

## ЛЕКЦІЯ №2. ДОДАТКОВІ КОМАНДИ ДРАЙВЕРІВ ПРИСТРОЇВ

### Команда 8. Виведення.

Ця команда використовується для пересилки даних в пристрій.

Етапи виконання:

1. Одержати адресу області передачі даних.
2. Одержати лічильник передач із заголовку запиту.
3. Передати запитаний об'єм інформації з області передачі даних в пристрій.
4. Повернути лічильник передач.
5. Встановити слово стану заголовку запиту.

Структура даних (загол. запита)

```
rh8 struc
```

```
rh8_rh db size rh dup ( ? )
```

```
rh8_media db ?
```

```
rh8_buf_ofs dw ?
```

```
rh8_buf_seg dw ? ;сегмент області передачі даних
```

```
rh8_count dw ? ;число переданих елементів
```

```
rh8_start dw ? ;номер початкового сектора (бл.)
```

```
rh8_volid_ofs dw ? ;зсув помітки диску.
```

```
rh8_volid_seg dw ?
```

```
rh8 ends
```

При виконанні цієї команди драйвер насамперед одержує покажчик на область передачі даних. (Змінні `rh8_buf_ofs` і `rh8_buf_seg` містять зсув і сегмент буферу з переданими даними). Потім драйвер витягає зі змінної `rh8_count` кількість переданих даних.

Для символічних пристроїв - число байтів, для блочних - секторів. Для блочних пристроїв драйвер повинен перетворити відносний номер сектора на номери голівки, доріжки і сектора.

Якщо запис пройшов успішно, драйвер установлює слово стану і передає керування системою. В іншому випадку драйвер повинен у слові стану заголовку

запиту встановити біт помилки і номер помилки. Крім цього, у змінну rh8\_count записується кількість даних, переданих до моменту виникнення помилки.

#### **Команда 9. Виведення з перевіркою.**

Ця команда виконується як символьними, так і блочними пристроями. Відмінність від команди виведення полягає в тому, що потрібно прочитати дані після того, як вони будуть записані на пристрій. Цю команду використовують, якщо необхідно переконатися, що дані були записані правильно. Структура даних збігається зі структурою даних команди 8.

Етапи виконання:

1. Якщо пристрій не може читати щойно записані дані, то перейти до процедури виведення.
2. Якщо пристрій може читати щойно записані дані, то встановити прапор читання. Перейти до процедури виведення, модифікованої так, щоб забезпечити читання даних при встановленому прапорі.

#### **Команда 10. Стан виведення.**

Виконується тільки символьними пристроями. Команда призначена для передачі в систему інформації про стан пристрою.

Структура даних збігається зі структурою даних команди 6.

Етапи виконання:

1. Одержати інформацію про стан пристрою.
2. Записати в біт ЗАЙНЯТО слова стану:
  - 0 - якщо пристрій вільний або буфер не повний;
  - 1 - якщо пристрій зайнятий або буфер заповнений.
3. Встановити слово стану заголовку запиту.

Перед виконанням операції виведення система посилає драйверу команду стану виведення. За результатами її виконання система вирішує, чи можна посилати відразу команду виведення або почекати і знову перевірити стан виведення

#### **Команда 11. Очищення виведення.**

Виконується тільки символьними пристроями. Використовується для очищення буферу виведення пристрою.

Етапи обробки команди:

1. Для пристроїв, що мають буфер, виконати операції з очищення буфера.
2. Встановити слово стану заголовку запиту.

Структура даних збігається зі структурою даних команди 6.

Щоб виконати цю команду драйвер виконує послідовність команд, що витягають символи з буфера пристрою. При відсутності буфера в пристрої драйвер не виконує ніяких дій.

#### **Команда 12. IOCTL - виведення.**

Ця команда виконується як блочними так і символічними драйверами, якщо в слові атрибутів установлений біт 14. Вона використовується для посилки даних драйверу безпосередньо з програми. Ці дані призначаються не для виведення на пристрій, а для керування пристроєм. Драйвер може використовувати їх будь-яким засобом.

Формат керуючої інформації повинен бути узгоджений між драйвером пристрою і програмою, яка використовує виведення функцією системи.

Структура даних для команди збігається зі структурою даних команди 3 (IOCTL - введення).

Етапи обробки:

1. Одержати адресу області передачі даних.
2. Одержати з заголовка запиту кількість переданих даних.
3. Декодувати керуючий рядок, що утримується в області передачі даних.
4. Встановити слово стану заголовку запиту.

Виконання команди починається з отримання адреси області передачі даних. Потім аналізується довжина командного рядка, який утримується в заголовку запиту. Цей розмір дозволяє драйверу визначити, чи правильно складений керуючий рядок. Після цього драйвер обробляє керуючий рядок, виконуючи запитані операції. Ці операції визначаються типом керованого пристрою. Наприклад, можна використовувати керуючі рядки для зміни швидкості передачі даних в пристроях і для керування форматом при роботі з принтером.

#### **Команда 13. Відкриття пристрою.**

Обробляється як символічними, так і блочними драйверами. Біт 11 у слові атрибутів заголовка драйвера повинен бути встановлений. Ця команда посилається від системи драйверу кожного разу, коли програма відкриває пристрій. Її можна

використовувати, щоб простежити, скільки разів був відкритий пристрій. Використовуючи її разом із командою закриття пристрою, можна встановити режим доступу до пристрою. Наприклад, щоб пристрій був одночасно доступний тільки одному користувачеві, можна заборонити його повторне відкриття.

Структура даних для цієї команди збігається зі структурою даних для команди 6 (стан введення).

Етапи виконання:

1. Збільшити на 1 лічильник відкриттів пристрою.
2. Для символьних пристроїв - видати керуючий рядок.
3. Встановити слово стану заголовку запиту.

Лічильник у драйвері збільшується при виконанні команди відкриття пристрою і зменшується при виконанні команди закриття пристрою. Це дозволяє драйверу визначити, чи вільний пристрій. Команда відкриття пристрою використовується для ініціалізації символьних пристроїв. Наприклад, можна ініціалізувати принтери посилкою команди, що встановлює початок сторінки, або завантажує стандартний шрифт.

Для блочних пристроїв лічильник відкриттів пристрою використовується інакше. При встановленому біті 11 слова атрибутів драйвер блочного пристрою повинен визначити, чи не було неприпустимої зміни диску. З цією метою можна використовувати лічильник відкриттів пристрою. Диск може бути змінений, якщо в лічильнику відкриттів пристроїв - нуль (тобто для диску немає відкритих файлів). Доки значення лічильника не дорівнює 0, диски не можна замінювати, тому що для них є відкриті файли.

#### **Команда 14. Закриття пристрою.**

Виконується при встановленому біті 11 у слові атрибутів як для символьних, так і для блочних пристроїв.

Структура даних збігається зі структурою даних команди 6.

Етапи обробки команди:

1. Зменшити на 1 лічильник відкриттів пристрою.
2. Встановити слово стану заголовку запиту.

При роботі з символьними пристроями команду можна використовувати для передачі пристрою необов'язкового рядку. Наприклад, при завершенні роботи

принтера можна подати команду прогнати сторінку. Пристрої CON:, AUX:, PRN: ніколи не закриваються. Якщо лічильник містить 0, то пристрій вільний.

### **Команда 15. Змінний носій.**

Ця команда обробляється тільки блочними пристроями. Біт 11 слова атрибутів заголовка запиту повинен бути встановлений.

Команда посилається системою, коли програма використовує підфункцію визначення типу пристрою 08h функції 44h. Ця команда використовується в програмах для визначення можливості зміни диску.

Структура даних збігається зі структурою даних команди 6.

Етапи виконання:

1. Записати в біт ЗАЙНЯТО слова стану пристрою:
  - 0 - якщо диск змінний;
  - 1 - диск незмінний.

2. Встановити слово стану заголовку запиту.

### **Команда 16. Виведення доки не зайнято.**

Використовується при роботі з символьними пристроями. Біт 13 слова атрибутів заголовку драйвера повинен бути встановлений.

Ця команда використовується спулерами принтерів для передачі в символьний пристрій вихідних даних доти, доки від пристрою не буде отримане повідомлення, що він зайнятий.

Структура даних:

```
rh16 struc
rh16_rh db size rh dup ( ? )
rh16_media db?
rh16_buf_ofs dw?
rh16_buf_seg dw?
rh16_count dw?
rh16 ends
```

Етапи виконання:

1. Одержати адресу області передачі даних.
2. Одержати кількість переданих даних.
3. Передавати запитаний об'єм даних з області передачі даних в пристрій доти, доки не буде отриманий сигнал ЗАЙНЯТО.

4. Повернути кількість переданих даних.
5. Встановити слово стану заголовку запиту.

### **Команди 17 і 18.**

Ці команди не визначені. Вони зарезервовані для майбутніх версій системи.

### **Команда 19. Узагальнений IOCTL.**

Використовується при роботі з блочними та символьними пристроями в системі при встановленому біті 0 слова атрибутів заголовку драйвера. Ця команда видається, якщо програма викликає підфункцію 0Dh функції 44h.

Структура даних:

```
rh19_struct
rh19_rh db size rh dup ( ? )
rh19_major db ?
rh19_minor db ?
rh19_SI dw ?           ;вміст регістра SI
rh19_DI dw ?           ;вміст регістра DI
rh19_pkt_ofs dw ?      ;зсув запиту
rh19_pkt_seg dw ?      ;узагальненого IOCTL
rh19                      ends
```

Етапи виконання:

1. Одержати старший і молодший функціональні коди.
2. Обробити запит на виконання процедури.
3. Повернути значення лічильника переданих даних.
4. Встановити слово стану заголовка запиту.

Призначена для виконання стандартних операцій по керуванню в/в. Молодші функціональні коди визначають операції, що не входять у саму систему. Наприклад, форматування диску виконується програмами-утілітами.

Старший і молодший функціональні коди утворюються зі змінних rh19\_major і rh19\_minor. Після одержання цих кодів потрібно переконатися, що старший код вірний.

Старші коди (коди категорій пристроїв):

- 01h - Послідовний пристрій.
- 03h - Консоль.
- 05h - Паралельний принтер.

08h - Диск.

Молодші коди для символьних пристроїв:

45h - Встановити лічильник числа ітерацій.

4Ah - Вибрати кодову сторінку.

4Ch - Почати підготування кодової сторінки.

4Dh - Закінчити підготування кодової сторінки.

65h - Прочитати лічильник числа ітерацій

6Ah - Запитати обрану кодову сторінку.

6Bh - Запитати список підготовлених сторінок.

Молодші коди для блочних пристроїв:

40h - Встановити параметри пристрою.

60h - Одержати параметри пристрою.

41h - Записати доріжку на логічному диску.

42h – Форматувати і перевірити доріжку на логічному диску.

61h - Прочитати доріжку на логічному диску.

62h - Перевірити доріжку на логічному диску.

46h - Встановити помітку диску.

66h - Одержати помітку диску.

68h - Вивести тип носія.

**Команди 20, 21, 22 не визначені. Зарезервовані.**

**Команда 23. Одержання логічного пристрою.**

Виконується драйверами блочних пристроїв. Біт 6 у слові атрибутів повинний бути встановлений.

Структура даних:

```
rh23 struc
rh23_rh db size rh dup  (?)
rh23_io db ?
;введення (номер пристрою)
;виведення (останній пристрій)
rh23_dev_cmd dw ? ;код помилки
rh23_dev_stat dw ? ;стан
rh23_reserved dd ?
rh23 ends
```

Етапи обробки:

1. Одержати вхідний код пристрою.
2. Повернути останній пристрій, до якого було звертання.
3. Встановити слово стану заголовку запиту.

При виконанні команди код логічного пристрою витягається зі змінної `rh23_io`. Драйвер визначає, чи призначені цьому логічному пристрою інші логічні диски. Якщо інші логічні диски не призначалися, то драйвером повертається 0 у змінній `rh23_io`. Інакше драйвер визначає логічний диск, до якого було останнє звертання. У змінну `rh23_io` записується 1 для пристрою A:; 2 - для B: і т.д. Іншими словами, команда запитує драйвер, які ще літери використовувалися для звертання до того ж фізичного пристрою.

#### **Команда 24. Встановлення логічного пристрою.**

Команда для блочних пристроїв при встановленому біті 6 слова атрибутів. Ця команда дозволяє визначити декілька літер для логічного пристрою.

Структура даних збігається зі структурою команди 23.

Етапи виконання:

1. Одержати вхідний код пристрою.
2. Зберегти цей код.
3. Встановити слово стану заголовку запиту.

Якщо драйвер не розпізнає ім'я логічного пристрою як альтернативну літеру для контрольованих пристроїв, він повертає 0 у змінній `rh24_io`. Літери, що позначають накопичувач, нумеруються починаючи з 1.

Використовувати зазначені можливості в програмі можна за допомогою виклику функції системи 44h для встановлення й одержання логічних дисків.

#### **Команда 25. IOCTL - запит.**

Обробляється як символьними, так і блочними пристроями, при встановленому біті 7 слова атрибутів. Ця команда використовується для виконання запиту до драйверів - чи підтримують вони конкретні функції IOCTL.

Структура даних як у ком. 19.

Етапи виконання:

1. Одержати молодший функц. код.
2. Встановити біт виконано в слові стану, якщо ця процедура підтримується.
3. Встановити слово стану заголовку запиту.



Якщо драйвер не підтримує процедуру з номером молодшого функціонального коду, то в слові стану встановлюється біт помилка, а код помилки - 3 (невідома команда).

#### Вихід із драйвера.

Перед передачею керування системі драйвер повинен встановити слово стану заголовку запиту і відновити регістри, збережені при виході. Перед виходом із драйвера завжди встановлюється біт виконано. Це показує системі, що команда виконана правильно. Команда стан введення, стан виведення, змінний носій і виведення, доки не зайнято установлюють біт зайнято.

Якщо драйвер визначить, що відбулася помилка, встановлюється біт помилка. Крім того, поле коду помилки повинно містити відповідний код.