

РЕФЕРАТ

Расчетно-пояснительная 13 с., 0 рис., ? ист.

Ключевые слова: драйвер, USB-мышь, яркость, цветовая температура.

В рамках данной курсовой работы был разработан драйвер для изменения яркости и цветовой температуры дисплея с использованием USB-мыши.

СОДЕРЖАНИЕ

РЕФЕРАТ	3
ВВЕДЕНИЕ	5
1 Аналитический раздел	6
1.1 Постановка задачи	6
1.2 Особенности шины USB	6
1.3 USB драйвер	8
1.4 Способы изменения яркости, цветовой температуры	8
1.5 Тип программного обеспечения	8
2 Конструкторский раздел	9
2.1 Диаграмма IDEF0	9
2.2 Инициализация устройства и драйвера	9
2.3 Алгоритмы	9
3 Технологический раздел	10
3.1 Выбор языка и среды программирования	10
3.2 Реализация драйвера	10
4 Исследовательский раздел	11
4.1 Демонстрация результатов работы программы	11
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	13

ВВЕДЕНИЕ

Одной из важнейших задач операционной системы является организация работы с устройствами ввода-вывода. Наиболее распространенными представителями таких устройств являются стандартные компьютерные мыши, клавиатуры, джойстики, мониторы; они позволяют пользователю интерактивно взаимодействовать с компьютерной системой.

Управление внешними устройствами драйверами, изменение их поведения осуществляется написанием драйверов. Ключевыми характеристиками дисплея компьютера являются яркость и цветовая температура (теплота). Чтобы задать необходимые пользователю (в частности, безопасные для его зрения) значения яркости и теплоты, целесообразно использовать устройство, позволяющее одновременно нажимать на клавиши и осуществлять скроллинг. Одним из таких устройств является USB-мышь.

1 Аналитический раздел

1.1 Постановка задачи

В соответствии с заданием на курсовую работу необходимо разработать драйвер для изменения яркости и цветовой температуры дисплея с использованием USB-мыши.

1.2 Особенности шины USB

Типы устройств:

b — блок-ориентированные устройства; c — байт-ориентированные устройства (символьные);

u — не буферизованное байт-ориентированное устройство; p — именованный канал.

Для хранения номеров устройств, в ядре используется специальный тип `dev_t`. Начиная с ядра 2.6.0 — `dev_t` 32-х разрядный, 12 отведены для старшего номера (все устройства делятся на наборы, которые определяются старшим номером: , например, все SCSI-диски имеют старший номер 8), 20 для младшего (младший номер конкретизирует устройство). Для получения старшей и младшей части используются макросы `MAJOR` и `MINOR`.

В Linux драйверы трех типов:

Встроенные в ядро. Соответствующие устройства распознаются системой автоматически, нужны для поддержки, например, монтирования корневой ФС и запуска ПК (материнская плата и т.д.) Реализованные, как загружаемые модули ядра. Используются для управления устройствами (звуковые и сетевые карты, SCSI адаптеры) Код поделен между ядром и специальной утилитой, управляющей устройством. Например, принтер: ядро осуществляет взаимодействие с параллельным портом, а формирование управляющих сигналов для принтера осуществляет демон печати `lpr`.

Схема взаимодействия прикладных программ с аппаратной частью компьютера (в Linux): устройство <-> ядро <-> специальный файл устройства <-> программа пользователя

Любому устройству, зарегистрировавшемуся в системе, выделяется специальный дескриптор в виде структуры `struct device`. Структура `struct device_driver` представляет универсальную модель драйвера устройства, которая отслеживает все драйверы, известные системе.

`usb_register_dev`

Устройства USB могут являться хабами, функциями или их комбинацией.

Хаб (Hub) - обеспечивает дополнительные точки подключения устройств к шине. Функции представляют собой устройства, способные передавать или принимать данные или управляющую информацию по шине. Функции USB предоставляют системе дополнительные возможности, например подключение акустической системы, мыши и т.п.

USB-шина является шиной, в которой имеется один мастер - host-контроллер (который еще называют root hub). USB-устройства всегда отвечают на запросы host-контроллера (usb-устройства никогда не могут посылать информацию самостоятельно; host-контроллер формирует запросы, а устройства отвечают).

Запросы имеют направление:

IN - хост отправляет запрос на прием данных OUT - хост отсылает данные устройству [1]

Конечные точки (endpoints). Это базовый объект связи интерфейса USB. Устройство может иметь до 16 конечных точек, нумерация начинается с 0 и заканчивается 15. Каждая конечная точка может включать в себя два буфера (адреса): входной и выходной. То есть устройство может обладать 32 адресами конечных точек. Каждая USB-функция должна содержать как минимум одну (нулевую) конечную точку с входным и выходным буфером.

Каналы (pipes). Хост определяет каналы, которые связаны с конечными точками функции. В отличие от конечной точки, которая имеет физическую сущность в нашем мире, канал является всего лишь логической концепцией, правилом. После установки канала, становится определенным и тип передачи данных, который он поддерживает.

Данные отправляются и принимаются посредством передач или сообщений, которые состоят из ряда транзакций, каждая из которых состоит из пакетов.

Существует 4 типа передач:

control — передача типа control является двунаправленным и предназначен для обмена с устройством короткими пакетами типа «вопрос-ответ». Обеспечивает гарантированную доставку данных. isochronous — изохронный канал имеет гарантированную пропускную способность (N пакетов за один период шины) и обеспечивает непрерывную передачу данных. interrupt — канал прерывания позволяет доставлять короткие пакеты без гарантии доставки и без подтверждений приема, но с гарантией времени доставки – пакет будет доставлен не позже, чем через N миллисекунд. bulk — поточная или сплошная передача используется устройствами, отправляющими и принимающими большое количество данных. Канал дает гарантию доставки каждого пакета.

1.3 USB драйвер

Структура USB-драйвера struct usb_driver (Основные точки входа: probe, disconnect, suspend, resume).

id_table используется для горячего подключения. Он содержит набор дескрипторов, и специализированные данные могут быть связаны с каждой записью. Эта таблица используется поддержкой горячего подключения как в режиме пользователя, так и в режиме ядра.

usb_alloc_urb, usb_fill_int_urb, usb_submit_urb

1.4 Способы изменения яркости, цветовой температуры

1.5 Тип программного обеспечения

драйвер + демон

2 Конструкторский раздел

2.1 Диаграмма IDEF0

2.2 Инициализация устройства и драйвера

2.3 Алгоритмы

3 Технологический раздел

3.1 Выбор языка и среды программирования

3.2 Реализация драйвера

4 Исследовательский раздел

4.1 Демонстрация результатов работы программы

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Corbet J., Rubini A., Kroah-Hartman G. Linux device drivers. — O'Reilly Media, Inc., 2005.