COMPUTER ORGANIZATION AND ARCHITECTURE

COA LAB 2023

NAME: YAMSANI KRISHNA VAMSHI

ROLL NO: 21CS01019

GROUP NUMBER: 13

ASSIGNMENT- 2

Kernel:

- Kernel is a core part of an operating system.
- Kernel adminsters and manages each operation on hardware and software.
- It is the part of the OS that always resides in computer memory and enables the communication between software and hardware components.
- A kernel of an OS is responsible for performing various functions and has control over the system.
- Main tasks of a kernel are device management,task management and memory management.
- Kernel panic is an undesirable event where kernel crashes resulting in crashing of the system.

Device Management

A kernel is responsible for controlling devices like keyboard,mouse,etc using device drivers. OS is able to communicate with any hardware devices with the help of programs known as device drivers.

Task Mangement

Kernel is responsible for prioritizing which process has to be executed.

Memory Management

Each process requires some memory to work, and the kernel enables the processes to safely access the memory as kernel has full access for memory.

Resource Management

A kernel shares resources between processes, ensuring that each process uniformly accesses the resource. It also provides synchronization and inter-process communication.

Types of Kernel:

1. Monolithic Kernel - Same memory space is used to implement user services and kernel services. The execution of processes is faster than other kernel types as it does not use separate user and kernel space. Ex: Linux

2. Micro Kernel - Different memory space is used to implement user services and kernel services. So microkernels can be managed easily. Ex: K42

3. Hybrid Kernel – They are mixture of both monolithic and micro kernel. Ex:Windows NT

4. Nanokernel - In *Nanokernel, the complete code of the kernel is very small, which means the code executing in the privileged mode of the hardware is very small*. Here the term nano defines a kernel that supports a nanosecond clock resolution. Ex: EROS.

5. Exokernel - Exokernel is still developing and is the experimental approach for designing OS.

Thread:

- A single sequential flow of instructions of a process is known as thread.
- Thread is often referred to as a lightweight process.
- Each thread from same process share all resources.
- They all have different program counter , stack and registers.
- Advantages of using threads:

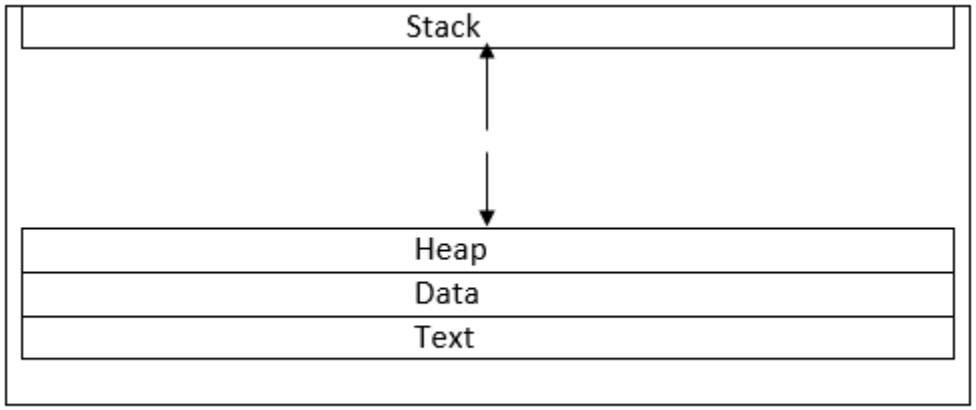
1. Time taken to create a new process is more than that of a creating a new thread.
2. Context switching is faster when working with threads.
3. It takes less time to terminate a thread than a process.

- User-level thread and kernel-level thread are types of threads.

1. OS does not recognize user-level thread.
2. If a user performs a user-level thread blocking operation, the whole process is blocked.
3. The kernel thread recognizes the operating system.
4. If a kernel thread performs a blocking operation, the Banky thread execution can continue.

Process:

- A process is a program in execution.A process is an 'active' entity as opposed to the program which is considered to be a 'passive' entity.
- A process has lot more than a program. It consists of 4 parts – Text,data,stack and heap.



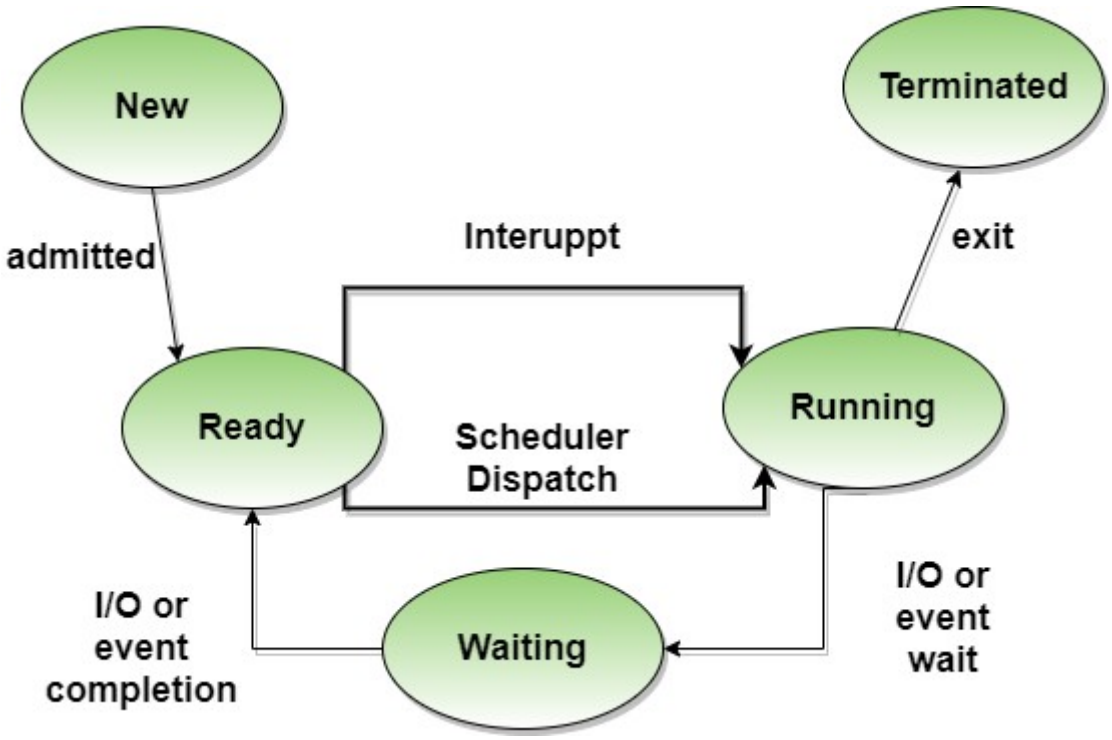
Process Diagram

- Stack stores temporary information such as method or function arguments, the return address, and local variables. Heap is used for dynamic allocate memory.
- Text consists of the information stored in the processor's registers as well as the most recent activity indicated by the program counter's value. Data contains both static and global variables.
- Operating system manages a process with the help of a PCB – process control block.
- A PCB consists of Process ID,Process State,Program Counter,CPU Registers,CPU Scheduling Information,Accounting and Business Information,Memory Management Information, and Input Output Status Information.
- Each process is uniquely identified by it’s Process ID which is assigned by OS.

- Process state tells us about the state of process. The valid states are New,Ready,Running,Waiting,and Terminated.

1. New : A Program which is going to be taken up by the Operating System directly into the Main Memory.
2. Ready : Processes which are in main memory and waiting for execution.
3. Running: Processes state it is running in CPU.
4. Waiting : Process which came out of CPU and waiting for response from another process or for a specific resource to be allocated or for user input.
5. Terminated: A process state is terminated when it has completed it's execution.

- Program counters stores last executed instruction in the program.



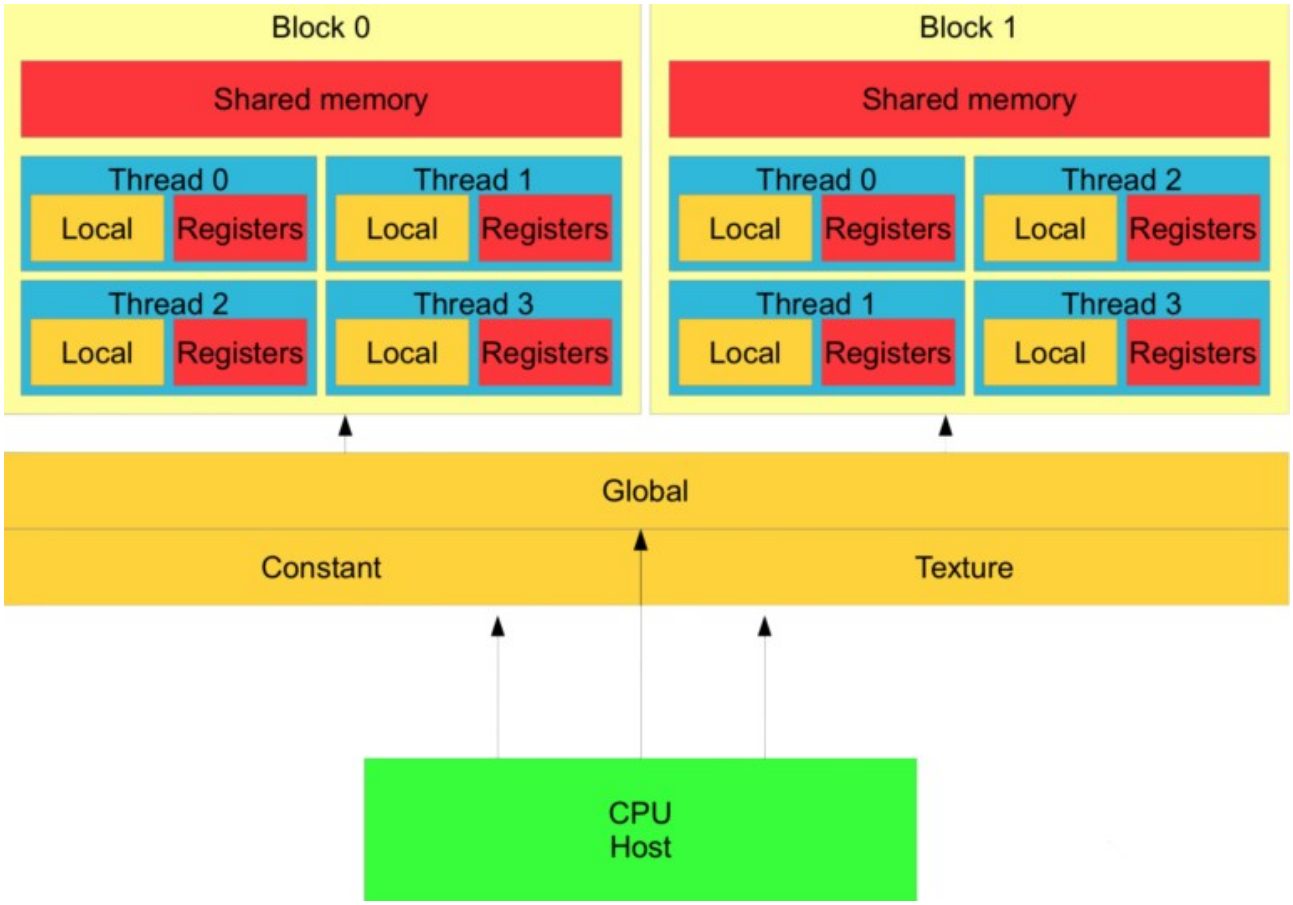
SIMD:

- SIMD means Single Instruction Multiple Data streams.
- It is a type of arcitecture in which a same instruction is run on a number of data.

- The shared memory unit must contain multiple modules so that it can communicate with all the processors simultaneously.
- Ex:Wireless MMX unit

GPU MEMORY HIERARCHY:

A pictorial representation of different types of memeory in a GPU.



SRAM:

- SRAM stands for Static Random Access Memory.
- It is used to store data as long as power is supplied.It is also a type of volatile memory.
- SRAM stores data with the help of flipflops.
- SRAM are used to store computer cache memory.

- Ex: SRAM chips are often used in cell phones, wearables and other consumer electronics.

DRAM:

- DRAM stands for Dynamic Random Access Memory.
- DRAM can store more amount of data but not for longer time.
- DRAM stores data with the help of capacitors. As charge on the capacitors decays so they have to be refreshed periodically to retain data.
- One bit storing cost of SRAM is more than DRAM.
- When compared with DRAM,SRAM performs better and use less power when it is idle but it can not store as much data as DRAM.
- Ex: DRAM is used for computer main memory.

Shared Memory:

- Shared memory is a memory shared between two or more processes.
- It is a type of IPC(inter-related process communication) technique which is used when two unrelated process have to communicate with each other.
- Shared memory is the fastest inter-process communication mechanism.
- Shared memory can be used by following below steps:

1. Request a memory segment that can be shared between processes to the operating system.
2. Associate a part of that memory or the whole memory with the address space of the calling process.

- shmget( ) and shmat( ) are functions used for IPC using shared memory.
- shmget() function is used to create the shared memory segment, while the shmat() function is used to attach the shared segment with the process's address space.

Constant Memory:

- Constant Memory is a read only cache.

- All threads can access it unlike shared memory which can only be accessed by threads in a block.
- A variable allocated in constant memory needs to be declared in CUDA by using the special `__constant__` identifier.

Scheduler:

- Schedulers are special system software which handle process scheduling in various ways.
- Process Scheduler are of 2 types – Preemptive and Non-preemptive.

1. Preemptive :

- Reasources are allocated for a process for a fixed amount of time.
- Process state are switched from waiting state to ready state or ready to running state.

2. Non – preemptive :

- Resources cannot be taken untill the process executes.
- Resources can be alloted for other process when current process stata changes from running to waiting stage.

- The OS maintains all Process Control Blocks (PCBs) in Process Scheduling Queues. Job , Ready and Device queues are process scheduling queues.

1. Job Queue : This queue keeps all the processes in the system.
2. Ready Queue : This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.
3. Device Queue : The processes which are blocked due to unavailability of an I/O device constitute this queue.

- Schedulers are of 3 types :

    1. Long-Term Scheduler – It is also known as job scheduler.A long-term scheduler determines which programs are admitted to the system for processing. It selects processes from the queue and loads them into memory for execution.
    2. Short-Term Scheduler - It is also called as CPU scheduler. Its main objective is to increase system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process.

3. Medium-Term Scheduler - Medium-term scheduling is a part of swapping. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-processes.

Warp:

- A warp is a set of 32 threads within a thread block such that all the threads in a warp execute the same instruction.
- SM (Streaming Multiprocessor) serially selects threads in a warp.
- Once a thread block is launched on a multiprocessor (SM), all of its warps are resident until their execution finishes.

Thread Block:

- A thread block is a representation of group of threads that can be executed serailly or in parallel.
- Number of threads in a thread block depends on architecture and computation compatability.
- Threads in a same block can communicate with each other with the help of shared memory.
- Group of thread blocks is called grid.
- All the blocks in the same grid contain the same number of threads.
- The number of threads in a block is limited, but grids can be used for computations that require a large number of thread blocks to operate in parallel.
- Every thread in CUDA is associated with a particular index so that it can calculate and access memory locations in an array.