

Szkozenie: The Linux Foundation
LFD450 Embedded Linux Development

FORMA SZKOLENIA	MATERIAŁY SZKOLENIOWE	CENA	CZAS TRWANIA
Stacjonarne	Tradycyjne	5800 PLN NETTO*	4 dni
Stacjonarne	Tablet CTAB	6200 PLN NETTO*	4 dni
Metoda dlearning	Tradycyjne	5800 PLN NETTO*	4 dni
Metoda dlearning	Tablet CTAB	5800 PLN NETTO*	4 dni

* (+VAT zgodnie z obowiązującą stawką w dniu wystawienia faktury)

LOKALIZACJE

Kraków - ul. Tatarska 5, II piętro, godz. 9:00 - 16:00

Warszawa - ul. Bielska 17, godz. 9:00 - 16:00

DOSTĘPNE TERMINY

2019-10-01 | 4 dni | Kraków (Promocja)

2019-10-01 | 4 dni | Kraków

2019-12-10 | 4 dni | Warszawa (Promocja)

2019-12-10 | 4 dni | Warszawa

Cel szkolenia:

Ten 4-dniowy kurs zapewni Ci wiedzę potrzebną do tworzenia embedded Linux product. Poznasz metody stosowane do adaptacji jądra Linuksa i bibliotek przestrzeni użytkownika oraz narzędzi do określonych środowisk osadzonych, takich jak te używane w elektronice użytkowej, wojsku, medycynie, przemyśle i motoryzacji.

Podczas tego kursu między innymi omawiana jest:

- Architektura jądra Linuksa, podkreślająca istotne punkty dla dostosowania jądra do niestandardowej platformy wbudowanej
- Techniki prawidłowego wymiarowania systemu w celu spełnienia ograniczeń projektu
- Mnogość zasobów dostępnych do budowy środowiska programistycznego dla projektów osadzonych.
- Dostępne opcje wypełniania bibliotek i przestrzeni użytkownika aplikacji w celu spełnienia celów i ograniczeń systemów wbudowanych.

Informacje zawarte w tym kursie będą działać z każdą główną dystrybucją Linuksa.

Plan szkolenia:

- Wstęp
 - Cele
 - Kim jesteś
 - Linux Foundation
 - Linux Foundation Training
 - Dystrybucje Linuksa
 - Platformy
 - Przygotowanie systemu
 - Zmiany w Linuksie
 - Dokumentacja i linki
 - Rejestracja
- Przygotowanie środowiska
 - Dystrybucje Linuksa
 - Instalacja maszyny wirtualnej
 - Procedury
- Jak pracować w projektach OSS **
 - Omówienie prawidłowego udziału
 - Badanie i zrozumienie DNA projektu
 - Dowiedz się, czym chcesz się zająć
 - Zidentyfikuj opiekunów i obserwuj ich pracę i metody
 - Uzyskaj wczesne wejście i pracę w otwartym środowisku
 - Contribute Incremental Bits, Not Large Code Dumps
 - Zostaw swoje ego w drzwiach: Wrzuć na luz
 - Bądź cierpliwy, rozwijaj relacje długoterminowe, bądź pomocny
- Koncepcje systemów wbudowanych i czasu rzeczywistego
 - Podstawowe koncepcje
 - Przyczyny ochrony
 - Off the Shelf (OTS)
 - Osadzone ostrzeżenia
 - Systemy operacyjne czasu rzeczywistego
 - Linux w czasie rzeczywistym
 - Niestandardowa pomoc sprzętowa
 - Zasoby
- Cross-Development Environments: cele i potrzeby

- Wprowadzenie
- Dlaczego to trudne?
- Rozważania dotyczące celu projektu
- Linki do dodatkowych dyskusji
- Laboratorium
- System Kbuild
 - Wprowadzenie
 - Kbuild Makefiles
 - Podstawy Kconfig
- Cross-Development Toolchain
 - The Compiler Triplet
 - Built-in Linux Distribution Cross Compiler
 - Linaro
 - CodeSourcery
 - crosstool-ng
 - Buildroot
 - OpenEmbedded
 - Projekt Yocto
 - Clang
 - Laboratorium
- Podstawowa konfiguracja Target Development Board
 - Cele laboratorium
 - Laboratorium
- Booting the Target Development Board z uSD
 - Cele laboratorium
 - Laboratorium
- Booting a Target Development Board over Ethernet
 - Cele laboratorium
 - Laboratorium
- Boot loaders and U-Boot
 - Etapy Boot Code
 - Przykłady GPL Boot Loaders
 - Das U-Boot
 - U-Boot Command Line
 - Środowisko U-Boot
 - Laboratorium

- Konfiguracja jądra, kompilacji i uruchamiania
 - Konfiguracja jądra dla Development Board
 - Laboratorium
- Sterowniki urządzeń **
 - Typy urządzeń
 - Węzły urządzeń
 - Character Drivers
 - Przykład
 - Laboratorium
- Device Trees
 - Co to są Device Trees?
 - Device Trees co robią a czego nie?
 - Składnia Device Tree
 - Omówienie przykładów Device Tree
 - Powiązania w Device Tree
 - Obsługa Device Tree w Boot Loaders
 - Korzystanie z danych Device Tree Data w sterownikach
 - Współistnienie i konwersja starych sterowników
 - Laboratorium
- Pakowanie docelowego systemu plików
 - Cele Osadzone systemu plików
 - Katalogi: ankieta
 - Wbudowane typy systemów plików
- Zbuduj Target Root Filesystem
 - Cele laboratorium
 - Laboratorium
- Wybór głównego systemu plików
 - SysV init vs. BusyBox init
 - udev vs. BusyBox mdev
 - Systemd
 - C Wybór biblioteki
- Konfigurowanie uClibc
 - Konfigurowanie uClibc dla NFS
 - Laboratorium
- Zbuduj BusyBox Utility Suite
 - Podstawowe czynności

- Integracja z Buildroot
- Laboratorium
- Monitorowanie jądra i debugowanie
 - Śledzenie i profilowanie
 - Ftrace, Trace-Command, Kernelshark
 - Perf
 - Korzystanie z perf
 - Sysctl
 - Klucz SysRq i wiadomości oops
 - Debugery jądra
 - Laboratorium
- Right-Sizing
 - Oft-Needed Embedded Components
 - Inwentaryzacja rozmiarów jądra
- Urządzenia technologii pamięci (systemy plików pamięci flash)
 - Czym są urządzenia MTD?
 - NAND vs. NOR vs. eMMC
 - Moduły sterownika i użytkownika
 - Systemy plików Flash
 - Laboratorium
- Skompresowane systemy plików
 - SquashFS
 - Wdrażanie w partycji MTD
 - Laboratorium
- Aktualizacje systemu
 - Kiedy należy zaktualizować?
 - Strategie aktualizacji
 - Wbudowane systemy aktualizacji
 - Laboratorium
- Real-Time Extensions
 - Przewidywalność i zapobieganie oraz blokady
 - Projekt PREEMPT_RT
 - Lista kontrolna w czasie rzeczywistym
- Zakończenie i ankieta
 - Ankieta ewaluacyjna

** Te sekcje mogą być uznane za częściowe lub w całości jako opcjonalne. Zawierają materiały

źródłowe, tematy specjalistyczne lub przedmioty zaawansowane. Instruktor może zdecydować się na ich realizację lub nie, w zależności od doświadczenia uczestników i ograniczeń czasowych.

Wymagania:

Kurs jest przeznaczony przede wszystkim dla doświadczonych programistów, programistów i inżynierów, którzy są zainteresowani nauką adaptacji Linuksa do systemu wbudowanego. Uczestnicy powinni być znać podstawowe narzędzia Linuksa, znać język programowania C i czuć się komfortowo w środowisku Linux lub UNIX.

Poziom trudności



Certyfikaty:

Uczestnicy otrzymają certyfikat The Linux Foundation.

Prowadzący:

Certyfikowany trener The Linux Foundation.