Memory leak in C++

Last Updated : 12 Mar, 2025

In C++, **memory leak** is a situation where the memory allocated for a particular task remains allocated even after it is no longer needed. This leads to the wastage of memory because it is unavailable for other tasks till the end of the program.

Why memory leak occurs in C++?

In C++, there is no automatic garbage collection. It means that any memory that is dynamically allocated by the programmer needs to be freed after its usage manually by the programmer. If the programmer forgets to free this memory, it will not be deallocated till the program lives and will be unavailable to other processes. This is called memory leak.

Example:

```
#include <stdlib.h>

void f() {

    // Allocate memory
    int* ptr = new int[10];

    // Return without freeing ptr
    return;
}

int main() {
    // do some tasks
}
```

Memory for an array of 10 integers is allocated using new, but the memory is never freed. This results in a memory leak since the allocated memory is no longer accessible but remains occupied.

Consequences of Memory Leakage

Different problems occur when memory leakage happens:

- **Reduced Performance:** Leaked memory is not available for other parts of your program or other programs running on the computer. Over time, if leaks are

significant, program and even the whole system can slow down due to lack of available memory.

- **Program Crashes:** If program keeps leaking memory, it might eventually use up all the available RAM. When this happens, the program might become unstable, start behaving erratically, or crash entirely.
- **Resource Depletion:** Memory is a limited resource and memory leaks can lead to resource depletion.
- **Long-Running Programs Suffer Most:** Memory leaks are often more of a problem in programs that run for a long time (like servers, daemons, etc.). Short-running programs might leak memory for little time, but long running program will keep that memory for a long period of time.

How to Find Memory Leaks?

Detecting memory leaks in C++ can be challenging due to the lack of automatic memory management. However, several strategies and tools can help detect and prevent memory leaks. There are basically two methods:

- **Manual Method**: It is basically going through the whole code and finding the memory which is not deallocated using corresponding delete operation.
- **Using Tools**: Tools like Valgrind, AddressSanitizer, etc. makes it easier to find the sources or memory leak without reviewing the whole code.

To know more methods to find memory leaks in C++, refer to the article - Detect Memory Leaks in C++

How to Avoid Memory Leaks?

The most basic method is to avoid memory leak is by careful usage of new and delete. For every time you allocate memory (e.g., with new), make sure there's a corresponding delete to free it when it's no longer needed.

There are also some methods that helps in reducing memory leaks:

- **Smart Pointers:** Use smart pointers wherever you can and avoid using raw pointers. They automatically manage memory and deallocate it when objects go out of scope.

- **Constructor and Destructor:** If you have complex memory allocation (like 2d array), wrap your resource in a class and then manually deallocate the memory using destructor which will be automatically called when the object goes out of scope.
- **Use Detection Tools**: Use memory leak detection tools such as **Valgrind** to look for memory leaks and handle them.

Next Article
Vector in C++ STL

Similar Reads

Deleting Memory in C

In C programming, we might allocate memory dynamically for various tasks but what happens when those pieces of memory are no longer needed? If not managed properly, they can lead to memory leaks,...

5 min read

memcpy() in C

The memcpy() function in C is defined in the <string.h> header is a part of the standard library in C. The memcpy() function is used to copy a block of memory from one location to another. Example: C#include...

2 min read

Vector in C++ STL

C++ vector is a dynamic array that stores collection of elements same type in contiguous memory. It has the ability to resize itself automatically when an element is inserted or deleted.Create a VectorBefore...

7 min read

vector swap() in C++

In C++, std::vector::swap() is a built-in function used to exchange the contents to two vectors of same type. This function does not copy, move or swap the individual elements, instead, it swaps the internal...

3 min read

Memory Model in C++ 11

Memory Model is a specification that describes how the program interacts with the memory. In C++ 11, a standardized memory model is created to provide the solution to issues surrounding concurrency,...

5 min read

Memset in C++

C++ memset() is a function that copies a single character for a specified number of times to the given bytes of memory. It is useful for filling a number of bytes with a given value starting from a specific...

5 min read

Vector data() in C++ STL

In C++, the vector data() is a built-in function used to access the internal array used by the vector to store its elements. In this article, we will learn about vector data() in C++.Letâ€™s take a look at an example...

2 min read

weak_ptr in C++

The weak_ptr is one of the smart pointers that provide the capability of a pointer with some reduced risks as compared to the raw pointer. The weak_ptr, just like shared_ptr has the capability to point to the...

3 min read

NULL Pointer in C++

A NULL Pointer in C++ indicates the absence of a valid memory address in C++. It tells that the pointer is not pointing to any valid memory location In other words, it has the value "NULL" (or 'nullptr' since...

4 min read

What is a Memory Heap?

What is Heap memory? Heaps are memory areas allocated to each program. Memory allocated to heaps can be dynamically allocated, unlike memory allocated to stacks. As a result, the heap segment can be...

5 min read