* What is the main purpose of an operating system? Discuss different types?

<https://www.geeksforgeeks.org/introduction-of-operating-system-set-1/?ref=lbp>

<https://www.geeksforgeeks.org/types-of-operating-systems/>

An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently.

An operating system is a software that manages computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

* What is a socket, kernel and monolithic kernel ?

<https://www.geeksforgeeks.org/socket-in-computer-network/>

<https://www.geeksforgeeks.org/difference-between-microkernel-and-monolithic-kernel/>

<https://www.tutorialspoint.com/what-is-the-concept-of-monolithic-kernel>

**What is a kernel ?**  
The [kernel](https://www.geeksforgeeks.org/kernel-in-operating-system/) is a computer program at the core of a computer’s operating system and has complete control over everything in the system. It manages the operations of the computer and the hardware.

**There are five types of kernels :**

1. A micro kernel, which only contains basic functionality;
2. A monolithic kernel, which contains many device drivers.
3. Hybrid Kernel
4. Exokernel
5. Nanokernel

**1.**[**Microkernel**](https://www.geeksforgeeks.org/microkernel-in-operating-systems/)**:**  
kernel manages the operations of the computer, In microkernel the user services and kernel services are implemented in different address space. The user services are kept in user address space, and kernel services are kept under kernel address space.

**2.**[**Monolithic kernel**](https://www.geeksforgeeks.org/monolithic-kernel-and-key-differences-from-microkernel/)**:**  
In Monolithic kernel, the entire operating system runs as a single program in kernel mode. The user services and kernel services are implemented in same address space.

**Differences between Microkernel and Monolithic Kernel :**

|  |  |
| --- | --- |
| **Microkernel** | **Monolithic kernel** |
| In microkernel user services and kernel, services are kept in separate address space. | In monolithic kernel, both user services and kernel services are kept in the same address space. |
| OS is complex to design. | OS is easy to design and implement. |
| Microkernel are smaller in size. | Monolithic kernel is larger than microkernel. |
| Easier to add new functionalities. | Difficult to add new functionalities. |
| To design a microkernel, more code is required. | Less code when compared to microkernel |
| Failure of one component does not effect the working of micro kernel. | Failure of one component in monolithic kernel leads to failure of entire system. |
| Execution speed is low. | Execution speed is high. |
| It is easy to extend Microkernel. | It is not easy to extend monolithic kernel. |
| **Example :** Mac OS X. | **Example :** Microsoft Windows 95. |

---------------------------------------------------------------------

\* Difference between process and program and thread? Different types of process.

<https://www.geeksforgeeks.org/difference-between-process-and-thread/>

| S.NO | Process | Thread |
| --- | --- | --- |
| 1. | Process means any program is in execution. | Thread means a segment of a process. |
| 2. | The process takes more time to terminate. | The thread takes less time to terminate. |
| 3. | It takes more time for creation. | It takes less time for creation. |
| 4. | It also takes more time for context switching. | It takes less time for context switching. |
| 5. | The process is less efficient in terms of communication. | Thread is more efficient in terms of communication. |
| 6. | Multiprogramming holds the concepts of multi-process. | We don’t need multi programs in action for multiple threads because a single process consists of multiple threads. |
| 7. | The process is isolated. | Threads share memory. |
| 8. | The process is called the heavyweight process. | A Thread is lightweight as each thread in a process shares code, data, and resources. |
| 9. | Process switching uses an interface in an operating system. | Thread switching does not require calling an operating system and causes an interrupt to the kernel. |
| 10. | If one process is blocked then it will not affect the execution of other processes | If a user-level thread is blocked, then all other user-level threads are blocked. |
| 11. | The process has its own Process Control Block, Stack, and Address Space. | Thread has Parents’ PCB, its own Thread Control Block, and Stack and common Address space. |
| 12. | Changes to the parent process do not affect child processes. | Since all threads of the same process share address space and other resources so any changes to the main thread may affect the behavior of the other threads of the process. |
| 13. | A system call is involved in it. | No system call is involved, it is created using APIs. |
| 14. | The process does not share data with each other. | Threads share data with each other. |

Define virtual memory, thrashing, threads.

<https://www.geeksforgeeks.org/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

* What is RAID ? Different types.

<https://www.geeksforgeeks.org/raid-redundant-arrays-of-independent-disks/>

RAID, or “Redundant Arrays of Independent Disks” is a technique which makes use of a combination of multiple disks instead of using a single disk for increased performance, data redundancy or both

What is a deadlock? Different conditions to achieve a deadlock.

<https://www.geeksforgeeks.org/introduction-of-deadlock-in-operating-system/>

<https://www.geeksforgeeks.org/introduction-of-deadlock-in-operating-system/>

What is fragmentation? Types of fragmentation.

<https://www.javatpoint.com/fragmentation-in-operating-system>

What is spooling ?

<https://www.javatpoint.com/spooling-in-operating-system>

What is semaphore and mutex (Differences might be asked)? Define Binary semaphore.

<https://www.geeksforgeeks.org/mutex-vs-semaphore/>

<https://www.geeksforgeeks.org/semaphores-in-process-synchronization/>

<https://www.geeksforgeeks.org/difference-between-counting-and-binary-semaphores/>

**Using Mutex:**

A mutex provides mutual exclusion, either producer or consumer can have the key (mutex) and proceed with their work. As long as the buffer is filled by the producer, the consumer needs to wait, and vice versa.

At any point of time, only one thread can work with the *entire* buffer. The concept can be generalized using semaphore.

**Using Semaphore:**

A semaphore is a generalized mutex. In lieu of a single buffer, we can split the 4 KB buffer into four 1 KB buffers (identical resources). A semaphore can be associated with these four buffers. The consumer and producer can work on different buffers at the same time.

Belady's Anomaly -> **Solution to fix Belady’s Anomaly:**

<https://www.geeksforgeeks.org/beladys-anomaly-in-page-replacement-algorithms/>

Starving and Aging in OS

<https://www.geeksforgeeks.org/starvation-and-aging-in-operating-systems/>

Differences between [Deadlock](https://www.geeksforgeeks.org/operating-system-process-management-deadlock-introduction/) and Starvation in OS are as follows:

1. Deadlock occurs when none of the processes in the set is able to move ahead due to occupancy of the required resources by some other process as shown in the figure below, on the other hand, Starvation occurs when a process waits for an indefinite period of time to get the resource it requires.
2. Another name for deadlock is **Circular Waiting**. Another name for starvation *is****Lived lock****.*
3. When deadlock occurs no process can make progress, while in starvation apart from the victim process other processes can progress or proceed.

Why does trashing occur?

<https://www.javatpoint.com/what-is-thrash>

What is paging and why do we need it?

<https://www.geeksforgeeks.org/paging-in-operating-system/>

Demand Paging, Segmentation

<https://www.geeksforgeeks.org/difference-between-demand-paging-and-segmentation/>

[Demand Paging](https://practice.geeksforgeeks.org/problems/demand-paging)**:**  
Demand paging is identical to the paging system with swapping. In demand paging, a page is delivered into the memory on demand i.e., only when a reference is made to a location on that page. Demand paging combines the feature of simple paging and implement virtual memory as it has a large virtual memory. Lazy swapper concept is implemented in demand paging in which a page is not swapped into the memory unless it is required.

[Segmentation](https://www.geeksforgeeks.org/segmentation-in-operating-system/)**:**  
Segmentation is the arrangement of memory management. According to the segmentation the logical address space is a collection of segments. Each segment has a name and length. Each logical address have two quantities segment name and the segment offset, for simplicity we use the segment number in place of segment name.

The difference between Demand Paging and Segmentation are as follows:

| S.No. | Demand Paging | Segmentation |
| --- | --- | --- |
| 1. | In demand paging, the pages are of equal size. | While in segmentation, segments can be of different size. |
| 2. | Page size is fixed in the demand paging. | Segment size may vary in segmentation as it grants dynamic increase of segments. |
| 3. | It does not allows sharing of the pages. | While segments can be shared in segmentation. |
| 4. | In demand paging, on demand pages are loaded in the memory. | In segmentation, during compilation segments are allocated to the program. |
| 5. | Page map table in demand paging manages record of pages in memory. | Segment map table in segmentation demonstrates every segment address in the memory. |
| 6. | It provides large virtual memory and have more efficient use of memory. | It provides virtual memory and maximum size of segment is defined by the size of memory. |

Real Time Operating System, types of RTOS.

<https://www.geeksforgeeks.org/real-time-operating-system-rtos/>

Real-time**operating systems (RTOS)** are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. such applications are industrial control, telephone switching equipment, flight control, and real-time simulations.

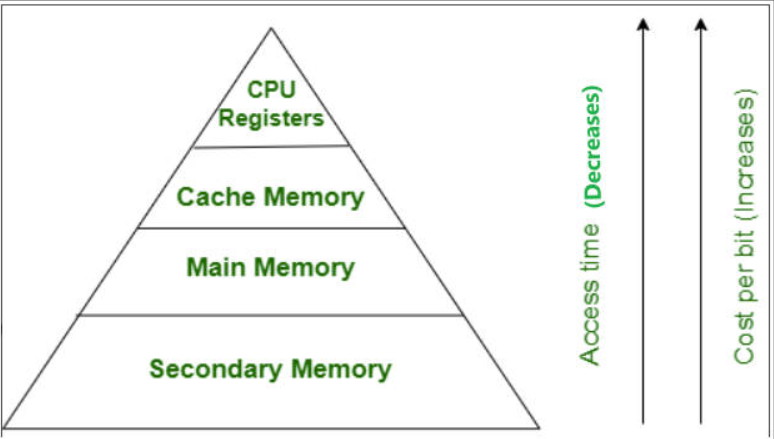
 With an RTOS, the processing time is measured in tenths of seconds. This system is time-bound and has a fixed deadline.

The processing in this type of system must occur within the specified constraints. Otherwise, This will lead to system failure.

Examples of the real-time operating systems: Airline traffic control systems, Command Control Systems, Airlines reservation system, Heart Pacemaker, Network Multimedia Systems, Robot etc

Difference between main memory and secondary memory.

<https://www.geeksforgeeks.org/difference-between-primary-and-secondary-memory/>



| ***Primary memory*** | ***Secondary memory*** |
| --- | --- |
| Primary memory is temporary. | Secondary memory is permanent. |
| Primary memory is directly accessible by Processor/CPU. | Secondary memory is not directly accessible by the CPU. |
| Nature of Parts of Primary memory varies, RAM- volatile in nature. ROM- Non-volatile. | It’s always Non-volatile in nature. |
| Primary memory devices are more expensive than secondary storage devices. | Secondary memory devices are less expensive when compared to primary memory devices. |
| The memory devices used for primary memory are semiconductor memories. | The secondary memory devices are magnetic and optical memories. |
| Primary memory is also known as Main memory or Internal memory. | Secondary memory is also known as External memory or Auxiliary memory. |
| Examples: RAM, ROM, Cache memory, PROM, EPROM, Registers, etc. | Examples: Hard Disk, Floppy Disk, Magnetic Tapes, etc. |

Static & Dynamic Binding

Explain Cache

<https://www.geeksforgeeks.org/cache-memory/>

A faster and smaller segment of memory whose access time is as close as registers are known as Cache memory.

In a [hierarchy](https://www.geeksforgeeks.org/memory-hierarchy-design-and-its-characteristics/) of memory, cache memory has access time lesser than primary memory. Generally, cache memory is very smaller and hence is used as a buffer.

Diff between direct mapping and associative mapping

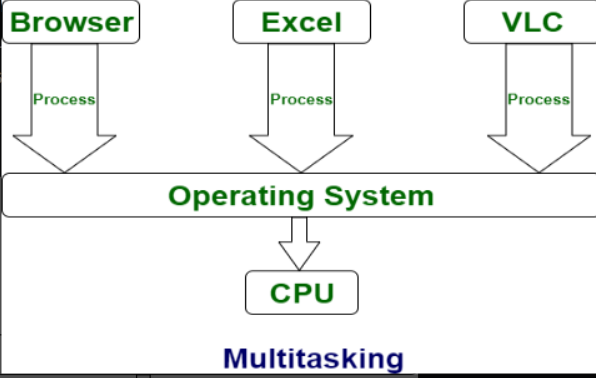
<https://www.geeksforgeeks.org/difference-between-direct-mapping-associative-mapping-set-associative-mapping/>

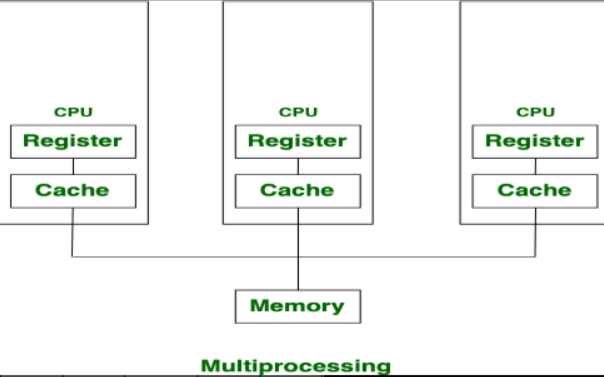
**Difference between Direct-mapping, Associative Mapping & Set-Associative Mapping :**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Direct-mapping** | **Associative Mapping** | **Set-Associative Mapping** |
| 1. | Needs only one comparison because of using direct formula to get the effective cache address. | Needs comparison with all tag bits, i.e., the cache control logic must examine every block’s tag for a match at the same time in order to determine that a block is in the cache/not. | Needs comparisons equal to number of blocks per set as the set can contain more than 1 blocks. |
| 2. | Main Memory Address is divided into 3 fields : TAG,**BLOCK** & WORD. The BLOCK & WORD together make an index. The least significant TAG bits identify a unique word within a block of main memory, the BLOCK bits specify one of the blocks and the Tag bits are the most significant bits. | Main Memory Address is divided into 1 fields : TAG & WORD. | Main Memory Address is divided into 3 fields : TAG, **SET** & WORD. |
| 3. | There is one possible location in the cache organization for each block from main memory because we have a fixed formula. | The mapping of the main memory block can be done with any of the cache block. | The mapping of the main memory block can be done with a particular cache block of any direct-mapped cache. |
| 4. | If the processor need to access same memory location from 2 different main memory pages frequently, cache hit ratio decreases. | If the processor need to access same memory location from 2 different main memory pages frequently, cache hit ratio has no effect. | In case of frequently accessing two different pages of the main memory if reduced, the cache hit ratio reduces. |
| 5. | Search time is less here because there is one possible location in the cache organization for each block from main memory. | Search time is more as the cache control logic examines every block’s tag for a match. | Search time increases with number of blocks per set. |

Diff between multitasking and multiprocessing

<https://www.geeksforgeeks.org/difference-between-multitasking-and-multiprocessing/>





|  |  |  |
| --- | --- | --- |
| S No. | Multi-tasking | Multiprocessing |
| 1. | The execution of more than one task simultaneously is known as multitasking. | The availability of more than one processor per system, that can execute several set of instructions in parallel is known as multiprocessing. |
| 2. | The number of CPU is one. | The number of CPUs is more than one. |
| 3. | It takes moderate amount of time. | It takes less time for job processing. |
| 4. | In this, one by one job is being executed at a time. | In this, more than one process can be executed at a time. |
| 5. | It is economical. | It is less economical. |
| 6. | The number of users is more than one. | The number of users is can be one or more than one. |
| 7. | Throughput is moderate. | Throughput is maximum. |
| 8. | Its efficiency is moderate. | Its efficiency is maximum. |
| 9. | It is of two types: Single user multitasking and Multiple user multitasking. | It is of two types: Symmetric Multiprocessing and Asymmetric Multiprocessing. |
| 10. | Number of user tasks is more than one. | Number of user tasks can be one or more than one. |