CO - 1

Session : 6,7

COURSE NAME : SYSTEM DESIGN AND INTRODUCTION TO CLOUD

COURSE CODE : 23AD2103A

TOPICS : SEGMENTATION, INTRODUCTION TO PAGING

TRANSLATION LOOK ASIDE BUFFER, SWAPPING, DEMAND PAGING, THRASHING

SESSION DESCRIPTION

- CPU SCHEDULING
- FCFS
- SJF
- SRT
- RR
- PRIORITY

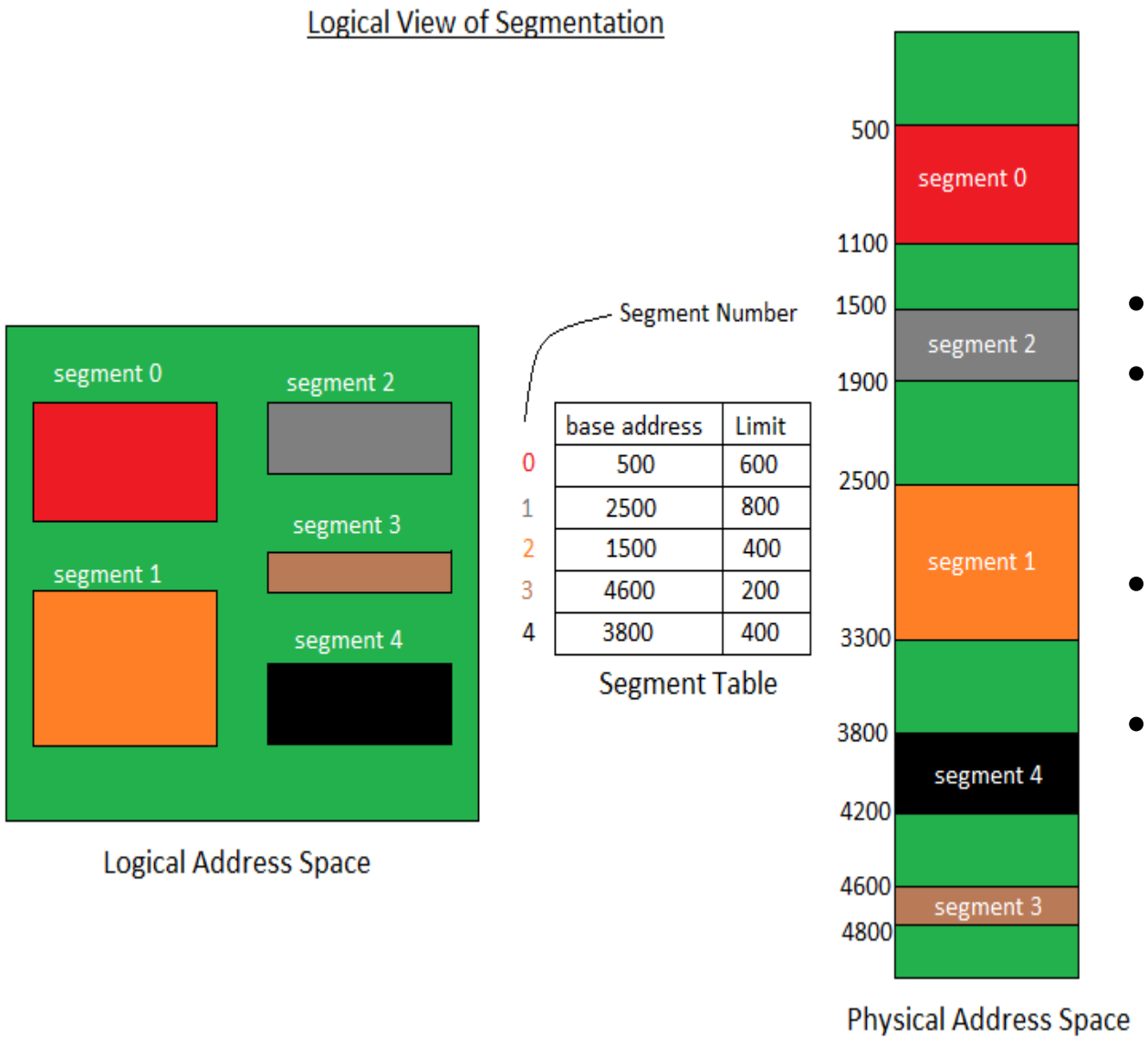
WHAT IS SEGMENTATION ?

- A Segment can be defined as a logical grouping of Instructions such as a subroutine, array or a data area.
- Every Program (job) is a collection of these segments.
- Each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions

WHAT IS SEGMENTATION ?

- Each Segment is actually a different logical address space of the Program
- Segmentation is a memory management scheme which supports the programmer's view of memory.
- Programmers never think of their programs as a linear array of words.
- Rather, they think of their programs as a collection of logically related entities such as subroutines or procedures, functions, global or local data areas, stacks etc.,

SEGMENTATION



- A process is divided into Segments.
- The chunks that a program is divided into which are not necessarily all the same sizes are called segments.
- Segmentation gives user's view of the process which paging does not give.
- Here the user's view is mapped to physical memory.

TYPES OF SEGMENTATION

- **Virtual memory segmentation**

  Each process is divided into a number of segments, not all of which are resident at any one point in time.

- **Simple segmentation**

  Each process is divided into a number of segments, all of which are loaded into memory at run time, though not necessarily contiguously.

TYPES OF SEGMENTATION

- There is no simple relationship between logical addresses and physical addresses in segmentation. A table stores the information about all such segments and is called Segment Table.
- **Segment Table** : It maps two-dimensional Logical address into one-dimensional Physical address. It's each table entry has:
- **Base Address :** It contains the starting physical address where the segments reside in memory.
- **Limit :** It specifies the length of the segment.

ADVANTAGES OF SEGMENTATION

- No Internal fragmentation.
- Segment Table consumes less space in comparison to Page table in paging.

DISADVANTAGES OF SEGMENTATION

- As processes are loaded and removed from the memory, the free memory space is broken into little pieces, causing External fragmentation.

DIFFERENCE BETWEEN PAGING AND SEGMENTATION

Paging

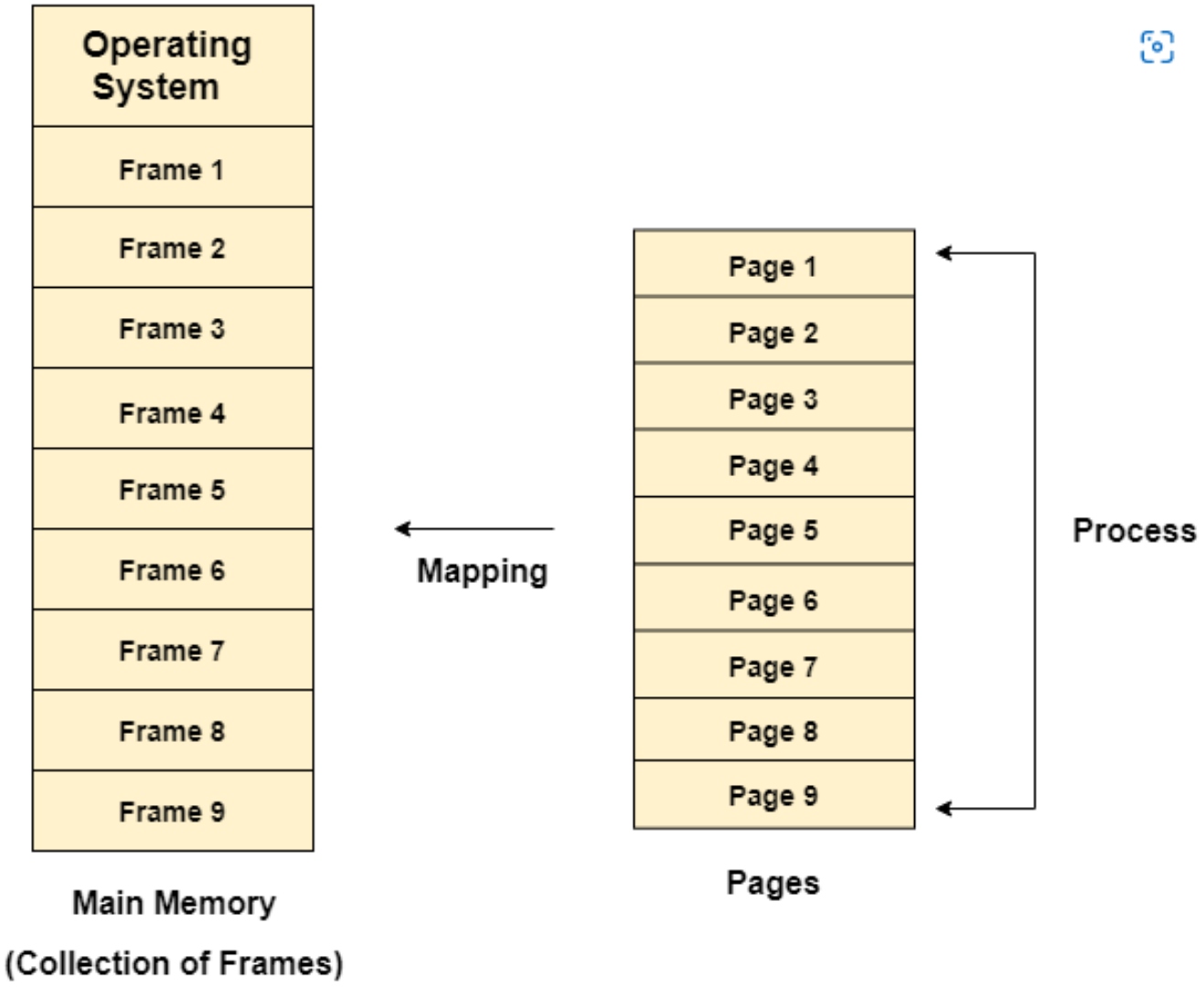
1. In Paging Scheme, the main memory is partitioned into frames(or blocks).
2. The logical address space is divided into pages by the compiler or memory management unit (MMU)
3. This Scheme suffer from internal Fragmentation or Page Breaks.
4. The OS maintains a free frame list; there is no need to search for free frame.

Segmentation

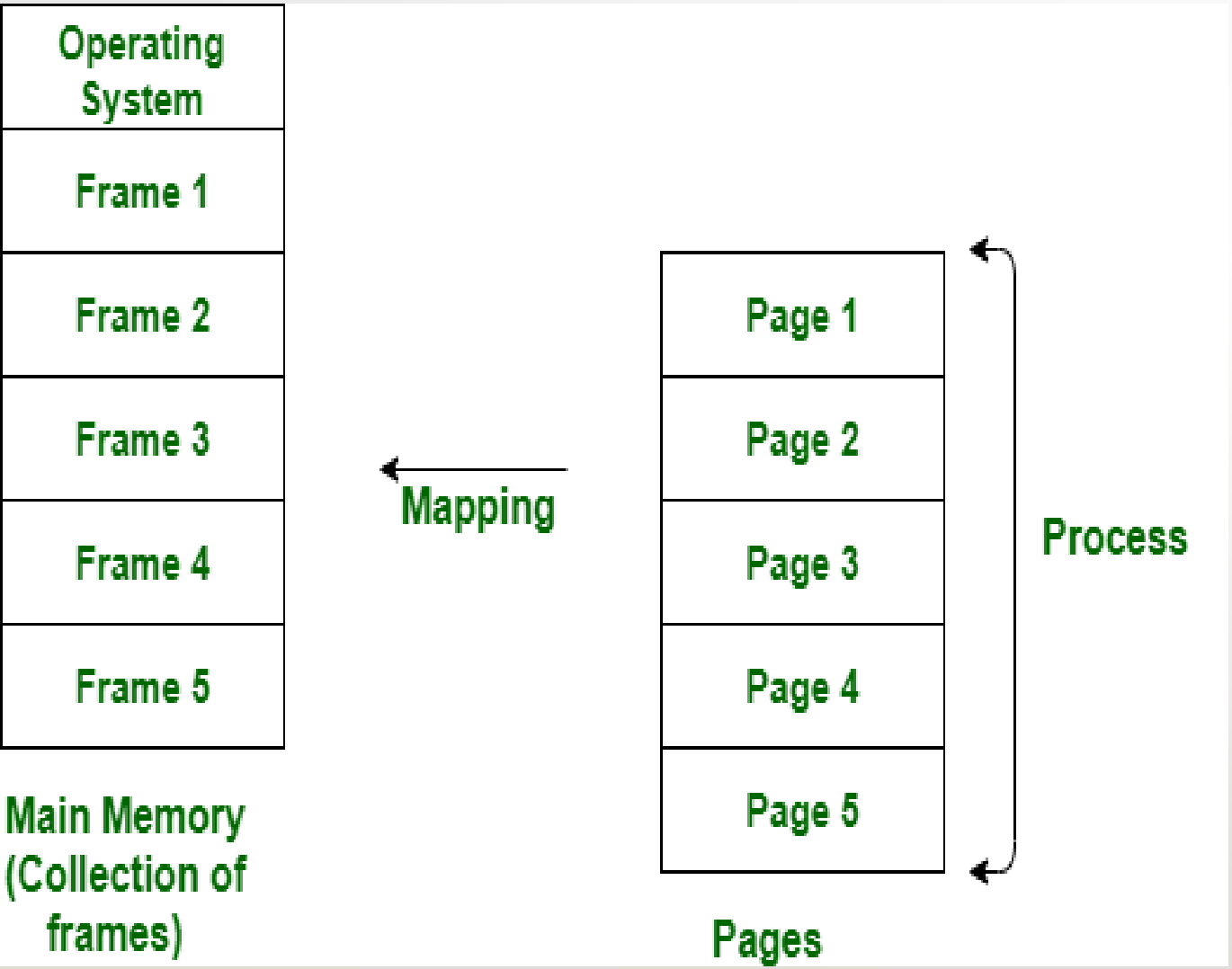
1. In Segmentation scheme, the main memory is partitioned into segments.
2. The logical address space is divided into segments as specified by the programmer.
3. This scheme suffers from external fragmentation.
4. The OS maintains the particulars of available memory.

Paging	Segmentation
5. The Operating System maintains a Page map Table for mapping between frames and pages.	5. The Operating System maintains a segment map table for mapping purpose.
6. This scheme does not support the users view of memory	6. This scheme supports user view of memory
7. Processor uses the page number and displacement to calculate absolute address (p, d)	7. Processor uses the segment number and displacement to calculate the absolute address (s, d)

PAGING



- Paging splits up address space into fixed-sized unit called a page.
- With paging, physical memory is also split into some number of pages called a page frame.
- Page table per process is needed to translate the virtual address to physical address



Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages.

The main idea behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.

ADVANTAGES & DISADVANTAGES OF PAGING

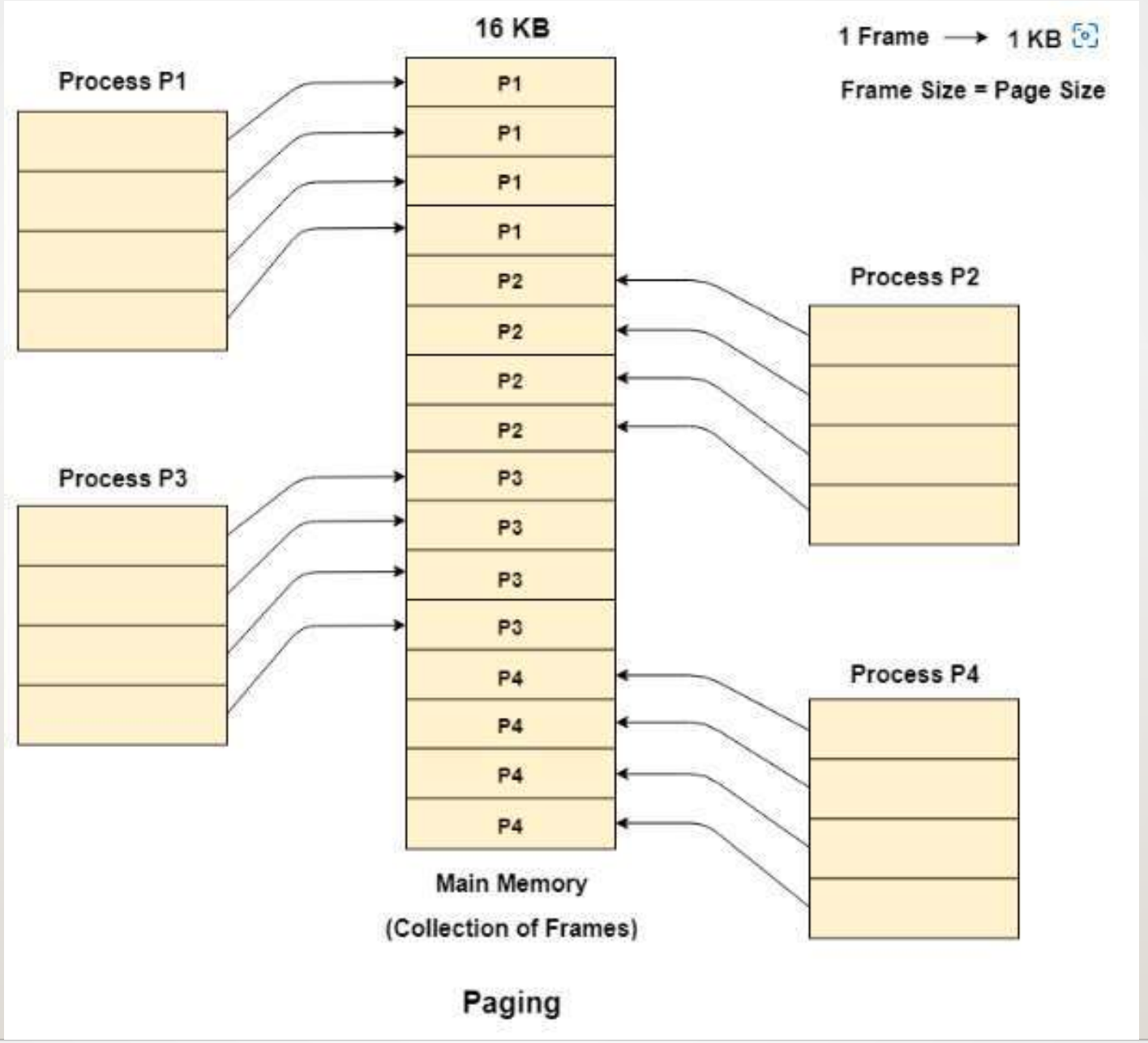
Advantages Of Paging

- Easy to use memory management algorithm
- No need for external Fragmentation
- Swapping is easy between equal-sized pages and page frames
- Allocating memory is easy and cheap

Dis-Advantages Of Paging

- May cause Internal fragmentation
- Page tables consume additional memory.
- Multi-level paging may lead to memory reference overhead.

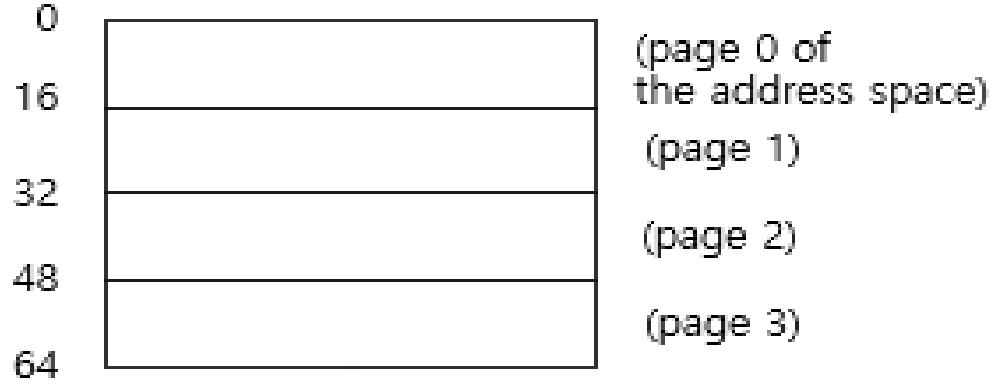
PAGING



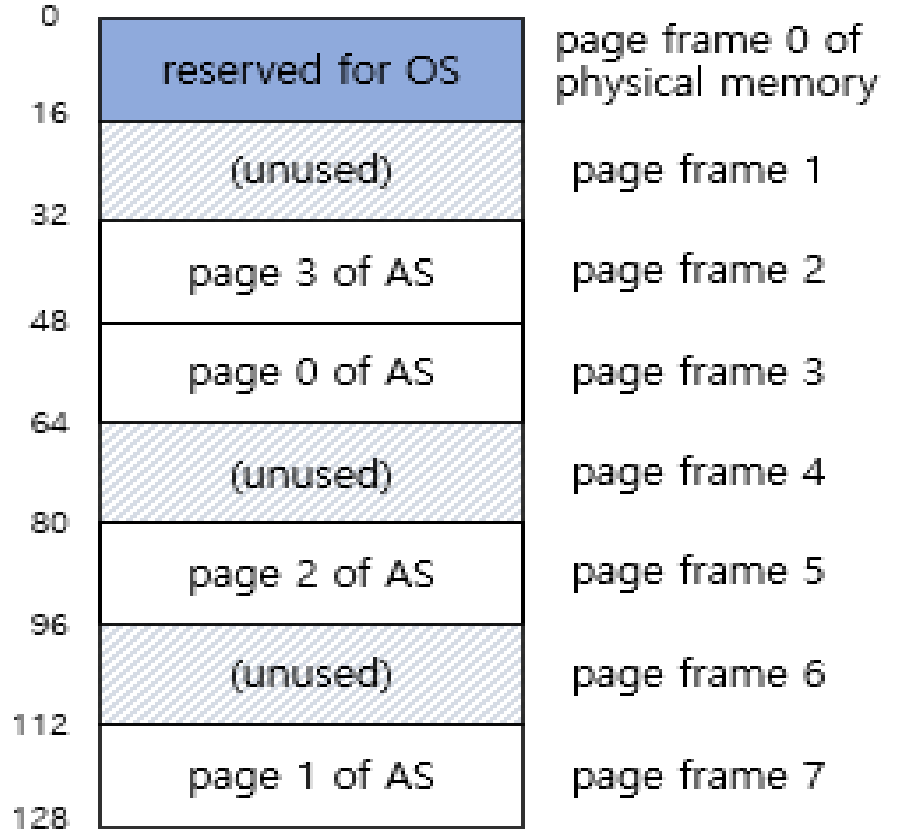
Paging

EXAMPLE :A SIMPLE PAGING

- 128-byte physical memory with 16 bytes page frames
- 64-byte address space with 16 bytes pages



A Simple 64-byte Address Space



64-Byte Address Space Placed In Physical Memory

DIFFERENCE BETWEEN PAGING AND SEGMENTATION

Paging

1. In Paging Scheme, the main memory is partitioned into frames(or blocks).
2. The logical address space is divided into pages by the compiler or memory management unit (MMU)
3. This Scheme suffer from internal Fragmentation or Page Breaks.
4. The OS maintains a free frame list; there is no need to search for free frame.

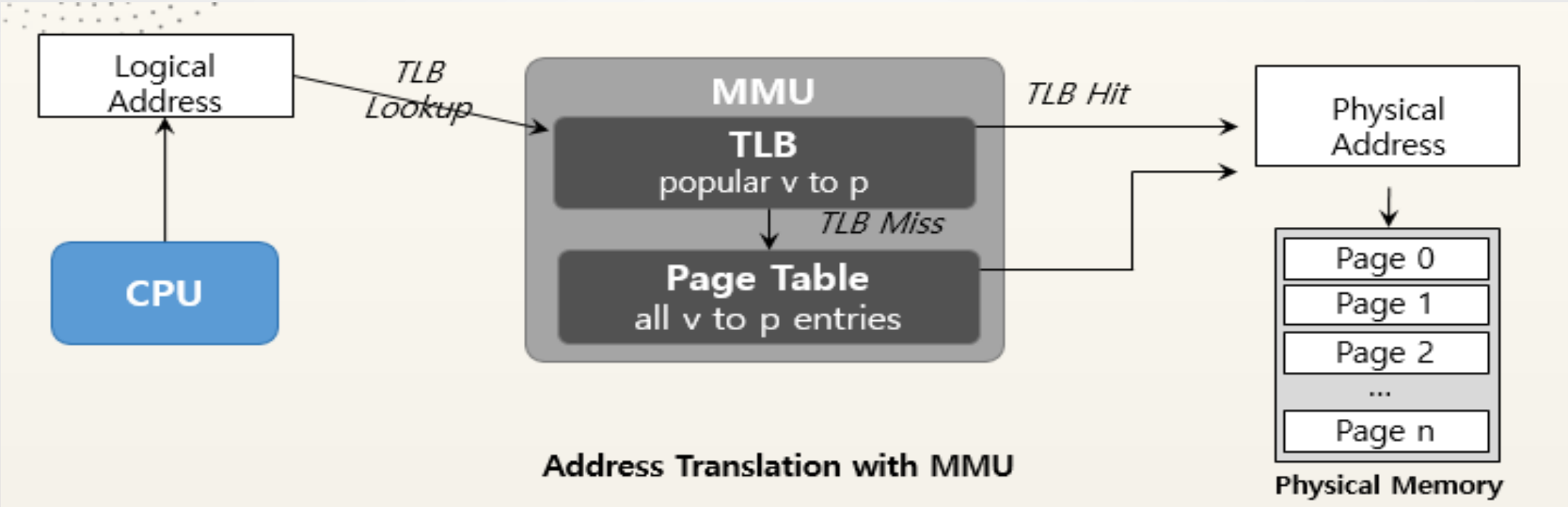
Segmentation

1. In Segmentation scheme, the main memory is partitioned into segments.
2. The logical address space is divided into segments as specified by the programmer.
3. This scheme suffers from external fragmentation.
4. The OS maintains the particulars of available memory.

Paging	Segmentation
5. The Operating System maintains a Page map Table for mapping between frames and pages.	5. The Operating System maintains a segment map table for mapping purpose.
6. This scheme does not support the users view of memory	6. This scheme supports user view of memory
7. Processor uses the page number and displacement to calculate absolute address (p, d)	7. Processor uses the segment number and displacement to calculate the absolute address (s, d)

TRANSLATION LOOK ASIDE BUFFER

- A translation look aside buffer (TLB) is **a memory cache that stores recent translations of virtual memory to physical addresses for faster retrieval**.
- When a virtual memory address is referenced by a program, the search starts in the CPU. First, instruction caches are checked.
- If the required memory is not in these very fast caches, the system has to look up the memory's physical address.
- At this point, TLB is checked for a quick reference to the location in physical memory.



Address Translation with MMU

- Part of the chip's memory-management unit(MMU).
- A hardware cache of **popular** virtual-to-physical address translation.

TRANSLATION LOOK ASIDE BUFFER

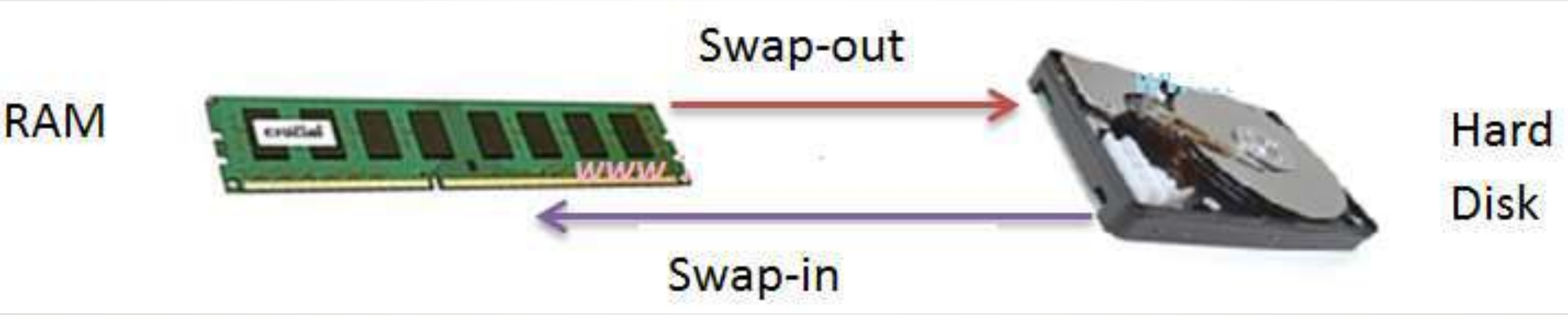
- When an address is searched in the TLB and not found, the physical memory must be searched with a memory page crawl operation.
- As virtual memory addresses are translated, values referenced are added to TLB.
- When a value can be retrieved from TLB, speed is enhanced because the memory address is stored in the TLB on processor.
- Most processors include TLBs to increase the speed of virtual memory operations through the inherent latency-reducing proximity as well as the high-running frequencies of current CPU's.

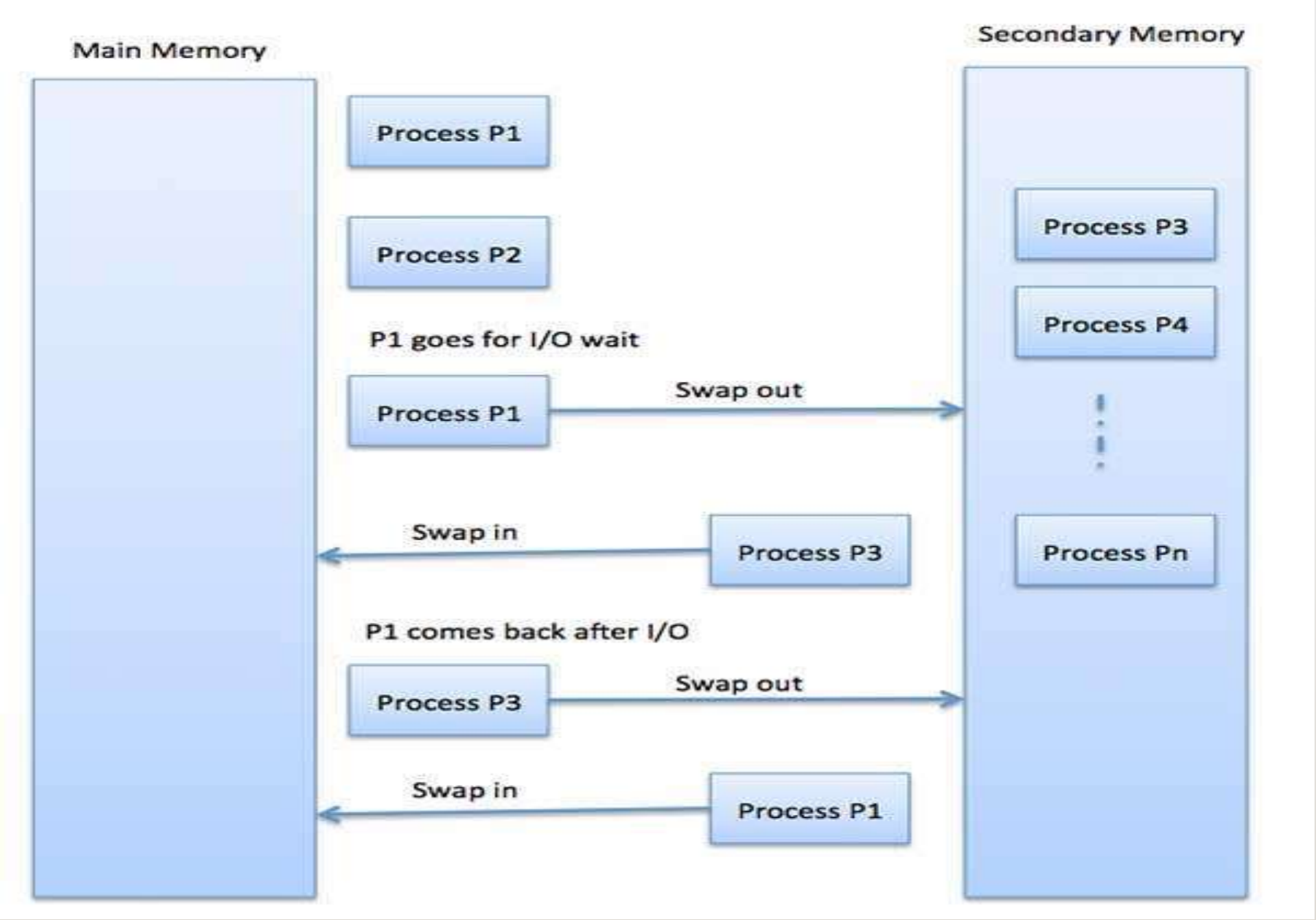
TRANSLATION LOOK ASIDE BUFFER

- TLBs also add the support required for multi-user computers to keep memory separate, by having a user and a supervisor mode as well as using permissions on read and write bits to enable sharing.
- TLBs can suffer performance issues from multitasking and code errors. This performance degradation is called a **cache thrash.**
- Cache thrash is caused by an ongoing computer activity that fails to progress due to excessive use of resources or conflicts in the caching system.

SWAPPING

- SWAPPING is the Process of temporarily removing inactive programs from the main memory of a computer system.
- 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 main memory.


RAM
Swap-out
Swap-in
Hard
Disk


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

WHAT IS DEMAND PAGING ?

- It is scheme in which a page is not loaded into the main memory from the secondary memory, until it is needed.
- So, in demand paging, pages are loaded only on demand and not in advance.
- The advantage here is that now lesser I/O is needed, less memory is needed, faster response and more users serviced now.

PURE DEMAND PAGING

- Pure Demand paging is the form of demand paging in which not even a single page is loaded into memory Initially.
- Therefore, the very first instance instruction cause a page fault in this case.
- This type of demand paging may significantly decrease the performance of a computer system by generally increasing the effective access time of memory.

THRASHING

- If the number of frames allocated to a low-priority process falls below the minimum number then we must suspend that process execution.
- We should then page out its remaining pages, freeing all its allocated frames.
- So, now swapping is required. We can find some process in a system that does not have "enough" frames.
- It is technically possible to reduce the number of allocated frames to the minimum, there is some(larger) number of pages in active use.

THRASHING

- If the Process does not have this number of frames then it will quickly page fault again and again.
- The process continues to fault, replacing pages for which It then faults and brings back in right away. Such a process spend more time in paging than executing.
- This high paging activity is called as Trashing. A process is said to be trashing if it is spending more time in paging then in execution.

Causes of Thrashing

Thrashing affects the performance of execution in the Operating system. Also, thrashing results in severe performance problems in the Operating system.

When the utilization of CPU is low, then the process scheduling mechanism tries to load many processes into the memory at the same time due to which degree of Multiprogramming can be increased. Now in this situation, there are more processes in the memory as compared to the available number of frames in the memory. Allocation of the limited amount of frames to each process.

Whenever any process with high priority arrives in the memory and if the frame is not freely available at that time then the other process that has occupied the frame is residing in the frame will move to secondary storage and after that this free frame will be allocated to higher priority process.

Effect of Thrashing

At the time, when thrashing starts then the operating system tries to apply either the **Global page replacement** Algorithm or the **Local page replacement** algorithm.

Global Page Replacement

The Global Page replacement has access to bring any page, whenever thrashing found it tries to bring more pages. Actually, due to this, no process can get enough frames and as a result, the thrashing will increase more and more. Thus the global page replacement algorithm is not suitable whenever thrashing happens.

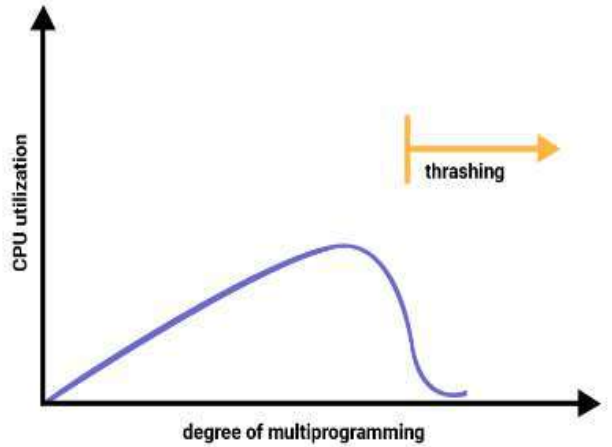


Figure: Thrashing

Local Page Replacement

Unlike the Global Page replacement, the local page replacement will select pages which only belongs to that process. Due to this, there is a chance of a reduction in the thrashing. As it is also proved that there are many disadvantages of Local Page replacement. Thus local page replacement is simply an alternative to Global Page replacement.

THANK YOU

Team – System Design & Introduction to Cloud