International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Memory Management Techniques: Static and Dynamic Memory Allocation

Supriya Pralhad Mali#, Sonali Dohe# and Priya Rangdal^

#Department of Computer Engineering, Government Residence Women Polytechnic, Tasgaon, Sangli (dist.) Maharashtra, India
^Government Polytechnic, Ahmednagar (dist.) Maharashtra, India

Received 07 Dec 2018, Accepted 08 Feb 2019, Available online 10 Feb 2019, **Vol.9, No.1 (Jan/Feb 2019)**

Abstract

In modern computer system, Memory management is one of the important function of the operating system. Dynamic Memory Allocation plays very vital role in Memory Management and becomes fundamental part of today's computer system. It is minimizing the cost, by providing efficient use of Memory. Memory Management is commonly one of the most significant part of Operating System. In this paper, we will describe Static Memory Allocation and Dynamic Memory Allocation in Memory Management, comparison between both.

Keywords*: OS: Operating System, DMA: Dynamic Memory Allocation, MM: Memory Management, DMM: Dynamic Memory Management, CPU: Central Processing Unit.*

1. Introduction

Now days the computer science firms are going through significant changes because of the development of new technologies, moving towards big data etc. Modern computer systems must adapt to requirements, such as efficient memory management, resource management, virtual memory management, efficient implementation process, good user interface, efficient inter process communication etc. To determine these significant requirements for modern operating system design, we need to concentrate on these requirements.

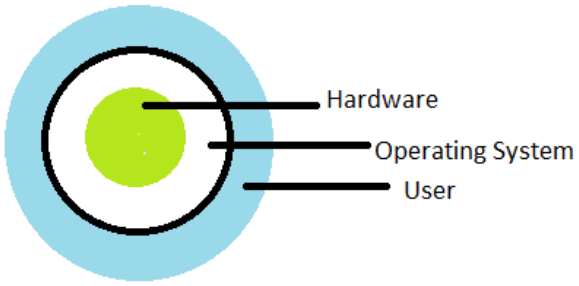

Fig.1 OS acts as intermediate between Hardware and User.

Operating system is an interface between hardware and application programs or end user. OS provides interface to run application programs on hardware. The design of OS is not easy task; we have to take care about above mentioned criteria. It is mostly developed in C, Assembly and C++ programming language. The basic functions of OS include Process Management, Memory Management, Storage Management, device management, Protection and Security as shown in fig.2.

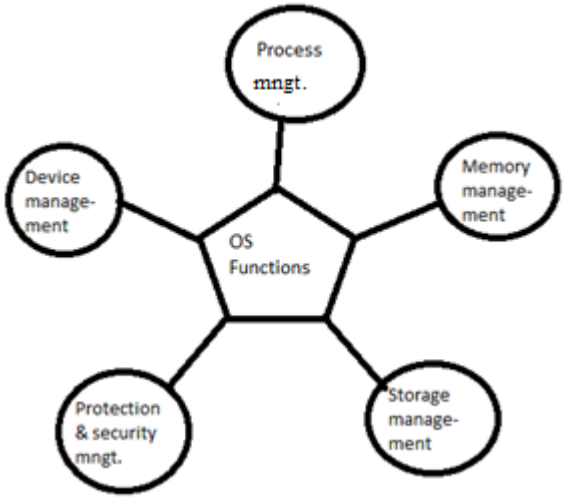

Fig.2 Functions of OS.

2. Memory Management Techniques

Memory management is the functionality of an operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution. Memory

*Corresponding author's ORCID ID: 0000-0002-7918-1435
DOI: https://doi.org/10.14741/ijcet/v.9.1.13

management keeps track of each and every memory location, regardless of either it is allocated to some process or it is free. It checks how much memory is to be allocated to processes. It decides which process will get memory at what time. It tracks whenever some memory gets freed or unallocated and correspondingly it updates the status.

Two techniques are used for allocating memory, as shown in fig.3

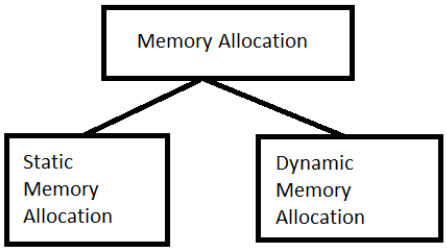

Fig.3 Memory Allocation Techniques.

Every memory allocation technique has its own pros and cons and it justify their performance for the purpose these techniques are developed. Our intent is to figure out which of these two techniques is best and work more effectively. First we will discuss about static memory allocation.

3. Static Memory Allocation

In Static memory allocation, memory is allocated at the beginning of execution to application's variables. In this, even if do not need most of the memory at particular instance of program or variable still we could not use allocated memory for any other purpose. Every day large amount of data is generating termed as Big data, this data need to be stored and maintain replication of it, to avoid data loss because of system crash or natural disaster and for fast accessing purpose. So, that we have to use available memory in efficient manner. Memory management issue comes because of inefficient way of allocating memory and de-allocating memory.

Now we see concept of static memory allocation with example, we declare static array to stored integer data, when we declare array of 500, required memory = 500*2 = 1000 bytes on 32 bit operating system and required memory = 500*4 = 2000 on 64 bit operating system. 1000 (on Windows) and 2000 (on Linux) has reserved for above declared array and we could not use that memory, even that array contains only one integer or no data. Suppose if array contains 500 integers, we have deleted 499 still we could not use that memory for other purpose.

This technique is called as static memory allocation. Above problem with memory management arises because of static memory allocation, in such situation static memory allocation fails to handle memory management efficiently. Still static memory allocation has few advantages over dynamic memory allocation like allocating fast speed, no extra algorithms, mostly not faced fragmentation problem required to achieve allocation task. Even these benefits with static memory allocation still mostly we prefer dynamic allocation because of its way of allocating memory.

Now we will discuss about dynamic memory allocation technique.

4. Dynamic Memory Allocation

Dynamic Memory Allocation is also known as Manual Memory Management. In DMA the memory is allocated at run time. It is allocated whenever program, application, data, variable demands with required amount of bytes.

We are going to present, manual memory allocation using 'C' programming language because it will very easy to show allocation and de-allocation. The programmer has direct access to memory at run time to control over memory using DMA. It is mostly explicit call to heap management functions. 'C' programming language has supports malloc(), calloc() and realloc() functions to allocate memory for our program or applications. These functions are called dynamic memory allocation functions; with the help of these functions we can allocate required size of memory at run time. The important mechanism about this strategy of allocation, once utilized allocated memory, and that memory is no longer requires for program, application or variable which can be again available to other purpose. 'C' language has provided free function to de-allocate unused memory (C++ Programming language used 'new' keyword to allocate memory and 'delete' keyword to free allocated memory).

This strategy also has some advantages and disadvantages. The most important advantage of this technique is that, when there is short memory, it plays better role for memory management. Because it allocates require bytes of memory only at run time and de-allocate memory after its use, it will be available for reuse. Keeping track of allocated and free memory is very difficult task but now it has maintained by modern operating system by providing some complexity to this.

Now, we will discuss dynamic memory allocation algorithm.

5. Dynamic Memory Allocation Algorithms

An available block of memory to store the process is called a hole. There are four algorithms used to allocate the memory dynamically:

1. First-fit: Allocate the first hole that is big enough. Searching start at the beginning of the set of holes .we

can stop searching as soon as we find a free hole that is large enough.

2. Next-fit: This behaves exactly as the first fit except that the scan begins from where the previous one left off.

3. Best-fit: Allocate the smallest hole big enough. We must search the entire list, unless the list is kept ordered by size. This strategy produces the smallest leftover hole.

4. Worst-fit: Allocate the largest hole. Again we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole from a best-fit approach.

A study done of the efficiency of next-fit, first-fit, and best-fit showed that in some cases, next-fit performs worse than first-fit or best-fit. When the mean size of the block is less than one-sixteenth the available memory, first-fit performs the best of all, with best-fit close to the performance of first-fit, and next-fit being substantially inferior to both of them.

6. Static vs Dynamic Memory Allocation

To compare static and dynamic memory allocation, we are going to take one simple example, so that we could find out how dynamic memory allocation uses memory efficiently than static memory allocation in some situations.

Suppose I have opened an organization, so need to maintain information of employees of organization. To implement employee information, we can do that in two ways, one using static memory allocation and second using dynamic memory allocation. Before implementing employee information, we have to think about growth of organization after 5-10 years, approximate how many employees will be on payroll. Consider at the beginning of this organization only 10 employees and after 10 years organization may have 5000 employee approximately. To implement employee information using static implementation, we can use array. We will declare array of structure for 5000 employee, because once developed its very difficult to change. Second way is to implement employee information using dynamic implementation, no need to allocate memory for 5000 memories at beginning. Whenever we will add new employee the memory is allocated at run time and we will delete employee we can recycled that memory after de-allocating if we don't that information in future.

Now we are going to address why dynamic allocation is better than static allocation. As per above example from beginning of inserting employee information using static technique, we have to allocate memory for 5000 employee even organization has 10 employee. The memory has wastage means we could not use allocated memory. While using dynamically implementation of employee information, the memory is allocated at run time whenever new employees will be added, the memory is allocated. This shows dynamic allocation efficiently used memory than static allocation. Second advantage is even if we delete employees from the organization we could not use that memory in case of static technique while in case of dynamic technique if we delete employee record the memory will available for us to use for other purpose.

Table 1 Comparison between Static and Dynamic Memory Allocation

Static Memory Allocation	Dynamic Memory Allocation
Allocation is done before program execution.	Allocation is done during program execution.
There is no memory reusability.	There is memory reusability.
Stack data structure is used for implementing static memory allocation	Heap data structure is used for implementing static memory allocation
Faster execution than dynamic.	Slower execution than static.
It is best if required size of memory is known in advance.	It is best if we don't have idea about how much memory required.

7. Dynamic Memory Allocation Work

Dynamic allocation plays vital role in memory management. Here is explanation, how does dynamic memory allocation works? C programming language is used to explain the working of the dynamic memory allocation technique.

Problem: Allocate memory for 5000 integers and after executing the process, de-allocate memory for recycled it.

Solution: Header file stdlib.h contains different library fuctions which includes functions like calloc() , malloc()and realloc() for allocation while free() for de-allocation of memory. The task is to allocate memory for 5000 integers; this can be done in two ways statically or dynamically. This is accomplished dynamically as follows: Suppose variable name: info (integer type). Dynamic allocation will require pointer variable instead of normal variable, symbol * (asterisk) shows pointer variable declaration. int * info; The above statement, declare 'info' as integer pointer. malloc() function, and also use sizeof() function will gives the number of bytes require for integer data type. It will help because size of integer is changes as per operating system (Linux:4 bytes and Windows: 2 bytes). info = (int *) malloc (5000 * sizeof(int)); malloc() function returns void pointer, so need to convert into our destination type means integer. Above statement will allocate memory for 15000 integers and 'info' pointer will point to first block of allocated memory. Now, next task is de-allocation of memory. free() function is used to de-allocate memory. Following statement will accomplish this task: free(info); It will de-allocate memory to reuse, but still 'info' pointer will points to that memory location. We can called 'info' pointer is a dangling pointer because it pointing to memory even it is de-allocated. Dangling pointer is a pointer which is pointing to nothing. NULL

value is used to remove dangling pointer: data = NULL; above statement will remove dangling pointer that means 'info' will point to null instead of any random memory. Dynamic memory allocation is performed by the memory allocator when the program is executing and the program (user) or application sends a request for additional memory. Some issues need to be addressed by the operating system:

1. Allocating variable-length dynamic storage when required
2. Free the storage when execution is completed.
3. Control the storage (e.g. reclaiming freed up memory for later use)

8. Problem with Dynamic Memory Allocation

DMA has two sides; one side is pros that we have seen above. Now we are going to see second side, cons of using DMA. There are some problems with DMA as follows:

Memory leak

Memory leak is a condition in which some programs or application which is continuously allocate memory without ever giving it up and finally run out of memory. It will affect data loss.

Dangling pointer

It is also called as premature free. After using the memory, the memory deallocate by using the function of memory deallocation (free () in C-language) but pointer will still points to that memory location. Whenever the same memory while allocating to other programs or applications then it will crash or behave randomly. To prevent dangling pointer one more task we have to perform with de-allocating memory is to assigning the NULL value to dangling pointer.

Time consuming

Memory management with the help of dynamic memory allocation is a time consuming process as compare to static memory allocation technique.

Memory Fragmentation

It is very chronic with dynamic memory allocation because of the external fragmentation in some situation. For instance, as we know using dynamic allocation technique the memory may not allocated continuously ,wherever available required size of memory are available ,that memory is allocated to the programs.

Suppose we have following memory structure which shown in fig 3. There is 70 bytes free space available. This 70 bytes of memory available in two parts, first is 20 bytes and second one 50 bytes. Remaining memory has allocated for other purpose. And request is come to memory manager, to allocate memory of 57 bytes but memory allocator could not allocate memory even it has 50 bytes of free memory, because of fragmentation.

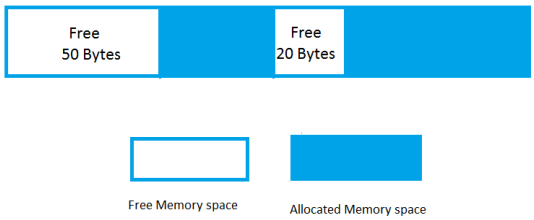

Fig.4 Representation of free and allocated memory space

Conclusions

In this paper, we presented role of memory allocation in OS. Along with this we discussed memory allocation techniques i.e. static and dynamic. And by comparing both the techniques we conclude that dynamic memory allocation is best strategy for allocating the memory when needed.

References

Amanjot Kaur Randhawa, Alka Bamotra[2017], Study of static and dynamic memory allocation, *International Journal of Innovative Computer Science & Engineering*, Volume 4 Issue 3; Page No. 127-131.

Meenu, Vinay Dhull ,Monika (2015)" computational study of static and dynamic memory allocation" *International Journal of Advanced Research in Computer Science and Software Engineering* 5(8)

Nilesh Vishwasrao Patil, Prabhudev S Irabashetti (2014), Dynamic Memory Allocation :Role in Memory Management ,*International Journal of Current Engineering and Technology*,Vol 4,No.2

Dharmender Aswal, Krishna Sharda, Mahipal Butola (2014)" research paper on DMA:Dynamic memory allocation" *IJIRT*, Volume 1, Issue 6.

Dipti Diwase, Shraddha Shah, Tushar Diwase, Priya Rathod. (2012). Survey Report on Memory Allocation Strategies for Real Time Operating System in Context with Embedded Devices. *IJERA Internet Computing* [Online]. 2(3), pp. 1151-1156. Available

M. Masmano, I. Ripoll, A. Crispo Dynamic Storage Allocation for real time embedded systems Universidad politecnica de Valencia, Spain.

Manish Mehta, David J. DeWitt Dynamic Memory Allocation for Multiple-Query Workloads Computer Science Department, University of Wisconsin-Madison.

Krishna M. Kavi, Merhan Rezaei, Ron Cytron An Efficient Memory Management Technique That Improves Localities University of Alabama in Huntsville and Washington University in Saint Louis.