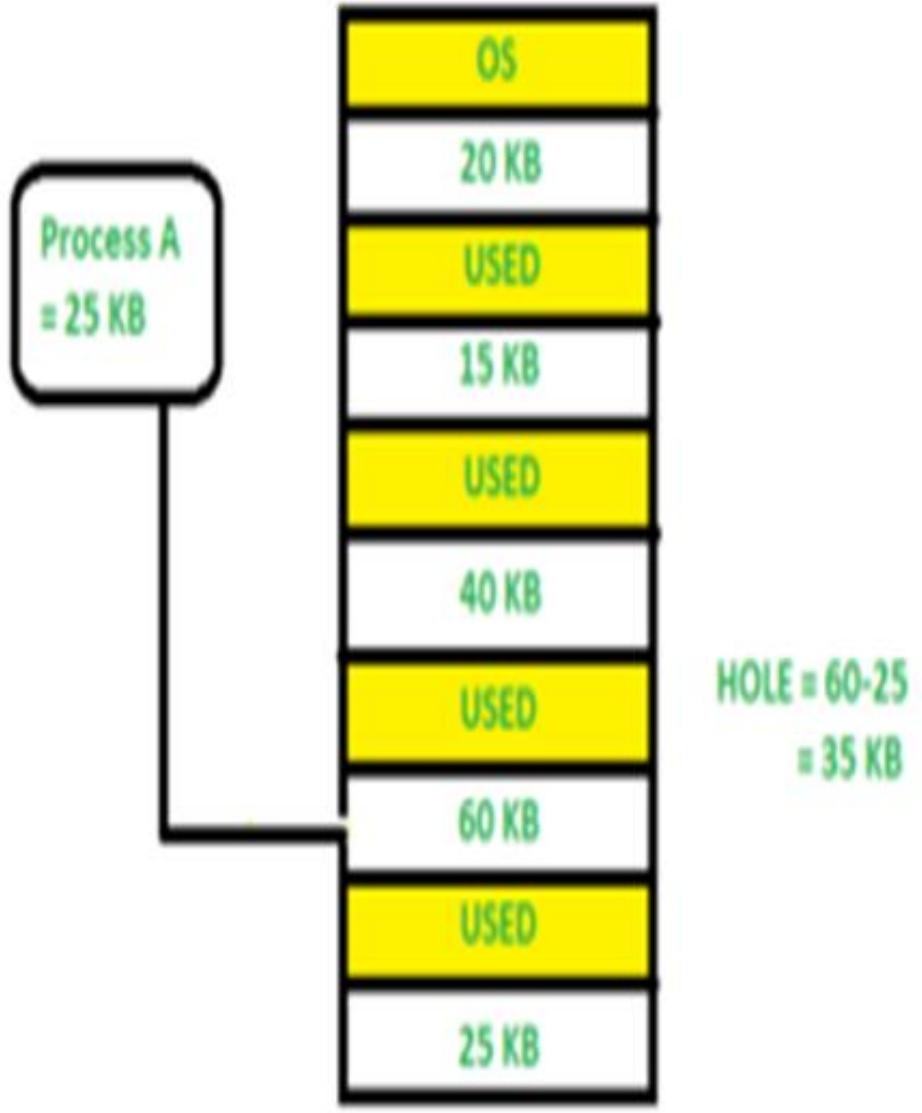
- **Allocate the process to the partition** which is *the first smallest sufficient partition among the free available partition.*
- **It searches** the entire list of holes to find the smallest hole **whose size is** greater than or equal to the size of the process.



Worst Fit

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Allocate the process to the partition which** is the largest sufficient *among the freely available partitions available in the main memory.*
- **It** is opposite *to the best-fit algorithm.*
- **It** searches *the entire list of holes* to find *the largest hole* and allocate *it to process.*



Lecture-27 By Vivek Dubey

Next Fit

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Next fit** *is similar to the first fit*
  - **but it will search**
    - *for the first sufficient partition*
      - *from the last allocation point*.

Lecture-27 By Vivek Dubey

Exercise

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Consider the requests from processes in given order 300K, 25K, 125K, and 50K. Let there be two blocks of memory available of size 150K followed by a block size 350K.
- Which of the following partition allocation schemes can satisfy the above requests?
  A) Best fit but not first fit.
  B) First fit but not best fit.
  C) Both First fit & Best fit.
  D) neither first fit nor best fit.

Lecture-27 By Vivek Dubey

Solutions

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Best Fit
  - 300K is allocated from a block of size 350K. 50 is left in the block. 25K is allocated from the remaining 50K block. 25K is left in the block. 125K is allocated from 150 K block. 25K is left in this block also. 50K can’t be allocated even if there is 25K + 25K space available.
- First it:
  - 300K request is allocated from 350K block, 50K is left out. 25K is be allocated from the 150K block, 125K is left out. Then 125K and 50K are allocated to the remaining left out partitions. So, the first fit can handle requests.
- So option B is the correct choice.

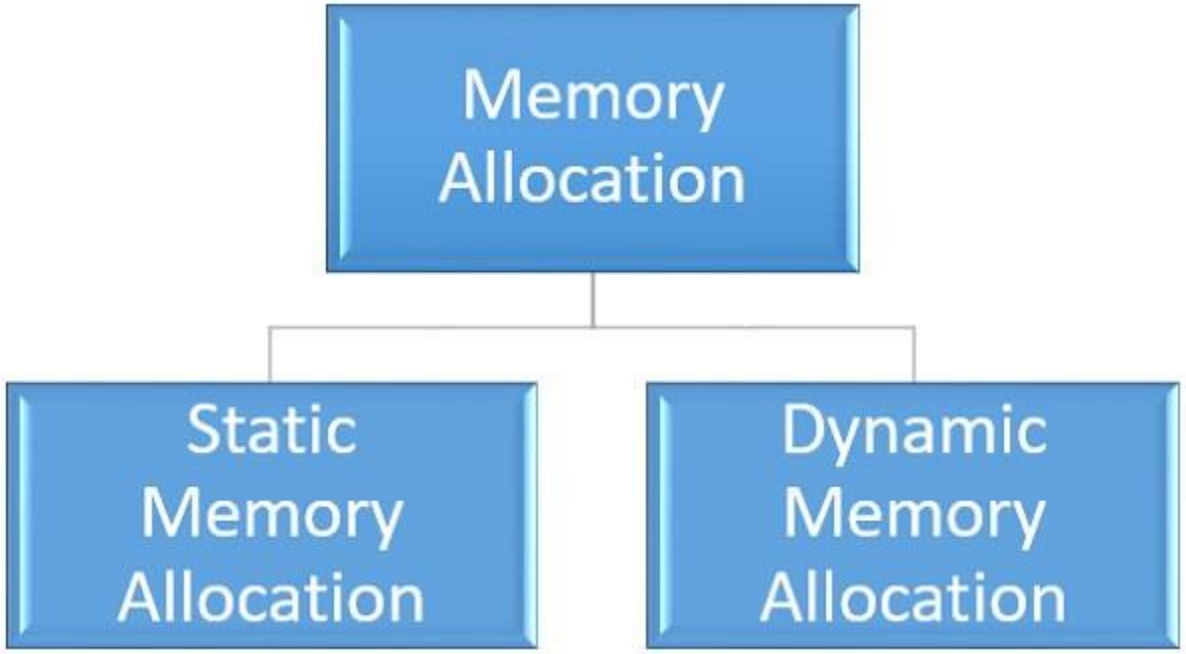
Lecture-27 By Vivek Dubey

What is Memory Allocation in OS?

Protected by PDF Anti-Copy Free
Upgrade to Pro Version to Remove the Watermark

Lecture-27 By Vivek Dubey

When a program or process *is to be executed, it needs some space* in the memory according to program or process requirements. ***This kind of method or approach*** *is called memory allocation.*



Static Memory Allocation

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- *In static memory allocation*, **the size of the data required** *by the process* must be known ***before the execution of the process initiates.***
- **If the data sizes** are not known *before the execution of the process*, ***then they have to be guessed***. If the data size guessed is larger than the required, ***then it leads to wastage of memory***.
- And, **if the guessed size** is smaller, ***then it leads to inappropriate execution of the process.***
- *All the memory allocation operation* **required for the process** is done *before the execution of the process has started*. So, it leads to ***faster*** *execution of a process*.

Lecture-27 By Vivek Dubey

Dynamic Memory Allocation

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- Dynamic memory allocation
  - provides **flexibility** to *the execution of the program*. **As it can decide** what amount of memory space will be required *by the program*. This reduces *memory wastage and improves* the performance of the system.
- Allocating memory dynamically
  - creates *an overhead over the system*. Some allocation operations are performed *repeatedly during the program execution* creating more overheads, leading in **slow** execution of the program.

Lecture-27 By Vivek Dubey

What is Contiguous Memory Allocation in OS and explain its types?

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- *In Contiguous memory allocation,* **when the process arrives** *from the ready queue to the main memory for execution,* **the contiguous memory blocks are allocated to** *the process according to its requirement.*

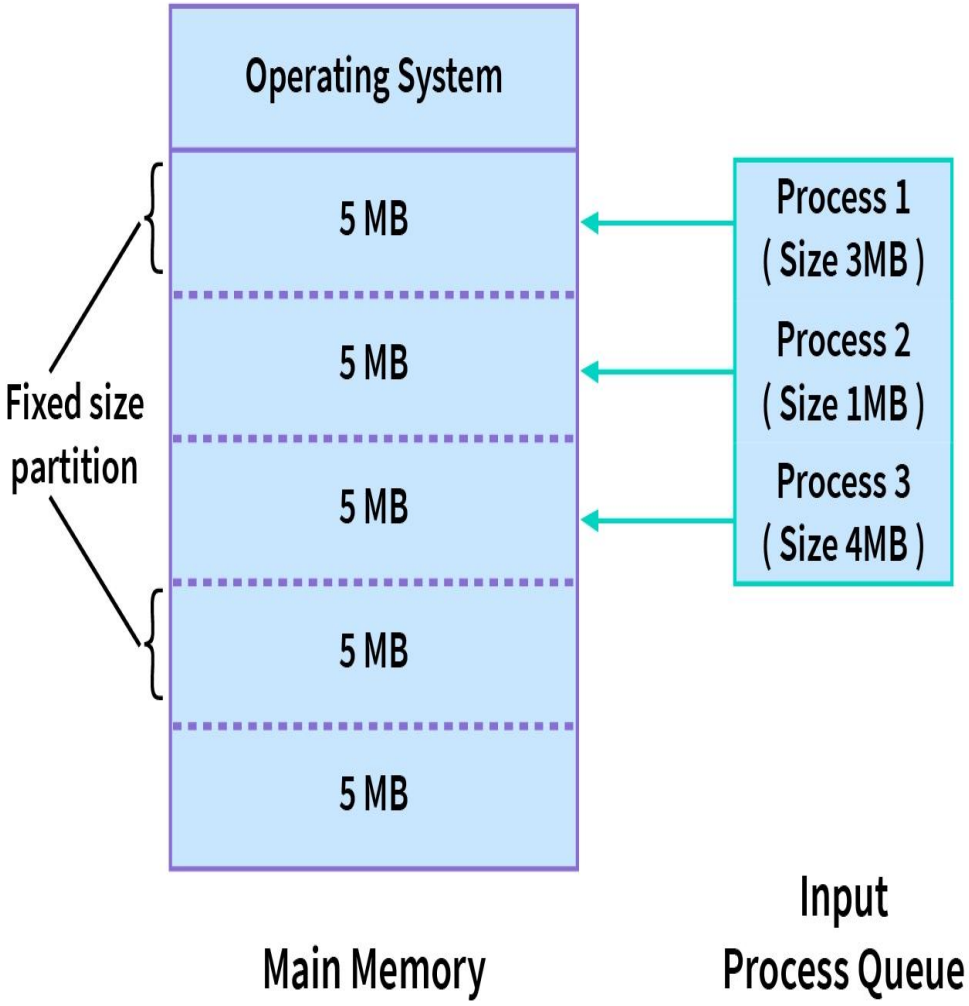
Lecture-27 By Vivek Dubey

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Fixed-size Partition Scheme

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- *In this type of contiguous memory allocation technique*, **each process is allotted a fixed size continuous block in the main memory.**
- *Because irrespective of the size of the process*, **each is allotted a block of the same size memory space.**
- **This technique is also called** *static partitioning.*



Lecture-27 By Vivek Dubey

Advantages of Fixed-size Partition

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

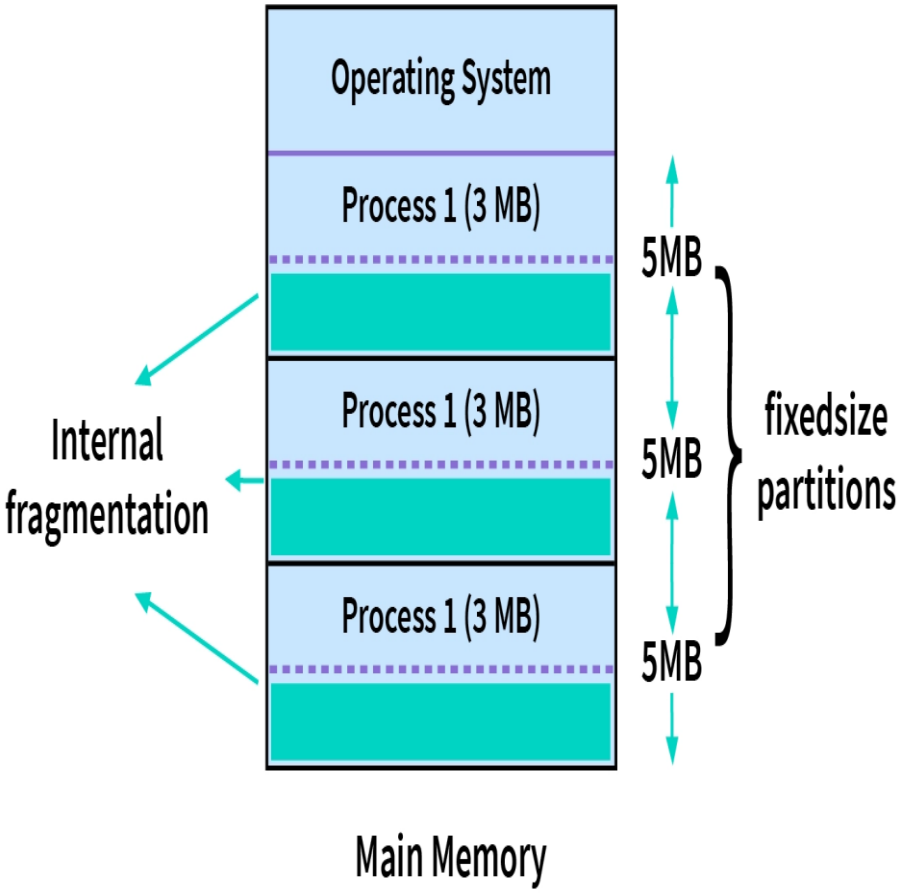
- **It is easy to keep track of** *how many blocks of memory are left*, **which in turn decides** *how many more processes can be given space in the memory.*
- **As at a time multiple processes** can be kept in the memory, **this scheme can be implemented** *in a system that needs multiprogramming.*
- **Because all of the blocks** are the same size, *this scheme is simple to implement.*

Lecture-27 By Vivek Dubey

Disadvantages of Fixed-size Partition

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **As the size of the blocks is fixed**, *we will not be able to allot space to a process* that has a greater size than the block.
- **The size of the blocks decides** *the degree of multiprogramming*, **and only that many processes** *can remain in the memory* at once as the number of blocks.
- **If the size of the block is greater** *than the size of the process*, **we have no other choice** *but to assign the process to this block*, **but this will lead** *to much empty space left behind in the block*.

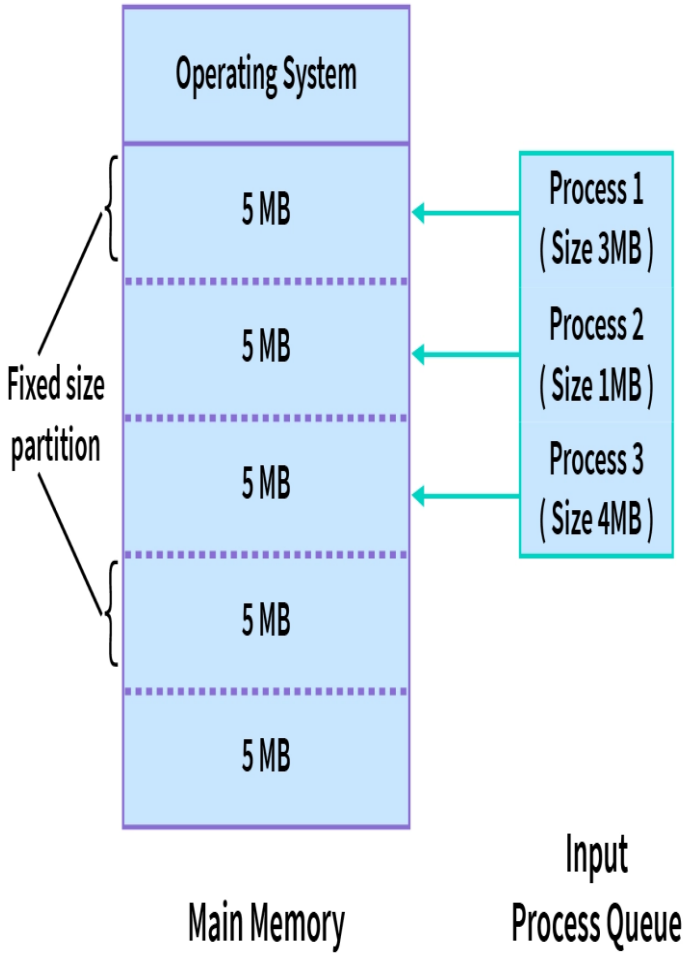


Lecture-27 By Vivek Dubey

Variable-size Partition Scheme

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **In this type of contiguous memory allocation technique**, *no fixed blocks or partitions are made in the memory*.
- **Instead, each process is allotted** *a variable-sized block depending upon its requirements.*
- That means, **whenever a new process wants** *some space in the memory*, if available, **this amount of space is allotted to it.**
- Hence, **the size of each block depends** *on the size and requirements of the process* which occupies it.



Lecture-27 By Vivek Dubey

Advantages of Variable-size Partition

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **As the processes have blocks of space allotted to them as per their requirements**, *there is no internal fragmentation*. Hence, there is no memory wastage in this scheme.
- **The number of processes that can be in the memory at once** will depend upon *how many processes are in the memory* and *how much space they occupy.* Hence, it will be different for different cases and will be dynamic.
- **As there are no blocks that are of fixed size**, *even a process of big size can be allotted space*.

Lecture-27 By Vivek Dubey

Disadvantages of Variable-size Partition

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

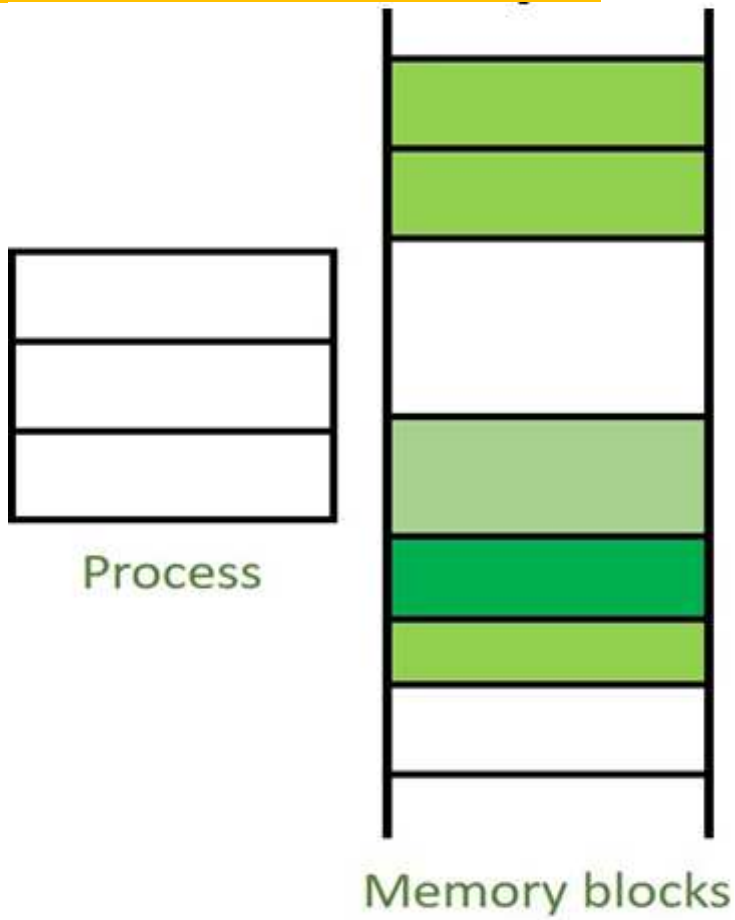
- **Because this approach is dynamic**, *a variable-size partition scheme* is difficult to implement.
- It *is difficult to keep track* of processes and the remaining space in the memory.

Lecture-27 By Vivek Dubey

Non-Contiguous Memory Allocation

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

- **Non-Contiguous memory allocation** is basically a method *on the contrary to contiguous allocation method*, allocates the memory space present in different locations *to the process as per it's requirements.*
- **As all the available memory space** is in a distributed pattern *so the freely available memory space is also scattered here and there.*
- **This technique of memory allocation** helps to reduce the wastage of memory, *which eventually gives rise to Internal and external fragmentation*.



Difference between
Contiguous and Non-contiguous Memory Allocation

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

CMA->Faster in Execution.
NCMA->Slower in Execution

CMA->It is easier for the OS to control.
NCMA-> It is difficult for the OS to control.

CMA->Wastage of memory is there.
NCMA->No memory wastage is there.

CMA->It could be visualized and implemented using Arrays.
NCMA->It could be implemented using Linked Lists.

CMA->Overhead is minimum as not much address translations are there while executing a process.
NCMA->More Overheads are there as there are more address translations.

CMA-> Contiguous memory allocation allocates consecutive blocks of memory to a file/process.
NCMA->Non-Contiguous memory allocation allocates separate blocks of memory to a file/process.

Lecture-27 By Vivek Dubey