Zephyr RTOS Programming (AC6401)

Course Duration: 40 Hours

TOC

Introduction

- The Linux Foundation
- The Linux Foundation Training
- The Linux Foundation Certifications
- Laboratory Exercises, Solutions and Resources
- Labs

Ch 1. Course Introduction

- Introduction and Course Flow

Ch 2. Introduction to Zephyr

- Zephyr ecosystem
- Installation and use of Zephyr

Ch 3. Zephyr Build System

- Toolchains and Zephyr SDK
- Zephyr CMake functions
- Application structure
- Code structure
- Lab: Getting started with Zephyr and using west

Ch 4. Configure Zephyr

- Kconfig from an application perspective
- Config fragments
- Device tree syntax
- Device tree overlay
- Lab: Writing a device tree overlay

Ch 5. West

- Repository management
- West commands
- Lab: Writing a custom west manifest

Ch 6. Zephyr Fundamentals

- Common Subsystems: GPIO, I2C
- DeviceTree specification structures dt_spec
- Preprocessor meta-programming macros
- Data Structures
- Shell
- Lab: Using X-Macros in Zephyr and understanding CONTAINER_OF

Ch 7. Thread Management

- Thread Fundamentals
- Main and Idle Threads
- System Initialization
- Delays and timeout
- Lab: Creating and managing threads
- Lab: Creating periodic threads

Ch 8. Tracing and Logging

- Scheduling Traces
- Thread analyzer
- Logging
- Lab: Creating config overlay for visual trace diagnostics with Tracealyzer

Ch 9. Memory Management

- Memory Overview
- Iterable Sections
- Dynamic memory managers
- Stack Memory Analysis
- Stack Overflow detection
- Lab: Understanding dynamic memory allocation in Zephyr
- Lab: Displaying threads information and detecting stack overflow

Ch 10. User Mode

- Memory Domains
- Syscalls

Ch 11. Traditional Multithreading Primitives

- Mutual Exclusion
- Critical Sections and Spinlocks
- Semaphores
- Polling
- Lab: The producer-consumer problem, synchronizing and avoiding concurrent access problems

Ch 12. Inter-Thread Communication

- Message Queues (MSGQ) & Queues
- Mailboxes
- Zephyr Bus (Zbus)
- Lab: Creating a print gatekeeper thread using message queue
- Lab: Synchronous communication using mailboxes

Ch 13. Interrupt Management

- Interrupts in zephyr
- Handler thread
- Workqueue
- Lab: Understanding how to wait on multiple events and interrupt safe APIs
- Lab: Understanding how to pass data using Queues from an interrupt to a thread
- Lab: Creating and submitting work items from interrupts to custom WorkQueue

Ch 14. Modules

- Creating a new module
- Module Structure
- Lab: Creating a basic module

Ch 15. Writing Kconfig Symbols

- Kconfig syntax
- Kconfig extension
- Lab: Creating and configuring a module that uses custom Kconfig options

Ch 16. Device Driver Architecture

- Device driver model
- Define and allocate devices
- Matching drivers with device trees
- Device tree bindings
- Standard and common properties
- Interrupts and devices
- Lab: Creating a driver that respects the Zephyr Device Driver Model and define devices
- Lab: Creating a driver that uses custom device tree and Kconfig

Ch 17. Power Management

- System Power Management
- Device Power Management
- Power domains
- Lab: Writing a driver compatible with power management subsystem

Closing and Evaluation Survey

- Evaluation Survey