

(IT620) DEVICE DRIVE PROGRAMMING

(ELECTIVE - IV)

Objective of the course :

That is a worthy goal, of course; the flow of new hardware products is not likely to slow down anytime soon, and somebody is going to have to make all those new gad gets work with Linux. But this course is also about how the Linux kernel works an how to adapt its workings to your needs or interests. Linux is an open system; wit this course, we hope, it is more open and accessible to a larger community of developers.

UNIT - I

An Introduction to Device Drivers The Role of the Device Driver, Splitting the Kernel, Classes of Devices and Modules, Security Issues, Version Numbering, License Terms, Joining the Kernel Development Community.

Building and Running Modules

Setting Up Your Test System, The Hello World Module, Kernel Modules Versus Applications, Compiling and Loading, The Kernel Symbol Table, Preliminaries, Initialization and Shutdown, Module Parameters, Doing It in User Space

UNIT - II

Char Drivers : The Design of scull, Major and Minor Numbers, Some Important Data Structures, Char Device Registration, open and release, scull's Memory Usage, read and write, Playing with the New Devices.

Debugging Techniques : Debugging Support in the Kernel, Debugging by Printing, Debugging by Querying, Debugging by Watching, Debugging System Faults, Debuggers and Related Tools

UNIT - III

Concurrency and Race Conditions. Pitfalls in scull, Concurrency and Its Management, Semaphores and Mutexes, Completions. Spinlocks, Locking Traps, Alternatives to Locking . Advanced Char Driver Operations ioctl, Blocking I/O, poll and select, Asynchronous Notification, Seeking a Device, Access Control on a Device File.

UNIT - IV

Time, Delays, and Deferred Work, Measuring Time Lapses, Knowing the Current Time, Delaying Execution, Kernel Timers, Tasklets, Workqueues

Allocating Memory, The Real Story of kmalloc, Lookaside Caches, get_free_page and Friends, vmalloc and Friends, Per-CPU Variables, Obtaining Large Buffers

UNIT - V

Communicating with Hardware, I/O Ports and I/O Memory, Using I/O Ports, An I/O Port Example, Using I/O Memory, Interrupt Handling, Preparing the Parallel Port, Installing an Interrupt Handler, Implementing a Handler, Top and Bottom Halves, Interrupt Sharing, Interrupt-Driven

I/O. Data Types in the Kernel, Use of Standard C Types, Assigning an Explicit Size to Data Items, Interface-Specific Types, Other Portability Issues, Linked Lists.

TEXT BOOK :

1. Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman, "Linux Device Drivers", 3rd ed., orielly, 2011.