UNIT IV

RTOS BASED EMBEDDED SYSTEM DESIGN

4.4 Device Management Functions

□ **Number of device driver ISRs in a system,**

□ **Each device or device function having a separate driver which is as per its hardware**

□ 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 request at the devices and the actions occur as per their priorities.

Manages IO Interrupts (requests) queues.

creates an appropriate kernel interface and API and that activates the control register specifications of the device. [Activates device controller through the API and kernel interface.]

□ 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/O s by polling from each device its the service need from each device.

(ii) 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

□ 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 drivers updating and upload of new device-functions

□ Backup and restoration

Device Types

□ Char device sand

□ Block devices

Set of Command Functions for the Device

Management Commands for Device

□ create

□ open

□ write

□ read

□ io

□ close and

□ delete

IO control Command for Device

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

Three arguments in io ct l ()

- First Argument: Defines the chosen device and its function by passing as argument the device descriptor (a number), for example, fd or s fd Example is fd =1 for read, fd = 2forwrite.
- 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 =io ctl (fd FIO BAUDRATE, 19200) is an instruction in RTOS V x Works.
- *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.

Device Driver ISR functions
ISR functions

- *Int lock* ( )to disable device-interrupts systems,
- int Unlock ( ) to enable device-interrupts,
- *int Connect* ( )to connect a C function to an interrupt vector
- Interrupt vector address for a device ISR points to its specified C function.
- int Context ( ) finds whether interrupt is called when an ISR was in execution

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.

API s 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) io ctl ( )

(iii) stop ( ) to cancel the output activity from the device.
(iv) strategy( )to permit a block *read or write* or character *read or write*