Assignment 1

Q-1: The issue of resource utilization shows up in different forms in different types of operating systems. List what resources must be managed carefully in the following settings:

a. Mainframe or minicomputer systems

b. Workstations connected to servers

c. Mobile computers

Ans:

a. Mainframes or minicomputer systems: memory and CPU resources, storage, network bandwidth.

b. Workstations connected to servers: memory and CPU resources

c. Mobile computers: power consumption, memory resources.

Q-2: Describe the differences between symmetric and asymmetric multiprocessing. What are three advantages and one disadvantage of multiprocessor systems?

Ans: Symmetric multiprocessing - In symmetric multiprocessing, each processor performs all tasks within the operating system. Symmetric multiprocessing means that all processors are peers; no boss–worker relationship exists between processors.

Asymmetric multiprocessing - In asymmetric multiprocessing, each processor is assigned a specific task. A boss processor controls the system.This scheme defines a boss–worker relationship. The boss processor schedules and allocates work to the worker processors.

Advantage of multiprocessor systems

1.Increased throughput - By increasing the number of processors, more work can be done in less time.

2.Cost saving - Multiprocessor systems can save money as compared to multiple single-processor systems because they can share peripherals, mass storage, and power supplies.

3.Increased reliability - The workload is distributed among several processors, so this system results in increased reliability. The failure of one processor will not stop the system, only slow it down.

Disadvantage of multiprocessor

If the CPU has an integrated memory controller, then adding CPUs can also increase the amount of memory addressable in the system. Either way, multiprocessing can cause a system to change its memory access model from uniform memory access (UMA) to non-uniform memory access (NUMA). UMA is defined as the situation in which access to any RAM from any CPU takes the same amount of time. With NUMA, some parts of memory may take longer to access than other parts, creating a performance penalty.

Q-3: How do clustered systems differ from multiprocessor systems? What is required for two machines belonging to a cluster to cooperate to provide a highly available service?

Ans: Clustered systems are typically constructed by combining multiple computers into a single system to perform a computational task distributed across the cluster. On the other hand, multiprocessor systems could be a single physical entity comprising of multiple CPUs. A clustered system is loosely coupled than a multiprocessor system.

Clustering is usually used to provide highly available service—that is, service will continue even if one system in the cluster fail. Generally, we obtain high availability by adding a level of redundancy in the system. A layer of cluster software runs on the cluster nodes. Each node can monitor one or more of the others (over the LAN). If the monitored machine fails, the monitoring machine can take ownership of its storage and restart the applications that were running on the failed machine.

Q-4: What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

Ans: The occurrence of an event is usually signaled by an interrupt from either the hardware or the software.When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location. The fixed location usually contains the starting address where the service routine for the interrupt is located.The interrupt service routine executes; on completion, the CPU resumes the interrupted computation.
A trap is a software generated interrupt. It can be caused by an error or by a request from a user program. A hardware generated interrupt is caused by hardware (such as I/O completion).
Yes, traps can be generated intentionally by a user program. User programs create traps for debugging purposes. A trap can be used to call the OS routines or to catch arithmetic errors.

Q-5: Consider an SMP system similar to the one shown in Figure 1.6. Illustrate with an example how data residing in memory could in fact have a different value in each of the local caches.

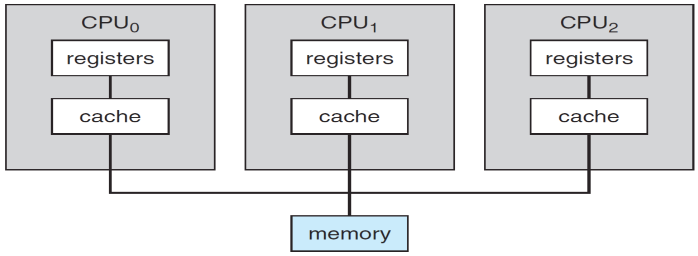

Figure 1.6 Symmetric multiprocessing architecture.

Ans: SMP means that all processors are peers; no master slave relationship exists between processors. A typical SNP architecture is that each processor has its own register, as well as a private or local cache; however, all processors share physical memory. The benefit of this model is that many processes can run simultaneously N processes can run if there are N CPUs- without causing a significant deterioration of performance. However, we must carefully control I/0 to ensure that the data reach the appropriate processor. Also, since the CPUs are separate, one may be sitting idle while another is overloaded, resulting in inefficiencies. For example, CPU-0 and CPU-1 read data A with value 10 from main memory into their local cache. Then CPU-0 updates A's value to 11. The Update does not occur in the local cache for CPU-1 because A resides in CPU-0's local cache, so the update only occurs in the local cache for CPU-0. In this way, there will be the different value in each of the local caches.

Q-6: Discuss, with examples, how the problem of maintaining the coherence of cached data manifests itself in the following processing environments:

a. Single-processor systems

b. Multiprocessor systems

c. Distributed systems

Ans: In a hierarchical storage structure, the same data may appear in different levels of the storage system. Let us consider the example: Suppose that an integer A that is to be incremented by 1 is located in file B, and file B resides on magnetic disk. The increment operation proceeds by first issuing an I/O operation to copy the disk block on which A resides to main memory. This operation is followed by copying A to the cache and to an internal register. Thus, the copy of A appears in several places: on the magnetic disk, in main memory, in the cache, and in an internal register in the following figure. Once the increment takes place in the internal register, the value of A differs in the various storage systems. The value of A becomes the same only after the new value of A is written from the internal register back to the magnetic disk.

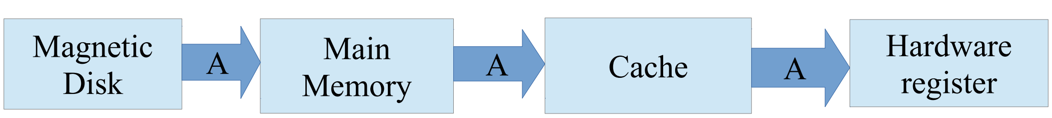

Figure: Migration of integer A from disk to register

Single processor systems: In a computing environment where only one process executes at a time, this arrangement poses no difficulties, since an access to integer A will always be to the copy at the highest level of the hierarchy. However, in a multitasking environment, where the CPU is switched back and forth among various processes, extreme care must be taken to ensure that, if several processes wish to access A, then each of these processes will obtain the most recently updated value of A.

Multiprocessor systems: It is more complicated in a multiprocessor environment since each of the CPUs also contains a local cache. In such an environment, a copy of A may exist simultaneously in several caches. Since the various CPUs can all execute in parallel, we must make sure that an update to the value of A in one cache is immediately reflected in all other caches where A resides. This situation is called cache coherency, and it is usually a hardware problem.

Distributed systems: In a distributed environment, the situation becomes even more complex. In this environment, several copies of the same file can be kept on different computers. Since the various replicas may be accessed and updated concurrently, some distributed systems ensure that, when a replica is updated in one place, all other replicas are brought up to date as soon as possible.