

ЧАСТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«Учебный Центр РТСофт»
ЧУ ДПО «УЦ РТСофт»

УЧЕБНЫЙ ПЛАН

дополнительного профессионального образования

«Разработка драйверов устройств для Linux»

Цель: получение теоретических и практических навыков по разработке драйверов для операционной системы Linux.

Категория слушателей: с высшим и средне-техническим образованием, инженеры программисты и разработчики встраиваемых систем.

Форма обучения: очная.

Форма контроля: зачет по результатам практических занятий и тестирования.

Продолжительность обучения: 40 часов.

Режим занятий: 8 академических часов в день.

Срок обучения: по договоренности с заказчиком.

Выдаваемый документ: «Удостоверение о повышении квалификации»

| № | Наименование разделов и дисциплин | Всего час | В том числе | | Форма контроля |
|-----|--|-----------|-------------|----------------------|----------------|
| | | | Лекции | Практические занятия | |
| 1. | Архитектура ядра Linux. | 1 | 1 | - | - |
| 2. | Исходный код и компиляция ядра. | 2 | 1 | 1 | - |
| 3. | Программирование на уровне ядра. Модули ядра. | 2 | 1 | 1 | зачет |
| 4. | Типы драйверов и символьный драйвер | 2 | 1 | 1 | зачет |
| 5. | Символьный драйвер. Read, Write | 2 | 1 | 1 | зачет |
| 6. | Файловая система proc | 2 | 1 | 1 | зачет |
| 7. | Потоки и методы синхронизации ядра | 2 | 1 | 1 | зачет |
| 8. | Реализация блокирующих операций | 2 | 1 | 1 | зачет |
| 9. | Символьный драйвер. ioctl | 2 | 1 | 1 | зачет |
| 10. | Операции ввода-вывода | 2 | 1 | 1 | зачет |
| 11. | Прерывания и обработчики прерываний | 2 | 1 | 1 | зачет |
| 12. | Модель драйверов linux | 1 | 1 | - | зачет |
| 13. | PCI драйвера | 1 | 1 | - | зачет |
| 14. | Символьный драйвер. mmap | 2 | 1 | 1 | зачет |
| 15. | Сетевой драйвер в Linux | 4 | 1 | 3 | зачет |
| 16. | Использование sysfs | 2 | 1 | 1 | зачет |
| 17. | Работа с dma | 4 | 1 | 3 | зачет |
| 18. | Подсистема usb и usb драйверы | 2 | 1 | 1 | зачет |
| 19. | Особенности драйвера для не x86 архитектур | 1 | 1 | 1 | - |
| 20. | Особенности драйверов для систем реального времени | 1 | 1 | - | - |
| | Итого | 40 | 20 | 20 | |

ЧАСТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«Учебный Центр РТСофт»
ЧУ ДПО «УЦ РТСофт»

УЧЕБНАЯ ПРОГРАММА
дополнительного профессионального образования
«Разработка драйверов устройств для Linux»

- Тема 1. Обзор архитектуры Linux и процесса разработки открытых систем.** Архитектура Linux. Монолитное и микроядро. Типы драйверов. Процесс разработки linux. Взаимодействие с linux сообществом при разработке компонентов ядра.
- Тема 2. Исходный код и компиляция ядра Linux.** Исходный код ядра Linux. Основная ветка ядра kernel.org. Обзор исходного кода ядра. Процесс компиляции ядра. Патчи. Работа с git. Наложение патчей, задание конфигурации ядра, компиляция и установка ядра linux.
- Тема 3. Модули ядра linux.** Модули ядра linux. Обзор программирования на уровне ядра. Взаимодействие между моделями ядра. Разработка модуля ядра.
- Тема 4. Типы драйверов и символьный драйвер.** Типы драйверов в Linux. Символьный драйвер. Ключевые структуры данных. Использование символьного драйвера из пользовательских программ. Разработка модуля ядра для символьного драйвера и использующей его пользовательской программы.
- Тема 5. Символьный драйвер. Read, Write.** Символьный драйвер, функции read, write. Использование символьного драйвера из пользовательских программ. Приватные данные в символьных драйверах. Разработка символьного драйвера и использующей его пользовательской программы.
- Тема 6. Файловая система proc.** Файловая система proc. Назначение и ключевые структуры данных. Использование proc в драйверах. Разработка модуля ядра использующего интерфейс /proc
- Тема 7. Потоки и методы синхронизации ядра.** Потоки ядра, SoftIRQ, Tasklets. Методы синхронизации в ядре. Разработка модуля ядра для иллюстрации использования потоков ядра и методов синхронизации.
- Тема 8. Реализация блокирующих операций.** Реализация блокирующих операций. Очереди ожидания. Разработка символьного драйвера с поддержкой блокирующих и неблокирующих операций.
- Тема 9. Символьный драйвер. Ioctl.** Назначение ioctl. Символьный драйвер, функции ioctl. Эволюция вызова ioctl в процессе развития ядра Linux. Использование символьного драйвера из пользовательских программ. Разработка символьного драйвера с функцией ioctl и использующей его пользовательской программы.

- Тема 10. Операции ввода-вывода.** .Операции ввода вывода с использованием портов и памяти. Разработка модуля ядра использующего операции ввода-вывода для портов и памяти.
- Тема 11. Прерывания и обработчики прерываний.** .Обработка прерываний в Linux. Разработка и установка обработчика прерываний в драйвере linux. Контекст прерываний. Ограничения на операции, допустимые в обработчиках прерываний. .Разработка драйвера с обработчиком прерываний.
- Тема 12. Модель драйверов Linux.** .Модель драйверов Linux. Ключевые структуры данных. Отображение модели драйверов через sysfs.
- Тема 13. PCI драйвер.** .Обзор шины PCI. PCI драйвер. Связь pci драйвера с моделью драйверов Linux. Ключевые структуры данных. MSI прерывания. .Разработка драйвера PCI драйвера.
- Тема 14. Символьный драйвер. Mmap.** .Информация по системе управлению памятью. Назначение mmap. Реализация mmap в ядре. Символьный драйвер с функцией mmap. Использование mmap из пользовательских программ. .Разработка символьного драйвера с функцией mmap и использующей его пользовательской программы.
- Тема 15. Сетевой драйвер.** .Сетевая подсистема Linux. Сетевой драйвер. Ключевые структуры данных. Методы повышения производительности сетевого драйвера. .Разработка сетевого драйвера драйвера.
- Тема 16. Файловая система sysfs.** .Файловая система sysfs. Назначение sysfs. Использование sysfs в драйверах. .Разработка драйвера с интерфейсом sysfs.
- Тема 17. Работа с DMA.** .Назначение прямого доступа к памяти. Поточковые и когерентный DMA. API для работы с DMA в linux. Примеры работы с dma драйверах Linux. .Разработка драйвера с dma.
- Тема 18. Подсистема USB и драйверы USB.** .Основные понятия связанные со стандартом USB. Обзор USB подсистемы. Типы usb драйверов. Ключевые структуры данных. Драйвер для usb устройства. .Разработка usb драйвера.
- Тема 19. Включение драйверов в ядро и драйверы для не x86 архитектур.** Разработка переносимых драйверов. Шина platform. Встраивание драйверов в ядро. .Встраивание драйвера в систему сборки Linux.
- Тема 20. Особенности драйверов для систем реального времени.** .Обзор rtpatch. Особенности разработки драйверов для работы с rtpatch.