# Курсовая работа по курсу «Операционные системы».

Рязанова Н.Ю.

Курсовая работа по курсу «Операционные системы» выполняется на седьмом семестре обучения.

**Целью проекта** является закрепление у студентов основных теоретических положений курса, приобретение навыков практической реализации задач системного программирования, на основе проектирования отдельных компонент операционных систем нулевого уровня привилегий.

**Оформление курсового проекта**

Курсовой проект оформляется в виде пояснительной записки объёмом 30-35 страниц и презентации (презентация не обязательна), демонстрация работы ПО.

Пояснительная записка должна содержать:

1. титульный лист с названием темы, фамилиями студента и руководителя курсовой работы ( бланк титульного листа );
2. техническое задание на курсовую работу (бланк кафедры);

К защите надо получить допуск руководителя курсовым проектированием.

Расчетно-пояснительная записка составляется по общим требованиям, предъявляемым в РПЗ.

**Расчетно-пояснительная записка должна содержать следующие разделы:**

**Введение.** (не нумеруется)

Во введении обосновывается актуальность поставленной задачи, приводится краткий обзор существующих подходов и методов ее реализации, а также даются краткие характеристики существующего программного обеспечения, полностью или частично реализующего заданные функции.

**I Аналитический раздел.**

В первом параграфе (1.1) выполняется постановка задачи на основе утвержденной руководителем КР темы и определяются требования и ограничения к разрабатываемому ПО. Первый параграф в соответствии с его назначением называется «Постановка задачи».

В следующих параграфах выполняется анализ поставленной задачи. При этом анализируются методы и/или способы ее решения. Проводится сравнительный анализ методов и/или способов ее решения и делается обоснованный выбор методов и/или способов решения поставленной задачи.

Например, описываются особенности USB шины, дерева USB, особенность архитектуры USB драйвера, конечных точек и типов передачи данных. При реализации мониторинга системных вызовов или работы системы выполняется анализ способов перехвата нужных функций ядра, их сравнение и обоснованный выбор способа и средств перехвата. Может определяться тип программного обеспечения, например, драйвер, реализованный в виде загружаемого модуля ядра, драйвер и приложение, взаимодействующее с драйвером или для Linux - драйвер и демон.

**II Конструкторский раздел.**

В конструкторском разделе демонстрируется последовательность выполняемых в ПО действий в виде IDEF0. Приводятся схемы алгоритмов по ГОСТу.

Для драйверов приводятся базовые структуры, например, struct device, struct usb\_driver и т.п., перечисляются и описываются точки входа драйвера.

Могут описываться взаимодействие модулей разработанного ПО через подсистему ввода/вывода (в случае разработки драйвера внешнего устройства) или другую подсистему, например, на основе событийной модели.

Может описываться библиотека или framework, ее хуки.

**III Технологический раздел.**

В первом параграфе обоснованно выбирается язык программирования, среда программирования, средства разработки и используемые библиотеки. Первый параграф в соответствии с назначением называется «Выбор языка и среды программирования».

В следующих параграфах в соответствии с материалом конструкторского раздела приводятся описания объявленных структур и кодов функций.

Например, для драйверов, демонов или загружаемых модулей ядра приводятся выбранные и разработанные структуры данных, коды точек входа и разработанные дополнительные функции, реализующие поставленную задачу. Для патчей описать особенности реализации и перекомпиляции ядра ОС.

Для Linux приводится make файл*.*

**IV Исследовательский раздел**

В разделе показываются и при возможности анализируются результаты работы разработанного ПО.

**Заключение или выводы (не нумеруется).**

В заключении кратко излагаются основные направления работы, рассмотренные методы или способы, предложенные особенности реализации, полученные результаты и выводы.

Например, в выводах можно кратко по пунктам перечисляются основные результаты работы (например, 1. Показано .. 2. Исследовано/ны .. 3.Разработана/ны структура .. 4. Определена целесообразность .. 5. Разработано программное обеспечение .. 6. Тестирование показало .. и т.п.)

***Замечание: после заголовка раздела не надо писать «В разделе рассматривается....». Это – лишнее, т.к. и так понятно.***

**Список используемых источников.**

**Пpиложение.**

В приложении к РПЗ дается краткое руководство пользователя, руководство системного программиста (если это необходимо), полный текст исходного кода драйвера, или загружаемого модуля ядра, или программного комплекса (с комментариями). Приложение к проекту может быть сдано в электронном виде.

**Замечания**

**1.** Следует избегать подробного изложения в записке банальных, широко известных положений, взятых из учебников, например, процесс это – программа в стации выполнения. Не следует приводить базовые определения, даваемые в курсе ОС. Графический материал: схема взаимодействия разработанной компоненты ОС с модулями ОС, схемы алгоритмов и структурные схемы должны выполняться по **ГОСТ**у. Отладочные и тестовые программы желательно строить таким образом, чтобы использовать их для последующей демонстрации основных пунктов проекта и работы программного обеспечения во время защиты.

**2.** В качестве темы нельзя брать: - драйвер-фильтр клавиатуры, изменяющий кодировку; - драйвер-фильтр мыши, меняющий правую и левую кнопки, идентификацию скрытых процессов, мониторинг действий пользователя с клавиатурой и мышью.

**3.** Для изменения функциональности устройств под ОС Linux нельзя использовать дескриптор отчета (Report Descriptor). Данная тема полностью проработана и описана в статье «Метод изменения поведения HID-устройств под управлением ОС Linux» авторы Сикорский О.С., Рязанова Н.Ю. (eLibrary),

|  |  |  |  |
| --- | --- | --- | --- |
| ЖУРНАЛ: | | | |
|  | | [НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ](https://www.elibrary.ru/contents.asp?id=34940386) Издательство: [Московский институт электроники и математики НИУ ВШЭ](https://www.elibrary.ru/publisher_titles.asp?publishid=9231) (Москва) ISSN: 2227-0973 | |
| АННОТАЦИЯ: | | | |
|  | | В статье рассматривается метод изменения поведения HID устройств. Поведение HID устройств описывается дескриптором отчета (Report Descriptor), поступающим от устройства в модуль HID core. Если устройство правильно реализует протокол своего класса, то для него не нужно разрабатывать специальный драйвер и система автоматически определит его назначение и начнет им управлять. При необходимости изменения поведения устройства нужно только заменить дескриптор отчета устройства новым дескриптором, в котором, например, может быть изменен тип устройства, назначение кнопок и стиков и т.п. Замена осуществляется с помощью HID драйвера устройства, реализуемого как загружаемый модуль ядра. | |

## Литература

1. Агуров П.В. Практика программирования USB. – СПб.: БХВ - Питербург, 2007. – 624с.

2. Банахан М., Раттер Э. Введение в операционную систему UNIX: Пер. с англ. М.: Радио и связь, 1986.

3. Вахалия Ю. UNIX изнутри. – СПб.: Питер, 2003. – 844 с.

4. Комиссарова В. Программирование драйверов для Windows. – СПб.: БХВ-Петербург, 2007.- 256 с.

5. Они У. Использование Microsoft Windows Driver Model. 2-е изд. (+CD); Для профессионалов. – СПб.: Питер, 2007. – 764 с.

6. Орвик П., Смит Г. Windows Driver Foundation: разработка драйверов: Пер. с англ. М.: Издательство «Русская редакция»; СПб.: «БХВ-Петербург», 2008 – 880 с.: ил.

7. Рудаков П.И., Финогенов К.Г. Программируем на языке ассемблера IBM PC. Части 2-4. М.: Энтроп, 1996.

8. Робачевский А.М. Операционная система UNIX. – СПб.: БХВ-Петербург, 2003.–528 с.

9. Солдатов В.П. Программирование драйверов Windows. Изд. 2-е, переработанное и доп. – М.: ООО «Бином-Пресс», 2004 г. – 480 с.

10. Соломон Д., Руссинович М. Внутреннее устройство MS Windows 2000. Питер: 2001

11. Соломон Д., Руссинович М. Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP, Windows 2000. Мастер класс. / Пер. с англ. – 4-е изд. – М.: Издательство «Русская редакция»; СПб.; Питер, 2006. – 992 с.

12. Сорокина С.И., Тихонов А.Ю., Щербаков А.Ю. Программирование драйверов и систем безопасности. – СПб.: БХВ-Петербург, 2003.

13. Теренс Чан Системное программирование на С++ для Unix. BHV, Киев, 1997

14. Фролов А.В., Фролов Г.В. Аппаратное обеспечение персонального компьютера. 15. Библиотека системного программиста 33 т. М.: Диалог-Мифи, 1997.

16. Шрайбер С. Недокументированные возможности Windows 2000. – СПб.: Питер, 2002

17. <http://msdn.microsoft.com/en-us/windows/hardware/gg463279>

18. Цирюлик О.И. Модули ядра Linux. Внутренние механизмы ядра.

<http://rus-linux.net/MyLDP/BOOKS/Moduli-yadra-Linux/kern-mod-index.html>

19. Создание собственных драйверов устройств под Linux <https://habr.com/ru/post/206148/>

20. Добавление драйвера устройства в Linux.

<https://itproffi.ru/dobavlenie-drajvera-ustrojstva-v-linux/>

21. Глава 9.1 Драйверы устройств [Linux center.ru](http://www.linuxcenter.ru/)  **»** [Библиотека Линукс центра](http://www.linuxcenter.ru/lib/)  **»** [Книги и руководства](http://www.linuxcenter.ru/lib/books/)  **»** [Linux для пользователя](http://www.linuxcenter.ru/lib/books/kostromin/)

<http://www.linuxcenter.ru/lib/books/kostromin/gl_09_01.phtml>

1. Перехват функций в ядре Linux с помощью ftrace [Электронный ресурс]. – Режим доступа: <https://m.habr.com/post/413241/>, свободный – (02.12.2020)
2. Модули Linux ядра, Олег Цилюрик [Электронный ресурс]. – Режим доступа: http://rus-linux.net/MyLDP/BOOKS/Moduliyadra-Linux/KERN-modul-4.95.pdf, свободный – (02.12.2020)
3. Loadable Kernel Module Programming and System Call Interception <https://www.linuxjournal.com/article/4378>
4. М. Джонс Анатомия загружаемых модулей ядра Linux. <https://www.ibm.com/developerworks/ru/library/llkm/index.html>
5. Исходные коды ядра Linux http://elixir.free-electrons.com