

## Описание курса «Основы сетевого программирования»

### Целевая аудитория

Курс «Основы сетевого программирования» предназначен для специалистов предприятий, занимающихся разработкой и тестированием программного обеспечения, студентов направлений 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии», 09.03.03 «Прикладная информатика», 09.03.04 «Программная инженерия», 10.03.01 «Информационная безопасность», 02.03.03 «Математическое обеспечение и администрирование информационных систем», а также всех, кто интересуется современными компьютерными технологиями.

### Предварительная подготовка

При отсутствии базовой подготовки рекомендуется прохождение курса «Основы сетевых технологий. Часть 1: Передача и коммутация данных в компьютерных сетях». Также для прохождения курса требуется знание основ программирования на языке С (предпочтительно) или на других языках программирования.

### Сертификаты

После прохождения курса, слушатель может сдать сертификационный экзамен в [авторизованном учебном центре](#) или [ближайшем офисе](#) и получить сертификат D-Link . Экзамен состоит из теста на портале дистанционного обучения D-Link.

### Описание курса

Длительность курса – 32 академических часа. Курс включает лекционную и практическую части.

Целью курса является приобретение знаний и навыков в области разработки прикладного сетевого программного обеспечения. Курс может использоваться как независимый или часть большого курса в средних специальных, профессиональных и высших образовательных учреждениях.

После прохождения курса слушатели смогут:

- получить базовые знания о сетевых технологиях;
- понимать основы передачи данных в компьютерных сетях;
- получить знания о использовании сетевых сокетов при программировании;
- разрабатывать клиентские и серверные программы, обменивающиеся данными потоковым способом;
- разрабатывать клиентские и серверные программы, обменивающиеся данными дейтаграммным способом;

- разрабатывать программы обменивающиеся данными по сети с использованием неблокирующих операций ввода/вывода;
- разрабатывать программы обменивающиеся данными по сети с использованием мультиплексирования операций ввода/вывода;
- разрабатывать программы, выполняющие передачу в компьютерной сети широковещательных сообщений;
- разрабатывать программы, реализующие функцию передачи данных в компьютерной сети на языке программирования C;
- разрабатывать программы, реализующие функцию передачи данных в компьютерной сети на языке программирования Java;
- разрабатывать программы, реализующие функцию передачи данных в компьютерной сети на языке программирования C#;
- разрабатывать программы, реализующие функцию передачи данных в компьютерной сети на языке программирования Python;
- понимать основы работы протоколов защищенной передачи данных SSL/TLS;
- разрабатывать программы, выполняющие защищенную передачу данных в компьютерной сети с использованием протоколов SSL/TLS;
- использовать протокол SSH для удаленной настройки сетевого компьютера;
- использовать протокол SSH для организации безопасной передачи по сети потоков данных;
- отлаживать сетевые приложения на локальном компьютере;
- разрабатывать программы, выполняющие передачу данных через виртуальные сетевые интерфейсы в Linux.

## **Оборудование**

Для проведения лабораторных работ по курсу требуется следующий комплект оборудования:

- для каждого слушателя 1 компьютер с операционной системой Ubuntu Linux.

## **Содержание курса**

### **1. Краткое введение в компьютерные сети**

- 1.1. Понятие компьютерной сети. Модель OSI
- 1.2. Типы адресов в компьютерных сетях
- 1.3. Физические адреса
- 1.4. Сетевые адреса
- 1.5. Доменные имена
- 1.6. Номера портов
- 1.7. Базовая настройка сетевых параметров в Linux

### **2. Сетевые сокеты**

- 2.1. Понятие сетевого сокета
- 2.2. Поточковые и дейтаграммные сокеты

- 2.3. Клиентские и серверные сокеты
- 2.4. Интернет-сокеты и другие виды сокетов
- 2.5. Пример программы с сетевыми сокетами

### **3. Поточковые сокеты**

- 3.1. Основные системные вызовы клиентских и серверных поточковых сокетов
- 3.2. Системный вызов `socket()`
- 3.3. Системный вызов `connect()`
- 3.4. Кодирование сетевых адресов
- 3.5. Системный вызов `bind()`
- 3.6. Системный вызов `listen()`
- 3.7. Системный вызов `accept()`
- 3.8. Системные вызовы `send()` и `recv()`
- 3.9. Системный вызов `close()`
- 3.10. Пример программирования поточковых клиента и сервера

### **4. Дейтаграммные сокеты. Дополнительные возможности сокетов**

- 4.1. Основные системные вызовы дейтаграммных сокетов
- 4.2. Системные вызовы `sendto()` и `recvfrom()`
- 4.3. Пример программирования дейтаграммных клиента и сервера
- 4.4. Неблокирующий ввод-вывод в сетевых сокетах
- 4.5. Мультиплексирование ввода-вывода в сетевых сокетах
- 4.5. Параметры сокетов. Системные вызовы `getsockopt()` и `setsockopt()`
- 4.6. Пример передачи широковещательных сообщений

### **5. Реализация сокетов в других операционных системах и языках программирования**

- 5.1. Реализация сокетов в Windows
- 5.2. Реализация сокетов в Java
- 5.3. Реализация сокетов в C#
- 5.4. Реализация сокетов в Python

### **6. Программирование SSL/TLS сокетов**

- 6.1. Общие сведения о протоколах SSL/TLS и библиотеке OpenSSL
- 6.2. Описание протокола SSL/TLS
- 6.3. Установка и начальная настройка библиотеки OpenSSL
- 6.4. Создание сертификата сервера
- 6.5. Основные программные объекты библиотеки OpenSSL
- 6.6. Передача данных через защищенное соединение с помощью библиотеки OpenSSL
- 6.7. Пример программирования SSL/TLS сокетов

### **7. Протокол SSH**

- SSH
  - 7.1. Назначение и основные возможности протокола SSH. Клиенты и серверы
  - 7.2. Основы работы с SSH-клиентом
  - 7.3. Настройка аутентификации клиента по публичному ключу

- 7.4. Настройка клиента и сервера OpenSSH
- 7.5. Перенаправление портов SSH

## **8. Виртуальные сетевые интерфейсы в Linux**

- 8.1. Виртуальные сетевые интерфейсы в Linux
- 8.2. tun/tap интерфейсы
- 8.3. dummy интерфейсы

## **Лабораторные работы**

Лабораторная работа №1. **Введение в программирование потоковых сокетов**

Лабораторная работа №2. **Программирование потоковых сокетов**

Лабораторная работа №3. **Программирование дейтаграммных сокетов**

Лабораторная работа №4. **Программирование TLS-сокетов**

Лабораторная работа №5. **Использование протокола SSH для удаленной настройки и организации безопасной передачи данных**