Grade: 65/65 A+

An Overview of Memory Management in Computing Systems

Alejandro Ricciardi

Colorado State University Global

CSC300: Operating Systems and Architecture

Joe Rangitsch

July 9, 2023

An Overview of Memory Management in Computing Systems

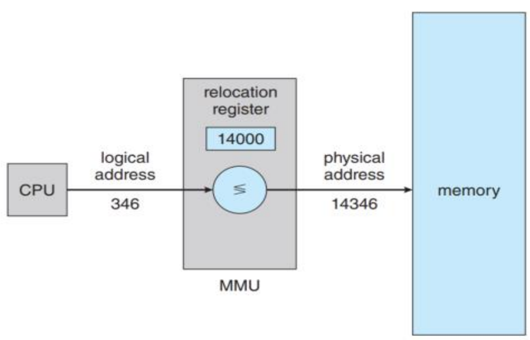
Memory management is a critical component of computer hardware and operating systems. It controls and coordinates a computer system's main memory to ensure that blocks of memory space are properly managed and allocated to processes. In other words, it ensures that all running processes have the memory they need to carry out their operations. Furthermore, memory management functions at three levels, which are the hardware, operating system, and program level (Sheldon, 2022). Moreover, memory management directly affects the overall system functionality, and it is an essential and critical component of any operating system's operation. In this essay, I will discuss memory management at the hardware level, which includes the physical components of a system's memory, and memory management at the operating system level, which includes memory allocation methods, which are virtual memory, paging, and segmentation techniques.

Memory Management at The Hardware Level

At the hardware level, memory management is concerned with the physical components of a system's memory (Sheldon, 2022), which are cache memory, Central Processing Unit (CPU) registers, main memory, and secondary memory. First, the cache memory is integrated into the CPU chip for fast access, it can have multiple levels referred to as L1, L2, and L3, and it is utilized by the CPU to quickly access frequently used data. Additionally, cache memory is volatile, meaning that it does not provide permanent storage. Second, registers are a type of volatile memory built directly into the processor or CPU that is used to store and manipulate data during the execution of processes (Hopkins, 2023). Third, the main memory, also called system memory or Random-Access Memory (RAM) is a volatile memory that is slower than cache memory but faster than the secondary memory. Furthermore, RAM is at the heart of memory management. The primary function of an operating system's memory management is to allocate and deallocate RAM to the various processes. Fourth, secondary memory is a nonvolatile memory that is mainly utilized as long-term storage. One of the roles of memory management is to transfer needed data between the RAM and the secondary memory by using either the swapping, segmenting, or paging methods. These methods are implemented by operating systems. Hence, memory management utilizes cash memory, CPU registers, main memory (RAM), and secondary memory collectively to efficiently store and retrieve data to support the CPU in executing processes. This is done at the hardware level by the memory management unit (MMU).

Figure 1

MMU



Note. From *Operating System Concepts*, by Silberschatz, Galvin, & Gagne, 2018, Figure 9.5. Copyright 2023 by Wiley.

The MMU's main role is to translate logical addresses into physical addresses (Sheldon, 2022). logical and physical addresses are sets of finite bits pointing to data stored in the RAM. The logical address, also called virtual address (not to be confused with virtual memory) is generated by the CPU during program execution. The physical address, on the other hand, points to a location in the RAM, and it is computed by the MMU. This safeguards or protects the physical memory from being directly accessed by a program (Silberschatz et al., 2018). In other words, programs cannot directly access the data stored on the RAM, they access it indirectly. Furthermore, different mapping schemes exist to map the logical address space to the physical address space. Figure 1 depicts a simple MMU scheme that is a generalization of the base register scheme. The base register or relocation register stores a base value that is used to compute the physical address from the logical address. In Figure 1 example, the CPU generated a logical address value of 346. Then the MMU generated a base address (Relocation Register) value of 14000 corresponding to the base value of the physical address space. Finally, the MMU translates the logical address into the physical address utilizing the base address, here 346 + 14000 = 14346. Thus, the MMU plays a crucial part in safeguarding and managing access to the RAM. Note that the MMU does not handle the transfer of data between the RAM and secondary memory; this is managed at the operating system level.

Memory Management at The Operation System Level

In multiprocessing systems, one of the most important roles of an operating system is to manage memory by removing inactive processes from the RAM. Overtime, operating systems utilized several technics to manage memory. Two of the earliest and now obsolete methods used by operating systems are fixed and dynamic partition techniques, they are referred to as contiguous memory allocation methods. Other methods include basic paging and simple segmentation. These are noncontiguous methods and are not used independently but are the base of the virtual memory technique (Stallings, 2018). Virtual memory is another memory management technique used by operating systems, it can utilize both paging and segmentation, either separately or in conjunction. Furthermore, the virtual memory technique is the operating system memory management method utilized in almost all modern operating systems. Thus, in multiprocessing systems environments, operating system memory management techniques -fixed and dynamic partition, paging, and segmentation techniques- played and play a crucial role in ensuring the efficiency and accessibility of the system memory.

First, the two contiguous memory allocation methods, the fixed and dynamic partition methods, are based on a swapping technique. In the swapping technique, inactive processes are moved to secondary memory to await reactivation and are replaced by new or reactivated processes. Furthermore, swapping utilizes replacement algorithms. Replacement algorithms choose free blocks of main memory that are equal to or larger than the process to be brought in (Stallings, 2018). The most used replacement algorithms are the first-fit, best-fit, and next-fit. "Best-fit chooses the block that is closest in size to the request. First-fit begins to scan memory from the beginning and chooses the first available block that is large enough. Next-fit begins to scan memory from the location of the last placement and chooses the next available block that is large enough." (Stallings, 2018, p. 321). To accommodate swapping, the fixed partition method divides the RAM into several same-size partitions at system generation. The strengths of this method are its simplicity of implementation and its low operating system overhead. Its drawbacks include a fixed maximum number of active processes and inefficient use of memory due to internal fragmentation. Internal fragmentation happens when any process, no matter how small, occupies an entire partition. In dynamic partitioning, on the other hand, when a process needs to be loaded into the RAM for the first time, a partition just large enough to hold the process is created. The benefit of dynamic partitioning is that it prevents internal fragmentation, as a result, it is a more efficient use of the RAM. Its drawbacks include inefficient use of the CPU due to the need for compaction to counter external fragmentation. External fragmentation occurs when free memory is broken into small, non-contiguous blocks over time. Thus, contiguous memory allocation methods had drawbacks, but they played a crucial role in the efficiency and accessibility of the system's memory in multiprocessing systems.

Second, the two noncontiguous memory allocation methods, paging, and segmentation methods. They are not used independently but are the base of the virtual memory technique. Virtual memory is a memory management technique where secondary memory can be used as if it were a part of RAM, i.e., it allows us to extend the use of main memory by using secondary memory. Furthermore, it can utilize both paging and segmentation methods, either separately or in conjunction. Virtual memory paging is the combination of paging and virtual memory, and Virtual memory segmentation is the combination of segmentation and virtual memory. First, virtual memory paging partitioned the RAM into small fixed-size chunks called frames (Stallings, 2018). A frame is a fixed-length block in the RAM. Furthermore, programs are segmented into pages that have the same length as the fixed-length frames. For example, 32-bit and 64-bit lengths are implemented in Windows operating systems. The benefits of virtual memory paging are no external fragmentation and virtual extension of the RAM, its drawback is an overhead of complex memory management. Second, virtual memory segmentation divides each process into several segments of the same or different sizes. These segments are then loaded into dynamically sized, contiguous RAM partitions. The size of these segments is determined by the program itself. The benefits of no internal fragmentation, a higher degree of multiprogramming, and a virtual extension of the RAM, its drawback is an overhead of complex memory management. Thus, the combination of paging and segmentation with virtual memory has one drawback, which is an overhead of complex memory management that is easily managed by modern computing systems and operating systems. Moreover, they play a crucial role in the efficiency and accessibility of the system's memory in modern multiprocessing systems.

Conclusion

To summarize, in the context of multiprocessing systems, memory management is a fundamental component of computer hardware and operating systems. Memory management functions at three levels, which are the hardware, operating system, and program level. In this essay, I discussed memory management at the hardware level and memory management at the operating system level. At the hardware level, memory management utilizes cash memory, CPU registers, main memory, and secondary memory collectively to efficiently store and retrieve data to support the CPU in executing processes. Additionally, the MMU plays a crucial part in safeguarding and managing access to the RAM. At the operating system level, the operating system's role is to manage memory by removing inactive processes from the RAM. Two of the earliest methods used by operating systems to manage memory were fixed and dynamic partition techniques, they are referred to as contiguous memory allocation methods. The contiguous memory allocation methods had drawbacks, but they played a crucial role in the efficiency and accessibility of the system's memory in multiprocessing systems. The methods utilized in modern multiprocessing systems are paging and segmentation methods. They are not used independently but are used in combination with the virtual memory technique. The virtual memory technique uses part of secondary memory as if it were a part of RAM. i.e., virtually extending the RAM capacity. Furthermore, the combinations of paging with virtual memory and segmentation with virtual memory have one drawback: an overhead of complex memory management, but they play a crucial role in the efficiency and accessibility of the system's memory in modern multiprocessing systems. Thus, memory management is essential for the overall functionality, security, and efficiency of multiprocessing systems, and it is an essential and critical component of any operating system.

References

Hopkins, J. (2023, May 24). What is a register in a CPU and how does it work? *Total Phase Blog.* htanother operating.com/blog/2023/05/what-is-register-in-cpu-how-does-it-work/

Sheldon, R. (2022, June 6). What is memory management in a computer environment? *WhatIs.com.* https://www.techtarget.com/whatis/definition/memory-management#:~:text=Memory%20management%20is%20the%20process,to%20carry%20out%20their%20operations.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* [PDF]. Wiley. Retrieved from: https://os.ecci.ucr.ac.cr/slides/Abraham-Silberschatz-Operating-System-Concepts-10th-2018.pdf

Stallings, W. (2018). *Operating systems: Internals and design principles*. Pearson.