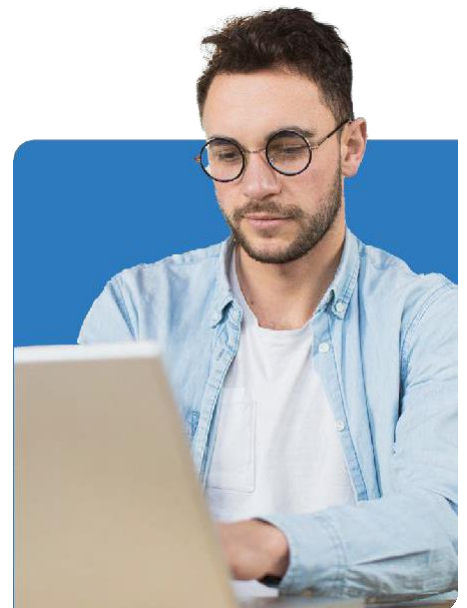


# ThriveDX™

## R-T Linux Embedded

---

# TDXAISD-106



# ThriveDX R-T Linux Embedded

## Time Commitment

4 days (total of 32 hours / 8 hours per day)

## Skill Level

Professional Level

## Course Category

AI | Software Development

This intensive hands-on course will teach you about Linux internals, kernel APIs Linux start-up process. Kernel driver structure, Daemons standard and customize, UNIX file system, UNIX process and threads structure, Inter process communication and synchronization methods and more.

## Prerequisites

The student should be an experienced C/C++ programmer, with knowledge on bash commands and working in a UNIX environment.

## Target Audience

Experienced web client developers



---

## Objectives

The objectives of this course is a comprehensive overview on Linux internals for both developers and architects, getting familiar with the variety of options available for Software architecture and design considerations. Giving tools to help answer standard questions for example: Should this be a process or a thread? Which communication mechanism is best using here? Should this code run in kernel space or user space? Etc.erequisites

## Program Structure

### Introduction Module 01

- Unix history
- Unix file system structure
- Unix kernel layers
- UNIX startup and initrd.
- Standard Daemons
- The /etc. scripts

### Basic IO Module 02

- Stream
- System calls
- File manipulation
- Links

### Signals and interrupts Module 03

- Signal handlers
- Timers
- Signal APIs
- Interrupt APIs
- Standard way to handle interrupts in user space
- The UNIX interrupt service routine



**TDX**

## **Process and threads Module 04**

- Process life cycle
- Fork
- Orphans & zombies
- Exec command
- System command
- Daemons

## **IPCS Module 05**

- ipcs
- Named pipes
- Message queues
- Sockets
- Shared memories
- Accept/read/wait APIs

## **Synchronization methods Module 06**

- Semaphores
- Semaphore calls
- Mutex

## **Debugging Linux kernel Module 07**

- Working with kernel drivers as file descriptors
- The Linux Scheduler
- Working on a working driver structure and example
- Linux procfs and sysfs



**TDX**

