21 Days Challenge on Linux Device Drivers Essentials

Description
Master the fundamentals of Linux device driver development in just 21 days! This intensive course is designed to take you from a basic understanding of Linux to confidently writing the Device Drivers. This 21 Days Journey serves as kick starter for building the solid fundamentals in Linux Device Drivers. Starting from Linux Fundamentals, Journey Progressively advances into all the Linux Device Driver Essentials.

Objective
The 21 Days Challenge on Linux Device Drivers Essentials attempts to serve multiple objectives:

- To enable participants understand the fundamental of Linux system & getting comfortability with working under Linux enviroment, so as to help them prepare for Linux Device Drivers
- To enable participants with solid Fundamentals of Linux System Programming which would enable them to get the comfortability with end-to-end System.
- To enable participants with solid Fundamentals of Linux Device Drivers

Target group:
Professionals/Students looking to get into Linux Device Drivers

Pre-requisite
Knowledge of C programming language

Methology
Every theoretical topic is accompanied by corresponding hands-on/assignment to get the deep understanding of the topic.

Day 1: Introduction
- What is Linux Driver?
- Why Linux Drivers?
- Roadmap to Linux Driver Developer

Day 2: Introduction to Linux & Its Architecture
- What is OS & its Need?
- What is Linux?
- Components of Linux

Day 3: Setting Up the Linux System
- Various way to setup the Linux
- Installing Ubunttu

Day 4: Linux Usage Basics
- Linux Directory Structure
- System Directories
- The Shell
- File Permissions

Day 5 & 6: Essential Commands in Linux

- File & Directory related commands (cp, mv, mkdir, rm etc)
- Zipping/unzipping the directory
- Get the CPU and Memory info of the system
- Symbolic links
- Mounting/Unmounting the partition
- Connecting to the remote system
- Sending a file over the network
- Miscellaneous Commands

Day 7: Embedded Linux Components

- Various Components that Constitute Embedded Linux

Day 8: Toolchain Fundamentals & Gcc Internals

- What is Toolchain
- Toolchain Components
- GCC & Its Friends

Day 9: Makefile

- Makefile & Make Utility
- Makefile Components
- Writing Your Makefile

Day 10: Git Basics

- Git & its need
- Clone & GIT Project
- Commiting the Changes
- Creating the Patches

Day 11: Editors in Linux

- Various Editors in Linux
- vim Basics
- Editing with vim
- Searching with vim

Day 12: System Calls

- W’s of System Calls
- System Calls & Library Functions
- System Call Examples

Day 13 & Day-14: Processes in Linux

- Process Overview
- Process Creation & Operations
- Waiting for Process Termination
- Zombie Processes

Day 15: Introduction to Linux Device Drivers

- What is Driver?
- Role of Drivers
- Linux Driver Ecosystem

Day 16: Linux Kernel

- Downloading the Kernel
- Linux Kernel Source Code Organization

Day 17: Linux Kernel Module & related commands

- Understanding the Kernel module & related commands
- Writing & Building a first Kernel module

Day 18: Writing Your First Driver

- Pre-Requisitess for Writing a Driver
- Standard Template for Writing a Driver
- Various Optimizations

Day 19: Character Driver Fundamentals

- What is Character driver?
- Major & Minor Number
- Flow from User Space to Kernel Space

Day 20: Character Driver Registration

- Registering & Unregistering the driver
- Writing a First Character Driver

Day 21: Interacting with User Space

- Exchanging the Data with User Space
- Do’s & Dont’s while interacting with User space