Санкт-Петербургский политехнический университет

Институт Компьютерных Наук и Технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа по дисциплине «Проектирование ОС и компонентов» на тему:

Разработка драйверов ОС Linux. Драйвер сетевой карты.

Выполнил: студент группы № 63501/3

Дедков С.В.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Душутина Е.В.

Санкт-Петербург

2016

# Оглавление

[Оглавление 2](#_Toc445736198)

[Цель работы 3](#_Toc445736199)

[Введение 4](#_Toc445736200)

# Цель работы

* Изучить общие принципы разработки драйверов ОС Linux. Написать тестовый модуль – каркас, для последующей разработки драйвера.
* Изучить принципы написания драйверов символьных устройств. Написать драйвер, для общего понимания работы драйверов в ОС Linux.
* Изучить принципы разработки драйверов для PCI устройств. Разработать драйвер сетевой карты.

# Введение

Драйвер - программное обеспечение, с помощью которого другое программное обеспечение (операционная система) получает доступ к аппаратному обеспечению некоторого устройства. (Драйвер, 2016)

Таким образом драйвер – программное обеспечение, которое реализует программный интерфейс аппаратуры, скрывая детали работы устройства. Таким образом, пользователю не приходиться вдаваться в детали аппаратных средств, достаточно выполнять стандартизированные вызовы, которые не зависят от специфики драйвера. Задача драйвера – перевести данные вызовы в специфичный для данного устройства операции.

В ОС Linux драйверы могут быть собраны отдельно от ядра и подгружены в процессе работы, когда это необходимо. Для этой цели используются LKM (Loadable Kernel Module) – загружаемые модули. (Jonathan Corbet, 2005)

В ОС Linux устройства разделяются на 3 типа:

* Символьные
* Блочные
* Сетевые

В данной работе рассматриваются драйвера для:

* символьного устройства – для понимания общих принципов разработки драйверов
* сетевого устройства – из задания к лабораторной работе. Используемая сетевая карта – TP-Link TF-3200. Версия ядра Linux – 2.6.38.8.

В ходе данной работы использовались Ubuntu 14.04.4 LTS и 11.04, которые основаны на Debian GNU/Linux. Соответственно использовался пакетный менеджер APT(Advanced Packaging Tool).

К примерам кода, приводимого в работе, указываются файлы листингов, а также версия ядра на котором модуль запускался.

Полные листинги находятся в репозитории по ссылке - <https://github.com/dsvgit/os_lab_syscalls>.

# Разработка каркаса драйвера – загружаемый модуль

Загружаемый модуль ядра (loadable kernel module, LKM) - в информатике - объектный файл, содержащий код, расширяющий возможности ядра операционной системы. Модули используются, чтобы добавить поддержку нового оборудования или файловых систем или для добавления новых системных вызовов. Когда функциональность, предоставляемая модулем, больше не требуется, он может быть выгружен, чтобы освободить память и другие ресурсы. (Загружаемый модуль ядра, 2016)

Модули ядра могут быть собраны и подгружены/выгружены во время работы ОС.

Реализация загрузки/выгрузки модуля ядра похожа на конструктор/деструктор объекта в терминах объектно-ориентированного программирования. Иными словами – модуль ядра должен реализовывать функцию загрузки и функцию выгрузки (что не всегда обязательно, например, для целей отладки). Функция регистрации модуля просто подготавливает работу модуля, для дальнейшей работы (регистрирует функции для последующего вызова, когда это будет необходимо).

Важно понимать, что код модуля ядра выполняется в пространстве ядра, привилегированном режиме работы процессора. Переход в этот режим осуществляется посредством системных вызовов (Jonathan Corbet, 2005).

Для работы с модулями потребуются заголовочные файлы текущей версии ядра, для их установки используется команда:

# sudo apt-get install linux-headers-$(uname -r)

Команда uname –r возвращает версию текущего ядра ОС.

При программировании в пространстве ядра не используются библиотеки, которые обычно подключаются при работе в пользовательском режиме. Таким образом программирование в режиме ядра предполагает ознакомление основными функциями и типами ядра ОС Linux.

Так же потребуются команды (Драйверы устройств в Linux, 2012):

* lsmod — список модулей, загруженных в текущий момент
* insmod <module\_file> — добавление / загрузка указанного файла модуля
* modprobe <module> — добавление / загрузка модуля вместе со всеми его зависимостями
* rmmod <module> — удаление / выгрузка модуля

Простейший модуль может содержать в себе функции регистрации и удаления модуля.

# Список литературы

Jonathan Corbet, A. R.-H. (2005). *Linux Device Drivers, Third Edition.* Sebastopol: O'Reilly.

*Драйвер*. (17 02 2016 г.). Получено из wikipedia: https://ru.wikipedia.org/wiki/Драйвер

Лав, Р. (2014). *Linux Системное программирование.* СПб: Питер.