

GUJARAT TECHNOLOGICAL UNIVERSITY

DEVICE DRIVER WRITING B.E. 8th Semester

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				ESE (E)	PA (M)	ESE Viva (V)	PA (I)	
4	0	2	6	70	30	30	20	150

L- Lectures; T- Tutorial/Teacher Guided Student Activity; P- Practical; C- Credit; ESE- End Semester Examination; PA- Progressive Assessment

Content:

Sr #	Topic	Teaching Hrs.
1	Linux Architecture: Linux - Introduction & basic commands (ls,cd,pwd,cal,arch,date etc..), Linux Directory structure (/mnt,/proc,/dev,/etc & more), Linux process – foreground & background (fg, bg), Advanced commands, vim editor, CVS with Practical Application, Shell script with Exercise, Make file concept	5
2	Data Structure Fundamentals: Structures,Types of Linked List, Stack, Queue, Trees, Graphs, Hash table,Preprocessors, Bit operators, Library Making and complexity, Debugging & optimization	5
3	Kernel Internals: Kernel - Introduction, init Process, Login, Syslog , procs & sysfs, Parts of Kernel, Linux Directory Structure	2
4	Cross compilation and Tool chain: Setting up the cross compilation tool chain on host machine, Using tool chain for compiling user space application for target board, Using tool chain for compiling kernel for target board, Using tool chain for compiling kernel modules for target board Device Driver Concepts Devices in Linux: Devices treated as files, The /dev directory Device classes – character, block, network, Pipe Creating device files with mknod User Space Driver APIs: Low-level API Streams API How Loadable Modules Work: Benefits of loadable modules Correct use of insmod, modprobe, rmmod, and lsmod, Passing parameters to a loadable module The GPL and your driver code Compiling, Loading And Exporting: Writing a simple module, Compiling modules Loading/unloading modules, Exporting symbols from a loadable module, Creating stacked loadable modules Character Devices: Major and minor numbers, Registering character device file Driver methods – the file operations table, Transferring data to/from User Space Tracing and Debugging: Printk for debugging, Device information in /proc strace to track system calls ksyms and ksymoops, Debuggers – gdb and kgdb Blocking and Wait Queues: Multi-tasking Schedule(), Wait Queues, Safe sleeping Poll()	16

	<p>Accessing Hardware: I/O ports vs. memory mapping, Allocating and mapping I/O space, Functions for reading and writing I/O ports, Barriers Accessing I/O from User Space</p> <p>Handling Interrupts: Interrupt Handler functions, Restrictions of kernel code running in interrupt context, Deferred interrupt handling tasklets and workqueues</p> <p>USB Drivers: USB structure and Topology, Driver Structure and organization, Endpoints, interfaces and configurations, USB Request Blocks(urbs)</p> <p>Managing Time: Timer interrupts and jiffies, Short Delays, Task queues, Kernel Timers</p> <p>Block Device Drivers: Block Device Drivers Header files, Registering block drivers, the block device operations structure</p> <p>Network Drivers: The net_device structure, Sockets, Naming scheme and registration, Network driver methods, NAPI the new API</p> <p>Adding a Driver to the Kernel Tree: Where to put it - kernel layout for drivers, Modifying the Makefile, Adding it to configuration options - the Kconfig file</p>	
5	<p>Implementation on Hardware (Panda board / Beagle board - xm): Reading Datasheet of Hardware, Booting Panda board, Formatting SD card, MLO(TI Primary bootloader), Uboot(Secondary bootloader), uImage(Kernel Image), Root File system creation, Boot Process of Pandaboard, Cross Compilation for ARM, Loading Hello Module, Implementing UART Driver or Network Driver</p>	2

Reference Books:

1. Linux Device drivers 3rd Edition - from O'Reilly publications - Authors :Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman
2. The Design of the UNIX Operating system - author :Maurice J Bach
3. Essential Linux Device Drivers by Sreekrishnan Venkateswaran - from Prentice Hall
4. Linux Kernel Development (3rd Edition) by Robert Love from Pearson publications
5. Understanding the Linux Kernel, Third Edition by Marco Cesati from O'Reilly publications

List of Practical's:

1. How to prevent ssh login? Implement, explain & how it works.
2. "Create an alias commands for the following:
 - a. listing of directory contents
 - b. listing of directories which are hidden.
 - c. to go to previous directory
 - d. to search for any string
 - e. fgrep
 - f. math calculator
 - g. to compare files line by line using diff and display output in color form."
3. Write a program to remove all comments from a C program. Don't forget to handle quoted strings and character constants properly. C comments don't nest.
4. Write a function htoi(s), which converts a string of hexadecimal digits (including an optional 0x or 0X) into its equivalent integer value. The allowable digits are 0 through 9, a through f, and A through F.
5. Write a pointer version of the function strcat. Strcat(s,t) copies the string t to the end of s.
6. "Write a program to implement menu driven for singly linked list, Doubly linked list, Circular linked list.
 - a. (Insertion, Deletion, Searching, Sorting, Display).

7. Menu Driven for all the sortings.)
8. Also create header files for each linked list."
9. Design and develop a system call which takes a value from user space, find the value is odd or even and according to the results return the decisive parameter in to in to user space.
10. Write Simple loadable kernel module, which takes integer, Char, String as command line Argument.
11. Write character device driver which can count the bytes of given character.
12. Figure 1 shows the scenario of a chat application. Understand the concept of chat application and implement using the Linux kernel module and user application.

Course Outcome:

The main objectives to give the subject Device Driver Writing are:

- To develop device drivers for Linux
- To give practical experience in developing Linux device drivers.
- The purpose and functionality of device drivers
- Compiling and linking device drivers

Instructional Method and Pedagogy:

1. Lectures will be taken in class room with the use of multi-media presentations, black board or mix of both.
2. Assignments based on above course content will be given to the students at the end of each module. Each assignment contains minimum 10 questions.
3. Mini Project based learning