REAL TIME OPERATING SYSTEMS

Lesson-6:
Device Management Functions
1. Device manager functions
Device Driver ISRs

- Number of device driver ISRs in a system,
- Each device or device function having a separate driver, which is as per its hardware
Device manager

- Software that manages the device drivers of each device
- Provides and executes the modules for managing the devices and their drivers ISRs.
- Effectively operates and adopts appropriate strategy for obtaining optimal performance for the devices.
- Coordinates between application-process, driver and device-controller.
Device manager

- Process sends a request to the driver by an interrupt; and the driver provides the actions by executing an ISR.
- Device manager polls the requests at the devices and the actions occur as per their priorities.
- Manages IO Interrupts (requests) queues.
Device manager

- Creates an appropriate kernel interface and API and that activates the control register specific actions of the device. [Activates device controller through the API and kernel interface.]
Device manager

- Manages the physical as well as virtual devices like the pipes and sockets through a common strategy.
Device management has three standard approaches

- Three types of device drivers:
  - (i) Programmed I/Os by polling from each device its the service need from each device.
  - Interrupt(s) from the device drivers device-ISR and
  - (iii) Device uses DMA operation used by the devices to access the memory.
- Most common is the use of device driver ISRs.
Device Manager Functions

- Device Detection and Addition
- Device Deletion
- Device Allocation and
- Registration
- Detaching and Deregistration
Device Manager Functions

- Restricting Device to a specific process
- Device Sharing
- Device control
- Device Access Management
- Device Buffer Management
- Device Queue, Circular-queue or blocks of queues Management
Device Manager Functions

- Device drivers updating and upload of new device-functions
- Backup and restoration
Device Types

- char devices and
- block devices
2. Set of Command Functions for the Device Management
Commands for Device

- create
- open
- write
- read
- ioctl
- close and
- delete
ioctl Command for Device

- (i) Accessing specific partition information
- (ii) Defining commands and control functions of device registers
- (iii) IO channel control
Three arguments in ioctl ( )

- First Argument: Defines the chosen device and its function by passing as argument the device descriptor (a number), for example, fd or sfd. Example is fd = 1 for read, fd = 2 for write.
- Second Argument: Defines the control option or uses option for the IO device, for example, baud rate or other parameter optional function.
- Third Argument: Values needed by the defined function are at the third argument.
Example

- Status = ioctl (fd, FIOBAUDRATE, 19200) is an instruction in RTOS VxWorks.
- \( fd \) is the device descriptor (an integer returned when the device is opened)
- FIOBAUDRATE is the function that takes value = 19200 from the argument.
- This at configures the device for operation at 19200-baud rate.
3. Device Driver ISR functions
ISR functions

- `intlock()` to disable device-interrupts systems,
- `intUnlock()` to enable device-interrupts,
- `intConnect()` to connect a C function to an interrupt vector
- Interrupt vector address for a device ISR points to its specified C function.
- `intContext()` finds whether interrupt is called when an ISR was in execution
4. Unix OS functions
UNIX Device driver functions

- Facilitates that for devices and files have an analogous implementation as far as possible.
  - `open ()`,
  - `close ()`,
  - `read ()`,
  - `write ()` functions analogous to a file `open`, `close`, `read` and `write` functions.
APIs and kernel interfaces in BSD (Berkley sockets for devices)

- `open`
- `close`
- `read`
- `write`
in-kernel commands

- (i) `select()` to check whether read/write will succeed and then select
- (ii) `ioctl()`
- (iii) `stop()` to cancel the output activity from the device.
- (iv) `strategy()` to permit a block read or write or character read or write
Summary
We learnt

- Device Manager initializes, controls, and drives the physical devices and virtual devices of the system.
- Main classes of devices are char devices and block devices.
- Device driver functions may be similar to file functions, open, read, lseek, write and close.
End of Lesson 6 of Chapter 8