

## Лекция №1. Основные КОМАНДЫ драйверов устройств

### *Обработка команд системы.*

Команда 0 - инициализация . Во время инициализации драйвер консоли вызывает функцию для управления экраном. Эта возможность объясняется тем, который перед установкой нашего драйвера, система загружает стандартный, который и используется для выполнения вызванных функций (BIOS). Но с момента инициализации и возврата управления системы все запросы будут обрабатываться новым драйвером.

Адрес точки разрыва это адрес первой свободной ячейки памяти с программой. Этот адрес нужна системе для того, чтобы определить, куда загружать драйверы устройств свои собственные программы после установки нового драйвера. Драйвер может задать адрес точки разрыва внутри программы драйвера. В этом случае система уничтожит часть программы драйвера. Процедура инициализации выполняется только один раз, то есть можно поместить в конец программы и в качестве адреса расторжения задать адрес начала этой процедуры.

Если что-то препятствует правильной работе драйвера консоли, то он сигнализирует системе о необходимости выгрузить драйвер. Это делается путем установления смещения точки разрыва в 0, а сегментного адреса в текущее значение cs. Это говорит системе о том, что такая доступная ячейка памяти находится в начале драйвера, наличие которого таким образом игнорируется.

Следовательно, при инициализации драйвер выводит сообщение на экран, устанавливает адрес точки разрыва в заголовке запроса и передает управление системе.

Команды с 1 по 3. Команды проверка носителя, создание BPB, IOCTL - введение не реализуются в этом драйвере. Для выполнения 2х команд в заголовке запроса устанавливается бит выполнено, при обработке третьей команды, происходит переход к установке бита ошибка.

команда 4 - введение. По этой команде осуществляется ввод символов с буфера клавиатуры с помощью INT16h и передача их в систему в буфере, определенном в заголовке запроса. Система передает драйверу через заголовок запроса количество символов, которые должны быть введены, а также адрес, по которому их надо

сохранить. Прерывания BIOS 16h возвращает ASCII-код символа в AL, а скан-код - в AH.

Для расширенных (имеющих только скан-код) клавиш INT16h возвращает 0 AL и скан-код в AH.

Когда нажимается клавиша, имеет ASCII-код, из буфера клавиатуры в системе возвращается только ее ASCII-код, а при нажатии расширенной клавиши система ожидает 2 значения: ASCII-код = 0 и скан-код.

Таким образом драйвер должен возвращать системе в составе заголовка запроса ASCII-код для всех клавиш и скан-код для расширения клавиш. Последовательность команд для чтения символов из буфера клавиатуры размещена в цикле для подсчета символов и возвращение в систему их количества. Система не запрашивает более чем один символ. После получения символа с помощью INT16h, он записывается в буфер данных. После этого работа драйвера заканчивается. При этом драйвер восстанавливает регистры ES и BX, потому что они нужны для адресации буфера данных, в котором система ожидает найти введенные символы. Затем происходит переход для установки бита выполнения в слове состояния и передача управления.

команда 5 - неразрушающее введение.

Эта команда позволяет системе проверить наличие символа в буфере клавиатуры без вытягивания его оттуда.

Она включена в состав команд в связи с тем, что программа может вызвать функцию системы проверки устройства ввода (0Bh). Для проверки состояния буфера клавиатуры драйвер использует функцию AH = 1 прерывания 16h. При этом либо система сообщается, что буфер пуст, или возвращается очередной символ, ожидает, без удаления его из буфера.

Если функция проверки состояния возвращает 0 как для ASCII, так и для скан-кода, то буфер клавиатуры пуст. При отсутствии символов в буфере устанавливается бит занят в слове состояния и возвращается управления системе.

команда 6 - Состояние ввода.

Устанавливается бит выполнено и завершается работа.

команда 7 - Очистка введения.

Используется для предотвращения использования программой введенных заранее символов. Например, команда FORMAT очищает буфер клавиатуры перед вопросом, форматировать диск.

Драйвер проверяет состояние клавиатуры с помощью прерывания BIOS 16h. Если в буфере символ, он оттуда извлекается, но не пересылается в систему. Так продолжается, пока буфер не опустеет.

**команда 8** - вывод. Для ее выполнения используется прерывание BIOS

10h. Реализуется счетчик выведенных символов на основе цикла, с которого производится вызов BIOS 10h с AH = 0Eh (функция вывода в режиме телетайпа). В этом же цикле вызывается локальная процедура, генерирует случайный цвет символа. Затем восстанавливаются значения регистров ES и BX, использовавшиеся при считывании символов из буфера данных.

**команда 9** - вывод с проверкой. Как правило, эта команда применяется

для устройств, которые могут читать только записанные данные. Ее назначение - проверить, правильно записанные данные. Здесь эта команда аналогична команде вывода.

**Команды 10-16.** Выполнение этих команд не нужно в драйвере консоли, но их обработка должна быть предусмотрена в связи с возможностью их случайной передачи драйвера.

**Выход при ошибке и общий выход** . В этой секции драйвер помещает в заголовок

запроса слово состояния, информирует систему о результатах работы драйвера. Система ожидает, что в любом случае будет установлен бит выполнено. Дополнительно м.б. установлены и другие биты: бит занято, биты ошибка.

***Команды драйверов устройств.***

Проанализируем каждую из 26 команд.

**Команда 0. инициализация** .

Это первая команда, которую обрабатывает драйвер. Система посылает команду инициализации сразу после загрузки драйвера в память.

После того, как команда выполнена, повторно она драйвера передана никогда не будет. Задача этой команды - позволить драйверу подготовить устройство к работе.

Выполняется установления ряда регистров и буферов, указателей и счетчиков. После инициализации драйвер готов к выполнению других команд.

Этапы выполнения команды:

1. Инициализация устройства, буферов и счетчиков.
2. Вывод дополнительного сообщения инициализации.
3. Установление числа устройств, обслуживаемых драйвером (для блочных устройств).
4. Установление адреса точки разрыва.
5. Установление указателя на таблицу адресов BPB (для блочных устройств).
6. Установление слова состояния.

Функции системы допускается только в процессе выполнения команды инициализации.

Разрешается использовать функции с номерами от 01h до 0Ch и 30h. Другие функции использовать нельзя, так как до этого момента сама система еще полностью не установлена.

Запишем структуру данных, используемой в команде

инициализации

(Динамическая часть заголовка запроса):

rh0 struc; инициализация (ком 0).	
rh0_rh db size rh dup (?);	фиксированная часть
rh0_nunits db?	; Число устройств (для блочных)
rh0_brk_ofs dw?	; смещение точки разрыва
rh0_brk_seg dw?	; сегмент точки разрыва
rh0_bpb_tbo dw?	; смещение указателя на массив BPB
rh0_bpb_tbs dw?	; сегмент указателя на массив BPB
rh0_drv_ltr db?	; первый доступный диск
; (Блочные ус.)	
rh0 ends	

В переменную rh0\_nunits, которая устанавливается блочным драйвером, заносится число устройств, которыми он управляет. Это число должно быть передано драйвером в систему. Оно заносится в первый байт поля имени устройства из заголовка запроса. В переменных rh0\_brk\_ofs и rh0\_brk\_seg содержится адрес точки разрыва, то есть адрес, которой заканчивается программа драйвера. Начиная с этого адреса, система может загружать в память

другие драйверы. процедура

инициализации

используется только один раз, значит ее можно расположить в конце драйвера, а ее начало определить как адрес расторжения. Переменные, содержащие адрес

расторжении должны определяться во всех драйверах. Если при инициализации драйвера возникают какие-то сложности, можно прекратить его загрузки, определив в качестве адреса расторжения, адрес начала драйвера. При этом для блочных драйверов необходимо обнулить переменную rh0\_nunits.

В переменных rh0\_bpb\_tbo и rh0\_bpb\_tbs содержится адрес (сегмент и смещение) таблицы BPB, которую драйвер, управляющий диском, должен передать в систему. Для системы необходима информация о типах дисковых накопителей, которыми управляет драйвер. Чтобы удовлетворить требованиям системы, нужно создать блок параметров BIOS для каждого типа диска. При этом должна быть сформирована таблица, содержащая адреса каждого блока параметров BIOS. Адрес этой таблицы, записанная в указанные переменные, передается системе. По этим сведениям системы и драйвер определяют, проводилась замена дисков, не удалены они и где расположена информация на каждом диске.

Переменная rh0\_drv\_ltr содержит следующую доступную букву накопителя. Блочные драйверы могут использовать эту информацию для вывода на экран, букв накопителей которыми эти драйверы управляют.

Перед завершением работы драйвера необходимо сформировать слово rh\_status.

#### **Команда 1. Проверка носителя .**

Эта команда действует только для блочных устройств. Она используется системой, если необходимо проверить, не был заменен диск. Система посылает команду проверки носителя перед выполнением каждой операции чтения или записи для любого диска.

Структура данных для этой команды:

rh1 struc;	Проверка носителя
rh1_rh db size rh dup (?);	
rh1_media db?	; Дескриптор носителя с блока
rh1_md_stat db?	, Состояние носителя (возвращается.)
rh1_valid_ofs dw?	; Смещение отметки диске
rh1_valid_seg dw?	; Сегмент пометки диске
rh1 ends;	

При определении того, не был заменен диск, выполняется следующая последовательность действий:

1. Одержання дескриптора носителя.
2. Определение того, заменено диск.

3. Востановления состояния носителя.

4. Установление слова состояния в заголовке запроса.

Для определения изменения диска можно использовать 3 основных метода: проверка времени, прошедшего с момента последнего обращения, проверка, выполняется аппаратно, проверка смены диска и сравнение информации о диске.

Первый метод не эффективен, потому что если время последнего обращения к диску превышает 2 сек., То нельзя быть уверенным в том, что он не был заменен.

Второй метод - лучший из трех. Накопители большой емкости выдают сигнал при открытии дисководов, поэтому этот факт можно зафиксировать и выполнить установку переменной состояния носителя. Этот сигнал подается все время, пока дисковод открыт.

Третий метод требует, чтобы драйвер сохранял информацию о диске. Эта информация включает байт дескриптора носителя, пометку диска и блок параметров BIOS. Если какой-то параметр изменился за период между последним и текущим обращением к диску, можно предположить, что диск заменен. Однако это средство ненадежное. Если сменить дискеты, одинаково отформатированные, в которых дескрипторы носителей совпадают, то этот метод приведет к ошибочному решению, что диск не изменен.

Но нельзя обойти эти сложности. Слово состояния носителя может иметь три значения:

- 1 - носитель изменен, 0 - неизвестно, изменен носитель, 1 - носитель не изменен.

Если невозможно определить, был ли заменен диск, слово состояния носителя `rh1_md_stat` заносится 0. Для дисков всех типов первая команда проверки носителя присваивает - 1 слову состояния смены носителя. Эта операция выполняется при первом обращении к электронному диску точно так же, как и при обращении к накопителю на гибких дисках, потому что в этот момент система не имеет точной информации о диске. Следующие команды проверки носителя при работе с жесткими и электронными дисками придают слову состояния изменения носителя 1.

Нужно помнить, что драйвер обеспечивает проверку изменения носителя данных, если бит 11 слова атрибутов в заголовке драйвера установлен.

Если значение переменной `rh1_md_stat` = - 1, то в переменные `rh_1_valid_ofs` и `rh1_valid_seg` записываются смещение и сегмент пометки диска предыдущего диска. При этом предполагается, что она сохраняется драйвером. В противном случае в эти переменные заносится адрес строки, содержащий имя "No Name", за которым следуют четыре

пробелы и 0h. Эта информация сообщает системе, что контроль метки тома не производится. Система использует информацию о пометку диска при изменении диска для того, чтобы определить, не нужно вернуть снятый диск. Это позволяет закончить работу с преждевременно снятой дискетой. При завершении работы с командой нужно установить слово состояния заголовке запроса.

### **Команда 2. Получение BPB.**

Эта команда выполняется только для драйверов блочных устройств. Система посылает эту команду драйверу, если необходима дополнительная информация о текущем диске. Эта необходимость возникает в двух случаях: если команда проверки носителя возвращает - 1 (диск изменено), или 0 (неизвестно, проводилась смена диска) и в буферах системы является информация, которая должна быть записана на диск.

Последовательность действий, выполняемых при обработке команды получения BPB.

1. Определяется расположение загрузочной записи на новом диске.
2. Загрузочная запись читается в память.
3. В загрузочной записи ищется блок параметров BIOS.
4. Возвращается указатель на новый BPB.
5. Если бит 11 слова атрибутов установлен, то определяется, где начинается каталог, в нем ищется пометка диска, после чего сохраняются старая и новая пометки диска.

6. Устанавливается слово состояния в заголовке запроса. Заголовок

запроса для команды Получение BPB:

```
rh2 struc
rh2_rh db size zh dup (?)
rh2_media db? ; Дескриптор носителя.
rh2_buf_ofs dw? ; Смещение области пересылки данных.
rh2_buf_seg dw? ; Сегменты области пересылки данных.
rh2_pbpbo dw? ; Смещение указателя на BPB
rh2_pbpbs dw? ; Сегмент указателя на BPB
rh2 ends
```

В функции драйвера входит чтение BPB с диска. После этого система с помощью переменной заголовка запроса rh2\_pbp\_bo и rh2\_pbp\_bs определяется указатель на новый BPB.

Блок параметров BIOS расположен в загрузочной записи. Гибких дисков это первый сектор диска, для жестких - первый сектор логического диска. Драйвер должен уметь определять начало логического диска (раздела) относительно

первого сектора жесткого. BPB располагается с байта 11 от начала загрузочной области.

Состав блока параметров BIOS:

- SS размер сектора в байтах (длина 26).
- AV единица выделения (размер кластера в секторах) (16).
- RS число зарезервированных секторов (26).
- NF число FAT на этом диске (16).
- DS размер каталога (число файлов) (26).
- TS общее число секторов (26).
- MD дескриптор носителя (16).
- FS число секторов в FAT (для каждой FAT) (26).
- ST количество секторов на диске (26).
- NH число головок.
- HS число скрытых секторов (26)
- LS число крупных секторов (46)

Адрес буфера, обусловлена сменными rh2\_buf\_ofs и rh2\_buf\_seg рассматривается по-разному в зависимости от установки бита 13 слова атрибутов заголовка драйвера. Установлен бит 13 говорит о том, что диск не является стандартным устройством IBM. Драйвера это значит, что данный буфер может быть использован как угодно. В противном случае буфер содержит начальный сектор FAT, первым элементом которого является дескриптор носителя.

### **Команда 3. IOCTL-ввода.**

Эта команда выполняется, если в слове атрибутов установлен бит 14. Строка IOCTL представляет собой данные, пересылаемые между программой и драйвером устройства. Эти данные не предназначены для отправки в устройство они являются средством взаимодействия исключительно с драйвером. Функция IOCTL-Введение используется для получения управляющей информации от драйвера. Обмен управляющими строками реализуется функцией 44h.

Последовательность действий выполняемых при обработке этой команды:

1. Одержаты адрес области передачи данных системе.
2. Одержаты счетчик передаваемых данных.
3. Записаты управляющий строку IOCTL в область передачи данных.
4. Повернуты значение счетчика.



## 5. Встановиты слово состояния заголовке запроса. Структура

данных:

rh3 struc

rh3\_rh db size rh dup (?)

rh3\_media db?

rh3\_buf\_ofs dw?

rh3\_buf\_seg dw?

rh3\_count dw? ; Число переданных элементов (секторов для блочного  
устройства, байтов для символьного).

rh3\_start dw? ; номер начального сектора (блочного устройства).

rh3 ends

Строка управляющих пересылаемых данных в драйвер через область передачи данных, не нужно переписывать в буферную область самого драйвера. Для доступа к данным драйвер может просто пользоваться указателем.

Формат управляющего строки должен быть согласован между программой и драйвером.

Пересылаться могут двоичные данные, символы ASCII или их комбинации.

Счетчик переданных данных в переменной rh3\_count является важной компонентой формата IOCTL-строк. Наличие этой переменной позволяет проверить правильность переданных данных, данные передают и принимают стороны должны использовать согласованный формат, число переданных байтов тоже должно быть согласовано.

Используя переменные rh3\_buf\_ofs и h3\_buf\_seg как указатель, драйвер может читать IOCTL-строки из области передачи данных или записывать их туда. По команде IOCTL-Введение драйвер должен вернуть IOCTL-строку в системе. Система, в свою очередь, возвращает его программе, спросила управляющую информацию.

После записи IOCTL-строки в область передачи данных, драйвер записывает в переменную rh3\_count число байтов в этой области. Затем устанавливается слово состояния в заголовке запроса и драйвер передает управление системой.

### **Команда 4. Введение.**

Команда Введение используется для пересылки данных с устройства в систему. Этапы обработки команды:

1. Одержаты адрес области передачи данных.
2. Одержаты счетчик переданных данных из заголовка запроса.
3. Прочитаты запрошенную количество данных с устройства.
4. Вернуть счетчик передаваемых данных.

## 5. Установить слово состояния заголовке запроса.

Структура данных для команды ввода:

```
rh4 struc
rh4_rh db size rh dup (?)
rh4_media db?
rh4_buf_ofs dw?
rh4_buf_seg dw?
rh4_count dw?
rh4_start dw?
rh4_valid_ofs dw?           ; Смещение отметки диске
rh4_valid_seg dw?          ; Сегмент пометки диске
rh4 ends
```

По команде введение данные с устройства читаются в область передачи данных, адрес которой содержится в переменных `rh4_buf_ofs` и `rh4_buf_seg`. Число переданных данных содержится в переменной `rh4_count`. Для символьных устройств в этой переменной записано число байтов, для блочных - секторов. В переменной `rh4_start` сохраняется номер начального сектора, если он не превышает 65535. В противном случае в переменную `rh4_start` записывается 0FFFFh, а 32 разрядный адрес сектора записывается в переменную `rh4_ls`. После завершения передачи данных драйвер определяет число переданных байтов или секторов и записывает это число в переменную `rh4_count`. Если передача прошла успешно, эта переменная не изменяется, так как передалась заданное количество данных. В противном случае значение переменной устанавливается в соответствии переданных байтов или секторов.

Это информирует систему, данные переданы полностью. В блочных драйверах, действие которых зависит от значения бита 11 слова атрибутов заголовке драйвера, должна быть реализована дополнительная обработка, то есть диск может быть заменен даже при наличии данных, которые должны быть на него записаны.

Если драйвер, получив команду ввода, устанавливает несоответствие дисков, то он прекращает выполнение команды и передает в систему сообщение об ошибке. Если установлено, что команда на ввод данных выдана для другого диска, драйвер возвращает код ошибки (0Fh) и старую пометку диске. Это позволяет системе спросить пользователя повторно поместить диск со старой пометкой диске.

Перед выходом в систему в слове состояния в заголовке запроса устанавливается бит "выполнено".

### **Команда 5. Введение, не разрушает.**

Эта команда используется только при управлении символьными устройствами. При использовании приложением функции системы 0Bh (получение состояния введения) Система посылает эту команду драйвера устройства, чтобы тот посмотрел на очередной символ. Система думает, что символьные устройства имеют входной буфер, в котором хранятся символы, которые вводятся. Драйвер читает очередной символ из этого буфера. При этом некоторые устройства позволяют прочитать символ, не удаляя его из буфера, в других устройствах при чтении символ удаляется из буфера. Термин "неразрушающий" означает, что символ должен оставаться доступным для следующей команды ввода. Не у всех устройств есть буфер. При его отсутствии драйвер должен фактически прочитать символ. Считан символ сохраняется для пересылки в системе и выполнения команды Введение. Драйвер также сохраняет символы вводимых с клавиатуры, с использованием прерывания BIOS 16h, которое возвращает два байта. Драйвер возвращает один байт и сохраняет другой. Команда ввода, не разрушает, просто вернет сохранен символ.

Этапы обработки команды:

1. Получить байт из устройства.
2. Установить слово состояния заголовке запроса. Структура

данных: rh5 struc

rh5\_rh db size rh dup (?)

rh5\_return db? ; символ, возвращается

rh5 ends

Драйвер получает байт из устройства и запоминает его в переменной rh5\_return. Если в буфере устройства нет символа, то драйвер устанавливает бит ЗАНЯТО в слове состояния. Перед выходом из драйвера устанавливается слово состояния запроса.

### **Команда 6. Состояние ввода.**

Эта команда выполняется только символьными устройствами. Она возвращает состояние входного буфера символьного устройства ввода, сообщая системе о наличии в нем символов, готовых для введения.

Этапы обработки команды:

1. Получить информацию о состоянии с устройства.

2. Записать в бит ЗАНЯТО слова состояния:

0 - если в буфере ввода устройства является символы или в устройстве нет буфера; 1 - если в буфере нет символа. Структура данных заголовка запроса: rh6 struc rh6\_len db?

, Длина запроса.

rh6\_unit db? , Номер устройству (блочного устройства).

rh6\_cmd db? ; Код команды

rh6\_status dw? ; возвращается

rh6\_res1 db? ; Резерв.

rh6\_res2 db? rh6

ends

### **Команда 7. Очистка введения.**

Команда сброса буфера ввода выполняется только символьными устройствами. Она освобождает буфер устройства.

Этапы выполнения:

1. Очистить буфер символьного устройства.

2. Установить слово состояния в заголовке запроса.

Заголовок запроса для этой команды совпадает с заголовком запроса для команды 6.

Для реализации этой команды нужно выполнить последовательность команд, обеспечивающие очистку буфера устройства. Большинство устройств не воспринимает управляющую информацию, вызывающую очистки буфера. Вместо этого драйвер просто считывает символы устройству до тех пор, пока состояние устройства не покажет, что в буфере больше нет символов. Перед завершением выполнения драйвер устанавливает слово состояния заголовке запроса.