MEMORY MANAGEMENT

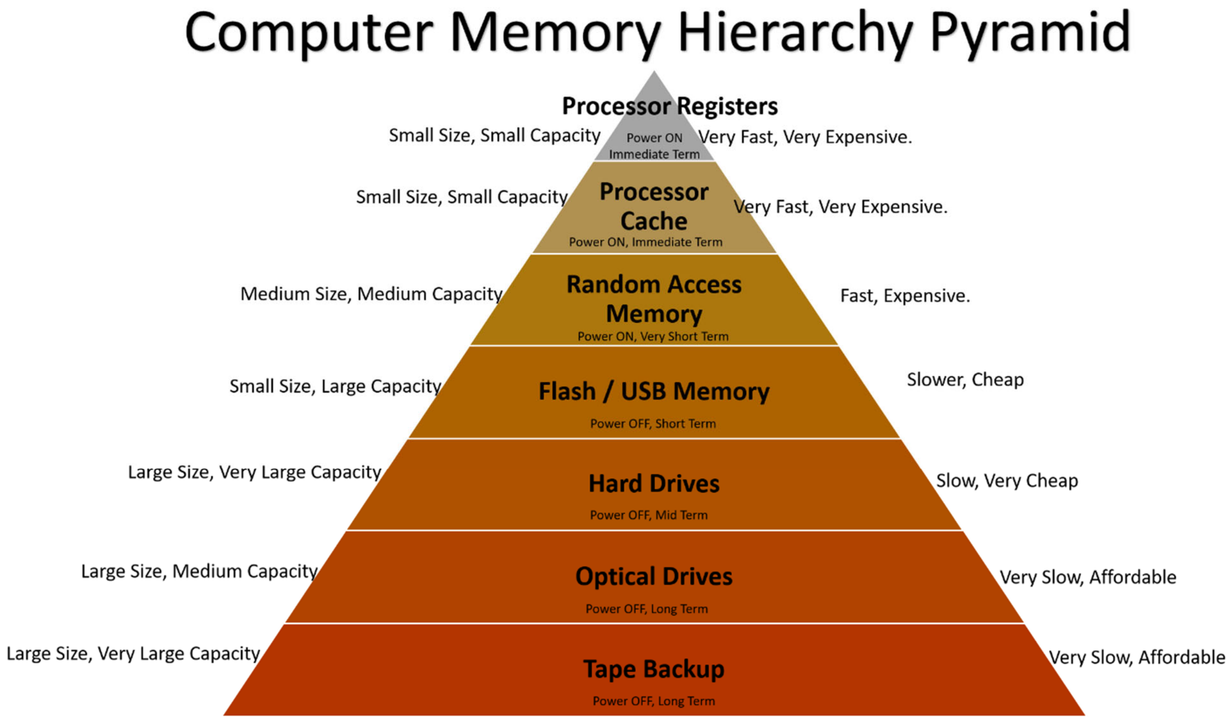
Memory is the electronic holding place for instructions and data that the computer's microprocessor can reach quickly. When the computer is in normal operation, its memory usually contains the main parts of the operating system and some or all of the application programs and related data that are being used. Memory is often used as a shorter synonym for random access memory (RAM).

Most desktop and notebook computers sold today include at least 4 Gigabytes of RAM, and are upgradeable to include more. The more RAM you have, the less frequently the computer has to access instructions and data from the more slowly accessed hard disk form of storage.

Memory is sometimes distinguished from storage, or the physical medium that holds the much larger amounts of data that won't fit into RAM and may not be immediately needed there.

Storage devices include hard disks, floppy disks, CD-ROM, and tape backup systems. The terms auxiliary storage, auxiliary memory, and secondary memory have also been used for this kind of data repository. Most computers have a memory hierarchy, with a small amount of very fast, expensive, volatile cache memory, some number of megabytes of medium-speed, medium-price, volatile main memory (RAM), and hundreds of thousands of megabytes of slow, cheap, non-volatile disk storage. It is the job of the operating system to coordinate how these memories are used.

The primary motive of a computer system is to execute programs. These programs, along with the information they access, should be in the main memory during execution.



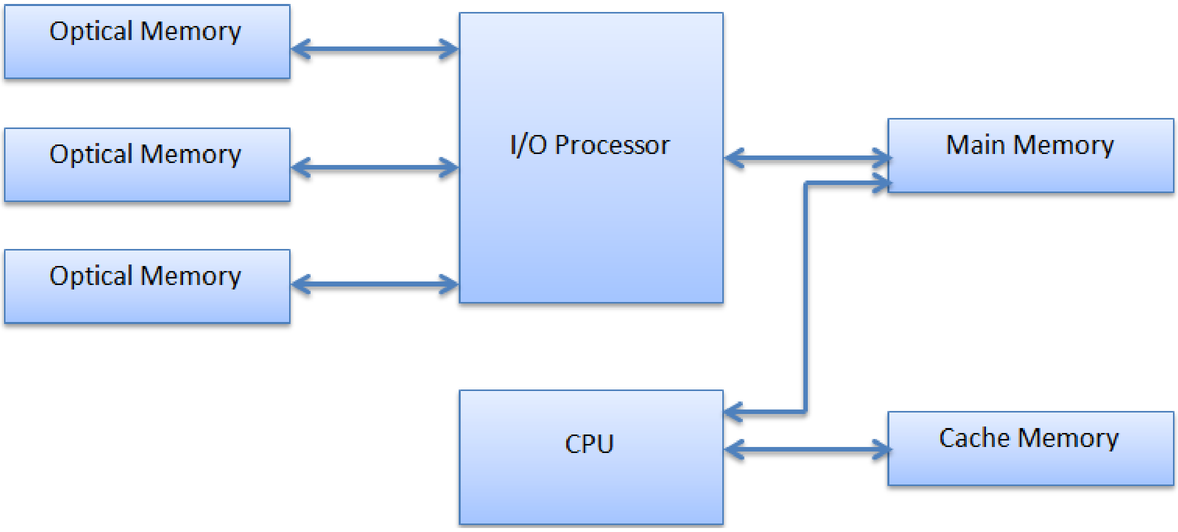


Figure: Memory hierarchy

Memory Management:

- Keep track of the status of memory locations, whether it is free or allocated.
- Allocation and Deallocation of memory before and after process execution.
- Permits computers with a small amount of main memory to execute programs larger than the size or amount of available memory. It does this by moving information back and forth between primary memory and secondary memory by using the concept of swapping.
- To minimize fragmentation issues.
- To proper utilization of main memory.
- Protection of memory for one process from other process.
- Memory managers should enable sharing of memory space between processes.

Logical Address space: An address generated by the CPU is known as “Logical Address”. It is also known as a Virtual address. Logical address space can be defined as the size of the process.

Physical Address space: An address seen by the memory unit (i.e., the one loaded into the memory address register of the memory) is commonly known as a “Physical Address”. A Physical address is also known as a Real address. The set of all physical addresses corresponding to these logical addresses is known as Physical address space.

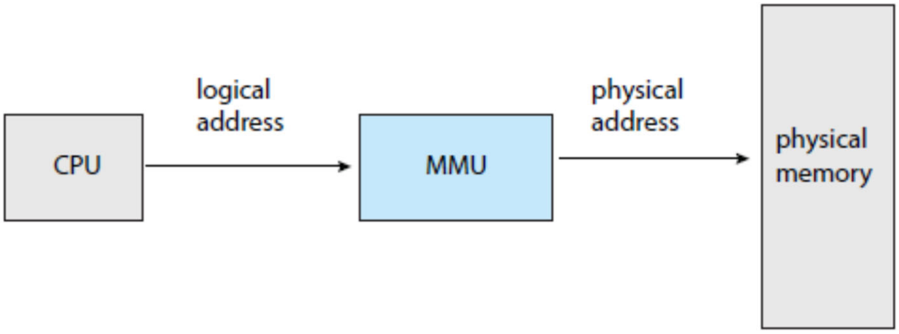


Figure: Memory management unit (MMU)

The run-time mapping from virtual to physical addresses is done by a hardware device called the **memory-management unit** (**MMU**). The base register is now called a **relocation register**. The value in the relocation register is added to every address generated by a user process at the time the address is sent to memory. For example, if the base is at 14000, then an attempt by the user to address location 0 is dynamically relocated to location 14000; an access to location 346 is mapped to location 14346.

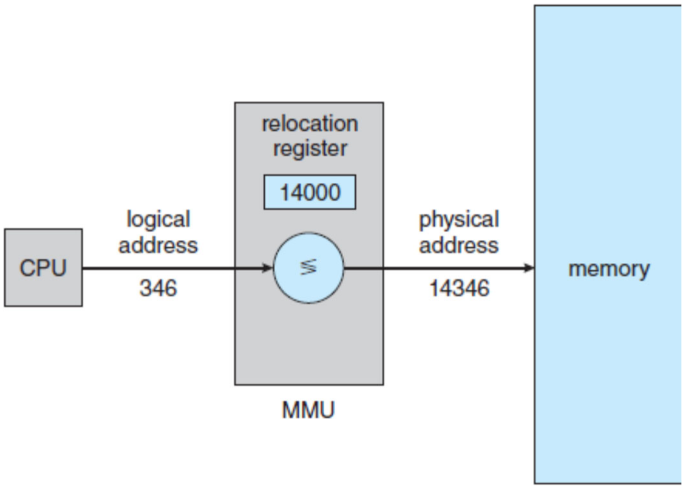


Figure: Dynamic relocation using a relocation register

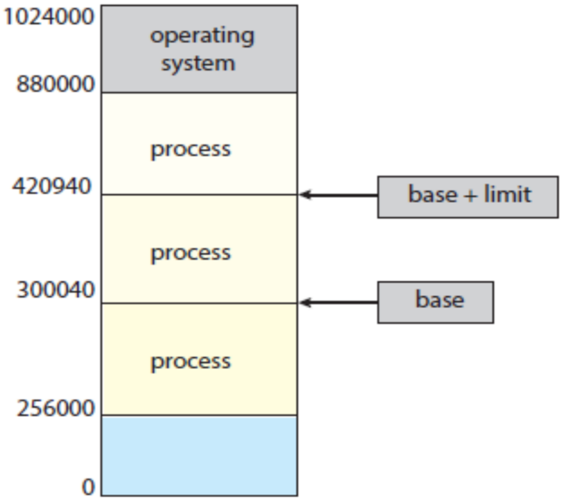


Figure: A base and a limit register define a logical address space

We now have two different types of addresses: logical addresses (in the range 0 to *max*) and physical addresses (in the range *R* + 0 to *R* + *max* for a base value *R*). The user program generates only logical addresses and thinks that the process runs in memory locations from 0 to *max.* However, these logical addresses must be mapped to physical addresses before they are used.

The set of all logical addresses generated by a program is referred to as a **logical address space**. The set of all physical addresses corresponding to these logical addresses is referred to as a **physical address space.**

The runtime mapping from virtual to physical address is done by the memory management unit (MMU) which is a hardware device. MMU uses following mechanism to convert virtual address to physical address.

- The value in the base register is added to every address generated by a user process, which is treated as offset at the time it is sent to memory. For example, if the base register value is 10000, then an attempt by the user to use address location 100 will be dynamically reallocated to location 10100.
- The user program deals with virtual addresses; it never sees the real physical addresses.

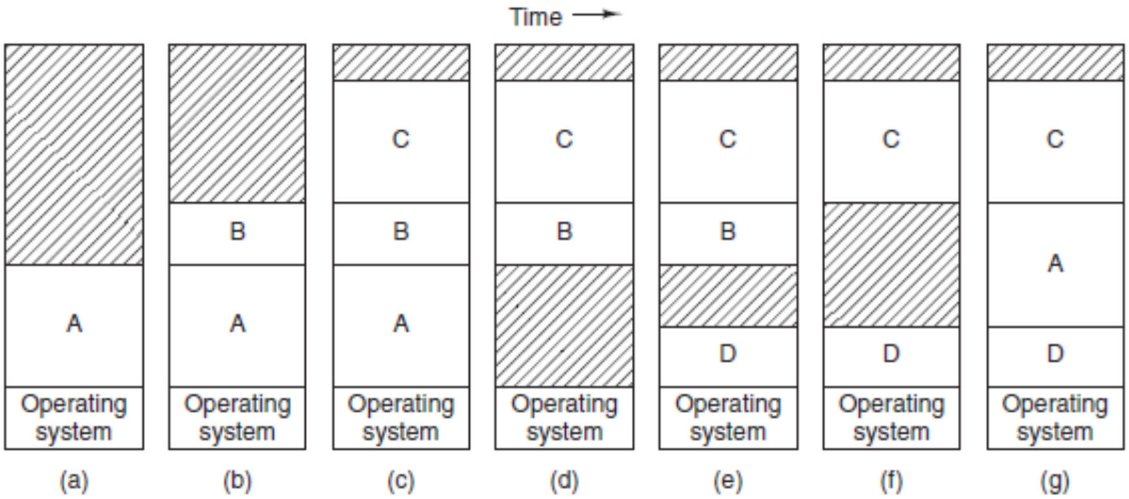


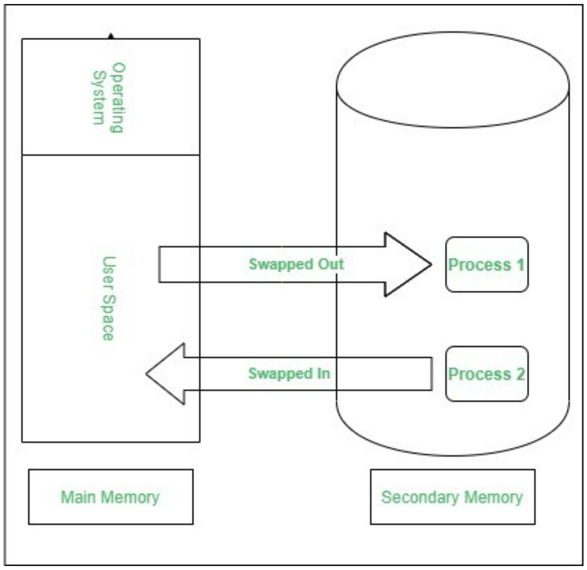
Figure: Memory allocation changes as processes come and leaves

Swapping

Swapping is a mechanism in which a process can be swapped temporarily out of main memory (or move) to secondary storage (disk) and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory.

The operation of a swapping system is illustrated in Fig. 3-4. Initially, only process *A* is in memory. Then processes *B* and *C* are created or swapped in from disk. In Fig. 3-4(d) *A* is swapped out to disk. Then *D* comes in and *B* goes out. Finally, *A* comes in again. Since *A* is now at a different location, addresses contained in it must be relocated, either by software when it is swapped in or (more likely) by hardware during program execution.

When swapping creates multiple holes in memory, it is possible to combine them all into one big one by moving all the processes downward as far as possible. This technique is known as **memory compaction**.


Operating System
User Space
Swapped Out
Swapped In
Process 1
Process 2
Main Memory
Secondary Memory