

Szkolenie: The Linux Foundation
LFD401 Developing Applications For Linux



DOSTĘPNE TERMINY

2021-05-10 | 4 dni | Virtual Classroom

2021-05-10 | 4 dni | Wirtualna sala

Cel szkolenia:

Ten czterodniowy kurs ma na celu pomóc doświadczonym programistom w szybkim przygotowaniu aplikacji w środowisku Linux. Podczas tego kursu zdobędziesz praktyczne doświadczenie z niezbędnymi narzędziami i metodami tworzenia aplikacji dla systemu Linux oraz poznasz funkcje i techniki unikalne dla systemu Linux.

Podczas tego kursu:

- Poznasz narzędzia i metody tworzenia programów w C i programowania systemów pod Linuxa.
- Nauczysz się techniki debugowania i zarządzania procesami.
- Poznasz wywołania systemowe specyficzne dla systemu Linux.
- I więcej.

Informacje zawarte w tym kursie będą kompatybilne z każdą główną dystrybucją Linux.

Plan szkolenia:

- Wprowadzenie
 - Cele
 - Kim jesteś
 - Linux Foundation
 - Linux Foundation Training
 - Dystrybucje Linux
 - Platformy
 - Przygotowanie system
 - Pobieranie i używanie maszyny wirtualnej
 - Rzeczy zmieniają się w Linux
 - Rejestracja kursu
- Czynności wstępne

- Procedury
- Standardy i LSB
- Jak pracować w projektach OSS **
 - Omówienie wniesienia prawidłowego udziału
 - Pozostań blisko głównej linii bezpieczeństwa i jakości
 - Badanie i zrozumienie DNA projektu
 - Określ czym chcesz się zająć
 - Zidentyfikowanie osób odpowiedzialnych, ich przepływu pracy i metod
 - Uzyskaj wczesne wejście i pracę w otwartym środowisku
 - Przekazuj przyrostowe bity, zamiast dużych fragmentów kodu
 - Zostaw swoje ego przed drzwiami: nie bądź wrażliwy
 - Bądź cierpliwy, rozwijaj relacje długoterminowe, bądź pomocny
- Kompilatory
 - GCC
 - Inne kompilatory
 - Główne opcje GCC
 - Preprocesor
 - Zintegrowane środowiska programistyczne (IDE)
 - Laboratorium
- Biblioteki
 - Biblioteki statyczne
 - Wspólne biblioteki
 - Łączenie z bibliotekami
 - Dynamiczny program ładujący
 - Laboratorium
- Make
 - Używanie Make i Makefile
 - Budowanie dużych projektów
 - Bardziej skomplikowane zasady
 - Wbudowane reguły
 - Laboratorium
- Kontrola źródła
 - Kontrola źródła
 - RCS i CVS
 - Subversion
 - Git

- Laboratorium
- Debugowanie i Core Dumps
 - Gdb
 - Co to jest Core Dump Files?
 - Produkcja Core Dumps
 - Badanie Core Dumps
 - Laboratorium
- Narzędzia debugowania
 - Electric Fence
 - Zdobywanie czasu
 - Profilowanie i wydajność
 - Valgrind
 - Laboratorium
- Wywołania systemowe
 - Wywołania systemowe, a funkcje biblioteki
 - Jak wykonywane są wywołania systemowe
 - Wartości zwrotne i numery błędów
 - Laboratorium
- Zarządzanie pamięcią i alokacją
 - Zarządzanie pamięcią
 - Dynamiczna alokacja
 - Tuning malloc()
 - Blokowanie stron
 - Laboratorium
- Pliki i systemy plików w Linux **
 - Pliki, katalogi i urządzenia
 - Wirtualny system plików
 - System plików ext2/ext3
 - Systemy plików dziennika
 - System plików ext4/
 - Laboratorium
- File I/O
 - UNIX File I/O
 - Otwieranie i zamykanie
 - Czytanie, pisanie i szukanie
 - Pozycyjne i wektorowe I/O

- Standardowa biblioteka I/O
- Obsługa dużych plików (LFS)
- Laboratorium
- Zaawansowane operacje na plikach
 - Funkcje statystyczne
 - Funkcje katalogu
 - Inotify
 - Mapowanie pamięci
 - Flock() i fcntl()
 - Tworzenie plików tymczasowych
 - Inne wywołania systemowe
 - Laboratorium
- Procesy - I
 - Co to jest proces?
 - Limity procesowe
 - Grupy procesowe
 - System plików proc
 - Metody komunikacji między procesami
 - Laboratorium
- Procesy - II
 - Używanie system() do tworzenia procesu
 - Używanie fork() do tworzenia procesu
 - Używanie exec() do utworzenia procesu
 - Korzystanie z polecenia clone()
 - Wyjście
 - Konstruktorzy i destruktory
 - Oczekiwanie
 - Procesy demon
 - Laboratorium
- Pipes i Fifos
 - Pipes i komunikacja między procesami
 - Popen() i pclose()
 - Pipe()
 - Nazwane powłoki (FIFO)
 - Splice(), vmsplice() i tee()
 - Laboratorium

- Asynchroniczne I/O **
 - Co to jest Asynchroniczne I/O?
 - API Asynchronicznego I/O POSIX
 - Implementacja Linux
 - Laboratorium
- Sygnały - I
 - Co to są sygnały?
 - Dostępne sygnały
 - Wysyłanie sygnałów
 - Alarmy, pauzy i uśpienie
 - Konfiguracja Signal Handler
 - Zestawy sygnałów
 - Sigaction()
 - Laboratorium
- Sygnały - II
 - Reentrancy i Signal Handler
 - Skoki i zwroty nielokalne
 - Siginfo i sigqueue()
 - Sygnały w czasie rzeczywistym
 - Laboratorium
- Wątki POSIX - I
 - Wielowątkowość w Linux
 - Podstawowa struktura programu
 - Tworzenie i niszczenie wątków
 - Sygnały i wątki
 - Forking a Threading
 - Laboratorium
- Wątki POSIX - II
 - Deadlocks i Race Conditions
 - Operacje Mutex
 - Semafony
 - Futexy
 - Operacje warunkowe
 - Laboratorium
- Sieci i gniazda
 - Warstwy sieciowe

- Co to są gniazda?
- Gniazda strumieniowe
- Gniazda datagramowe
- Surowe gniazda
- Kolejność bajtów
- Laboratorium
- Gniazda - adresy i hosty
 - Struktura adresów gniazd
 - Konwersja adresów IP
 - Informacje o gospodarzu
 - Laboratorium
- Gniazda - porty i protokoły
 - Informacje o porcie usług
 - Informacje o protokole
 - Laboratorium
- Gniazda - klienci
 - Podstawowa sekwencja klientów
 - Socket()
 - Connect()
 - Close() i shutdown()
 - Klient UNIX
 - Klient internetowy
 - Laboratorium
- Gniazda - serwery
 - Podstawowa sekwencja serwerów
 - Bind()
 - Listen()
 - Accept()
 - Serwer UNIX
 - Serwer internetowy
 - Laboratorium
- Gniazda - operacje wejścia/wyjścia
 - Write(), read()
 - Send(), recv()
 - Sendto(), recvfrom()
 - Sendmsg(), recvmsg()

- Sendfile()
- Socketpair()
- Laboratorium
- Gniazda – opcje
 - Pobieranie i ustawianie opcji gniazd
 - Fcntl()
 - Lcntl()
 - Getsockopt() i setsockopt()
 - Laboratorium
- Gniazda Netlink **
 - Co to są gniazda Netlink?
 - Otwieranie gniazd Netlink
 - Wiadomości Netlink
 - Laboratorium
- Gniazda – multipleksowanie i serwery współbieżne
 - Gniazda multipleksowe i asynchroniczne I/O
 - Select()
 - Poll()
 - Pselect() i ppoll()
 - Epoll
 - Sygnałowe i asynchroniczne I/O
 - Serwery współbieżne
 - Laboratorium
- Komunikacja między procesami
 - Metody IPC
 - POSIX IPC
 - System V IPC **
 - Laboratorium
- Pamięć współdzielona
 - Co to jest pamięć współdzielona?
 - Pamięć współdzielona POSIX
 - Pamięć współdzielona System V **
 - Laboratorium
- Semafore
 - Co to jest semafor?
 - Semafore POSIX

- Semafore systemowe V **
- Laboratorium
- Kolejki komunikatów
 - Czym są kolejki komunikatów?
 - Kolejki komunikatów POSIX
 - Kolejki komunikatów systemu V **
 - Laboratorium
- Ankieta zamknięcia i oceny

** Te sekcje mogą być uznane za częściowo lub w całości opcjonalne. Zawierają materiały źródłowe, tematy specjalistyczne lub przedmioty zaawansowane. Instruktor może zdecydować się na ich omówienie lub nie, w zależności od doświadczenia grupy i ograniczeń czasowych.

Wymagania:

Ten kurs przeznaczony jest dla doświadczonych programistów. Uczestnicy powinni być biegli w programowaniu w języku C i zaznajomieni z podstawowymi narzędziami Linux oraz edytorami tekstu.

Poziom trudności



Certyfikaty:

Uczestnicy otrzymają certyfikaty podpisane przez The Linux Foundation.

Prowadzący:

Certyfikowany trener The Linux Foundation.