

Chapter 2

Memory Management: Early Systems

At a Glance

Instructor's Manual Table of Contents

- Overview
- Objectives
- Teaching Tips
- Quick Quizzes
- Class Discussion Topics
- Additional Projects
- Additional Resources
- Key Terms

Lecture Notes

Overview

The management of main memory is critical. In fact, from a historical perspective, the performance of the entire system has been directly dependent on two things: how much memory is available and how it is optimized while jobs are being processed. This chapter introduces the Memory Manager and four types of memory allocation schemes: single-user systems, fixed partitions, dynamic partitions, and relocatable dynamic partitions. The best-fit and the first-fit memory allocation schemes are discussed. The importance of deallocation of memory in a dynamic partition system and the importance of the bounds register in memory allocation schemes are explained. The chapter also discusses the role of compaction and how it improves memory allocation efficiency. These early memory management schemes are seldom used by today's operating systems, but they are important to study because each one introduced fundamental concepts that helped memory management evolve.

Learning Objectives

After completing this chapter, the student should be able to describe:

- The basic functionality of the three memory allocation schemes presented in this chapter: fixed partitions, dynamic partitions, relocatable dynamic partitions
- Best-fit memory allocation as well as first-fit memory allocation schemes
- How a memory list keeps track of available memory
- The importance of deallocation of memory in a dynamic partition system
- The importance of the bounds register in memory allocation schemes
- The role of compaction and how it improves memory allocation efficiency

Teaching Tips

Single-User Contiguous Scheme

1. Provide students with an overview of a single-user contiguous scheme. Point out that this scheme required a minimal amount of work done by the operating system's Memory Manager, because a program was loaded in its entirety into memory and allocated as much contiguous space in memory as it needed. However, the problem was that if the program didn't fit the available memory space, it could not be executed.
2. Point out that jobs are processed sequentially in a single-user system. Explain the algorithm to load a job in a single-user system, as shown on page 33 in the text.
3. Outline the major disadvantages of a single-user contiguous scheme, including the fact that it does not support multiprogramming and it is not cost effective.

Fixed Partitions

1. Explain the concept of a fixed-partition scheme along with its various features. Point out its major advantage over a single-user contiguous scheme, which is that it allowed multiprogramming.
2. Be sure to mention that partition sizes remained static unless and until the computer system was shut down, reconfigured, and restarted.
3. Outline the critical factor that was introduced with this scheme, which is protection of the job's memory space.
4. Explain the algorithm to load a job in a fixed-partition scheme, as listed on page 34 in the text. Point out that this algorithm required matching of job size with the size of the partition. Use Table 2.1 and Figure 2.1 on page 35 to clarify the concept of fixed-partition.
5. Outline various disadvantages of the fixed-partition scheme, such as that it still requires the entire program to be stored contiguously and that jobs are allocated space on the basis of first available partition of required size. Note that this scheme works well only if all of the jobs are of the same size or if the sizes are known ahead of time, and arbitrary partition sizes lead to the problem of memory waste or internal fragmentation – a major drawback to the fixed partition memory allocation scheme.

Teaching Tip	Fixed partitioning allowed multiprogramming, but arbitrary partition sizes led to the problem of internal fragmentation.
---------------------	--

Dynamic Partitions

1. Explain the concept of a dynamic partition scheme. Point out its advantages over fixed partitions, such as the fact that jobs are given only as much memory as they request when they are loaded for processing, which leads to relatively less memory waste than fixed partition schemes.
2. Point out various disadvantages of a dynamic partition scheme, such as the fact that it still requires an entire program to be stored contiguously and it fully utilizes memory only when the first jobs are loaded. Emphasize that all subsequent allocation of memory leads to memory waste or external fragmentation.
3. Use Figure 2.2 on page 37 of the text to clarify the concept of dynamic partition.

Quick Quiz 1

1. Which of the following memory management schemes does not allow multiprogramming?
 - a. Single-user contiguous scheme
 - b. Fixed partition
 - c. Dynamic partition
 - d. Relocatable dynamic partitionsAnswer: a

2. Which of the following are the disadvantages of a fixed partition scheme (choose all that apply)?
 - a. Requires that the entire program be loaded into memory
 - b. Requires that the entire program be stored contiguously
 - c. Requires that the entire program remain in memory until the job is completed
 - d. Does not allow multiprogrammingAnswer: a, b, and c

3. The phenomenon of partial usage of fixed partitions and the coinciding creation of unused spaces within the partition is called _____.
Answer: internal fragmentation

Best-Fit Versus First-Fit Allocation

1. Point out to the students that partitions in both fixed and dynamic memory allocation schemes may be allocated on the basis of first-fit memory allocation or best-fit memory allocation.
2. Explain the difference between these two allocation processes. Point out the advantages and disadvantages of each, such as first-fit memory allocation is faster in making allocation but leads to memory waste, while best-fit memory allocation makes the best use of memory space but is slower in making allocation.
3. Provide examples to clarify the concept of first-fit memory allocation and best-fit memory allocation schemes, using Figures 2.3 and 2.4 on pages 39-40 in the text.
4. Explain the algorithms of both schemes as listed on pages 41-42 in the text. Use Table 2.2 and Table 2.3 to clarify these concepts. Point out that the algorithm for best-fit is slightly more complex because the goal is to find the smallest memory block into which the job will fit, and the entire table must be searched before the allocation can be made.
5. Discuss briefly some of the hypothetical allocation schemes, such as next-fit that starts searching from the last allocated block for the next available block when a new job arrives, and worst-fit that allocates the largest free available block to the new job.

Teaching Tip	To simulate the best-fit, worst-fit, and first-fit strategies for memory allocation, refer to the following Web site: http://courses.cs.vt.edu/csonline/OS/Lessons/MemoryAllocation/Lesson.html .
---------------------	--

Deallocation

1. Define the concept of the deallocation process. Explain the process for both a fixed partition system and a dynamic partition system.
2. Point out that for fixed-partition systems, it is a straightforward process: Memory Manager resets the status of the memory block where the job was stored to “free” when the job is completed. Note that the system may use binary codes to reflect the status.
3. Point out that in dynamic partition systems, the algorithm tries to combine free areas of memory whenever possible. Outline three alternative situations possible in dynamic partition systems: when the block to be deallocated is adjacent to another free block, when the block to be deallocated is between two free blocks, and when the block to be deallocated is isolated from other free blocks.
4. Explain the algorithm to deallocate memory blocks as listed on page 44 in the text. Mention that the deallocation algorithm is prepared for all three eventualities with a set of nested conditionals.

Case 1: Joining Two Free Blocks

1. Use Table 2.4 and Table 2.5 on page 45 of the text to clarify the case of joining two free blocks. Point out that the list must be changed to reflect the starting address of the new free block, and the memory block size for the new free space must be changed to show its new size; that is, the combined total of the two free partitions.

Case 2: Joining Three Free Blocks

1. Use Table 2.6 and Table 2.7 on page 46 of the text to clarify the case of joining three free blocks. Point out that the list must be changed to reflect the starting address of the new free block (smallest beginning address), sizes of the three free partitions must be combined, and the status of the entry (last of the three) that is combined must be changed to a null entry.

Case 3: Deallocating an Isolated Block

1. Explain deallocation of a space that is isolated from all other free areas using Tables 2.8 through 2.11 on pages 47–48 in the text. Point out that in this case, the system must search the table for a null entry. Make sure the students understand that a null entry in the busy list occurs when a memory block between two other busy memory blocks is returned to the free list.

Relocatable Dynamic Partitions

1. Explain the concept of relocatable dynamic partitions. Point out that in this scheme, Memory Manager relocates programs to gather together all of the empty blocks and compacts them to make one block of memory large enough to accommodate some or all of the jobs waiting to get in.
2. Explain the concept of compaction. Discuss the requirements to achieve compaction: every program in memory must be relocated so they are contiguous and the operating system must distinguish between addresses and data values. Use Figures 2.5 to 2.8 on pages 50–52 in the text to clarify the relocatable dynamic partitions scheme.
3. Outline lists that need to be updated after compaction. Discuss briefly special-purpose registers used for relocation, namely the bounds register and the relocation register. Use Figure 2.8 to clarify the concept of relocation registers.
4. Make sure that students understand that with compaction, more overhead is incurred. Discuss how and when compaction needs to be done in order to optimize both processing time and memory use while keeping overhead as low as possible. Review the three approaches described on pages 53-54 in the text.

Teaching Tip	To learn about the IBM history of memory management technology, refer to the article at: www.research.ibm.com/journal/rd/255/ibmrd2505N.pdf .
---------------------	---

Quick Quiz 2

1. Which of the following memory management schemes optimizes fragmentation?
 - a. Single-user contiguous scheme
 - b. Fixed partition
 - c. Dynamic partition
 - d. Relocatable dynamic partitionsAnswer: d
2. The _____ is used to store the highest location in memory accessible by each program.
Answer: bounds register
3. _____ is the process of collecting fragments of available memory space into contiguous blocks by moving programs and data in a computer's memory or disk.
Answer: Compaction

Class Discussion Topics

1. Have students discuss the limitations of memory management schemes in early systems. What do they think was the worst aspect of all of these schemes and why? Why were these schemes sufficient for the first three generation of computers?
2. Have students discuss the compaction process. What do they think is the negative side of this process? When and how often should it be done? Discuss the various alternatives and their pros and cons.

Additional Projects

1. Have students research the algorithm for next-fit and worst-fit schemes.
2. Have students make a list of the steps needed in order to implement relocation for a program, including loop sequences, decision sequences, branching sequences, and data references.

Additional Resources

1. Microsoft.com:
www.microsoft.com
2. IBM.com:
www.ibm.com
3. A History of Operating Systems:
www.osdata.com/kind/history.htm
4. Windows Products and Technologies History:
www.microsoft.com/windows/WinHistoryIntro.mspx
5. Ravenbrook – The Memory Management Reference:
www.memorymanagement.org

Key Terms

- **Address:** a number that designates a particular memory location.
- **Best-fit memory allocation:** a main memory allocation scheme that considers all free blocks and selects for allocation the one that will result in the least amount of wasted space.
- **Bounds register:** a register used to store the highest location in memory legally accessible by each program.

- **Compaction:** the process of collecting fragments of available memory space into contiguous blocks by moving programs and data in a computer's memory or disk. Also called *garbage collection*.
- **Deallocation:** the process of freeing an allocated resource, whether memory space, a device, a file, or a CPU.
- **Dynamic partitions:** a memory allocation scheme in which jobs are given as much memory as they request when they are loaded for processing, thus creating their own partitions in main memory.
- **External fragmentation:** a situation in which the dynamic allocation of memory creates unusable fragments of free memory between blocks of busy, or allocated, memory.
- **First come first served (FCFS):** a nonpreemptive process scheduling policy that handles jobs according to their arrival time; the first job in the READY queue is processed first.
- **First-fit memory allocation:** a main memory allocation scheme that searches from the beginning of the free block list and selects for allocation the first block of memory large enough to fulfill the request.
- **Fixed partitions:** a memory allocation scheme in which main memory is sectioned off, with portions assigned to each job.
- **Internal fragmentation:** a situation in which a fixed partition is only partially used by the program; the remaining space within the partition is unavailable to any other job and is therefore wasted.
- **Kilobyte (K):** a unit of memory or storage space equal to 1,024 bytes or 210 bytes.
- **Null entry:** an empty entry in a list.
- **Relocatable dynamic partitions:** a memory allocation scheme in which the system relocates programs in memory to gather together all of the empty blocks and compact them to make one block of memory that's large enough to accommodate some or all of the jobs waiting for memory.
- **Relocation:** (1) the process of moving a program from one area of memory to another; or (2) the process of adjusting address references in a program, by either software or hardware means, to allow the program to execute correctly when loaded in different sections of memory.
- **Relocation register:** a register that contains the value that must be added to each address referenced in the program so that it will be able to access the correct memory addresses after relocation.
- **Static partitions:** another term for *fixed partitions*.