MANAV RACHNA
vidyanatariksha

MANAV RACHNA
UNIVERSITY
(FORMERLY MANAV RACHNA COLLEGE OF ENGINEERING
NAAC ACCREDITED 'A' GRADE INSTITUTION)
Declared as State Private University under section 2f of the UGC act, 1956

DEPARTMENT OF COMPUTER SCIENCE & TECHNOLOGY

Subject Name: Principles of Operating Systems **Subject Code:** **CSH206B-T**

Tutorial :3

Name: Harsh Mittal
Class: BTECH CSE 4B
Roll no: 2K19CSUN01082

1. Differentiate between
   a. Kernel and Operating System
   b. Kernel and Shell
   c. Monolithic Kernel and Micro Kernel
   d. System Call and System Program

A1. a)

SNo.	Kernel	Operating System
1	Kernel is a part of Operating System.	Operating System is a system software.
2	Kernel acts as an interface between applications and hardware.	Operating System acts as an interface between user and hardware.
3	An Operating System needs Kernel to run.	A computer needs Operating System to run.
4	Kernel is the first program to load when Operating System loads.	Operating System is the first program to load when the computer boots up.

b)

SNo.	Shell	Kernel
1	Shell allows the users to communicate with the Kernel.	Kernel controls all the tasks of the system.
2	It is the interface between kernel and user.	It is the core of the Operating System.
3	It carries out commands on a group of files by specifying a pattern to match.	It performs memory management.
4	Its types are: Bourne Shell, C shell, Korn shell, etc.	Its types are: Monolithic Kernel, Micro Kernel, Hybrid Kernel, etc.

c)

SNo.	Micro Kernel	Monolithic Kernel
1	Micro Kernel is smaller in size.	Monolithic Kernel is larger than Micro Kernel.
2	Its execution time is slow.	Its execution time is fast.
3	To write a Micro Kernel, more code is required.	To write a Monolithic Kernel, less code is required.
4	Examples: QNX, Symbian, L4Linux, etc.	Examples: Linux, BSDs, etch

d)

SNo.	System Call	Function Program
1	System calls provide the interface between a running program and the operating system.	System programs provide a convenient environment for program development and execution.
2	Generally available as assembly-language instructions.	Most users' view of the operation system is defined by system programs, not the actual system calls.
3	Languages defined to replace assembly language for systems programming allow system calls to be made directly (e.g., C, C++).	System Programs can be divided into: File manipulation, Status information, File modification, Programming language support, Program loading and execution, Communications, etc.

2. Define the Following: -
   a. Trap
   b. IVT
   c. ISR

a) A trap, also known as an exception or a fault, is typically a type of synchronous interrupt caused by an exceptional condition (e.g., breakpoint, division by zero, invalid memory access). A trap usually results in a switch to kernel mode, wherein the operating system performs some action before returning control to the originating process. A trap in a kernel process is more serious than a trap in a user process, and in some systems is fatal. In some usages, the term trap refers specifically to an interrupt intended to initiate a context switch to a monitor program or debugger.

b) An interrupt vector table (IVT) is a data structure that associates a list of interrupt handlers with a list of interrupt requests in a table of interrupt vectors. Each entry of the interrupt vector table, called an interrupt vector, is the address of an interrupt handler. While the concept is common across processor architectures, IVTs may be implemented in architecture-specific fashions. For example, a dispatch table is one method of implementing an interrupt vector table.

c) An ISR (also called an interrupt handler) is a software process invoked by an interrupt request from a hardware device. It handles the request and sends it to the CPU, interrupting the active process. When the ISR is complete, the process is resumed.

3. The services and functions provided by an operating system can be divided into two main categories. Briefly describe the two categories and discuss how they differ.

   One class of services provided by an operating system is to enforce protection between different processes running concurrently in the system. Processes are allowed to access only those memory locations that are associated with their address spaces. Also, processes are not allowed to corrupt files associated with other users. A process is also not allowed to access devices directly without operating system intervention. The second class of services provided by an operating system is to provide new functionality that is not supported directly by the underlying hardware. Virtual memory and file systems are two such examples of new services provided by an operating system.

4. What is an Interrupt? Provide some examples where hardware and software interrupt is generated.
   An interrupt is a signal sent to the processor that interrupts the current process. It may be generated by a hardware device or a software program.
   A hardware interrupt is often created by an input device such as a mouse or keyboard. For example, if you are using a word processor and press a key, the program must process the input immediately. Typing "hello" creates five interrupt requests, which allows the program to display the letters you typed. Similarly, each time you click a mouse button or tap on a touchscreen, you send an interrupt signal to the device.

5. Explain the following: -
   a. Dual Mode Operation
   b. Memory Protection
   c. I/O Protection

   Dual-mode operation forms the basis for I/O protection, memory protection and CPU protection. In dual-mode operation, there are two separate modes: monitor mode (also called 'system mode' and 'kernel mode') and user mode. In monitor mode, the CPU can use all instructions and access all areas of memory. In user mode, the CPU is restricted to unprivileged instructions and a specified area of memory. User code should always be executed in user mode and the OS design ensures that it is. When responding to system calls, other traps/exceptions, and interrupts, OS code is run. The CPU automatically switches to monitor mode whenever an interrupt or trap occurs. So, the OS code is run in monitor mode.

   Input/output protection: Input/output is protected by making all input/output instructions privileged. While running in user mode, the CPU cannot execute them; thus, user code, which runs in user mode, cannot execute them. User code requests I/O by making appropriate system calls. After checking the request, the OS code, which is running in monitor mode, can actually perform the I/O using the privileged instructions.

   Memory protection: Memory is protected by partitioning the memory into pieces. While running in user mode, the CPU can only access some of these pieces. The boundaries for these pieces are controlled by the base register and the limit register (specifying bottom bound and number of locations, respectively). These registers can only be set via privileged instructions.

6. What are the five major activities of an operating system in regard to file management, Process Management?

   The five main major activities of an operating system in regard to process management are:

   - Creation and deletion of user and system processes.
   - Suspension and resumption of processes.
   - A mechanism for process synchronization.
   - A mechanism for process communication.
   - A mechanism for deadlock handling.

   The five main major activities of an operating system in regard to file management are:

   - The creation and deletion of files.
   - The creation and deletion of directions.
   - The support of primitives for manipulating files and directions.
   - The mapping of files onto secondary storage.
   - The backup of files on stable storage media.

7. Draw the Memory Structure for some processes and operating system with their legal memory addresses. The base and limit registers are loaded with the addresses 1050 and 1000, respectively. Suppose P1 and P2 reference the memory locations 2040 and 3052, respectively. Check if the processes will be allowed to execute.

   P1 is first checked against the base register 1050. In this case, the reference memory location of P1, i.e., 2040, is greater than base register. Now, P1 is checked against the sum of base and limit registers, i.e., 1050 + 1000 = 2050. Since it is less than 2050, it will be allowed to execute. On the other hand, if P2 references a memory location 3052, it is not allowed to execute because it violates the second criterion, i.e., m < limit + base. So the control is passed to the operating system as it attempts to access the memory location of P3.

8. Which of the following instructions should be privileged?
   a. Switch from User Mode to Kernel Mode – unprivileged
   b. Updating Base and Limit Register - unprivileged
   c. Clear Memory Location - unprivileged
   d. Read a clock - unprivileged
   e. Interrupts are disabled - privileged
   f. Executing a loop to enter user data - privileged
   g. Load a value in processor register - unprivileged
   h. Send a file to printer to print - unprivileged