Memory Management in Operating System

Last Updated : 13 Jan, 2025

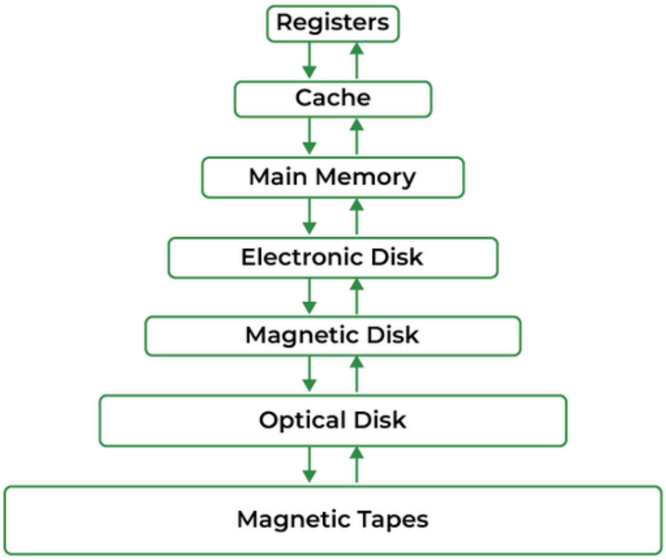
The term memory can be defined as a collection of data in a specific format. It is used to store instructions and process data. The memory comprises a large array or group of words or bytes, each with its own location. The primary purpose of a computer system is to execute programs. These programs, along with the information they access, should be in the main memory during execution. The CPU fetches instructions from memory according to the value of the program counter.

To achieve a degree of multiprogramming and proper utilization of memory, memory management is important. Many memory management methods exist, reflecting various approaches, and the effectiveness of each algorithm depends on the situation.

Before we start Memory management, let us know what is main memory is.

What is Main Memory?

The main memory is central to the operation of a Modern Computer. Main Memory is a large array of words or bytes, ranging in size from hundreds of thousands to billions. Main memory is a repository of rapidly available information shared by the CPU and I/O devices. Main memory is the place where programs and information are kept when the processor is effectively utilizing them. Main memory is associated with the processor, so moving instructions and information into and out of the processor is extremely fast. Main memory is also known as RAM (Random Access Memory). This memory is volatile. RAM loses its data



Main Memory

What is Memory Management?

Memory management mostly involves management of main memory. In a multiprogramming computer, the Operating System resides in a part of the main memory, and the rest is used by multiple processes. The task of subdividing the memory among different processes is called Memory Management. Memory management is a method in the operating system to manage operations between main memory and disk during process execution. The main aim of memory management is to achieve efficient utilization of memory.

Why Memory Management is Required?

- Allocate and de-allocate memory before and after process execution.
- To keep track of used memory space by processes.
- To minimize fragmentation issues.
- To proper utilization of main memory.
- To maintain data integrity while executing of process.

read more about - Requirements of Memory Management System

Now we are discussing the concept of Logical Address Space and Physical Address Space

Logical and Physical Address Space

address space can be defined as the size of the process. A logical address can be changed.

- **Physical Address Space:** An address seen by the memory unit (i.e. the one loaded into the memory address register of the memory) is commonly known as a “Physical Address”. A Physical address is also known as a Real address. The set of all physical addresses corresponding to these logical addresses is known as Physical address space. A physical address is computed by MMU. The run-time mapping from virtual to physical addresses is done by a hardware device Memory Management Unit(MMU). The physical address always remains constant.

Static and Dynamic Loading

Loading a process into the main memory is done by a loader. There are two different types of loading :

- **Static Loading:** Static Loading is basically 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. So, the size of a process is limited to the size of physical memory. 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 the unused routine is never loaded. This loading is useful when a large amount of code is needed to handle it efficiently.

Static and Dynamic Linking

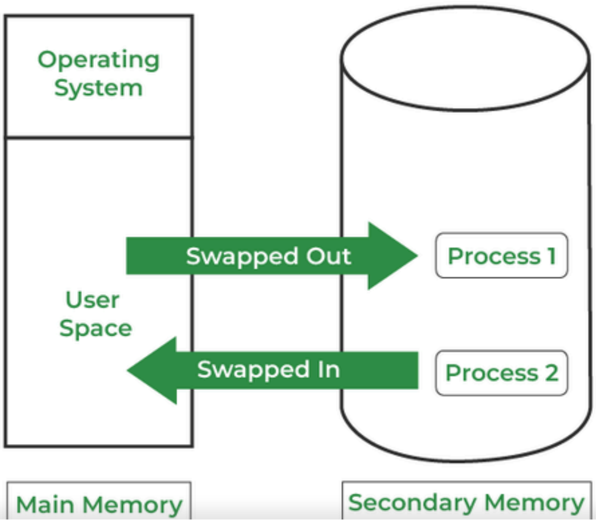
To perform a linking task a linker is used. A linker is a program that takes one or more object files generated by a compiler and combines them into a single executable file.

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.

Swapping

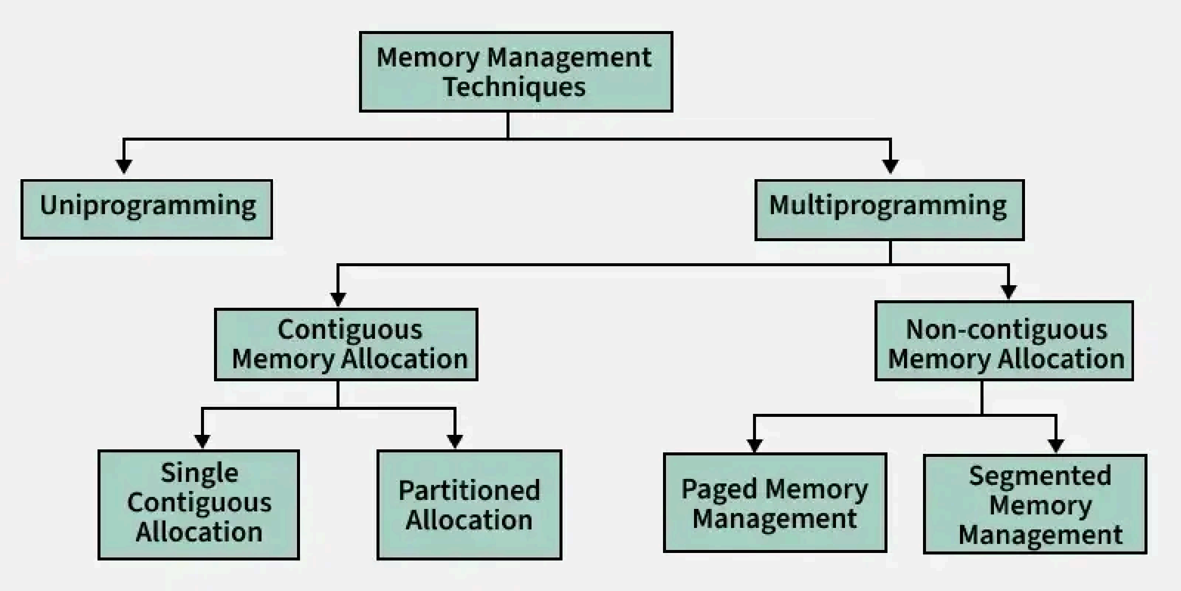
When a process is executed it must have resided in memory. Swapping is a process of swapping a process temporarily into a secondary memory from the main memory, which is fast compared to secondary memory. A swapping allows more processes to be run and can be fit into memory at one time. The main part of swapping is transferred time and the total time is directly proportional to the amount of memory swapped. Swapping is also known as roll-out, or roll because if a higher priority process arrives and wants service, the memory manager can swap out the lower priority process and then load and execute the higher priority process. After finishing higher priority work, the lower priority process swapped back in memory and continued to the execution process.



Memory Management Techniques

Memory management techniques are methods used by an operating system to efficiently allocate, utilize, and manage memory resources for processes. These techniques ensure smooth execution of programs and optimal use of system memory

Different Memory Management techniques are:


Memory Management Techniques
Uniprogramming
Multiprogramming
Contiguous Memory Allocation
Non-contiguous Memory Allocation
Single Contiguous Allocation
Partitioned Allocation
Paged Memory Management
Segmented Memory Management

Memory Management with Monoprogramming (Without Swapping)

This is the simplest memory management approach the memory is divided into two sections:

- One part of the operating system
- The second part of the user program

Fence Register	
operating system	user program

- In this approach, the operating system keeps track of the first and

- Interrupt vectors are often loaded in low memory therefore, it makes sense to load the operating system in low memory
- Sharing of data and code does not make much sense in a single process environment
- The Operating system can be protected from user programs with the help of a fence register.

Multiprogramming with Fixed Partitions (Without Swapping)

- A memory partition scheme with a fixed number of partitions was introduced to support multiprogramming. this scheme is based on contiguous allocation
- Each partition is a block of contiguous memory
- Memory is partitioned into a fixed number of partitions.
- Each partition is of fixed size

Example: As shown in fig. memory is partitioned into 5 regions the region is reserved for updating the system the remaining four partitions are for the user program.

Fixed Size Partitioning

Operating System
p1
p2
p3
p4

Partition Table

Once partitions are defined operating system keeps track of the status

Sample Partition Table

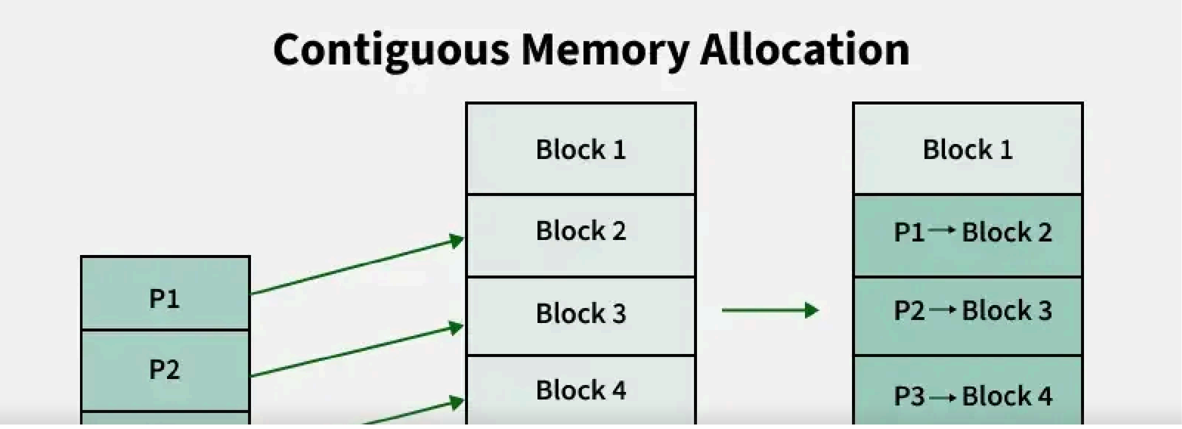
Starting Address of Partition	Size of Partition	Status
0k	200k	allocated
200k	100k	free
300k	150k	free
450k	250k	allocated

Logical vs Physical Address

An address generated by the CPU is commonly referred to as a logical address. the address seen by the memory unit is known as the physical address. The logical address can be mapped to a physical address by hardware with the help of a base register this is known as dynamic relocation of memory references.

Contiguous Memory Allocation

Contiguous memory allocation is a memory management method where each process is given a single, continuous block of memory. This means all the data for a process is stored in adjacent memory locations.



read more about - Contiguous Memory Management Techniques

Partition Allocation Methods

To gain proper memory utilization, memory allocation must be allocated efficient manner. One of the simplest methods for allocating memory is to divide memory into several fixed-sized partitions and each partition contains exactly one process. Thus, the degree of multiprogramming is obtained by the number of partitions.

- **Fixed partition allocation:** Memory is divided into fixed-sized partitions during system initialization. Each partition can hold only one process.

- **Dynamic Partition Allocation:** In this allocation strategy, Memory is divided into variable-sized partitions based on the size of the processes.

When it is time to load a process into the main memory and if there is more than one free block of memory of sufficient size then the OS decides which free block to allocate.

There are different Placement Algorithm:

1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit

- read more about - Fixed (or static) Partitioning in Operating System

- read more about - Variable (or Dynamic) Partitioning in Operating System

- read more about - Partition Allocation Methods in Memory Management

Non-contiguous memory allocation is a memory management method where a process is divided into smaller parts, and these parts are stored in different, non-adjacent memory locations. This means the entire process does not need to be stored in one continuous block of memory.

Techniques of Non-Contiguous Memory Allocation are:

- Paging
- Segmentation

read more about - Non-Contiguous Memory Allocation

Fragmentation

Fragmentation is defined as when the process is loaded and removed after execution from memory, it creates a small free hole. These holes can not be assigned to new processes because holes are not combined or do not fulfill the memory requirement of the process. In the operating systems two types of fragmentation are:

- **Internal fragmentation:** Internal fragmentation occurs when memory blocks are allocated to the process more than their requested size. Due to this some unused space is left over and creating an internal fragmentation problem. **Example:** Suppose there is a fixed partitioning used for memory allocation and the different sizes of blocks 3MB, 6MB, and 7MB space in memory. Now a new process p4 of size 2MB comes and demands a block of memory. It gets a memory block of 3MB but 1MB block of memory is a waste, and it can not be allocated to other processes too. This is called internal fragmentation.

- **External fragmentation:** In External Fragmentation, we have a free memory block, but we can not assign it to a process because blocks are not contiguous. **Example:** Suppose (consider the above example) three processes p1, p2, and p3 come with sizes 2MB, 4MB, and 7MB respectively. Now they get memory blocks of size 3MB, 6MB, and comes and demands a 3MB block of memory, which is available, but we can not assign it because free memory space is not contiguous. This is called external fragmentation.

read more about - Fragmentation

Comment More info

Advertise with us

Next Article

Implementation of Contiguous Memory Management Techniques

Similar Reads

Disk Management in Operating System

Disk management is one of the critical operations carried out by the operating system. It deals with organizing the data stored on the...

15+ min read

Best Ways for Operating System Memory Management

Prerequisite - Partition Allocation Methods Contiguous Memory Allocation :Probably the least difficult technique for the memory portion ...

15+ min read

Device Management in Operating System

The process of implementation, operation, and maintenance of a device by an operating system is called device management. When we use...

15+ min read

Levels of Memory in Operating System

Memory hierarchy of a computer system it handles differences in speed. "Hierarchy" is a great way to say "order of thinks" like top to bottom, fast...

Resource Management in Operating System

Resource Management in Operating System is the process to manage all the resources efficiently like CPU, memory, input/output devices, and...

15+ min read

Swap-Space Management in Operating system

Swapping is a memory management technique used in multi-programming to increase the number of processes sharing the CPU. It is ...

15+ min read

Free Space Management in Operating System

Free space management is a critical aspect of operating systems as it involves managing the available storage space on the hard disk or other...

15+ min read

Requirements of Memory Management System

Memory management keeps track of the status of each memory location, whether it is allocated or free. It allocates the memory dynamically to th...

15+ min read

Virtual Memory in Operating System

Virtual memory is a memory management technique used by operating systems to give the appearance of a large, continuous block of memory t...

15+ min read

Components of Operating System

An Operating system is an interface between users and the hardware of a computer system. It is a system software that is viewed as an organized...

15+ min read

Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305

Company

About Us
Legal
Privacy Policy
Careers
In Media
Contact Us
GfG Corporate Solution
Placement Training Program

Explore

Job-A-Thon Hiring Challenge
GfG Weekly Contest
Offline Classroom Program
DSA in JAVA/C++
Master System Design
Master CP
GeeksforGeeks Videos

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview Questions
Competitive Programming

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths

Web Technologies

HTML
CSS
JavaScript
TypeScript

Deep Learning

Tailwind CSS

Python Tutorial

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

Databases

SQL
MYSQL
PostgreSQL
PL/SQL
MongoDB

Preparation Corner

Company-Wise Recruitment Process
Aptitude Preparation
Puzzles
Company-Wise Preparation

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets

Machine Learning/Data Science

Complete Machine Learning & Data Science Program - [LIVE]
Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
Data Science Training Program - [LIVE]
Data Science Course with IBM Certification

Programming Languages

C Programming with Data Structures
C++ Programming Course
Java Programming Course
Python Full Course

Clouds/Devops

GATE 2026

GATE Rank Predictor