Unit 2: Memory system

Memory Hierarchy Design and its Characteristics

In the Computer System Design, Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time. The Memory Hierarchy was developed based on a program behavior known as locality of references.

Memory Hierarchy is Required in the System

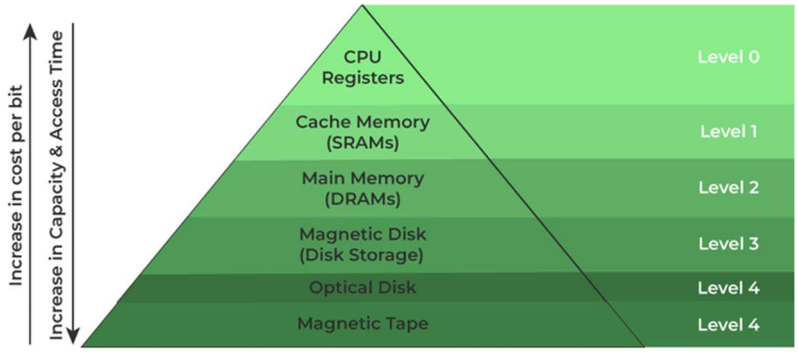
Memory Hierarchy is one of the most required things in Computer Memory as it helps in optimizing the memory available in the computer. There are multiple levels present in the memory, each one having a different size, different cost, etc. Some types of memory like cache, and main memory are faster as compared to other types of memory but they are having a little less size and are also costly whereas some memory has a little higher storage value, but they are a little slower. Accessing of data is not similar in all types of memory, some have faster access whereas some have slower access.

Types of Memory Hierarchy

This Memory Hierarchy Design is divided into 2 main types:

1. **External Memory or Secondary Memory:** Comprising of Magnetic Disk, Optical Disk, and Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via an I/O Module.

2. **Internal Memory or Primary Memory:** Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.



Memory Hierarchy Design

1. Registers:- Registers are small, high-speed memory units located in the CPU. They are used to store the most frequently used data and instructions. Registers have the fastest access time and the smallest storage capacity, typically ranging from 16 to 64 bits.

2. Cache Memory:- Cache memory is a small, fast memory unit located close to the CPU. It stores frequently used data and instructions that have been recently accessed from the main memory. Cache memory is designed to minimize the time it takes to access data by providing the CPU with quick access to frequently used data.

3. Main Memory:- Main memory, also known as RAM (Random Access Memory), is the primary memory of a computer system. It has a larger storage capacity than cache memory, but it is slower. Main memory is used to store data and instructions that are currently in use by the CPU.

Types of Main Memory

Static RAM: Static RAM stores the binary information in flip flops and information remains valid until power is supplied. It has a faster access time and is used in implementing cache memory.

Dynamic RAM: It stores the binary information as a charge on the capacitor. It requires refreshing circuitry to maintain the charge on the capacitors after a few milliseconds. It contains more memory cells per unit area as compared to SRAM.

Secondary Storage: Secondary storage, such as hard disk drives (HDD) and solid-state drives (SSD), is a non-volatile memory unit that has a larger storage capacity than main memory. It is used to store data and instructions that are not currently in use by the CPU. Secondary storage has the slowest access time and is typically the least expensive type of memory in the memory hierarchy.

Magnetic Disk: Magnetic Disks are simply circular plates that are fabricated with either a metal or a plastic or a magnetized material. The Magnetic disks work at a high speed inside the computer and these are frequently used.

Magnetic Tape: Magnetic Tape is simply a magnetic recording device that is covered with a plastic film. It is generally used for the backup of data. In the case of a magnetic tape, the access time for a computer is a little slower and therefore, it requires some amount of time for accessing the strip.

Characteristics of Memory Hierarchy

Capacity: It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.

Access Time: It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increases.

Performance: Earlier when the computer system was designed without a Memory Hierarchy design, the speed gap increased between the CPU registers and Main Memory due to a large difference in access time. This results in lower performance of the system and thus, enhancement was required. This enhancement was made in the form of Memory Hierarchy Design because of which the performance of the system increases. One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.

Cost Per Bit: As we move from bottom to top in the Hierarchy, the cost per bit increases i.e. Internal Memory is costlier than External Memory.

Advantages of Memory Hierarchy

- It helps in removing some destruction, and managing the memory in a better way.
- It helps in spreading the data all over the computer system.
- It saves the consumer's price and time.

Cache Memory in Computer Organization

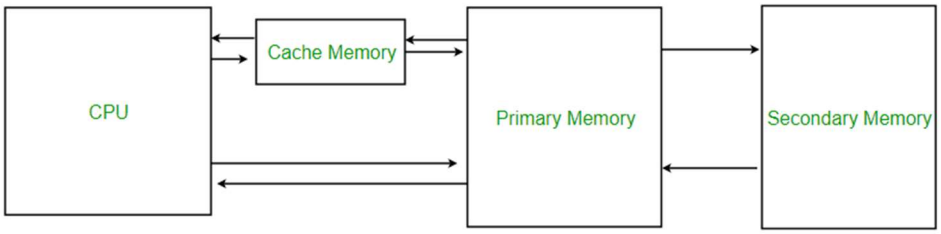
Cache memory is a small, high-speed storage area in a computer. The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations. There are various independent caches in a CPU, which store instructions and data. The most important use of cache memory is that it is used to reduce the average time to access data from the main memory.

By storing this information closer to the CPU, cache memory helps speed up the overall processing time. Cache memory is much faster than the main memory (RAM). When the CPU needs data, it first checks the cache. If the data is there, the CPU can access it quickly. If not, it must fetch the data from the slower main memory.

Characteristics of Cache Memory

- Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU.
- Cache Memory holds frequently requested data and instructions so that they are immediately available to the CPU when needed.
- Cache memory is costlier than main memory or disk memory but more economical than CPU registers.
- Cache Memory is used to speed up and synchronize with a high-speed CPU.

Understanding cache memory and its role in computer architecture is crucial for excelling in exams like GATE, where computer organization is a core topic. To deepen your understanding and enhance your exam preparation, consider enrolling in the GATE CS Self-Paced Course . This course offers in-depth coverage of computer architecture, including detailed explanations of cache memory and its optimization, helping you build the expertise needed to perform well in your exams.



Level 1 or Register: It is a type of memory in which data is stored and accepted that are immediately stored in the CPU. The most commonly used register is Accumulator, Program counter , Address Register, etc.

Level 2 or Cache memory: It is the fastest memory that has faster access time where data is temporarily stored for faster access.

Level 3 or Main Memory: It is the memory on which the computer works currently. It is small in size and once power is off data no longer stays in this memory.

Level 4 or Secondary Memory: It is external memory that is not as fast as the main memory but data stays permanently in this memory.

Cache Performance

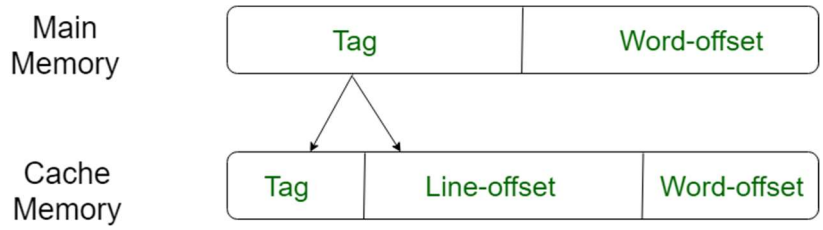
When the processor needs to read or write a location in the main memory, it first checks for a corresponding entry in the cache.

If the processor finds that the memory location is in the cache, a Cache Hit has occurred and data is read from the cache.
If the processor does not find the memory location in the cache, a cache miss has occurred. For a cache miss, the cache allocates a new entry and copies in data from the main memory, then the request is fulfilled from the contents of the cache.
The performance of cache memory is frequently measured in terms of a quantity called Hit ratio.
Hit Ratio(H) = hit / (hit + miss) = no. of hits/total accesses
Miss Ratio = miss / (hit + miss) = no. of miss/total accesses = 1 - hit ratio(H)

We can improve Cache performance using higher cache block size, and higher associativity, reduce miss rate, reduce miss penalty, and reduce the time to hit in the cache.

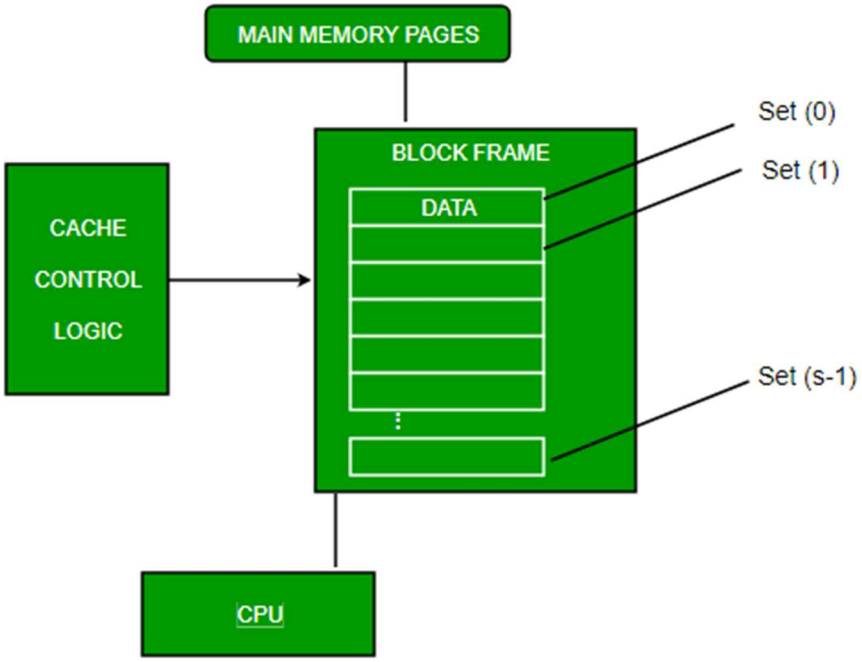
Cache Mapping
There are three different types of mapping used for the purpose of cache memory which is as follows:
Direct Mapping
Associative Mapping
Set-Associative Mapping

1. Direct Mapping
The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. or In Direct mapping, assign each memory block to a specific line in the cache. If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed. An address space is split into two parts index field and a tag field. The cache is used to store the tag field whereas the rest is stored in the main memory. Direct mapping`s performance is directly proportional to the Hit ratio.
i = j modulo m
Where i = cache line number
j = main memory block number
m = number of lines in the cache



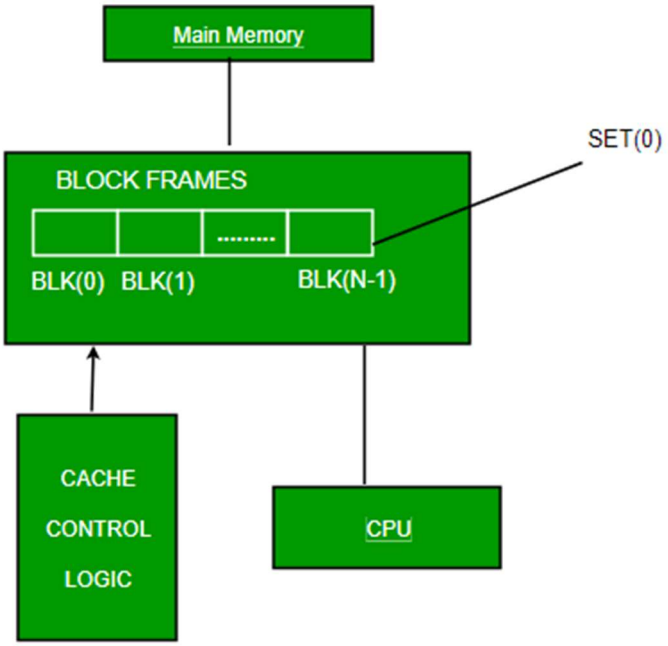
1. Direct Mapping

For purposes of cache access, each main memory address can be viewed as consisting of three fields. The least significant w bits identify a unique word or byte within a block of main memory. In most contemporary machines, the address is at the byte level. The remaining s bits specify one of the 2 s blocks of main memory. The cache logic interprets these s bits as a tag of s-r bits (the most significant portion) and a line field of r bits. This latter field identifies one of the m=2 r lines of the cache. Line offset is index bits in the direct mapping.



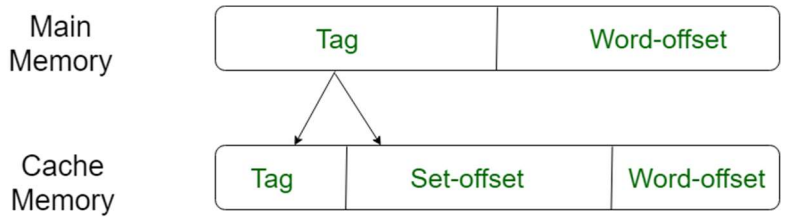
2. Associative Mapping

In this type of mapping, associative memory is used to store the content and addresses of the memory word. Any block can go into any line of the cache. This means that the word id bits are used to identify which word in the block is needed, but the tag becomes all of the remaining bits. This enables the placement of any word at any place in the cache memory. It is considered to be the fastest and most flexible mapping form. In associative mapping, the index bits are zero.



3. Set-Associative Mapping

This form of mapping is an enhanced form of direct mapping where the drawbacks of direct mapping are removed. Set associative addresses the problem of possible thrashing in the direct mapping method. It does this by saying that instead of having exactly one line that a block can map to in the cache, we will group a few lines together creating a set . Then a block in memory can map to any one of the lines of a specific set. Set-associative mapping allows each word that is present in the cache can have two or more words in the main memory for the same index address. Set associative cache mapping combines the best of direct and associative cache mapping techniques. In set associative mapping the index bits are given by the set offset bits. In this case, the cache consists of a number of sets, each of which consists of a number of lines.



Relationships in the Set-Associative Mapping can be defined as:

m = v * k

i= j mod v

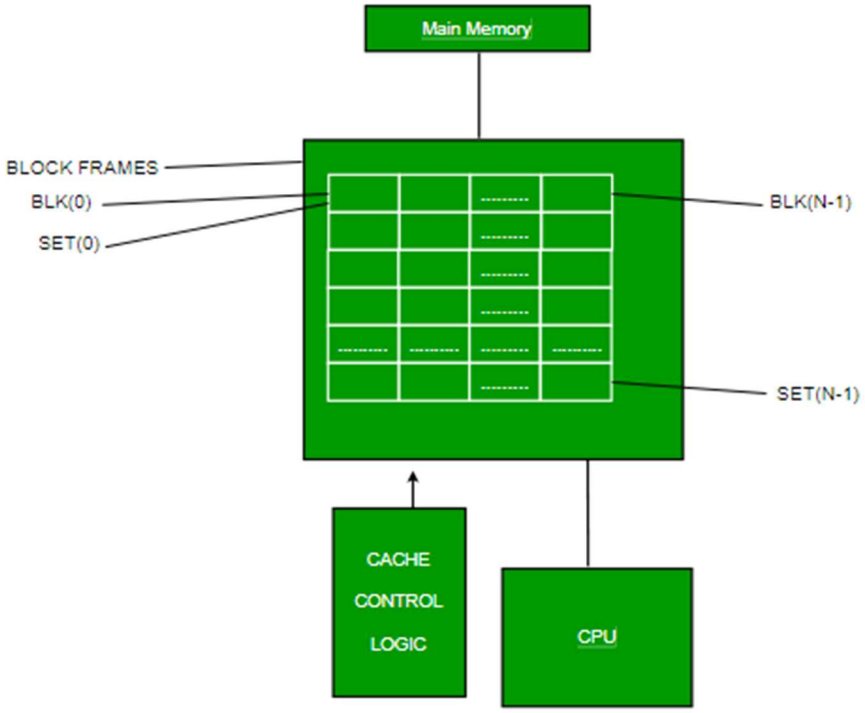
where, i = cache set number

j = main memory block number

v = number of sets

m = number of lines in the cache number of sets

k = number of lines in each set



Application of Cache Memory

Here are some of the applications of Cache Memory.

Primary Cache: A primary cache is always located on the processor chip. This cache is small and its access time is comparable to that of processor registers.

Secondary Cache: Secondary cache is placed between the primary cache and the rest of the memory. It is referred to as the level 2 (L2) cache. Often, the Level 2 cache is also housed on the processor chip.

Spatial Locality of Reference: Spatial Locality of Reference says that there is a chance that the element will be present in close proximity to the reference point and next time if again searched then more close proximity to the point of reference.

Temporal Locality of Reference: Temporal Locality of Reference uses the Least recently used algorithm will be used. Whenever there is page fault occurs within a word will not only load the word in the main memory but the complete page fault will be loaded because the spatial locality of reference rule says that if you are referring to any word next word will be referred to in its register that's why we load complete page table so the complete block will be loaded.

Advantages

1. Cache Memory is faster in comparison to main memory and secondary memory.
2. Programs stored by Cache Memory can be executed in less time.
3. The data access time of Cache Memory is less than that of the main memory.
4. Cache Memory stored data and instructions that are regularly used by the CPU, therefore it increases the performance of the CPU.

Disadvantages

1. Cache Memory is costlier than primary memory and secondary memory .
2. Data is stored on a temporary basis in Cache Memory.
3. Whenever the system is turned off, data and instructions stored in cache memory get destroyed.
4. The high cost of cache memory increases the price of the Computer System.

Virtual Memory in Operating System

Virtual Memory is a storage allocation scheme in which secondary memory can be addressed as though it were part of the main memory. The addresses a program may use to reference memory are distinguished from the addresses the memory system uses to identify physical storage sites and program-generated addresses are translated automatically to the corresponding machine addresses.

Virtual Memory

Virtual memory is a memory management technique used by operating systems to give the appearance of a large, continuous block of memory to applications, even if the physical memory (RAM) is limited. It allows the system to compensate for physical memory shortages, enabling larger applications to run on systems with less RAM.

A memory hierarchy, consisting of a computer system's memory and a disk, enables a process to operate with only some portions of its address space in memory. A virtual memory is what its name indicates- it is an illusion of a memory that is larger than the real memory. We refer to the software component of virtual memory as a virtual memory manager. The basis of virtual memory is the noncontiguous memory allocation model. The virtual memory manager removes some components from memory to make room for other components.

The size of virtual storage is limited by the addressing scheme of the computer system and the amount of secondary memory available not by the actual number of main storage locations.

Working of Virtual Memory

It is a technique that is implemented using both hardware and software. It maps memory addresses used by a program, called virtual addresses, into physical addresses in computer memory.

All memory references within a process are logical addresses that are dynamically translated into physical addresses at run time. This means that a process can be swapped in and out of the main memory such that it occupies different places in the main memory at different times during the course of execution.

A process may be broken into a number of pieces and these pieces need not be continuously located in the main memory during execution. The combination of dynamic run-time address translation and the use of a page or segment table permits this.

If these characteristics are present then, it is not necessary that all the pages or segments are present in the main memory during execution. This means that the required pages need to be loaded into memory whenever required. Virtual memory is implemented using Demand Paging or Demand Segmentation.

Types of Virtual Memory

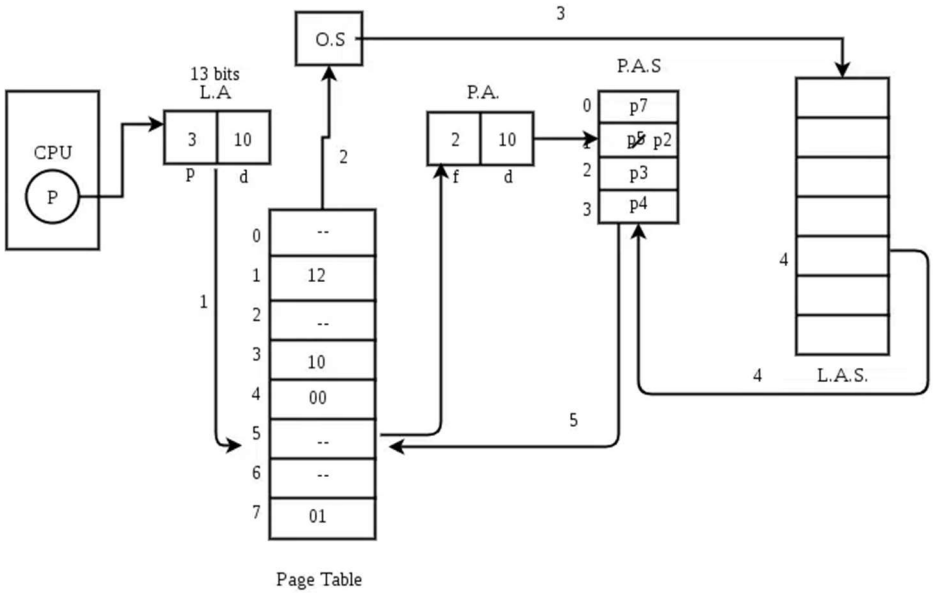
In a computer, virtual memory is managed by the Memory Management Unit (MMU), which is often built into the CPU. The CPU generates virtual addresses that the MMU translates into physical addresses.

There are two main types of virtual memory:

- Paging
- Segmentation

1. **Paging:-** Paging divides memory into small fixed-size blocks called pages. When the computer runs out of RAM, pages that aren't currently in use are moved to the hard drive, into an area called a swap file. The swap file acts as an extension of RAM. When a page is needed again, it is swapped back into RAM, a process known as page swapping. This ensures that the operating system (OS) and applications have enough memory to run.

Demand Paging: The process of loading the page into memory on demand (whenever a page fault occurs) is known as demand paging. The process includes the following steps are as follows:



- If the CPU tries to refer to a page that is currently not available in the main memory, it generates an interrupt indicating a memory access fault.
- The OS puts the interrupted process in a blocking state. For the execution to proceed the OS must bring the required page into the memory.
- The OS will search for the required page in the logical address space.
- The required page will be brought from logical address space to physical address space. The page replacement algorithms are used for the decision-making of replacing the page in physical address space.
- The page table will be updated accordingly.
- The signal will be sent to the CPU to continue the program execution and it will place the process back into the ready state.
- Hence whenever a page fault occurs these steps are followed by the operating system and the required page is brought into memory.

Page Fault Service Time

The time taken to service the page fault is called page fault service time. The page fault service time includes the time taken to perform all the above six steps.

Let Main memory access time is: m

Page fault service time is: s

Page fault rate is : p

Then, Effective memory access time = (p*s) + (1-p)*m

Segmentation

Segmentation divides virtual memory into segments of different sizes. Segments that aren't currently needed can be moved to the hard drive. The system uses a segment table to keep track of each segment's status, including whether it's in memory, if it's been modified, and its physical address. Segments are mapped into a process's address space only when needed.

Combining Paging and Segmentation:- Sometimes, both paging and segmentation are used together. In this case, memory is divided into pages, and segments are made up of multiple pages. The virtual address includes both a segment number and a page number.

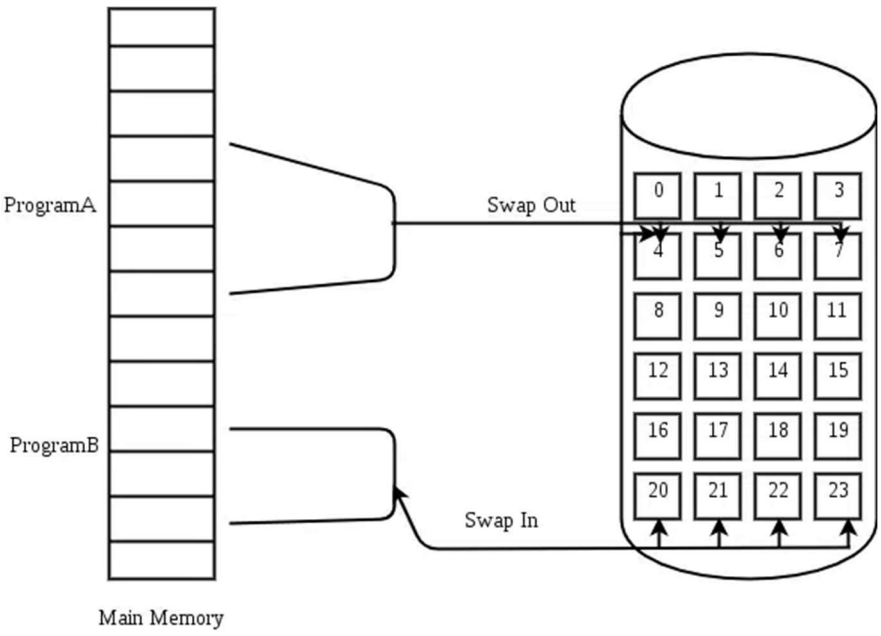
Virtual Memory vs Physical Memory

When talking about the differences between virtual memory and physical memory, the biggest distinction is speed. RAM is much faster than virtual memory, but it is also more expensive. When a computer needs storage for running programs, it uses RAM first. Virtual memory, which is slower, is used only when the RAM is full.

Feature	Virtual Memory	Physical Memory (RAM)
Definition	An abstraction that extends the available memory by using disk storage	The actual hardware (RAM) that stores data and instructions currently being used by the CPU
Location	On the hard drive or SSD	On the computer's motherboard
Speed	Slower (due to disk I/O operations)	Faster (accessed directly by the CPU)
Capacity	Larger, limited by disk space	Smaller, limited by the amount of RAM installed

Feature	Virtual Memory	Physical Memory (RAM)
Cost	Lower (cost of additional disk storage)	Higher (cost of RAM modules)
Data Access	Indirect (via paging and swapping)	Direct (CPU can access data directly)
Volatility	Non-volatile (data persists on disk)	Volatile (data is lost when power is off)

Swapping is a process out means removing all of its pages from memory, or marking them so that they will be removed by the normal page replacement process. Suspending a process ensures that it is not runnable while it is swapped out. At some later time, the system swaps back the process from the secondary storage to the main memory. When a process is busy swapping pages in and out then this situation is called thrashing.



Performance in Virtual Memory

Let p be the page fault rate(0 <= p <= 1).

if p = 0 no page faults

if p =1, every reference is a fault.

Effective access time (EAT) = (1-p)* Memory Access Time + p * Page fault time.

Page fault time = page fault overhead + swap out + swap in +restart overhead

The performance of a virtual memory management system depends on the total number of page faults, which depend on "paging policies" and "frame allocation"

Frame Allocation:- A number of frames allocated to each process in either static or dynamic.

- **Static Allocation:** The number of frame allocations to a process is fixed.
- **Dynamic Allocation:** The number of frames allocated to a process changes.

Paging Policies

- **Fetch Policy:** It decides when a page should be loaded into memory.
- **Replacement Policy:** It decides which page in memory should be replaced.
- **Placement Policy:** It decides where in memory should a page be loaded.

Applications of Virtual memory

Virtual memory has the following important characteristics that increase the capabilities of the computer system. The following are five significant characteristics of Lean.

Increased Effective Memory: One major practical application of virtual memory is, virtual memory enables a computer to have more memory than the physical memory using the disk space. This allows for the running of larger applications and numerous programs at one time while not necessarily needing an equivalent amount of DRAM.

Memory Isolation: Virtual memory allocates a unique address space to each process and that also plays a role in process segmentation. Such separation increases safety and reliability based on the fact that one process cannot interact with and or modify another's memory space through a mistake, or even a deliberate act of vandalism.

Efficient Memory Management: Virtual memory also helps in better utilization of the physical memories through methods that include paging and segmentation. It can transfer some of the memory pages that are not frequently used to disk allowing RAM to be used by active processes when required in a way that assists in efficient use of memory as well as system performance.

Simplified Program Development: For case of programmers, they don't have to consider physical memory available in a system in case of having virtual memory. They can program 'as if' there is one big block of memory and this makes the programming easier and more efficient in delivering more complex applications.

To Manage Virtual Memory

Here are 5 key points on how to manage virtual memory:

1. Adjust the Page File Size:- Automatic Management: All contemporary operating systems including Windows contain the auto-configuration option for the size of the empirical page file. But depending on the size of the RAM, they are set automatically, although the user can manually adjust the page file size if required.

Manual Configuration: For tuned up users, the setting of the custom size can sometimes boost up the performance of the system. The initial size is usually advised to be set to the minimum value of 1. To set the size of the swap space equal to 5 times the amount of physical RAM and the maximum size 3 times the physical RAM.

2. Place the Page File on a Fast Drive:- SSD Placement: If this is feasible, the page file should be stored in the SSD instead of the HDD as a storage device. It has better read and write times, and the virtual memory may prove beneficial in an SSD.

Separate Drive: Regarding systems having multiple drives involved, the page file needs to be placed on a different drive than the os and that shall in turn improve its performance.

3. Monitor and Optimize Usage:- Performance Monitoring: Employ the software tools used in monitoring the performance of the system in tracking the amounts of virtual memory. High page file usage may signify that there is a lack of physical RAM or that virtual memory needs a change of settings or addition in physical RAM.

Regular Maintenance: Make sure there is no toolbar or other application running in the background, take time and uninstall all the tool bars to free virtual memory.

4. Disable Virtual Memory for SSDs (with Sufficient RAM):- Sufficient RAM: If for instance your system has a big physical memory, for example 16GB and above then it would be advised to freeze the page file in order to minimize SSD usage. But it should be done, in my opinion, carefully and only if the additional signals that one decides to feed into his applications should not likely use all the available RAM.

5. Optimize System Settings:- System Configuration: Change some general properties of the system concerning virtual memory efficiency. This also involves enabling additional control options in Windows such as adjusting additional system setting option on the operating system, or using other options in different operating systems such as Linux that provides different tools and commands to help in adjusting how virtual memory is utilized.
Regular Updates: Ensure that your drivers are run in their newest version because new releases contain some enhancements and issues regarding memory management.

Advantages of Virtual Memory

2. More processes may be maintained in the main memory: Because we are going to load only some of the pages of any particular process, there is room for more processes. This leads to more efficient utilization of the processor because it is more likely that at least one of the more numerous processes will be in the ready state at any particular time.
3. A process may be larger than all of the main memory: One of the most fundamental restrictions in programming is lifted. A process larger than the main memory can be executed because of demand paging. The OS itself loads pages of a process in the main memory as required.
4. It allows greater multiprogramming levels by using less of the available (primary) memory for each process.
5. It has twice the capacity for addresses as main memory.
6. It makes it possible to run more applications at once.
7. Users are spared from having to add memory modules when RAM space runs out, and applications are liberated from shared memory management.
8. When only a portion of a program is required for execution, speed has increased.
9. Memory isolation has increased security.
10. It makes it possible for several larger applications to run at once.
11. Memory allocation is comparatively cheap.
12. It doesn't require outside fragmentation.
13. It is efficient to manage logical partition workloads using the CPU.
14. Automatic data movement is possible.

Disadvantages of Virtual Memory

1. It can slow down the system performance, as data needs to be constantly transferred between the physical memory and the hard disk.

2. It can increase the risk of data loss or corruption, as data can be lost if the hard disk fails or if there is a power outage while data is being transferred to or from the hard disk.
3. It can increase the complexity of the memory management system, as the operating system needs to manage both physical and virtual memory.

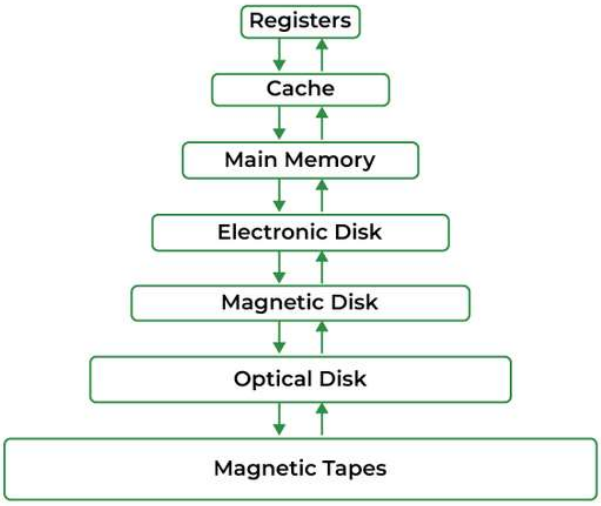
Memory Management

The term memory can be defined as a collection of data in a specific format. It is used to store instructions and process data. The memory comprises a large array or group of words or bytes, each with its own location. The primary purpose of a computer system is to execute programs. These programs, along with the information they access, should be in the main memory during execution. The CPU fetches instructions from memory according to the value of the program counter.

To achieve a degree of multiprogramming and proper utilization of memory, memory management is important. Many memory management methods exist, reflecting various approaches, and the effectiveness of each algorithm depends on the situation.

Main Memory

The main memory is central to the operation of a Modern Computer. Main Memory is a large array of words or bytes, ranging in size from hundreds of thousands to billions. Main memory is a repository of rapidly available information shared by the CPU and I/O devices. Main memory is the place where programs and information are kept when the processor is effectively utilizing them. Main memory is associated with the processor, so moving instructions and information into and out of the processor is extremely fast. Main memory is also known as RAM (Random Access Memory). This memory is volatile. RAM loses its data when a power interruption occurs.



Memory Management:- In a multiprogramming computer, the Operating System resides in a part of memory, and the rest is used by multiple processes. The task of subdividing the memory among different processes is called Memory Management. Memory management is a method in the operating system to manage operations between main memory and disk during process execution. The main aim of memory management is to achieve efficient utilization of memory.

Requirements of Memory Management

- Allocate and de-allocate memory before and after process execution.
- To keep track of used memory space by processes.
- To minimize fragmentation issues.
- To proper utilization of main memory.
- To maintain data integrity while executing of process.

Logical and Physical Address Space:- Logical Address Space: An address generated by the CPU is known as a "Logical Address". It is also known as a Virtual address. Logical address space can be defined as the size of the process. A logical address can be changed.

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. A physical address is computed by MMU. The run-time mapping from virtual to physical addresses is

done by a hardware device Memory Management Unit(MMU). The physical address always remains constant.

Static and Dynamic Loading:- Loading a process into the main memory is done by a loader. There are two different types of loading:

1. **Static Loading:** Static Loading is basically loading the entire program into a fixed address. It requires more memory space.
2. **Dynamic Loading:** The entire program and all data of a process must be in physical memory for the process to execute. So, the size of a process is limited to the size of physical memory. To gain proper memory utilization, dynamic loading is used. In dynamic loading, a routine is not loaded until it is called. All routines are residing on disk in a relocatable load format. One of the advantages of dynamic loading is that the unused routine is never loaded. This loading is useful when a large amount of code is needed to handle it efficiently.

Static and Dynamic Linking:- To perform a linking task a linker is used. A linker is a program that takes one or more object files generated by a compiler and combines them into a single executable file.

Static Linking: In static linking, the linker combines all necessary program modules into a single executable program. So there is no runtime dependency. Some operating systems support only static linking, in which system language libraries are treated like any other object module.

Dynamic Linking: The basic concept of dynamic linking is similar to dynamic loading. In dynamic linking, "Stub" is included for each appropriate library routine reference. A stub is a small piece of code. When the stub is executed, it checks whether the needed routine is already in memory or not. If not available then the program loads the routine into memory.

Swapping:- When a process is executed it must have resided in memory. Swapping is a process of swapping a process temporarily into a secondary memory from the main memory, which is fast compared to secondary memory. A swapping allows more processes to be run and can be fit into memory at one time. The main part of swapping is transferred time and the total time is directly proportional to the amount of memory swapped. Swapping is also known as roll-out, or roll because if a higher priority process arrives and wants service, the memory manager can swap out the lower priority process and then load and execute the higher priority

process. After finishing higher priority work, the lower priority process swapped back in memory and continued to the execution process.