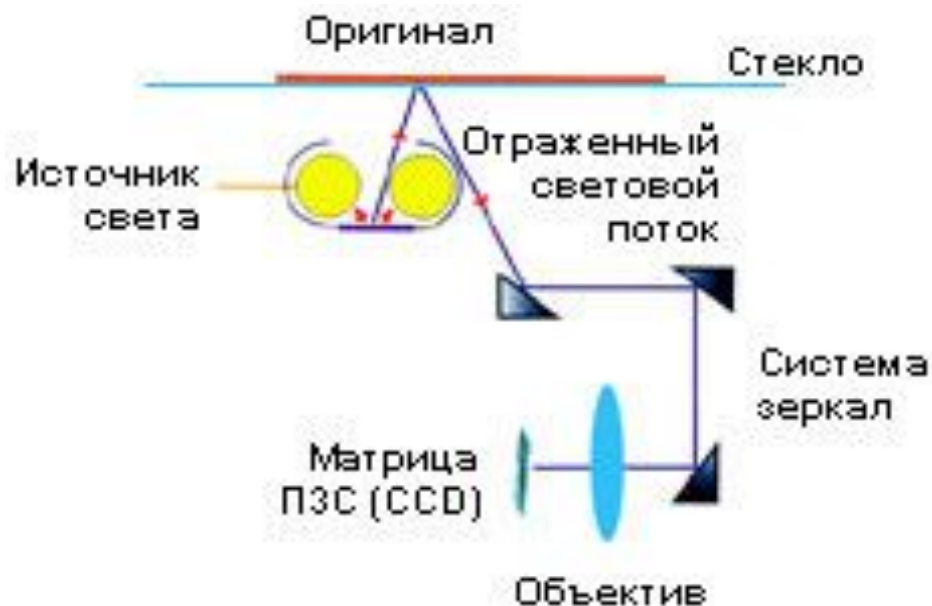




**ТЕМА: ТИПЫ  
СКАНЕРОВ И ИХ  
ТЕХНИЧЕСКИЕ  
ХАРАКТЕРИСТИКИ**

**Сканер (Scanner)** – устройство ввода в ПК информации в виде текстов, рисунков слайдов, фотографий на плоских носителях, а также изображения объёмных объектов небольших размеров.



**Сканирование** представляет собой цифровое кодирование изображения, заключающееся в преобразовании аналогового сигнала яркости в цифровую форму.

## *Виды сканеров:*

- ручной
- листопротяжный
- планшетный
- Барабанный



Ручные сканеры – обычные или самодвижущиеся – обрабатывают полосы документа шириной около 10 см и представляют интерес, прежде всего для владельцев мобильных ПК.

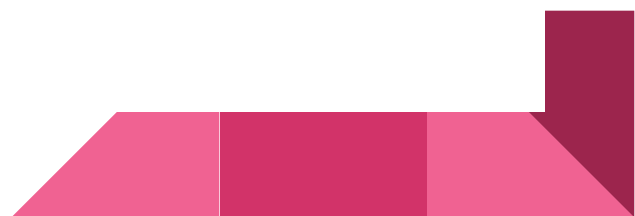
**В листопротяжном сканере,** страницы документа при считывании пропускаются через специальную щель с помощью направляющих роликов (последние зачастую становятся причиной перекоса изображения при вводе).

**Планшетные сканеры:** документ – кладут на специальное стекло, под которым

пе



-ци<sup>1</sup>



**Барабаннные сканеры** - оригиналы размещаются на внутренней или внешней (в зависимости от модели) стороне прозрачного цилиндра, который называется барабаном. За один его оборот считывается одна линия пикселей, так что процесс сканирования очень напоминает работу токарно-винторезного станка. Проходящий через слайдузкий луч света, который создается мощным лазером, с помощью системы зеркал попадает на ФЭУ (фотоэлектронный умножитель), где оцифровывается.

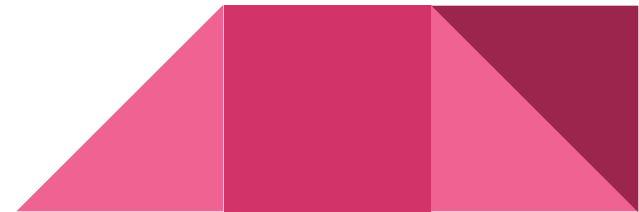
## Барабанный сканер



## **Основные характеристики сканера:**

**Оптическое разрешение** - измеряется в точках на дюйм (dots per inch, dpi). Характеристика, показывающая, чем больше разрешение, тем больше информации об оригинале может быть введено в компьютер и подвергнуто дальнейшей обработке.

**Глубина цвета** – это характеристика, обозначающая количество цветов, которое способен распознать сканер.



**Тип подключения:**

**По типу интерфейса сканеры делятся всего на четыре категории:**

**Сканеры с параллельным или последовательным интерфейсом, подключаемые к LPT- или COM-порту**

Эти интерфейсы самые медленные и постепенно себя изживают. Если ваш выбор все-таки пал на подобный сканер, заранее настройтесь на появление проблем, связанных с конфликтом сканера с LPT-принтером, если таковой имеется.

**Сканеры с интерфейсом USB**


Стоят чуть-чуть дороже, но работают значительно быстрее. Необходим компьютер с USB-портом. Проблемы с установкой также могут возникнуть, но обычно они легко устранимы.

**Сканеры со SCSI-интерфейсом**

С собственной интерфейсной платой для шины ISA или PCI либо подключаемые к стандартному SCSI-контроллеру. Эти сканеры быстрее и дороже представителей двух предыдущих категорий и относятся к более высокому классу.

**Сканеры с ультрасовременным интерфейсом FireWire(IEEE 1394)**

Специально разработанным для работы с графикой и видео. Такие модели представлены на рынке относительно недавно



# Оптическая система планшетного сканера

**Ксеноновые газоразрядные лампы** отличаются чрезвычайно малым временем прогрева, высокой стабильностью излучения, небольшими размерами и долгим сроком службы.

С другой стороны, они требуют высокого напряжения, потребляют большой ток и имеют неидеальный спектр, что пагубно сказывается на точности цветопередачи.

**Люминесцентные лампы** с горячим катодом обладают очень ровным, управляемым в определенных пределах спектром и малым временем прогрева. В качестве недостатков можно назвать крупные габариты и относительно короткий срок службы. С другой стороны, они требуют высокого напряжения, потребляют большой ток и имеют неидеальный спектр, что пагубно сказывается на точности цветопередачи.

Люминесцентные лампы с горячим катодом обладают очень ровным, управляемым в определенных пределах спектром и малым временем прогрева.

В качестве недостатков можно назвать крупные габариты и относительно короткий срок службы.

**Светодиоды (LED)** применяются, как правило, в CIS-сканерах, не требуют времени для прогрева и обладают небольшими габаритами и энергопотреблением. В большинстве случаев используются трехцветные светодиоды, меняющие с большой частотой спектр излучаемого света.





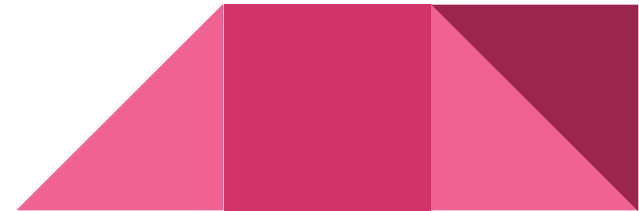
## ***Стек драйверов для системных устройств ввода***

Драйвера клавиатуры, независимо от схем физического подключения, используют системные драйвера класса клавиатуры для обработки не зависящих от аппаратной части операций. Данные драйвера называются **драйверами класса**, так как обеспечивают требуемые системой, но не зависящие от аппаратной реализации требования к конкретному классу устройств.

Соответствующий **функциональный драйвер** (драйвер порта) реализует зависящую от конкретного устройства поддержку выполнения операций ввода-вывода. В ОС Windows для x86-платформ реализован единый драйвер системной клавиатуры (i8042) и мыши.

## ***Стек драйверов для Plug and Play PS/2-клавиатуры***

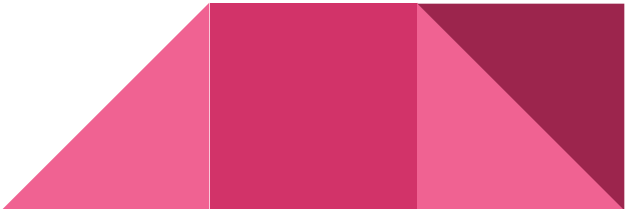
- Стек драйверов содержит (сверху вниз):
- Kbdclass — верхнеуровневый фильтр-драйвер класса клавиатуры;
- опциональный верхнеуровневый фильтр-драйвер класса клавиатуры;
- i8042prt — функциональный драйвер клавиатуры;
- корневой драйвер шины.



В ОС Windows 2000 и старше драйвером класса клавиатуры является драйвер **Kbdclass**, основными задачами которого являются:

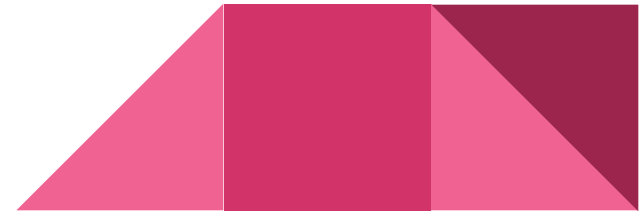
- обеспечение общих и аппаратно-независимых операций класса устройств;
- поддержка Plug and Play, поддержка управления питанием и Windows Management Instrumentation (WMI);
- поддержка операций для legacy-устройств;
- одновременное выполнение операций более чем одного устройства;
- реализация class service callback routine, которая вызывается функциональным драйвером для передачи данных из входного буфера устройства в буфер данных драйвера класса устройств.

В ОС Windows 2000 и старше функциональным драйвером для устройств ввода, использующих PS/2-порт (клавиатуры), является драйвер **i8042prt**, основные функции которого следующие:

- обеспечение аппаратно-зависимых одновременных операций PS/2-устройств ввода (клавиатуры и мыши разделяют общие порты ввода вывода, но используют разные прерывания, процедуры обработки прерываний (ISR) и процедуры завершения обработки прерываний);
  - поддержка Plug and Play, поддержка управления питанием и Windows Management Instrumentation (WMI);
  - поддержка операций для legacy-устройств;
  - вызов class service callback routine для классов клавиатуры и мыши для передачи данных из входного буфера данных i8042prt в буфер данных драйвера класса;
  - вызов набора функций обратного вызова, которые могут реализовать драйвера-фильтры высокого уровня для гибкого управления устройством.
- 

В целом стек устройств (правильнее говорить о стеке объектов устройств) PS/2-клавиатуры состоит из:

- физического объекта устройства клавиатуры (PDO), созданного драйвером шины (в данном случае, шины PCI) – `\Device\00000066`;
- функционального объекта устройства клавиатуры (FDO), созданного и присоединенного к PDO драйвером `i8042prt` — неименованный объект (`unnamed`);
- опциональных фильтр-объектов устройства клавиатуры, создающихся фильтр-драйверами клавиатуры, разрабатываемыми сторонними разработчиками;
- верхнеуровневого фильтр-объекта устройства класса клавиатуры, созданного драйвером класса `Kbdclass` – `\Device\KeyboardClass0`.

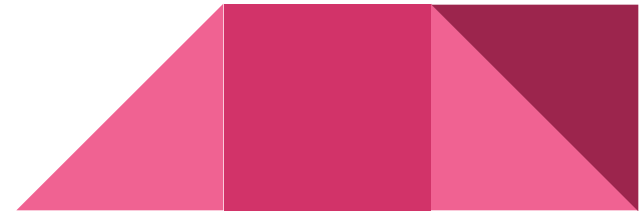


# Обработка клавиатурного ввода приложениями

Подсистема Microsoft Win32 получает доступ к клавиатуре, используя поток необработанного ввода (Raw Input Thread, **RIT**), который является частью системного процесса csrss.exe

**IRQ 1** (interrupt service routine, ISR) - процедуру обработки прерывания, зарегистрированную в системе драйвером i8042prt.

У стека клавиатуры всегда есть по крайней мере один ожидающий завершения IRP, и находится он в очереди драйвера Kbdclass



# Клавиатурные ловушки

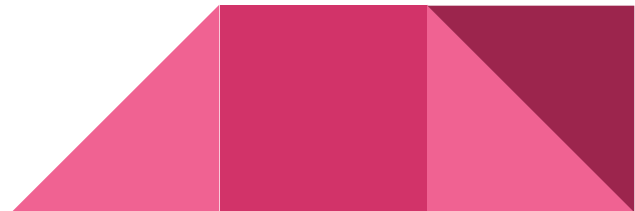
В операционной системе Microsoft Windows **ловушкой**, или **хуком** (hook) называется механизм перехвата событий с использованием особой функции (таких как передача сообщений Windows, ввод с мыши или клавиатуры) до того, как они дойдут до приложения. Эта функция может затем реагировать на события и, в некоторых случаях, изменять или отменять их.

Функции, получающие уведомления о событиях, называются **фильтрующими функциями** и различаются по типам перехватываемых ими событий.

Прикрепление одной или нескольких фильтрующих функций к какому-нибудь хуку называется **установкой хука**.

Для установки и удаления фильтрующих функций приложения используют функции Win32 API SetWindowsHookEx и UnhookWindowsHookEx.

**Цепочка хуков** — это список указателей на фильтрующие функции (специальные функции обратного вызова, определяемые приложением).



# Общая схема обработки

Итак, алгоритм прохождения сигнала от нажатия пользователем клавиш на клавиатуре до появления символов на экране можно представить следующим образом:

- Операционная система при старте создает в системном процессе `csrss.exe` поток необработанного ввода и системную очередь аппаратного ввода.
- Поток необработанного ввода в цикле посылает запросы чтения драйверу класса клавиатуры, которые остаются в состоянии ожидания до появления событий от клавиатуры.
- Когда пользователь нажимает или отпускает клавишу на клавиатуре, микроконтроллер клавиатуры фиксирует нажатие/отпускание клавиши и посылает в центральный компьютер скан-код нажатой клавиши и запрос на прерывание.
- Системный контроллер клавиатуры получает скан-код, производит преобразование скан-кода, делает его доступным на порту ввода-вывода `60h` и генерирует аппаратное прерывание центрального процессора.
- Контроллер прерываний вызывает процедуру обработки прерывания `IRQ 1`, — `ISR`, зарегистрированную в системе функциональным драйвером клавиатуры `i8042prt`.
- Процедура `ISR` считывает из внутренней очереди контроллера клавиатуры появившиеся данные, переводит скан-коды в коды виртуальных клавиш (независимые значения, определенные системой) и ставит в очередь вызов отложенной процедуры `I8042KeyboardIsrDpc`.
- Как только это становится возможным, система вызывает `DPC`, которая в свою очередь вызывает процедуру обратного вызова `KeyboardClassServiceCallback`, зарегистрированную драйвером класса клавиатуры `Kbdclass`.
- Процедура `KeyboardClassServiceCallback` извлекает из своей очереди ожидающий завершения запрос от потока необработанного ввода и возвращает в нем информацию о нажатой клавише.
- Поток необработанного ввода сохраняет полученную информацию в системной очереди аппаратного ввода и формирует на ее основе базовые клавиатурные сообщения Windows `WM_KEYDOWN`, `WM_KEYUP`, которые ставятся в конец очереди виртуального ввода `VIQ` активного потока.
- Цикл обработки сообщений потока удаляет сообщение из очереди и передает его соответствующей оконной процедуре для обработки. При этом может быть вызвана системная функция `TranslateMessage`, которая на основе базовых клавиатурных сообщений создает дополнительные «символьные» сообщения `WM_CHAR`, `WM_SYSCHAR`, `WM_DEADCHAR` и `WM_SYSDEADCHAR`.