CMPSCI 577 Operating Systems Design and Implementation **Spring 2020**

Lecture 16: 03/26

Lecturer: ***Prashant Shenoy*** *Scribe:* ***Shashwat***

16.1 Minix I/O and device drivers

This lecture discusses about minix subsystem, device independent IO subsystem code within minix and also device drivers within minix and how they are implemented

16.2 Minix I/O Subsystem

Hardware interfaces with the rest of the machine through interrupts and is the lowest layer of minix. Interrupt handlers are found within the kernel and interface with the device drivers. There is one device driver for each device and includes all the device specific code. Device independent OS includes high level abstraction of the devices such as block devices or character devices.

Uniform Interfacing for device drivers
A) Without a standard driver interface
B) With a standard driver interface

The interface should be generic so that support to a large class of devices could be provided irrespective of the hardware details.

16.3 User-Space I/O Software

All of the device drivers and most of the kernel tasks are implemented as user space processes. Device driver sets up the device registers and waits for the interrupt to come.

16.4 Minix Interrupt handlers and I/O

A) Minix device drivers run in user space unlike monolithic kernels
B) Minix kernel handles the interrupts and forwards the request to the specific device driver

Predictable interrupts: These are the ones which we know are going to arrive because of an I/O request outstanding.

Unpredictable interrupts: Examples include network packets arriving on ethernet interface card and a keyboard interrupt.

16.5 Driver Access to kernel

A) System task can allow user processes to access segments from other address spaces
B) Minix kernel calls support access to I/O ports
C) Handles predictable interrupts
D) Unpredictable interrupts

16.6 Minix Drivers

A) Separate device driver for each class of I/O device.
B) Drivers run as full-fledged user processes.
C) Communicate with system task using message passing
D) Simple drivers are written in one source file

16.7 Device Driver Messages

Devices in MINIX have a major and a minor number, major number tells about the classes of the devices. Minor number refers to a specific device within that class.

Drivers are of two types 1-Block device driver- RAM disk device driver, Hard disk driver 2- Character device driver- Terminal driver and keyboard driver

16.8 Device Independent I/O

Vfs contains all device independent code and implements device independent layer in MINIX

16.9 Device driver operations

Each driver needs to handle 6 operations: open, close , read , write, IOCTL: changes device parameters, SCATTERED _IO:read/write multiple blocks

RAM disk: These are volatile temporary storage devices used during boot process to create root device.The OS is bootstrapped using a RAM disk and then the hard disk is accessed and remaining files can then be accessed. Minix Ram disk driver uses 6 drivers in one.

Hard disk driver: Have a geometry of cylinder, track or sector and typically have disk address of type cylinder number, track number, sector number

Terminals: These are different kinds of monitors where the output from a machine can be seen.

Memory mapped terminal: It writes directly into video RAM. Video RAM is an extension of physical RAM.

RS-232 Terminals: These communicates with a computer over a communication line 1 bit at a time

Input software: a) Central buffer pool b) dedicated buffer for each terminal

Terminal driver message types: Read from the terminal, write to the terminal, Set terminal parameters from IOCTL, IO occurred during last clock tick, cancel previous request, open a device, close a device.

Keyboard driver: It scans the input buffer with corresponding key actions for a line of text entered at the keyboard.