

# Memory Management

---

## Introduction

So far we have seen how all the code and data associated with a program is converted to process and then allocated to the CPU. This is the **Virtualisation** piece of the OS. We have seen CPU virtualization. Now, it's time to virtualize or abstract the usage of main memory.

---

### Crux of this Section

Nowadays, main memory is used by multiple processes simultaneously. So, the question arises how we have reached this level. How did we start and progress in terms of memory usage? Moreover, since we have started accommodating a lot of processes simultaneously, how do we protect processes from one another?

---

## Flow of Lecture

This module comprises of following lectures:

1. Introduction of Memory Management
2. Memory Allocation and Main Memory

In this module, we will be answering following questions and discussing following concepts:

1. How was the memory managed in early systems and how did we progress to the present state of accommodating multiple processes to main memory.
2. Accommodation of multiple processes came up with the issue of protection of processes from one another. How was the memory allocated to solve this problem?
3. We will discuss four ways to allocate memory to processes: Fixed Partitioning, Dynamic Partitioning, Segmentation and Paging. During our discussion on memory allocation techniques, we will come across the following terms: Memory Management Unit and Translation Lookaside Buffers.
4. We will also discuss various techniques to find the most optimal free space for the process.



5. Finally, we will talk about how the secondary memory can be used to accommodate the “memory pages” and increase the degree of multiprogramming. Bringing secondary memory into the equation leads to swapping of “memory pages” between main and secondary memory and we will discuss various page replacement algorithms for the same.