

Device Drivers

COMP755

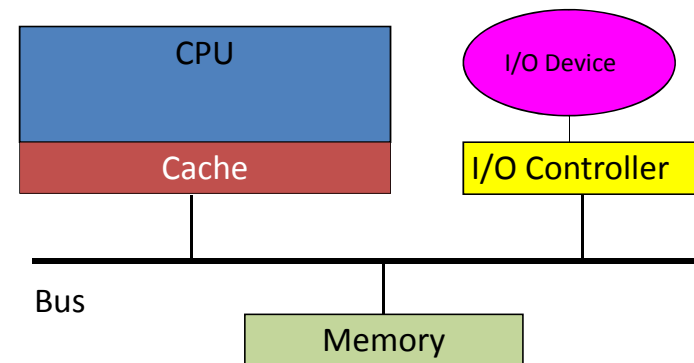
“Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers?”

Linus Torvalds

Hardware Operation

- The I/O controller for a device will have a port address to receives commands and another port to check the status of the device.
- Each device determines what commands it uses and how those commands are formatted
- A device can be made to do something by storing a specially formatted number in the command port.
- The status of a device is checked by reading the status port.

Basic Computer Components



Linux devices

- Under Unix, devices appear as special virtual files in the file system.
- Users can access a device as if it were a file using the regular file operations such as open, read, write and close.
- Devices usually appear in the `/dev` directory

What is a Device Driver?

- A device driver is software that interfaces between the operating system and a hardware device.
- The device driver issues commands to a device to perform the operations requested by the OS.
- When a device completes an operation, the device driver may invoke a routine in the OS.
- Device drivers often service interrupts from their device.

Generic to Hardware Specific

- Device drivers provide public methods to perform common (or not so common) actions, such as read, write, etc.
- The device driver sends hardware specific commands to the device to make it perform the generic request.

Why Device Drivers?

- There are zillions of different widgets that you might plug into a computer
- Each one may have its own set of commands and statuses
- The creators of the OS cannot be expected to know how to control every device in the world
- The manufacturer of the device will generally write the device driver.

Character or Block

- Linux distinguishes between character and block oriented device drivers
- A character device can read or write a single character
 - Serial port
 - Mouse (read only device)
- Block devices always read or write data in multiples of a block
 - Disk drives

Device Driver Design

Device drivers often have two layers

- Logical
 - Process requests for a class of devices
 - Often written by OS vender
- Physical
 - Communicates with a specific device
 - Handles the details of how that device functions
 - Usually written by device manufacturer

Permanent or Loadable

- Some device drivers are loaded at boot time and remain loaded all the time.
- Other device drivers can be loaded and unloaded. Some are loaded automatically.
- The Linux command **lsmod *drivername*** will load the specified driver.
- The command **rmmod *drivername*** will unload the driver.

Kernel and User Mode

- Device drivers traditionally execute in kernel mode.
- They often need that privilege to directly access the hardware.
- If a device driver has a bug and causes a fault in kernel mode, it will often crash the system

Vender Tension

- Microsoft claims that most “Blue Screens of Death” are caused by faulty drivers.
- The OS vendor does not create the drivers and has little control over their quality.
- Some device drivers can execute in user mode

Windows Driver Foundation (WDF)

- A set of Microsoft tools that aid in the creation of device drivers for Windows
- Comprised of the Kernel Mode Driver Framework (KMDF) and User Mode Driver Framework (UMDF).
- The goal is to require the programmer to know only a few concepts to write a simple driver and then incrementally add complexity

User-Mode Driver Framework

- Runs at user level instead of Kernel level
- Failure of a user mode driver will not crash the system. The driver will be restarted.
- UMDF driver is a Dynamic Link Library (DLL)
- Accesses the OS through an API

WDF Tools

- The Static Driver Verifier (SDV) is capable of performing deep validation of code paths.
- SDV can find driver problems that span multiple function calls
- PReFast for Drivers (PFD) performs more shallow analysis of driver operations. PFD can check for buffer overruns and other common programming errors.

Virtual device drivers

- A driver can control a device that does not really exist.
- Useful for virtual machine environments
- A virtual disk driver might actually read and write to a file on the host system
- A virtual device driver might simulate an old device using a new device

Open Source Drivers

- Creating a driver requires detailed information on how the device works.
- There are many open source drivers available
- CUPS is a printing system for Unix-like systems

Assignment

- Read (*listen*) about Android and/or device drivers.
- Send me some new facts about Android or device drivers that were not mentioned in the lecture.
- The facts will be judged for quality from trivial to significant.
- Your fact must be unique from anything submitted by another student.

Submission

- Email your facts to me before Thursday, November 19, at midnight or use the digital drop box.
- Your description of the "fact" must be clear and understandable.
- Facts must be defined in one paragraph (<600 characters) unless you really need more.
- You must provide a reference.