**Operating Systems Assignment**

**1 -> Rank the following storage systems from slowest to fastest: a. Hard-disk drives b. Registers c. Optical disk d. Main memory e. Nonvolatile memory f. Magnetic tapes g. Cache**

Registers :- It is the smallest and fastest storage as it sits inside the CPU besides the RAM. It has an access time of just a fraction of a nanosecond.

Cache Memory :- It is also a very fast storage system but slightly slower than the Registers. It is located between the CPU and main memory.

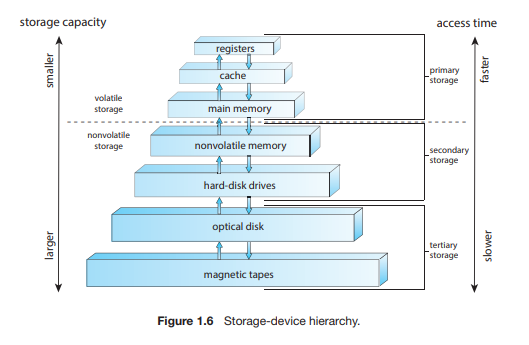
Main Memory :- It is the primary and the heart of the working storage. It is fast but volatile.

Non-Volatile Memory :- It includes flash memory and SSD. It is fast in terms of processing and much faster than HDDs.

Hard-Disks Drives :- It uses magnetic rotators with read and write head. It provides direct access to the memory but it has latency in processing because it takes time for the head to rotate and retrieve data.

Optical Disk :- It uses laser to read and write data on a smooth reflective surface. It is much slower than HDDs because of the spin-up and sleek delays.

Magnetic Tapes :- It has very high storage capacity but it is very slow in processing because it had sequential access only.



**2 -> Consider an SMP system similar to the one shown in Figure 1.8. Illustrate with an example how data residing in memory could in fact have a different value in each of the local caches.**

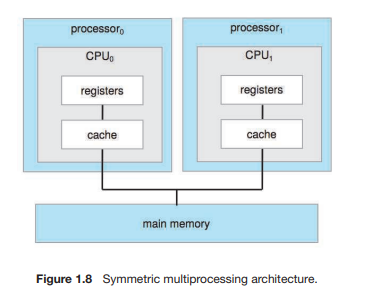
SMP refers to Symmetric Multiprocessing System. In this type of system, there are multiple processors which are connected to same memory. Each processor has its own stored cache. In this type of systems, the problem that arises is when one processor updates the cache data, other processor might still have old values in cache or memory. This situation is called to be cache coherence problem.

When system does not handle it well, processor end up working with different versions of same data. In order to tackle this situation, hardware uses cache coherence protocols that update the other caches or declare the outdated values invalid.

Eg: Suppose Memory location x=0

* Processor 1 and 2 have x in their caches
* Processor 1 changes value of x to 1, but does not update it in the memory
* In this case processor 1 has x=1, whereas 2 has x=0

This example shows that how data can have different values in different caches until cache coherence protocols fix it.



**3 -> Discuss, with examples, how the problem of maintaining coherence of cached data manifests itself in the following processing environments: a. Single-processor systems b. Multiprocessor systems c. Distributed systems**

Cache coherence problems refers to problems which arises when there are multiple available copies of the same data. These problems can happen in 3 types of systems which are:

1 – Single Processor Systems :- In single CPU system, cache coherence problems happen when hard disk or network card updates memory using direct memory assists (DMA). In this case, CPU is using old data in cache, while memory has been updated by device.

* Eg: Network card writes new data in memory but CPU using old cache data.

2 – Multi Processor Systems :- Here the same above problem is more common due to the presence of multiple processor having their own cache and still connected to one memory. In this type, when one processor updates variable, other may still be using old value in caches. This leads to incorrect code/program behaviour.

* Eg: CPU 1 has x=1 value with CPU 2 has x=2 in cache

3 – Distributed Systems :- In this type, multiple copies of data can be present on different machines which are connected to main system through maybe cables or network. When one machine updates the data, others may use outdated data and this type of problem is usually handled by server by validating cache data.

* Eg: Two people using different machines and storing file on server. In this case, if server does not notify the users about one another, there are chances that files may get mixed up and the users work on different versions of same file

