

Szkolenie: The Linux Foundation LFD430 Developing Linux Device Drivers



DOSTĘPNE TERMINY

2021-05-17 | 4 dni | Virtual Classroom

2021-05-17 | 4 dni | Wirtualna sala

Cel szkolenia:

Ten 4-dniowy kurs nauczy Cię, jak tworzyć sterowniki urządzeń dla systemów Linux, oparte na podstawowej znajomości jądra Linuxa. Dowiesz się również o różnych typach sterowników urządzeń Linuxowych, a także o odpowiednich interfejsach API i metodach, za pomocą których jądro współpracuje z urządzeniami.

Podczas tego kursu poznasz między innymi:

- Różne rodzaje sterowników urządzeń używanych w Linuxie
- Odpowiednie interfejsy API, za pomocą których urządzenia (zarówno sprzętu i oprogramowania) współpracują z jądrem
- Niezbędne moduły i techniki opracowywania i debugowania sterowników Linuxa

Metody i informacje zawarte w tym kursie będą działały z każdą znaną dystrybucją Linuxa.

Plan szkolenia:

- Wprowadzenie
 - Cele
 - Kim jesteś
 - The Linux Foundation
 - Linux Foundation Training
 - Dystrybucje Linuxa
 - Platformy
 - Przygotowywanie systemu
 - Pobieranie i używanie maszyny wirtualnej
 - Rzeczy zmieniają się w Linuxie
 - Dokumentacja i linki
 - Rejestracja kursu

- Praktyki wstępne
 - Procedury
 - Wersje jądra
 - Źródła jądra i korzystanie z Git
 - Wdrażanie własnego jądra
 - Sprzęt
 - Drzewo przemieszczania
- Jak pracować w projektach OSS * *
 - Omówienie sposobu prawidłowego udziału
 - Badanie i zrozumienie DNA projektu
 - Dowiedz się czym chcesz się zajmować
 - Identyfikacja opiekunów projektów i ich przepływów pracy i metod
 - Uzyskanie wczesnego wejścia i pracy w otwartym środowisku
 - Przekazuj przyrostowe bity, a nie duże zrzuty kodu
 - Zostaw swoje ego za drzwiami
 - Bądź cierpliwy, rozwijaj relacje długoterminowe, bądź pomocny
 - Laboratoria
- Sterowniki urządzeń
 - Typy urządzeń
 - Mechanizm vs. Polityka
 - Unikanie binarne obiekty blob
 - Zarządzanie zasilaniem
 - Jak aplikacje używają sterowników urządzeń
 - Wprowadzenie do połączeń systemowych z dostępem do urządzeń
 - Numery błędów
 - Printk()
 - devres: zarządzane zasobami urządzeń
 - Laboratoria
- Moduły i sterowniki urządzeń
 - Makra modułu driver ()
 - Moduły i hot plug
 - Laboratoria
- Zarządzanie pamięcią i alokacja
 - Pamięć wirtualna i fizyczna
 - Strefy pamięci
 - Tabele stron

- kmalloc ()
- __get_free_pages ()
- vmalloc ()
- Slabs i alokacje pamięci podręcznej
- Laboratoria
- Urządzenia znakowe
 - Węzły urządzenia
 - Numery główne i pomocnicze
 - Rezerwowanie numerów głównych/pomocniczych
 - Uzyskiwanie dostępu do węzła urządzenia
 - Rejestrowanie urządzenia
 - Udev
 - dev_printk ()
 - file_operations struktura
 - Punkty wejścia kierowcy
 - Struktury pliku i węzła
 - Różne sterowniki znaków
 - Laboratoria
- Funkcje jądra
 - Komponenty jądra
 - Przestrzeń użytkownika a kernel-Space
 - Co to są wywołania systemowe?
 - Dostępne wywołania systemowe
 - Planowanie algorytmów i struktur zadań
 - Kontekst procesu
 - Laboratoria
- Przenoszenie między użytkownikiem a przestrzenią jądra
 - Przenoszenie między przestrzeniami
 - put(get)_user() i copy_to(from)_user()
 - Bezpośredni transfer: I/O jądra i mapowanie pamięci
 - I/O Jądra
 - Mapowanie stron użytkownika
 - Mapowanie pamięci
 - Funkcje przestrzeni użytkownika dla mmap()
 - Punkt wejścia sterownika dla mmap()
 - Uzyskiwanie dostępu do plików z jądra

- Laboratoria
- Przerwania i wyjątki
 - Czym są przerwania i wyjątki?
 - Wyjątki
 - Przerwania asynchroniczne
 - MSI
 - Włączanie/wyłączanie przerw
 - Czego nie można zrobić w czasie przerwania
 - Struktury danych IRQ
 - Instalowanie obsługi przerw
 - Laboratoria
- Pomiary czasu
 - Rodzaje pomiarów czasu
 - Jiffies
 - Uzyskiwanie bieżącego czasu
 - Źródła zegara
 - Zegar czasu rzeczywistego
 - Programowalny licznik interwałów
 - Licznik sygnatury czasowej
 - HPET
 - Tickless
 - Laboratoria
- Czasomierze jądra
 - Wstawianie opóźnień
 - Co to są czasomierze jądra?
 - Funkcje czasomierza o niskiej rozdzielczości
 - Implementacja czasomierza o niskiej rozdzielczości
 - Przekazniki czasowe o wysokiej rozdzielczości
 - Korzystanie z czasomierzy o wysokiej rozdzielczości
 - Laboratoria
- ioctls
 - Czym są ioctls?
 - Punkt wejścia sterownika dla polecenia funkcji ioctl
 - Definiowanie polecenia funkcji ioctl
 - Laboratoria
- Ujednolicony model urządzenia i sysfs

- Ujednolicony model urządzenia
- Podstawowe konstrukcje
- Urządzenia
- sysfs
- kset i kobject
- Laboratoria
- Firmware
 - Co to jest firmware?
 - Ładowanie oprogramowania firmware
 - Laboratoria
- Kolejki snu i oczekiwania
 - Co to są kolejki oczekiwania?
 - Usypianie i budzenie
 - Szczegóły dotyczące usypiania
 - Szczegóły budzenia
 - Sondowania
 - Laboratoria
- Obsługa przerw: funkcje deferrable i sterowniki użytkownika
 - Górna i dolna części
 - Softirqs
 - Tasklets
 - Kolejki robocze
 - Nowy interfejs API kolejki pracy
 - Tworzenie wątków jądra
 - Programy obsługi przerw gwintowanych
 - Przerwanie obsługi w przestrzeni użytkownika
 - Laboratoria
- Sprzęt I/O
 - Busy i porty
 - Bariery pamięci
 - Rejestrowanie portów I/O
 - Odczytywanie i zapisywanie danych z rejestrów I/O
 - Przydzielanie i mapowanie pamięci I/O
 - Uzyskiwanie dostępu do pamięci I/O
 - Dostęp użytkownika – ioperm (), iopl (), /dev/port
 - Laboratoria

- PCI
 - Co to jest PCI?
 - Sterowniki urządzeń PCI
 - Lokalizowanie urządzeń PCI
 - Uzyskiwanie dostępu do przestrzeni konfiguracji
 - Uzyskiwanie dostępu do funkcji i/O i przestrzeni pamięci
 - Złącze PCI Express
 - Laboratoria
- Sterowniki platformy * *
 - Czym są sterowniki platformy?
 - Główne struktury danych
 - Rejestrowanie urządzeń platformy
 - Przykład
 - Dane platformy na stałe
 - Nowy sposób: drzewa urządzeń
 - Laboratoria
- Bezpośredni dostęp do pamięci (DMA)
 - Co to jest DMA?
 - DMA bezpośrednio do użytkownika
 - DMA i przerwania
 - Ograniczenia pamięci DMA
 - Maski DMA
 - API DMA
 - Zasoby DMA
 - Mapowanie rozproszenia/zbierania
 - Laboratoria
- Sterowniki sieciowe I: podstawy
 - Warstwy sieciowe i hermetyzacja danych
 - Warstwa DATALINK
 - Sterowniki urządzeń sieciowych
 - Załadunek/rozładunek
 - Otwieranie i zamykanie
 - Laboratoria
- Sterowniki sieciowe II: struktury danych
 - net_device struktura
 - net_device_ops struktura

- sk_buff struktura
- Funkcje bufora gniazd
- netdev_printk () i powiązania
- Laboratoria
- Sterowniki sieciowe III: Transmisja i odbiór
 - Przesyłanie danych i limity czasu
 - Odbieranie danych
 - Statystyki
 - Laboratoria
- Sterowniki sieciowe IV: wybrane tematy
 - Multicast * *
 - Zmiany w stanie łącza
 - ioctl
 - NAPI i łagodzenia przerw
 - Szczegóły NAPI
 - TSO i TOE
 - MII i ethtool * *
- Sterowniki USB
 - Czym jest USB?
 - Topologia USB
 - Terminologie
 - Punkty końcowe
 - Deskryptory
 - Klasy urządzeń USB
 - Obsługa USB w Linuxie
 - Rejestrowanie sterowników urządzeń USB
 - Przenoszenie danych
 - Przykład sterownika USB
 - Laboratoria
- Zarządzanie zasilaniem
 - Zarządzanie zasilaniem
 - ACPI i APM
 - Stany zasilania systemu
 - Funkcje wywołania zwrotnego
 - Laboratoria
- Sterowniki blokowe

- Czym są sterowniki blokowe?
- Buforowanie
- Rejestrowanie sterownika bloku
- gendisk struktury
- Obsługa żądań
- Laboratoria
- Badanie końcowe i zakończenie
 - Ankieta ewaluacyjna

** Sekcje te mogą być brane pod uwagę w części lub w całości jako opcjonalne. Zawierają one zarówno materiały referencyjne tła, tematy specjalistyczne, jak i tematy zaawansowane. Instruktor może zdecydować się na ich omówienie lub nie w zależności od doświadczenia w klasie i ograniczeń czasowych.

Wymagania:

Znajomość podstawowych interfejsów jądra i metod, takich jak pisanie, kompilowanie, wczytywanie i zwalnianie modułów, używanie elementów podstawowych synchronizacji oraz podstawy alokacji i zarządzania pamięcią, zagadnienia te są omówione w kursie [LFD420 Linux Kernel Internals and Development](#).

Poziom trudności



Certyfikaty:

Uczestnicy uzyskają certyfikat podpisany przez The Linux Foundation.

Prowadzący:

Certyfikowany trener The Linux Foundation.