REAL TIME OPERATING SYSTEMS

Lesson-1:

OPERATING SYSTEM SERVICES GOAL, MODES AND STRUCTURE

1. OS Services Goal

Goal

The OS Service Goal – Perfection and correctness during a service

OS Services Goal

- **(i)** Facilitating easy sharing of resources as per schedule and allocations—

No processing task or thread uses any resource until it has been allocated by the OS at a given instance

Meaning of Resource of a system

- processor (s), memory, I/O
- devices, virtual devices,
- system timer, software timers
- keypad, displays, printer and other such resources, which processes (tasks or threads) request from the OS.

OS Services Goal…

- **(ii)** Facilitating easy implementation of the application program with the given system-hardware through the system software

OS Services Goal…

- (iii) Optimally scheduling the processes on one (or more CPUs if available) by providing an appropriate context switching mechanism

OS Services Goal…

- (iv) Maximizing the system performance to let different processes (or tasks or threads) share the resources most efficiently with protection and without any security breach.

OS Services Goal…

- (v) Providing management functions for the processes (tasks or threads), memory, file, physical and virtual devices and I/Os and for other functions for which it is designed
- (vi) Providing management and organisation functions for the I/Os, devices and files and file -like devices

OS Services Goal…

- (vii) Providing interoperability of the application on different networks
- (viii) Providing a common set of interfaces that integrates various devices and applications through standard and open systems

OS Services Goal…

- (ix) Providing easy interfacing and management functions for the network protocols and network
- (x) Providing portability of the application on different hardware configurations

2. User and Supervisory mode structure

Processor in the system running in two modes

- Clock, called system clock.
- At every clock tick of the system-clock, there is system interrupt.
- On each system interrupt, the system time updates, the system context switches to the supervisory mode from user mode.
- After completing the supervisory functions in the OS, the system context switches back to user mode

User mode

- User process─ permitted to run and use only a subset of functions and instructions in OS.
- Use of OS functions in user mode─ either by sending a message to the waiting process associated in the OS kernel space or by system call (calling an OS function).

User mode

- The use of hardware resources including memory is not permitted without OS making the call to the OS functions, called system call.

User mode

- User function call, which is not a system call, is not permitted to read and write into the protected memory allotted to the OS functions, data, stack and heap
- That protected memory space is also called kernel space. Hence execution of user functions calls is slower than the execution of the OS functions (on system call) due to need to spend time in first checking the access permission to the protected space

Supervisory mode

- Also called kernel mode
- OS runs in protected mode the privileged functions and instructions in protected mode that are the privileged ones and the OS (more specifically, the kernel) is only one permitted to access the hardware resources and protected area memory
- Kernel space functions and processes execute faster than the user space functions and processes.

Supervisory mode

- Only a system call is permitted to read and write into the protected memory allotted to the OS functions, data, stack and heap

3. Example— Improving system performance by running application program threads in supervisory (kernel) mode

RTOS Windows CE and several RTOSes

- Run all the threads run in the supervisory mode (kernel mode). Therefore, the threads executes fast.
- Improves the system performance.
- If the threads are to execute in user mode, as in Unix or in non-real time OS, then the execution slows down due to time spent in checks on the code access to the protected kernel space

3. Structure

Top to down Structural Layers

- Software Application Programming Interface (API)
- System software other than the one provided at the OS
- OS Interface
- OS
- Hardware–OS Interface
- Hardware

Top to down Structural Layers

- OS is the middle in-between layer between the application software and system hardware

4. Kernel Services in the OS

Kernel

- Process, memory and IO managers are essential components of a kernel
- Kernel may include file and device management functions as part of the kernel in a given OS
- Kernel in certain OSes is without File and Device management functions as part of the kernel in the given OS and any other needed functions other than process, memory and IO device managers, which are essential components of a kernel are not provided for at the kernel

Process Management Kernel Services in an OS

- Creation to deletion of Processes
- Process structure maintenance
- Processing resource requests
- Scheduling Processes
- Inter process Communication (IPC) (communication between Tasks, ISRs, OS functions)

Memory Management Kernel Services in an OS

- Allocation and de-allocation between Tasks, ISRs, OS functions

Kernel Services in an OS

- File Management
- Device Management,
- Device Drivers
- I/O Management
- Interrupts Control (by handling ISR) Mechanism

Summary

We learnt

- OS goal is perfection, correctness, portability, interoperability, and providing a common set of interfaces for the system, resources and orderly access and control when managing the processes

We learnt

- On each system clock interrupt, the system time updates, the system context switches to supervisory mode from user mode. After completing the supervisory functions in the OS, the system context switches back to user mode

We learnt

- OS structure consists of kernel and other service functions outside the kernel. System software includes the kernel. The OS lets the application run on the hardware
- OS is the middle in-between layer between the application software and system hardware
- Process, memory and device managers, which are essential components of a kernel

We learnt

- Kernel -_A basic unit of any OS that includes the functions for processes, memory, task scheduling, inter process communication, management of devices, IOs, and interrupts and may include the file systems and network subsystems in certain OSes

We learnt

- OS lets a process execute on the CPU—some process at OS for a resource-management mechanism lets it use the system-memory and other system-resources such as network, file, display or printer

We learnt

- Kernel functions are management of processes, process and resources scheduling memory, file, devices, IOs, network and handling the interrupts

End of Lesson 1 of Chapter 8