Project Title

Dynamic Memory Allocator .NET Desktop Application

Submitted by

Ayesha Islam Qadri (2020-CE-36)

Fatima Sohail Shaukat (2020-CE-37)

Ayesha Shafique (2020-CE-39)

Submitted to

Ma'am Darakhshan Abdul Ghaffar

Course

CMPE-331L Operating Systems

Semester:

5th

Date

6th December 2022

Department of Computer of Engineering

University of Engineering and Technology, Lahore

Table of Contents

Problem Statement

Fixed memory allocation is therefore defined as the system of dividing memory into non-overlapping sizes that are fixed, unmovable, and static. A process may be loaded into a partition of equal or greater size and is confined to its allocated partition. If we have comparatively small processes with respect to the fixed partition sizes, this poses a big problem. [1] This results in occupying all partitions with lots of unoccupied space left. Within the fixed partition context, this is known as internal fragmentation. An alternate solution to address these problems is dynamic memory allocation. Still, dynamic memory allocation can cause internal fragmentation. There will be external fragmentation despite the absence of internal fragmentation. So, our main concern is to remove internal and external fragmentation by staying under the umbrella of contiguous memory management techniques.

Abstract

Besides the responsibility of managing processes, the operating system must efficiently manage the primary memory of the computer. The part of the operating system which handles this responsibility is called the memory manager. Since every process must have some amount of primary memory to execute, the performance of the memory manager is crucial to the performance of the entire system. The memory manager is responsible for allocating primary memory to processes and for assisting the programmer in loading and storing the contents of the primary memory. Managing the sharing of primary memory and minimizing memory access time are the basic goals of the memory manager. The real challenge of efficiently managing memory is seen in the case of a system that has multiple processes running at the same time. [2]

Objective

Memory Management Techniques are basic techniques that are used in managing the memory in the operating system. Memory Management Techniques are classified into two categories; Contiguous and Non-contiguous. The contiguous technique is further divided into fixed and dynamic memory allocation. In our project, our main focus is dynamic memory allocation and handling of external fragmentation. For this purpose, we will implement First fit, Best fit, and Worst fit. To overcome the problem of external fragmentation, compaction technique or non-contiguous memory management techniques are used. We will implement the compaction technique as our coalescing policy. Compaction here means moving all the processes toward the top or toward the bottom to make free available memory in a single continuous place. In a nutshell, we will make a desktop application for better visualization of dynamic memory allocation.

Scope

Dynamic Memory Allocation is a part of the Contiguous allocation technique. It is used to alleviate the problem faced by Fixed Memory allocation. In contrast with fixed memory allocation, partitions are not made before the execution or during system configuration. Initially, RAM is empty and partitions are made during the run-time according to the process's need instead of partitioning during system configuration. The size of the partition will be equal to the incoming process. The partition size varies according to the need of the process so that internal fragmentation can be avoided to ensure efficient utilization of RAM. [3]

Features

Our desktop-based dynamic memory allocation application has the following features

- Users can specify Memory Size.
- Users can add holes and specify their sizes and starting addresses in memory.
- Users can add and remove processes.
- Users can select a memory allocation technique for each process allocation in memory.
- Users can visualize the memory on each update and deletion.
- Users can handle external fragmentation using coalescing policy.
- All exceptions will be handled in a good manner.

Explanation

For both fixed and dynamic memory allocation schemes, the operating system must keep a list of each memory location noting which are free and which are busy. Then as new processes come into the system, the free partitions must be allocated. These partitions may be allocated in 3 ways:

1. **First-Fit Memory Allocation**

This algorithm searches along the list looking for the first segment that is large enough to accommodate the process. The segment is then split into a hole and a process. This method is fast as the first available hole that is large enough to accommodate the process is used.

2. **Best-Fit Memory Allocation**

Best fit searches the entire list and uses the smallest hole that is large enough to accommodate the process. The idea is that it is better not to split up a larger hole that might be needed later. The best fit is slower than the first fit as it must search the entire list every time. It has also been shown that the best fit performs worse than the first fit as it tends to leave lots of small gaps.

3. **Worst-Fit Memory Allocation**

As the best fit leaves many small, useless holes it might be a good idea to always use the largest hole available. The idea is that splitting a large hole into two will leave a large enough hole to be useful. It has been shown that this algorithm is not very good either.

Languages and Framework

We're going to implement our project in .NET Framework 4.6 using C#. All implementation is done using Visual Studio 2022. [4]

Dotnet API

We will use the following .NET APIs in our project:

- System Namespace
- System.Drawing Namespace
- System.Windows.Forms Namespace
- System.Collections.Generic Namespace [5]

Flowchart

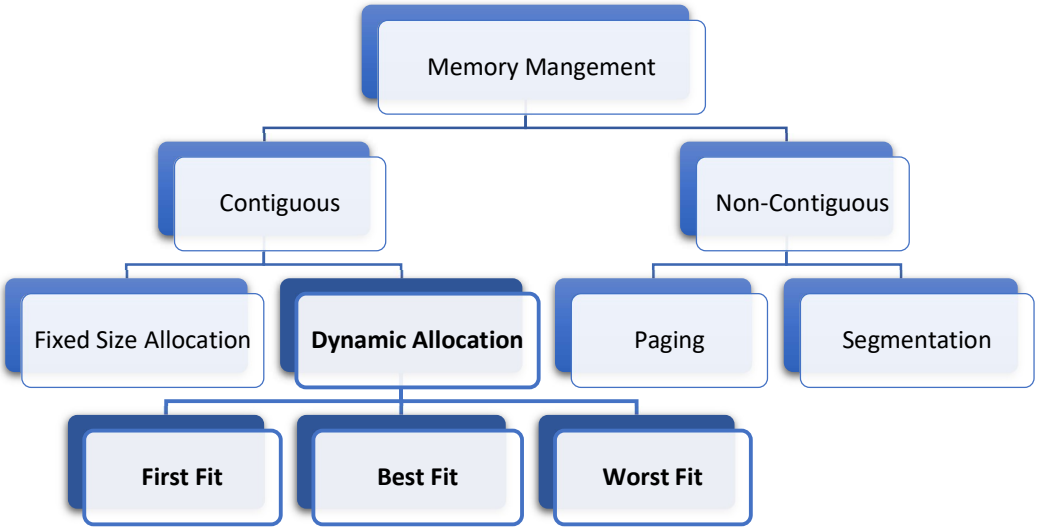

Figure 1 Memory Management in OS

References

[1] https://study.com/academy/lesson/what-is-memory-partitioning-definition-concept.html

[2] International Journal of Emerging Engineering Research and Technology Volume 3, Issue 9

[3] https://www.geeksforgeeks.org/variable-or-dynamic-partitioning-in-operating-system/

[4] https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/

[5] https://learn.microsoft.com/en-us/dotnet/api/?view=netframework-4.6&preserve-view=true

Comments

_____

Signature