Difference Between Static and Dynamic Memory Allocation
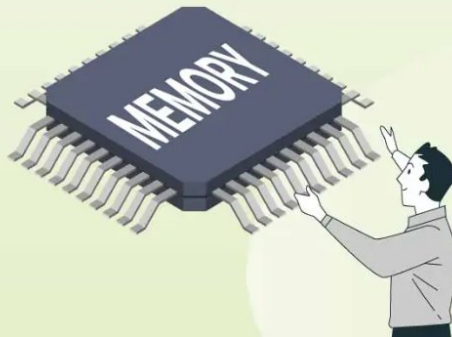
Shiksha Online ✔

Updated on Oct 25, 2023 09:08 IST

This article includes the difference between static and dynamic memory allocation with examples and applications

Managing computer memory is a difficult task. Even professionals must study how to allocate and deallocate memory regularly. Each operating system creates a specific area of memory for applications to use, and each can be allocated a limited amount of memory. The way computer memory is allocated impacts system performance and user accessibility. When memory allocation is done at compile time, it is called static memory allocation. The compiler allocates memory for variables. It is called **dynamic memory allocation** when memory allocation occurs at run time or runtime. In this article we will discuss Difference between static and dynamic memory allocation in os.

Table of contents

Difference between Static and Dynamic memory allocation

Parameter	Static Memory Allocation	Dynamic Memory Allocation
Compile time/Run time	When the allocation of memory performs at the compile time, then it is known as static memory.	When the memory allocation is done at the execution or run time, then it is known as dynamic memory allocation.
Memory reusability	With static memory allocation there is no memory reusability.	Dynamic memory allocation enables memory reuse and frees memory when it is not needed.
Memory allocation	Once memory is allocated, it can be used throughout the application.	Memory can be allocated at any point in your application.
Speed	Faster than dynamic memory allocation.	Slower than static memory allocation.
Memory allocation change	Static memory allocation memory cannot be changed while the application is running.	Dynamic memory allocation allows memory to change while the program is running.

shiksha

Depth First Search Algorithm

Data structures like trees and graphs are traversed or explored by the depth-first search algorithm, or DFS. The algorithm starts at the root node (in the case of a graph,**...read more**

ArrayList vs. LinkedList

ArrayList and LinkedList are linear data structure and are part of collection framework present in java.util packages. In this article, we will briefly discuss the difference between ArrayList and LinkedList**...read more**

Difference between Malloc and Calloc

Malloc and Calloc are dynamic memory allocation methods in C language. You will learn the differences between these two. And will also see the applications of them.

What is Static memory allocation?

Static memory allocation is performed when the compiler compiles your program and produces an object file. The linker merges all these object files to create a single executable file. The loader loads and executes this single executable file into the main memory. Static memory allocation requires knowing the size of the data it needs before it starts running.

You May Like – [Difference Between Paging And Segmentation](#)

Example of static memory allocation

- **Array declaration:** It is an example of static memory allocation. The size of the array must be known in advance. Arrays cannot be resized after memory has been allocated.

- **Real-time systems:** Real-time systems often require deterministic behavior and predictable memory usage. Static memory allocation allows for upfront memory allocation, reducing the risk of unexpected delays or interruptions during runtime.

- Performance-critical applications: Static memory allocation can perform better than

dynamic allocation. With static allocation, memory addresses are known at compile-time, enabling efficient memory access and reducing runtime overhead.

- Device drivers: Device drivers interact with hardware devices and often use static memory allocation. It allows them to manage memory buffers for data exchange with devices efficiently and ensures reliable and predictable memory access.

Application of Static memory allocation

Array-An array is a collection of similar data stored in contiguous locations. This simplest data structure allows direct access to each data item using only the index number.

What is Dynamic memory allocation?
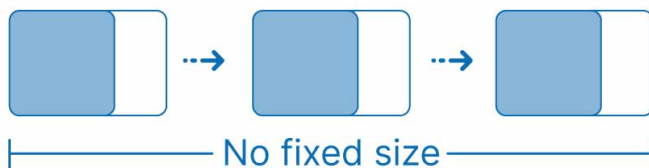
Dynamic Memory Allocation

Dynamic memory allocation is performed during program execution. Memory is allocated to an instance of a program when it is first used during program execution. Since the actual size of the data required is known at run time, the program allocates the correct amount of memory, reducing memory waste.

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. If the program is large enough, then a dynamic memory allocation is performed on the different parts of the program, which is to be used currently. This reduces memory wastage and improves the performance of the system.

Example of dynamic memory allocation

we use to allocate memory at runtime dynamically are −

- malloc () − allocates a block of memory in bytes at runtime.
- calloc () − allocating non-stop blocks of memory at run time.

Application of Dynamic Memory Allocation

This dynamic memory allocation is commonly used for

- **Linked lists**: This data structure is a set of linearly connected nodes. Each node has at least two values. Node data (content) and node next/reference (referring to the node being "linked").
- **Heap**: A heap is a particular type of tree that satisfies heap properties. The relationship between parent node P and child node C is constant throughout the tree.
- **Queue**: A queue is defined as a linear data structure with open ends, and operations are performed in first-in-first-out (FIFO) order.
- **Stack**: A stack is a restricted data structure. Elements can only be added and removed from the top of the stack. Push adds an element to the top of the stack. Pop removes an element from the top. A helpful analogy is a stack of books. You can also add new books by removing the top book.

Key Points about Dynamic memory allocation and Static memory allocation

- Static memory allocation is efficient because it allocates memory to the process before it starts running. Therefore, the program runs faster because there is no overhead of memory allocation operations while the program is running. On the other hand, the dynamic memory allocation scheme allocates memory to the process while running.
- With static memory allocation, the required memory size must be known before running

the program. With dynamic memory allocation, you don't know how much your program needs, so the expected amount of memory is allocated to your process. This leads to wasted memory.

- Dynamic memory allocation is performed at program runtime. So avoid wasting memory and allocate the correct amount of memory to your program.

- Dynamic memory allocation has the overhead of memory allocation operations during program execution, slowing program execution.

- Dynamic memory allocation provides flexibility because if the program is large enough, memory allocation operations are performed in different parts of the program, reducing memory waste.

Conclusion

Managing computer memory can be tricky since many factors must work harmoniously. Memory management is essential for running applications, optimizing system performance, and restricting access to pornographic websites, among other things. Fortunately, most computer users have little to no knowledge in this field as they accept how their computers work based on years of development by experts in the field of computing.

Related reads

❝Learning About Time Sharing Operating System

❝Importance of Java Memory Management

❝Memory Management Techniques in Operating System

❝Malloc in C Programming: How to Allocate Memory Dynamically

❝ Memory Management in Python

❝ Insertion Sort Algorithm (With Code)

❝ Selection Sort Algorithm in C

You can also explore – **Data Structures and Algorithms Online Courses & Certifications**

FAQs

What are the advantages of dynamic memory allocation?

When should I use static or dynamic memory allocation?

Which programming languages support static and dynamic memory allocation?