# UNIVERSITY INSTITUTEOF ENGINEERING

## Bachelor of Engineering (Computer Science & Engineering)

## Operating System (CST-328)

**Subject Coordinator: Er. Puneet kaur (E6913)**

Introduction to Operating System

Font size 24

DISCOVER . LEARN . EMPOWER

# Lecture 15

# Swapping

# ,

# Fragmentation

# &

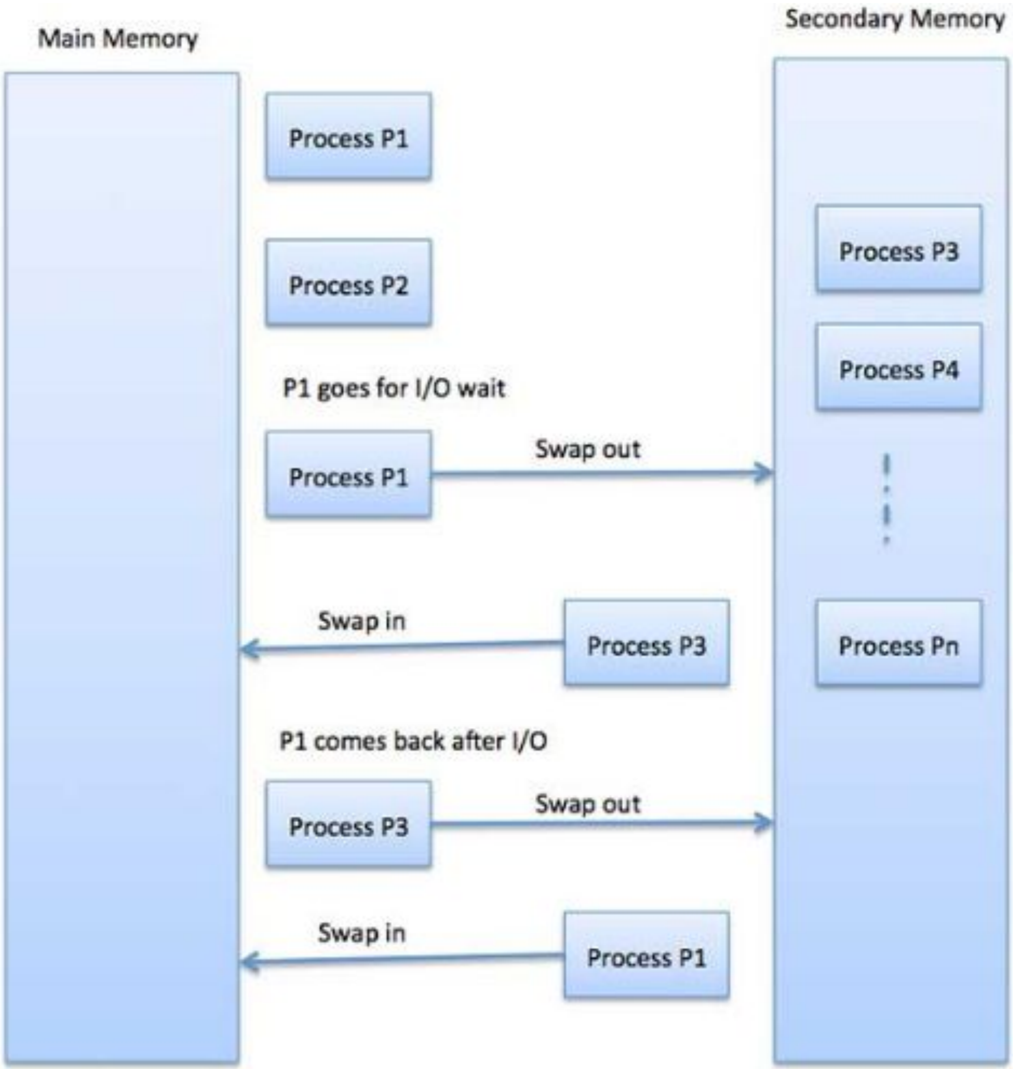
# Compaction

# Swapping

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.

A process can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution

**Swapping is also known as a technique for memory compaction**.



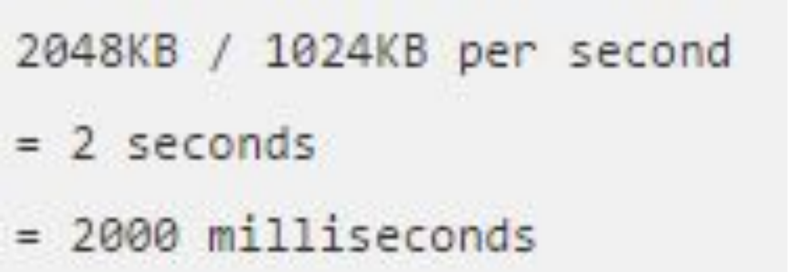
# Swapping

- **Backing store** – fast disk large enough to accommodate copies of all memory images for all users; must provide direct access to these memory images
- **Roll out, roll in** – swapping variant used for priority-based scheduling algorithms; lower-priority process is swapped out so higher-priority process can be loaded and executed
- Major part of swap time is transfer time; total transfer time is directly proportional to the amount of memory swapped
- Does the swapped out process need to swap back in to same physical addresses? Depends on address binding method

# Swapping Time

- 100MB process swapping to hard disk with transfer rate of 50MB/sec Swap out time of 2000 ms
- Plus swap in of same sized process
- Total context switch swapping component time of 4000ms (4 seconds)

- 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
```

# Swapping Advantage/Disadvantage

**Advantages of Swapping**

1. It helps the CPU to manage multiple processes within a single main memory.
2. It helps to create and use virtual memory.
3. Swapping allows the CPU to perform multiple tasks simultaneously. Therefore, processes do not have to wait very long before they are executed.
4. It improves the main memory utilization.

**Disadvantages of Swapping**

1. If the computer system loses power, the user may lose all information related to the program in case of substantial swapping activity.
2. If the swapping algorithm is not good, the composite method can increase the number of Page Fault and decrease the overall processing performance.

**Note:**

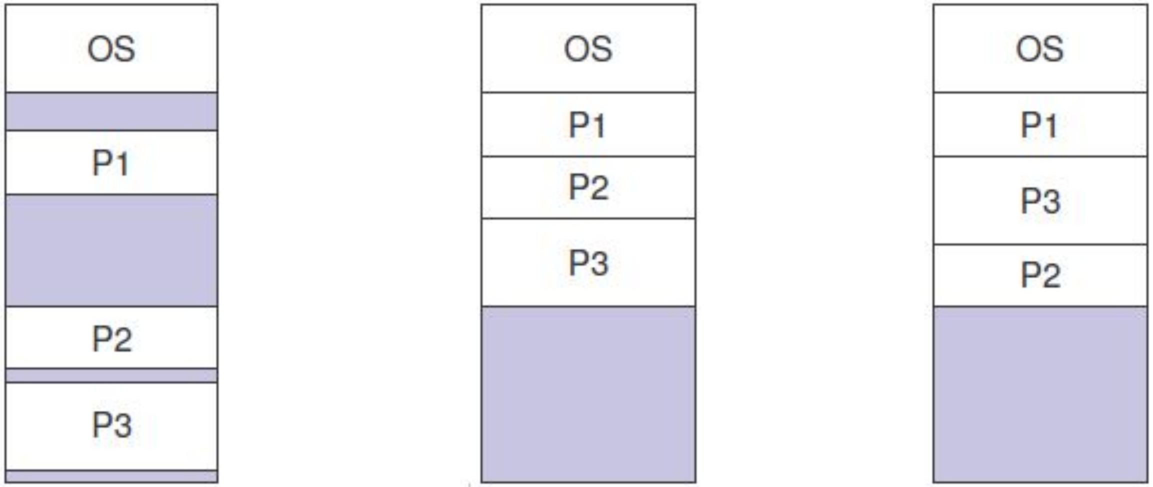
1. In a single tasking operating system, only one process occupies the user program area of memory and stays in memory until the process is complete.
2. In a multitasking operating system, a situation arises when all the active processes cannot coordinate in the main memory, then a process is swap out from the main memory so that other processes can enter it.

# Fragmentation

- **External Fragmentation** – total memory space exists to satisfy a request, but it is not contiguous
- **Internal Fragmentation** – allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used
- First fit analysis reveals that given *N* blocks allocated, 0.5 *N* blocks lost to fragmentation 1/3 may be unusable -> **50-percent rule**

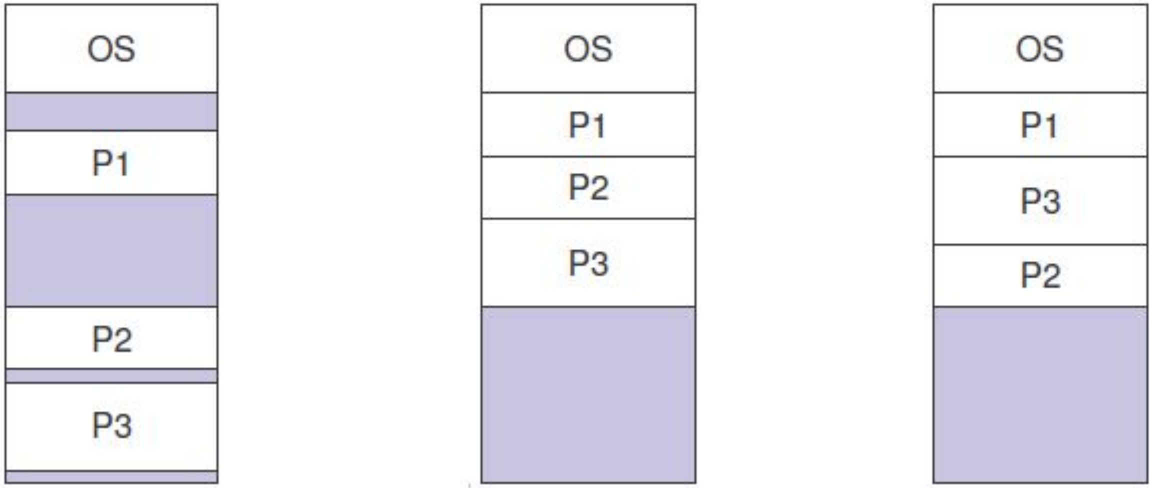
# Compaction

If a program is moved out or terminates, it creates a hole, (i.e. a contiguous unused area) in main memory. When a new process is to be moved in, it may be allocated one of the available holes. It is quite possible that main memory has too many small holes at a certain time. In such a situation none of these holes is really large enough to be allocated to a new process that may be moving in. The main memory is too fragmented. It is, therefore, essential to attempt compaction. Compaction means OS re-allocates the existing programs in contiguous regions and creates a large enough free area for allocation to a new process.



# Compaction

- Reduce external fragmentation by **compaction** Shuffle memory contents to place all free memory together in one large block
- Compaction is possible *only* if relocation is dynamic, and is done at execution time
  - I/O problem Latch job in memory while it is involved in I/O
  - Do I/O only into OS buffers



# Swapping Practice example

**Que:** Suppose the user process's size is 2048KB and is a standard hard disk where swapping has a data transfer rate of 1Mbps. Calculate how long it will take to transfer from main memory to secondary memory.

**Sol:** User process size is 2048Kb

Data transfer rate is 1Mbps = 1024 kbps

Time = process size / transfer rate

= 2048 / 1024

= 2 seconds

= 2000 milliseconds

Now taking swap-in and swap-out time, the process will take 4000 mi lliseconds.

# Conclusion

This lecture makes the student familiar with Memory allocation techniques like contiguous allocation, partitioned allocation and dynamic allocation techniques.

# Video Link

https://www.youtube.com/watch?v=SqYigYLFvcI

https://www.youtube.com/watch?v=ZN-baY3x85o

https://www.youtube.com/watch?v=buRdtPIieOM

# References

https://www.tutorialspoint.com/operating_system/os_memory_management.htm

https://www.studytonight.com/operating-system/memory-management

https://www.guru99.com/os-memory-management.html

https://www.geeksforgeeks.org/partition-allocation-methods-in-memory-management/

https://www.javatpoint.com/os-memory-management-introduction

http://www2.latech.edu/~box/os/ch08.pdf

https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/8_MainMemory.html#:~:text=8.3%20Contiguous%20Memory%20Allocation,allocated%20to%20processes%20as%20needed.

http://www.csdl.tamu.edu/~furuta/courses/99a_410/slides/chap08