Assignment (Video- 13 to 15): Database System Impl. (COP6726)

Name: Vikas Chaubey**, UFID:** 3511 5826**, Email:** vikas.chaubey@ufl.edu

1) **Application Space vs Kernel Space:** Modern operating system divides main memory present in the computer central processing unit into two parts one is kernel space and other is user space. This segregation by the operating system is done to achieve memory protection and hardware protection from erroneous software behavior kernel space is usually the memory space reserved to run the kernel of operating system. On the other hand, the user space or application space is the memory segment allocated to run programs that runs outside the operating systems kernel. These programs could generally be various programs and libraries that operating system used to interact with the kernel. it involves software programs which perform input/output operation, manipulates file system objects and application software. Kernel is the core of any operating system; kernel has complete control over all the processes running in the computer system. kernel connects the software processes with the hardware units. It is the part of operating system code which always resides in the memory. When the computer system is started then kernel is the first program which is loaded into memory apart from bootloader, once the kernel is loaded it handles rest of the computer startup processes such as memory, peripherals and input and output requests from software. kernel translates these requests into data processing instructions for the central processing unit for processing. on the main memory the kernel space is reserved to load the kernel onto the memory when the operating system is booted. kernel space mainly stores processes like kernel extensions and most device drivers. All the software processes are loaded into the same memory for processing in central processing unit, that creates the problem because without any segregation mechanism all programs can read the stored instructions for all other programs, because practically they reside on the same memory space. hence this virtual segregation is done, and memory is divided into two different spaces logically and that provides isolation for kernel processes and normal software application processes running on the same computer system. Even all the processes running in the application space are also segregated virtually so that they cannot access memory space of other processes running in same application space. This is the primary basis of memory protection in today's computer systems.

2) **Context Switching:** context switching is the feature of the modern multitasking operating system, context switching allows multiple processes running in computer system to share the same central processing unit. When multiple processes are running in the computer system then in order to make parallel execution of processes possible multiple processes has to utilize the same available processing unit for processing.it is done so that processor idle time could be reduced and system can make use of processor efficiently and overall performance of the system could be increased.so when multiple processes are executed parallelly and operating system provides access to other process for the central processor when one process is idle. in this scenario current state of the executing process has to be saved before the control is provided to other process.

Context Switching involves storing the context or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier. The main triggers which cause context switching are multitasking operations in computer system, interrupt handling and user and kernel mode switching.

3) **Soft Threading:** in computer systems threading could be divided into two types first is hardware threading and the other type is software threading. hardware threads can be interpreted as CPU cores. that is how many CPU cores are available for parallel processing in computer system. but hardware threads are a feature of certain modern processors to utilize processor more efficiently under certain circumstances. Sometimes these hardware threads are exposed by the operating system or to the operating system which appears to be additional cores in the computer system.it is done to make utilization of processor more efficient and this concept of threading is called hyperthreading. On the other hand, software threads are abstraction to the hardware to make multiprocessing possible. All the program or processes running in the computer system make use of same resources available in the system, in order to make process execution faster the processor idle time has to be reduced, processor can be idle at times when a program utilizing current processor is in idle state this can happen when the process is waiting from responses from other processes to proceed furtgher.in this case processes become stuck And CPU becomes idle. This could be avoided by assigning CPU to a different process which is waiting for execution access to processor. Software threads are a way to run all tasks in parallel by allocating same resources to different processes for limited time. although allocation of resources is sequential, but it appears that treads are running in parallel. these software threads are managed by operating system.

4) **Iterator Model:** In computer programming iterator is an object which is used to access containers. Containers are the grouping of the **data** items. These data items could be lists, maps and other data structures which store programming data in a certain way. The iterator object can access those data elements in these data containers but they themselves do not perform iteration. The iterators to the containers are provided via a containers interface. An iterator is behaviorally similar to a database cursor. Iterators date to the CLU programming language. In programming languages iterators are based out of iterator patterns. The iterator decouples the algorithm from the container. Iterator model provides a good abstraction on aggregate objects because it hides the actual presentation of the aggregate object while traversing the data elements. The other advantage of this abstraction is that it allows to add additional traversing operation without changing its interface.