DEVICE DRIVERS AND TERRUPTS SERVICE MECHANISM

Lesson-13: Multiple device functions

Multiple functions in a Timer device

- A *timer* device control registers needs to be programmed by a device function. A timer device performs does timing functions as well as counting functions. It also performs does the delay function and periodic system calls.

Multiple functions in a Transceiver device

- A *transceiver* device control registers needs to be programmed by a device function
- A *transceiver* device does transmitting as well as receiving functions.
- It may not be just a repeater. It may also do the *jabber control* and *collision control*.
- Jabber control means prevention of that it prevent continuous streams of unnecessary bytes in case of some system fault.
- Collision control means that it must first sense the network bus availability then only transmit.

Multiple functions in a device

- Voice-data-fax modem device has transmitting as well as receiving functions for voice, fax as well as data.

DEVICE DRIVER

- A program that controls the functioning of a device.
- Every device, whether it be a peripheral such as printer, LCD display unit, keypad or an internal device such as timer, must have a driver.

Common driver or the separate drivers

- A common driver or the separate drivers for each device function are required.
- Device-drivers are the important routines in a system
- Routines generic functions, such as create ( ), open ( ), connect ( ), listen ( ), accept ( ), read ( ), write ( ), close ( ), delete ( ) for use by high level programmers

Device Driver

- System has number of physical devices. A device may have multiple functions. Each device function is controlled by the driver
- The driver provides a software layer (interface) between application and the actual device.

Device Application programmer using the generic commands

- The driver facilitates the use of device by using the generic commands sent by application program-steps.
- Once a driver is available, application programmer does not need to know any thing about the mechanism, addresses, registers, bits and flags used by the device.

Device Driver Function

- The driver function is to translate a generic command in an application (user software) into specific set of commands that the device understands.
- Generic command means a command used for the number of devices or device groups.

DEVICE DRIVER

- Many times, the drivers, such as the keypad driver, LCD display unit driver, modem driver, are available to an application program from the operating system.

DEVICE DRIVER

- Several times, drivers, such as LCD display unit driver for the multilingual display, not available at the OS have to written by the application programmer or sourced from other programmer for an embedded system

Driver routine as a black box

- Simple commands from a task or function can then drive the device.
- Once a driver is available, application developer does not have to know the any thing about the mechanism, addresses, registers, bits and flags used by the device.
- For example, consider a case when system clock is to be set to tick every 10000 µs (100 times each second).

Driver routine OS_Ticks (100) as a black box

- The user of OS for the application simply makes a call to function like OS_Ticks (100)
- It is not necessary for the user this function to know which timer device will do that.
- What are the addresses, which, will be used by the driver? Which will be the device register where value 100 for the ticks registers?
- What are the control bits that will be set or reset?

Driver routine OS_Ticks (100)

- When runs, simply interrupts the system and executes the software interrupt (SWI) instruction which calls the signaled routine (driver) for the system ticking device.
- Then the driver, which executes will takes 100 as *input* and will configures the real time clock to let the system clock tick each 10000 µs and generate continuously the system clock interrupts continuously every 10000 µs to get 100 ticks each second

Device driver accesses to the device

- Accesses the device through interrupt (s), and drives a peripheral device or internal device through generic commands.
- User don’t have to write codes when OS provides the drivers.
- A device’s file descriptor is shared by the commands and interrupt service routines, which execute those commands.

Application Program Access to a Device

- Access to a device is by using generic commands of the driver.
- When a command executes in application program, device driver generates an interrupt (using an SWI instruction, such as INT *n* or SWI *m*) to the device occurs.
- An interrupt service routine corresponding to that command is then executed.

DEVICE DRIVER

- Several times, drivers, such as LCD display unit driver for the multilingual display, not available at the OS have to written by the application programmer or sourced from other programmer for an embedded system

A Device Driver Command

- A device may be configured (commanded), for example, configured to read or write or read and write both through the port address (es).
- Device may be set to function in different modes. For example, mode 0 for one data transfer rate, 1 for another.

File descriptor

- A general term File descriptor refers to device descriptor data structure
- A data structure accessed by the programmer using an integer number, fd

Driver Generic commands examples

- fd = open (“device1”, O_RDWR, 0) in an application.
- Means configure the device at address device1 for the read as well as write operations and set the device for mode 0 functions. fd1 is device1 descriptor pointer, which open function returns.

Descriptor

Routine addresses for the open, write, read, close and other functions
Device port address(es)
Device Status, Mode, Present config.
Read buffer start and end addresses
Write buffer start and end addresses
Current cursor position address(es) for read or write or read-write.

Cursor

- Current cursor address is an address from address at the buffer from where the read (operation) will be done on next read command or to where the write (operation) will be done in next write operation. For example, cursor in an LCD display means the address (position) for the *write* (then display) from which line number (*row*) which character number (*column*).

Driver Generic commands example 2

- write (*fd1*, device1, 1024) in the application means write into the device buffer 1024 Bytes from current cursor position. Device if set in mode 0 then transmit from the buffer at the rate defined for its mode 0 operation. Device descriptor is at fd1.

Driver Generic commands example 3

- msg1 = read (*fd1*, device1, 32) in the application means (a) device is device at address (b) descriptor is at fd1 (c) read and save at msg1 from the device buffer the 32 Bytes from current cursor position.

Driver Generic commands example 4

- fd1 = close (device1) in an application means close the device1 and save the device current status after close operation. The device can now be reused only after first executing open function.

Device Types

- For each type of device, there is a set of the generic commands.
- For example, for char device one set of commands and for block device there can be another set.

Summary

We learnt

(i) The driver accepts generic commands from an application or OS program and then translates each command into a set of specialized commands for the device.

We learnt

- (ii) The driver provides a software layer (interface) between application and the actual device. (iii) A device is configured and initialized by the function ‘open’ of the driver.

We learnt

- (v) A device is deactivated by the function ‘close’ at the driver.

End of Lesson 13 of Chapter 4