ЛЕКЦІЯ №1. ОСНОВНІ КОМАНДИ ДРАЙВЕРІВ ПРИСТРОЇВ

Обробка команд системи.

Команда 0 - ініціалізація. Під час ініціалізації драйвер консолі викликає функцію для керування екраном. Ця можливість пояснюється тим, який перед встановленням нашого драйвера, система завантажує стандартний, який і використовується для виконання викликаних функцій (BIOS). Але з моменту ініціалізації і повернення керування системи усі запити будуть оброблятися новим драйвером.

Адреса точки розірвання це адреса першої вільної чарунки пам'яті за програмою. Ця адреса потрібна системі для того, щоб визначити, куди завантажувати драйвери пристроїв або свої власні програми після установлення нового драйвера. Драйвер може задати адресу точки розірвання усередині програми драйвера. У цьому випадку система знищить частину програми драйвера. Процедура ініціалізації виконується тільки один раз, тобто можна помістити в кінець програми й у якості адреси розірвання задати адресу початку цієї процедури.

Якщо щось перешкоджає правильній роботі драйвера консолі, то він сигналізує системі про необхідність вивантажити драйвер. Це робиться шляхом установлення зсуву точки розірвання в 0, а сегментної адреси в поточне значення cs. Це говорить системі про те, що така доступна чарунка пам'яті знаходиться на початку драйвера, наявність якого таким чином ігнорується.

Отже, при ініціалізації драйвер виводить повідомлення на екран, встановлює адресу точки розірвання в заголовку запиту і передає керування системі.

Команди з 1 по 3. Команди перевірка носія, створення BPB, IOCTL - введення не реалізуються в цьому драйвері. Для виконання 2х команд у заголовку запиту встановлюється біт виконано, при обробці третьої команди, відбувається перехід для встановлення біта помилка.

Команда 4 - введення. По цій команді здійснюється введення символів із буфера клавіатури за допомогою INT16h і передача їх у систему у буфері, визначеному в заголовку запиту. Система передає драйверу через заголовок запиту кількість символів, які повинні бути введено, а також адресу, за якою їх треба

зберегти. Переривання BIOS 16h повертає ASCII-код символу в AL, а скан-код - у AH.

Для розширених (маючих тільки скан-код) клавіш INT16h повертає 0 у AL і скан-код у AH.

Коли натискається клавіша, що має ASCII-код, із буфера клавіатури в системі повертається тільки її ASCII-код, а при натисканні розширеної клавіші система очікує 2 значення: ASCII-код=0 і скан-код.

Таким чином драйвер повинен повертати системі у складі заголовка запиту ASCII-код для всіх клавіш і скан-код для розширенних клавіш. Послідовність команд для читання символів із буферу клавіатури розміщена в циклі для підрахунку символів і повернення в систему їхньої кількості. Система не запитує більш ніж один символ. Після одержання символу за допомогою INT16h, він записується в буфер даних. Після цього робота драйвера закінчується. При цьому драйвер відновлює регістри ES і BX, тому що вони потрібні для адресації буферу даних, у якому система очікує знайти введені символи. Потім відбувається перехід для встановлення біту виконано в слові стану і передача керування.

Команда 5 – неруйнуюче введення.

Ця команда дозволяє системі перевірити наявність символу в буфері клавіатури без витягування його звідти.

Вона включена до складу команд у зв'язку з тим, що програма може викликати функцію системи перевірки пристрою введення (0Bh). Для перевірки стану буфера клавіатури драйвер використовує функцію AH=1 переривання 16h. При цьому або система повідомляється, що буфер порожній, або повертається черговий символ , що очікує , без видалення його з буферу.

Якщо функція перевірки стану повертає 0 як для ASCII, так і для скан-коду, то буфер клавіатури порожній. При відсутності символів у буфері встановлюється біт зайнято в слові стану і повертається керування системі.

Команда 6 - Стан введення.

Встановлюється біт виконано і завершується робота.

Команда 7 - Очищення введення.

Використовується для запобігання використання програмою введених заздалегідь символів. Наприклад, команда FORMAT очищує буфер клавіатури перед питанням, чи форматувати диск.

Драйвер перевіряє стан клавіатури за допомогою переривання BIOS 16h. Якщо в буфері є символ, він звідти витягається, але не пересилається в систему. Так продовжується, доки буфер не спорожніє.

Команда 8 - виведення. Для її виконання використовується переривання BIOS 10h. Реалізується лічильник виведених символів на основі циклу, з якого проводиться виклик BIOS 10h із AH=0Eh (функція виведення в режимі телетайпа). У цьому ж циклі викликається локальна процедура, що генерує випадковий колір символу. Потім відновлюються значення регістрів ES і BX, що використовувалися при зчитуванні символів із буфера даних.

Команда 9 - виведення із перевіркою. Як правило, ця команда застосовується для пристроїв, що можуть читати щойно записані дані. Її призначення - перевірити, чи правильно записані дані. Тут ця команда аналогічна команді виведення.

Команди 10-16. Виконання цих команд не потрібно в драйвері консолі, але їх обробка повинна бути передбачена у зв’язку з можливістю їх випадкової передачі драйверу.

Вихід при помилці і загальний вихід. У цій секції драйвер поміщає в заголовок запиту слово стану, що інформує систему про результати роботи драйвера. Система очікує, що в будь-якому випадку буде встановлений біт виконано. Додатково м.б. встановлено й інші біти: біт зайнято, біти помилка.

Команди драйверів пристроїв.

Проаналізуємо кожну з 26 команд.

Команда 0. Ініціалізація.

Це перша команда, яку опрацьовує драйвер. Система посилає команду ініціалізації відразу після завантаження драйвера в пам'ять.

Після того, як команда виконана, повторно вона драйверу передана ніколи не буде. Задача цієї команди – дозволити драйверу підготувати пристрій до роботи.

Виконується установлення ряду регістрів і буферів, покажчиків і лічильників. Після ініціалізації драйвер готовий до виконання інших команд.

Етапи виконання команди:

1. Ініціалізація пристрою, буферів і лічильників.

2. Виведення необов'язкового повідомлення ініціалізації.

3. Встановлення числа пристроїв, що обслуговуються драйвером (для блочних пристроїв).

4. Встановлення адреси точки розірвання.

5. Встановлення покажчика на таблицю адрес BPB (для блочних пристроїв).

6. Встановлення слова стану.

Використання функцій системи припускається тільки в процесі виконання команди ініціалізації. Дозволяється використовувати функції з номерами від 01h до 0Ch і 30h. Інші функції використовувати не можна, тому що до цього моменту сама система ще цілком не встановлена.

Запишемо структуру даних, використовуваної в команді ініціалізації (динамічна частина заголовка запиту):

```
rh0 struc ;ініціалізація (ком 0).
rh0_rh db size rh dup(?) ;          фіксована частина
rh0_nunits db ?               ;число пристроїв (для блочних)
rh0_brk_ofs dw ?              ; зсув точки розірвання
rh0_brk_seg dw ?              ; сегмент точки розірвання
rh0_bpb_tbo dw ?              ; зсув покажчика на масив BPB
rh0_bpb_tbs dw ?              ; сегмент покажчика на масив BPB
rh0_drv_ltr db ?              ; перший доступний диск
;( блочні прис.)
rh0 ends
```

У змінну rh0_nunits, яка встановлюється блочним драйвером, заноситься число пристроїв, якими він управляє. Це число повинно бути передано драйвером у систему. Воно заноситься в перший байт поля імені пристрою з заголовка запиту. У змінних rh0_brk_ofs і rh0_brk_seg утримується адреса точки розірвання, тобто адреса, якою закінчується програма драйвера. Починаючи з цієї адреси, система може завантажувати в пам'ять інші драйвери. Процедура ініціалізації використовується тільки один раз, значить її можна розташувати наприкінці драйвера, а її початок визначити як адресу розірвання. Змінні, що містять адресу

розірвання повинні визначатися у всіх драйверах. Якщо при ініціалізації драйвера виникають якісь складності, можна припинити його завантаження, визначивши у якості адреси розірвання, адресу початку драйвера. При цьому для блочних драйверів необхідно обнулити змінну rh0_nunits.

У змінних rh0_bpb_tbo і rh0_bpb_tbs утримується адреса (сегмент і зсув) таблиці BPB, яку драйвер, що управляє диском, повинен передати в систему. Для системи необхідна інформація про типи дискових накопичувачів, якими управляє драйвер. Щоб задовільнити вимогам системи, потрібно створити блок параметрів BIOS для кожного типу диска. При цьому повинна бути сформована таблиця, що містить адреси кожного блока параметрів BIOS. Адреса цієї таблиці, записана у зазначені змінні, передається системі. За цими відомостями системи і драйвер визначають, чи проводилася заміна дисків, чи не видалені вони і де розташована інформація на кожному диску.

Змінна rh0_drv_ltr містить наступну доступну літеру накопичувача. Блочні драйвери можуть використовувати цю інформацію для виведення на екран, літер накопичувачів якими ці драйвери керують.

Перед завершенням роботи драйвера необхідно сформувати слово rh_status.

Команда 1. Перевірка носія.

Ця команда діє тільки для блочних пристроїв. Вона використовується системою, якщо необхідно перевірити, чи не був замінений диск. Система посилає команду перевірки носія даних перед виконанням кожної операції читання або запису для будь-якого диска.

Структура даних для цієї команди:

```
rh1 struc ;                    Перевірка носія
rh1_rh db size rh dup( ? ) ;
rh1_media db ?                 ;дескриптор носія з блока
rh1_md_stat db ?               ;стан носія (повертається.)
rh1_volid_ofs dw ?             ;зсув помітки диску
rh1_volid_seg dw ?             ;сегмент помітки диску
rh1 ends ;
```

При визначенні того, чи не був замінений диск, виконується така послідовність дій:

1.Одержання дескриптора носія.

2.Визначення того, чи замінено диск.

3.Встановлення стану носія.

4. Встановлення слова стану в заголовку запиту.

Для визначення зміни диска можна використовувати 3 основних методи: перевірка часу, що пройшов із моменту останнього звертання, перевірка, що виконується апаратно, перевірка зміни диска і порівняння інформації про диск.

Перший метод не ефективний, тому що якщо час останнього звертання до диску перевищує 2 сек., то не можна бути впевненим у тому, що він не був замінений.

Другий метод - кращий з трьох. Накопичувачі великої ємності видають сигнал при відкритті дисководу, тому цей факт можна зафіксувати і виконати встановлення змінної стану носія. Цей сигнал подається увесь час, доки дисковод відкритий.

Третій метод потребує, щоб драйвер зберігав інформацію про диск. Ця інформація включає байт дескриптора носія, помітку диску і блок параметрів BIOS. Якщо якийсь параметр змінився за період між останнім і поточним звертанням до диску, можна припустити, що диск замінений. Проте цей засіб ненадійний. Якщо перемінити дискети, однаково форматовані, у яких дескриптори носіїв збігаються, то цей метод призведе до помилкового рішення, що диск не змінено.

Але можна обминути ці складнощі. Слово стану носія може мати три значення: - 1 - носій змінено, 0 - невідомо, чи змінено носій, 1 - носій не змінено.

Якщо неможливо визначити, чи був замінений диск, у слово стану носія rh1_md_stat заноситься 0. Для дисків усіх типів перша команда перевірки носія привласнює - 1 слову стану зміни носія. Ця операція виконується при найпершому звертанні до електронного диску точно так само, як і при звертанні до накопичувача на гнучких дисках, тому що в цей момент система не має точної інформації про диск. Наступні команди перевірки носія при роботі з жорсткими й електронними дисками привласнюють слову стану зміни носія 1.

Потрібно пам'ятати, що драйвер забезпечує перевірку зміни носія даних, якщо біт 11 слова атрибутів у заголовку драйвера встановлений.

Якщо значення змінної rh1_md_stat = - 1,то в змінні rh_1_volid_ofs і rh1_volid_seg записуються зсув і сегмент помітки диску попереднього диску. При цьому припускається, що вона зберігається драйвером. У противному випадку в ці змінні заноситься адреса рядка, який містить ім'я “No Name”, за яким йдуть чотири

прогалини і 0h. Ця інформація повідомляє системі, що контроль мітки тому не проводиться. Система використовує інформацію про помітку диску при зміні диска для того, щоб визначити, чи не потрібно повернути знятий диск. Це дозволяє закінчити роботу з передчасно знятою дискетою. При завершенні роботи з командою потрібно встановити слово стану заголовку запиту.

Команда 2. Одержання BPB.

Ця команда виконується тільки для драйверів блочних пристроїв. Система посилає цю команду драйверу, якщо необхідна додаткова інформація про поточний диск. Ця необхідність виникає в двох ситуаціях: якщо команда перевірки носія повертає - 1 (диск змінено), або 0 (невідомо, чи проводилася зміна диска) і в буферах системи є інформація, що повинна бути записана на диск.

Послідовність дій, що виконуються при опрацюванні команди одержання BPB.

1. Визначається розташування завантажувального запису на новому диску.

2. Завантажувальний запис читається в пам'ять.

3. У завантажувальному записі шукається блок параметрів BIOS.

4. Повертається покажчик на новий BPB.

5. Якщо Біт 11 слова атрибутів встановлений, то визначається, де починається каталог, у ньому шукається помітка диску, після чого зберігаються стара і нова помітки диску.

6. Встановлюється слово стану в заголовку запиту.

Заголовок запиту для команди Одержання BPB:

```
rh2 struc
rh2_rh db size zh dup(?)
rh2_media db ?              ;Дескриптор носія.
rh2_buf_ofs dw ?            ;Зсув області пересилки даних.
rh2_buf_seg dw ?            ;Сегмент області пересилки даних.
rh2_pbpbo dw ?              ;Зсув покажчика на BPB
rh2_pbpbs dw ?              ;Сегмент покажчика на BPB
rh2 ends
```

У функції драйвера входить читання BPB із диску. Після цього система за допомогою змінної заголовка запиту rh2_pbp_bo і rh2_pbp_bs визначається покажчик на новий BPB.

Блок параметрів BIOS розташований у завантажувальному записі. Для гнучких дисків це перший сектор диску, для жорстких - перший сектор логічного диску. Драйвер повинен вміти визначати початок логічного диску (розділу) стосовно

першого сектора жорсткого. BPB розташовується з байта 11 від початку завантажувальної області.

Склад блока параметрів BIOS:

- SS розмір сектора в байтах (довжина 2б).
- AV одиниця виділення (розмір кластера в секторах) (1б).
- RS число зарезервованих секторів (2б).
- NF число FAT на цьому диску (1б).
- DS розмір каталога (число файлів) (2б).
- TS загальне число секторів (2б).
- MD дескриптор носія (1б).
- FS число секторів у FAT (для кожної FAT) (2б).
- ST кількість секторів на диску (2б).
- NH число голівок.
- HS число схованих секторів (2б)
- LS число великих секторів (4б)

Адреса буфера, обумовлена змінними rh2_buf_ofs і rh2_buf_seg розглядається по-різному в залежності від встановлення біту 13 слова атрибутів заголовка драйвера. Встановлений біт 13 говорить про те, що диск не є стандартним пристроєм IBM. Для драйвера це значить, що даний буфер може бути використаний як завгодно. У противному випадку буфер містить початковий сектор FAT, першим елементом якого є дескриптор носія.

Команда 3. IOCTL-введення.

Ця команда виконується, якщо в слові атрибутів установлений біт 14. Рядок IOCTL являє собою дані, що пересилаються між програмою і драйвером пристрою. Ці дані не призначені для посилки в пристрій; вони є засобом взаємодії винятково з драйвером. Функція IOCTL-Введення використовується для одержання керуючої інформації від драйвера. Обмін керуючими рядками реалізується функцією 44h.

Послідовність дій які виконуються при обробці цієї команди:

1.Одержати адресу області передачі даних системі.

2.Одержати лічильник переданих даних.

3.Записати керуючий рядок IOCTL в область передачі даних.

4.Повернути значення лічильника.

5.Встановити слово стану заголовку запиту.

Структура даних:

```
rh3 struc
rh3_rh db size rh dup(?)
rh3_media db ?
rh3_buf_ofs dw ?
rh3_buf_seg dw ?
rh3_count dw ?     ;число переданих елементів (секторів для блочного
; пристрою, байтів для символьного).
rh3_start dw ?     ; номер початкового сектору (блочного пристрою).
rh3 ends
```

Рядок керуючих даних, що пересилається в драйвер через область передачі даних, не потрібно переписувати в буферну область самого драйвера. Для доступу до даних драйвер може просто користуватися покажчиком.

Формат керуючого рядка повинний бути узгоджений між програмою і драйвером. Пересилатися можуть двійкові дані, символи ASCII або їх комбінації.

Лічильник переданих даних у змінній rh3_count є важливою компонентою формату IOCTL-рядків. Наявність цієї змінної дозволяє перевірити вірність переданих даних, дані передають і приймають сторони повинні використовувати узгоджений формат, число переданих байтів теж повинно бути узгоджено.

Використовуючи змінні rh3_buf_ofs і h3_buf_seg як покажчик, драйвер може читати IOCTL-рядки з області передачі даних або записувати їх туди. За командою IOCTL-Введення драйвер повинен повернути IOCTL-рядок у системі. Система, у свою чергу, повертає його програмі, що запитала керуючу інформацію.

Після запису IOCTL-рядка в область передачі даних, драйвер записує в змінну rh3_count число байтів у цій області. Потім установлюється слово стану в заголовку запиту і драйвер передає керування системою.

Команда 4. Введення.

Команда Введення використовується для пересилки даних з пристрою в систему. Етапи обробки команди:

1.Одержати адресу області передачі даних.

2.Одержати лічильник переданих даних із заголовку запиту.

3.Прочитати запитану кількість даних з пристрою.

4. Повернути лічильник переданих даних.

5. Встановити слово стану заголовку запиту.

Структура даних для команди введення:

```
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_volid_ofs dw ?       ;зсув помітки диску
rh4_volid_seg dw ?       ;сегмент помітки диску
rh4 ends
```

По команді введення дані з пристрою читаються в область передачі даних, адреса якої утримується в змінних rh4_buf_ofs і rh4_buf_seg. Число переданих даних утримується в змінній. rh4_count. Для символьних пристроїв у цій змінній записане число байтів, для блочних - секторів. У змінній rh4_start зберігається номер початкового сектора, якщо він не перевищує 65535. В іншому випадку в змінну rh4_start записується 0FFFFh, а 32 розрядна адреса сектора записується в змінну rh4_ls. Після завершення передачі даних драйвер визначає число переданих байтів або секторів і записує це число в змінну rh4_count. Якщо передача пройшла успішно, ця змінна не змінюється, тому що передалася задана кількість даних. У противному випадку значення змінної встановлюється у відповідності переданих байтів або секторів.

Це інформує систему, що дані передані неповністю. У блочних драйверах, дія яких залежить від значення біту 11 слова атрибутів заголовку драйвера, повинна бути реалізована додаткова обробка, тобто диск може бути заміненим навіть при наявності даних, що повинні бути на нього записані.

Якщо драйвер, отримавши команду введення, установлює невідповідність дисків, то він припиняє виконання команди і передає в систему повідомлення про помилку. Якщо встановлено, що команда на введення даних видана для іншого диску, драйвер повертає код помилки (0Fh) і стару помітку диску. Це дозволяє системі запитати користувача повторно помістити диск зі старою поміткою диску.

Перед виходом у систему у слові стану в заголовку запиту встановлюється біт “виконано”.

Команда 5. Введення , що не руйнує.

Ця команда використовується тільки при керуванні символьними пристроями. При використанні прикладною програмою функції системи 0Bh (одержання стану введення) Система посилає цю команду драйверу пристрою, щоб той подивився на черговий символ. Система думає, що символьні пристрої мають вхідний буфер, у якому зберігаються символи, які вводяться. Драйвер читає черговий символ із цього буферу. При цьому деякі пристрої дозволяють прочитати символ, не видаляючи його з буфера, в інших пристроях при читанні символ видаляється з буфера. Термін ”неруйнуючий” означає, що символ повинен залишатися доступним для наступної команди введення. Не у всіх пристроях є буфер. При його відсутності драйвер повинен фактично прочитати символ. Зчитаний символ зберігається для пересилки в системі і виконання команди Введення. Драйвер також зберігає символи, що вводяться з клавіатури, з використанням переривання BIOS 16h, яке повертає два байти. Драйвер повертає один байт і зберігає інший. Команда введення , що не руйнує , просто поверне збережений символ.

Етапи опрацювання команди:

1. Одержати байт з пристрою.

2. Встановити слово стану заголовку запиту.

Структура даних:

```
rh5 struc
rh5_rh db size rh dup( ? )
rh5_return db ?                    ; символ, що повертається
rh5             ends
```

Драйвер одержує байт з пристрою і запам'ятовує його в змінній rh5_return. Якщо в буфері пристрою немає символу, то драйвер встановлює біт ЗАЙНЯТО в слові стану. Перед виходом із драйверу встановлюється слово стану запиту.

Команда 6. Стан введення.

Ця команда виконується тільки символьними пристроями. Вона повертає стан вхідного буфера символьного пристрою введення, повідомляючи систему про наявність у ньому символів, готових для введення.

Етапи опрацювання команди:

1. Одержати інформацію про стан з пристрою.

2. Записати в біт ЗАЙНЯТО слова стану:

0 - якщо в буфері введення пристрою є символи або в пристрої немає буфера;

1 - якщо в буфері немає символів.

Структура даних заголовка запиту:

```
rh6 struc
rh6_len db ?        ;довжина запиту.
rh6_unit db ?       ;номер пристрію (блочного пристрою).
rh6_cmd db ?        ;код команди
rh6_status dw ?     ;повертається
rh6_res1 db ?       ;резерв.
rh6_res2 db ?
rh6         ends
```

Команда 7. Очищення введення.

Команда скидання буфера введення виконується тільки символьними пристроями. Вона звільняє буфер пристрою.

Етапи виконання:

1. Очистити буфер символьного пристрою.

2. Встановити слово стану в заголовку запиту.

Заголовок запиту для цієї команди збігається з заголовком запиту для команди 6.

Для реалізації цієї команди потрібно виконати послідовність команд, що забезпечують очищення буферу пристрою. Більшість пристроїв не сприймає керуючу інформацію, що викликає очищення буфера. Замість цього драйвер просто зчитує символи з пристрію доти, доки стан пристрою не покаже, що в буфері більше немає символів. Перед завершенням виконання драйвер установлює слово стану заголовку запиту.