Time: 45 minutes CS2006 Operating Systems, Spring 2024 Marks: 15
Roll No: Quiz-1 20-2-2024

1. Write the suitable option [application code/ kernel code] for the following statements.[2]
   a. Has no standard libraries...............**Kernel Code**
   b. Must be monolithic, statically linked library.........**Kernel Code**
   c. Has no libc (malloc, pthreads, string handling, etc)............**Kernel Code**
   d. Standard libraries can be too slow in.........**Kernel Code**
2. Why are traps sometimes referred to as software interrupts? Briefly explain.[3]

   Solution:

   Traps are sometimes referred to as software interrupts because they are mechanisms used by an operating system to handle exceptional conditions, such as errors or specific events.Traps are initiated by software and are often used for tasks like handling system calls, memory protection violations, or other exceptional conditions that require immediate attention from the operating system.
3. Give any two major drawbacks that you think can take place, if , without modules, the entire kernel loaded into memory to boot. [2.5]

   Solution:

   Waste of memory (embedded systems)
   Slower boot time

4. What is policy and mechanism for operating system structure give an example? [2.5]

   Solution:

- **Policy: What** needs to be done?
  - Example: Interrupt after every 100 seconds
- **Mechanism: How** to do something?
  - Example: timer
- Important principle: separate policy from mechanism
- The separation of policy from mechanism is a very important principle, it allows maximum flexibility if policy decisions are to be changed later.
  - Example: change 100 to 200

5. What is dll and where does operating system store it? [2.5]

   Solution:

   Dll are dynamic linked libraries which are required at running time. As these are only called at run time , therefore these are stored in virtual memory that is disk.

6. Describe the advantages of loadable kernel modules. [2.5]

   Solution:

   The overall result resembles a layered system in that each kernel section has defined, protected interfaces; but it is more flexible than a layered system, because any module can call any other module. The approach is also similar to the microkernel approach in that the primary module has only core functions and knowledge of how to load and communicate with other modules; but it is more efficient, because modules do not need to invoke message passing in order to communicate.