

Лабораторная работа № 13 (5)

Тема: Разработка программ с использованием сервисных функций ОС MS DOS (прерывание 21h).

Цель: Освоить разработку программ с использованием сервиса предоставляемого ОС MS DOS посредством программных прерываний.

Краткая теория

Прерывание – событие, заставляющее процессор переключиться с выполнения одной программы на другую (после завершения выполнения текущей команды), а затем вернуться в исходную программу (на выполнение следующей команды). Это происходит «незаметно» для исходной программы.

Прерывания делятся на три вида:

- Аппаратные прерывания (внешние прерывания).
- Исключения (внутренние прерывания).
- Программные прерывания.

Аппаратное прерывание - это асинхронное событие, генерируемое внешним (по отношению к процессору) устройством. Посредством аппаратных прерываний аппаратура «информирует» центральный процессор о том, что произошло какое-либо событие, требующее немедленной реакции, например:

- нажатие клавиши клавиатуры,
- завершение асинхронной операции ввода-вывода,
- сигнал таймера и т.д.

Каждый тип аппаратных прерываний имеет собственный номер (irq – interrupt query), однозначно определяющий источник прерывания. Обработка аппаратного прерывания не должна учитывать, какой процесс является текущим.

Аппаратные прерывания подразделяются на маскируемые и немаскируемые.

- Немаскируемые прерывания – системные прерывания на которые процессор должен отреагировать в любом случае.
- Маскируемые прерывания – реакция на них может быть заблокирована. Эти прерывания разрешены если IF=1, и запрещены если IF=0.

Управление разрешением обработки маскируемых прерываний осуществляется с помощью команд:

- CLI – запретить обработку маскируемых прерываний,
- STI – разрешить обработку маскируемых прерываний.

Исключение – синхронное событие, возникающее в результате попытки выполнения программой команды, которая по каким-то причинам не может быть выполнена до конца. Примеры исключений:

- деление на ноль,
- переполнение стека,
- ошибка обращения к памяти и т.д.

Программное прерывание (пользовательское прерывание) - это синхронное событие, которое может быть повторено при выполнении одного и того же программного кода. Программное прерывание вызывается явно используя команду INT.

Основное предназначение программных прерываний – обращение к сервисам BIOS (Basic Input Output System) и операционной системы (DOS, Linux и т.д.).

Адреса обработчиков прерываний (вектор прерываний) в реальном режиме располагаются по нулевому адресу (0000h:0000h) и занимают 1Кб ($256 \times 4 = 1024$), где

- 256 – максимальное количество прерываний,

– 4 – размер одного адреса (сегмент:смещение).

Во всех программных прерываниях в АН указывается номер функции. В таблице 1 приведен перечень основных программных прерываний BIOS и MS DOS.

Таблица 1 – Прерывания BIOS и MS DOS

№	Тип	Описание	№	Тип	Описание
00h	HRD	Деление на 0	16h	BIOS	Ввод – вывод клавиатуры
01h	HRD	Пошаговое прерывание	17h	BIOS	Ввод – вывод принтера
02h	HRD	Немаскируемое прерывание	18h	BIOS	ROM Basic
03h	HRD	Прерывание по точке останова	19h	BIOS	Загрузка
04h	HRD	Прерывание при переполнении	1ah	BIOS	Ввод – вывод таймера и часов
05h	HRD	Печать экрана	1bh	BIOS	Обработчик Ctrl-Break
08h	HRD	Прерывание таймера	1ch	BIOS	Прерывание по таймеру
09h	HRD	Прерывание клавиатуры	20h	DOS	Завершить программу
0eh	HRD	Прерывание дисководов	21h	DOS	Сервис DOS
10h	BIOS	Видео сервис	25h	DOS	Абсолютное чтение диска
11h	BIOS	Список оборудования	26h	DOS	Абсолютная запись диска
12h	BIOS	Размер используемой памяти	27h	DOS	Создание резидента
13h	BIOS	Дисковый ввод – вывод	28h	DOS	Квант времени DOS
14h	BIOS	Ввод – вывод через COM	2eh	DOS	Выполнить команду DOS
15h	BIOS	Расширенный сервис AT	2fh	DOS	Мультиплексное прерывание

Дополнительную информацию по прерываниям можно посмотреть в электронном справочнике HELP.RUS.

В прерывании 21h реализовано большинство сервисных функций, предоставляемых операционной системой MS DOS прикладным программам. Здесь содержатся функции:

- Консольного ввода – вывода информации.
- Файлового ввода и вывода.
- Работы с динамической памятью.
- Работы с датой и временем.
- и др.

В таблице 2 приведен перечень основных функций прерывания 21h

№	Описание	№	Описание
00h	Завершить программу	3ch	Создать файл
01h	Ввод символа с клавиатуры	3dh	Открыть файл
02h	Вывод символа на экран	3eh	Закрыть файл
07h	Ввод символа без эха и проверки Ctrl-Break	3fh	Читать данные из файла
08h	Ввод символа без эха, но с проверкой Ctrl-Break	40h	Записать данные в файл
09h	Вывод строки	41h	Удалить файл
0ah	Буферизированный ввод строки	42h	Установить позицию в файле
0bh	Проверка стандартного ввода	43h	Установить атрибуты файла
08h	Очистка буфера ввода, а затем ввод	48h	Выделить динамический блок памяти
25h	Установить вектор прерывания	49h	Освободить динамический блок памяти
2ah	Получить системную дату	4ah	Сжать или расширить блок памяти
2bh	Установить системную дату	4bh	Вызывать программу
2ch	Получить системное время	4ch	Завершить программу
2dh	Установить системное время	56h	Переименовать файл
35h	Получить вектор прерывания	62h	Получить адрес PSP данной программы

Дополнительную информацию по функциям прерывания 21h можно посмотреть в электронном справочнике HELP.RUS.

Для каждой программы при ее запуске создается блок PSP (Program segment prefix – Префикс программного сегмента), который содержит следующую информацию:

Смещение	Размер, байт	Описание
00h	2	INT 20h – можно использовать для выхода
02h	2	Вершина доступной памяти системы в параграфах
04h	1	Резерв
05h	5	FAR CALL к диспетчеру функций DOS
06h		Доступные байты в программном сегменте (только для COM)
0ah	4	Адрес завершения
0eh	4	Адрес обработки Ctrl-Break
12h	4	Обработчик критических ошибок
16h	16h	Резервная область DOS
2ch	2	Сегментный адрес окружения DOS
2eh	2eh	Резервная область DOS
5ch	10h	Форматированная область параметров 1 (как FCB для 1-го пар.)
6ch	14h	Форматированная область параметров 2 (как FCB для 2-го пар.)
80h	1	Длина области неформатированных параметров (UPA)
81h	7fh	Область неформатированных параметров

Структура PSP находится перед самой программой. При загрузке сегментные регистры ES и DS установлены на нее. Для COM-программ область PSP располагается в начале сегмента загруженной программы (поэтому во всех программах COM-формата генерируется смещение на 100h байт с помощью директивы ORG). В процессе выполнения программы адрес PSP можно получить с помощью функции 62h прерывания 21h.

ПРИМЕР: Написать программу копирования файла. Программа запрашивает ввод имен исходного и целевого файлов. Осуществляет копирование данных из исходного файла в целевой файл.

```
.486
model small
Data SEGMENT use16
    ASSUME ds: Data
mess1 db 'Source file name: ','$'
mess2 db 0dh, 0ah, 'Destination file name: ','$'
mess3 db 0dh, 0ah, 'Error open source file!$'
mess4 db 0dh, 0ah, 'Error create destination file!$'
mess5 db 0dh, 0ah, 'Read file error!$'
mess6 db 0dh, 0ah, 'Write file error!$'
file1 db 100, 100 dup(0)
file2 db 100, 100 dup(0)
hand1 dw 0
hand2 dw 0
buff db 512 dup(?)
Data ENDS
Stk SEGMENT use16 STACK
    ASSUME ss: Stk
    db 256 dup(0)
Stk ENDS
Code SEGMENT use16
    ASSUME cs: Code
start:
    mov     ax, data           ;В AX адрес сегмента данных
    mov     ds, ax            ;DS устанавливаем на AX
;----- Ввод имени и открытие исходного файла -----
    mov     ah, 09h           ;Вывод приглашения ввода имени
```

```

        mov     dx, offset mess1 ;исходного файла
        int     21h
        mov     ah, 0ah          ;Ввод имени файла
        mov     dx, offset file1
        mov     si, dx
        inc     si
        int     21h
        xor     ax, ax           ;Запись вместо 0dh значения 0
        mov     al, ds:[si]
        add     si, ax
        mov     byte ptr ds:[si+1], 0
        mov     ah, 3dh          ;Открытие файла для чтения
        add     dx, 2
        xor     al, al
        int     21h
        jnc     next0
        mov     ah, 09h          ;Если файл открыть не удалось
        mov     dx, offset mess3
        int     21h
        jmp     exit0
next0:                                ;Успешное открытие файла
        mov     word ptr hand1, ax
;----- Ввод имени и открытие результирующего файла -----
        mov     ah, 09h          ;Вывод приглашения ввода имени
        mov     dx, offset mess2 ;целевого файла
        int     21h
        mov     ah, 0ah          ;Ввод имени файла
        mov     dx, offset file2
        mov     si, dx
        inc     si
        int     21h
        xor     ax, ax           ;Запись вместо 0dh значения 0
        mov     al, ds:[si]
        add     si, ax
        mov     byte ptr ds:[si+1], 0
        mov     ah, 3ch          ;Создание файла
        add     dx, 2
        xor     cx, cx
        int     21h
        jnc     next1
        mov     ah, 09h          ;Если файл создать не удалось
        mov     dx, offset mess4
        int     21h
        jmp     exit0
next1:                                ;Успешное создание файла
        mov     word ptr hand2, ax
;----- Цикл чтения и записи -----
loop0:   mov     ah, 3fh          ;Чтение из исходного файла
        mov     bx, hand1
        mov     dx, offset buff
        mov     cx, 512
        int     21h
        jnc     next2
        mov     ah, 09h          ;Вывод сообщения, если ошибка
        mov     dx, offset mess5
        int     21h
        jmp     next4
next2:   mov     cx, ax           ;Запись в целевой файл
        mov     ah, 40h
        mov     bx, hand2
        mov     dx, offset buff
        int     21h
        jnc     next3
        mov     ah, 09h          ;Вывод сообщения, если ошибка

```

```

                mov     dx, offset mess6
                int     21h
                jmp     next4
next3:          cmp     ax, 512             ;Проверка на конец файла
                jnz     next4
                jmp     loop0             ;Переход к следующей итерации
next4:          ;Закрытие файлов
                mov     ah, 3eh
                mov     bx, hand1
                int     21h
                mov     ah, 3eh
                mov     bx, hand2
                int     21h
exit0:          ;Завершение программы
                mov     ax, 4c00h
                int     21h
Code ENDS
end start

```

Ход работы

Во всех заданиях этой лабораторной работы ввод исходных данных осуществляется с клавиатуры с помощью функции 0ah прерывания 21h, а вывод данных – с помощью функции 09h прерывания 21h. Часть входных данных программы может передаваться через параметры командной строки. Во всех заданиях предполагается, что размер исходного файла не превышает 64Кб. Программу генерации исходного файла данных можно написать на языке С.

Задание 1

Разработать программу согласно варианту задания. При обработке использовать динамически выделяемую память запрещено. Все файлы содержат информацию в бинарном виде (если другое не оговорено в условии задания). Имена файлов вводятся с клавиатуры или кодируются жестко, если другое не оговорено в условии задания. Варианты заданий:

1	Дан файл, содержащий целые 16-ти разрядные числа. Вывести на экран максимальное значение в файле и количество раз, которое оно встречается. Имя файла вводит пользователь.
2	Дан файл, содержащий текст. Определить сколько знаков препинания содержится в файле. Полученное значение вывести на экран. Имя файла вводит пользователь.
3	Дан файл, содержащий целые 16-ти разрядные числа. Записать в новый файл все положительные значения из исходного файла. Имя исходного файла вводит пользователь. Имя результирующего файла – result.dat.
4	Дан файл, содержащий текст. Определить количество строк в файле. Полученное значение вывести на экран. Имя исходного файла вводит пользователь.
5	Дан файл, содержащий целые 16-ти разрядные числа. Вычислить сумму четных положительных значений и записать ее в текстовом виде в новый файл. Имя исходного файла вводит пользователь. Имя нового файла – res.txt.
6	Дан файл, содержащий текст. Определить наиболее часто встречающуюся в нем цифру. Полученное значение вывести на экран. Имя исходного файла вводит пользователь.
7	Дан файл, содержащий целые 16-ти разрядные числа. Записать в новый файл все значения из исходного файла кратные K. Число K кодируется жестко в программе. Имя исходного файла вводит пользователь. Имя нового файла – res.dat.
8	Дан файл, содержащий произвольный текст. Записать в новый файл количество букв латинского алфавита в исходном файле (заглавные и строчные буквы подсчитываются отдельно). Имя исходного файла вводит пользователь. Имя нового файла – res.dat.
9	Дан файл, содержащий целые 8-ми разрядные числа. Записать в новый файл все значения из исходного файла, в которых ровно 3 разряда в бинарном представлении установлены в 1. Имя исходного файла вводит пользователь. Имя нового файла – res.dat.

10	Дан файл, содержащий целые 16-ти разрядные числа. Записать в файл 1.txt все значения у которых младшие два разряда установлены в единицу. Имя исходного файла вводит пользователь.
11	Дан файл, содержащий текст. Определить сколько букв латинского алфавита содержится в файле. Полученное значение вывести на экран. Имя файла вводит пользователь.
12	Дан файл, содержащий целые 16-ти разрядные числа. Записать в файл 1.txt все значения, у которых количество установленных бит в старшем байте равно количеству установленных бит в младшем байте. Имя исходного файла вводит пользователь.
13	Дан текстовый файл. Определить и записать в файл res.dat количество слов начинающихся с заглавной буквы латинского алфавита. Имя исходного файла вводит пользователь.
14	Дан текстовый файл. Определить и записать в файл res.dat количество пробелов, табуляций и переводов строк (подсчитывать отдельно). Имя исходного файла вводит пользователь.
15	Дан файл, содержащий целые 8-ми разрядные числа. Записать в файл 1.dat все положительные значения, а в файл 2.dat – все отрицательные значения из исходного файла. Имя исходного файла вводит пользователь.
16	Дан файл, содержащий текст. Определить наиболее часто встречающуюся в нем букву латинского алфавита (независимо от регистра). Полученное значение вывести на экран. Имя исходного файла вводит пользователь.
17	Дан файл, содержащий текст. Определить самую длинную строку в файле и вывести ее на экран. Имя исходного файла вводит пользователь. Предполагается, что в строки в файле не превышают 100 символов.
18	Дан файл, содержащий целые 8-ми разрядные числа. Вычислить и записать в файл res.txt в текстовой форме сумму положительных четных значений. Имя исходного файла вводит пользователь.
19	Дан файл, содержащий целые 16-ти разрядные числа. Записать в файл 1.txt все значения у которых старшие три разряда установлены в единицу. Имя исходного файла вводит пользователь.
20	Дан файл, содержащий 16-ти разрядные целые числа. Вывести на экран минимальное значение в файле и количество раз, которое оно встречается. Имя файла вводит пользователь.
21	Дан файл, содержащий целые 8-ми разрядные числа. Вычислить и записать в файл res.txt в текстовой форме сумму значений, кратных 7. Имя исходного файла вводит пользователь.
22	Дан файл, содержащий целые 8-ми разрядные числа. Записать в новый файл все значения из исходного файла, в которых ровно 5 разрядов в бинарном представлении установлены в 0. Имя исходного файла вводит пользователь. Имя нового файла – res.dat.
23	Дан файл, содержащий текст. Определить сколько цифр содержится в файле. Полученное значение вывести на экран. Имя файла вводит пользователь.
24	Дан текстовый файл. Определить и записать в файл res.dat количество слов начинающихся с цифры. Имя исходного файла вводит пользователь.
25	Дан файл, содержащий целые 16-ти разрядные числа. Вывести на экран сумму всех значений из исходного файла кратных K. Число K кодируется жестко в программе. Имя исходного файла вводит пользователь. Имя нового файла – res.dat.
26	Дан файл, содержащий целые 16-ти разрядные числа. Записать в новый файл все отрицательные значения из исходного файла. Имя исходного файла вводит пользователь. Имя результирующего файла – result.dat.
27	Дан файл, содержащий целые 16-ти разрядные числа. Вывести на экран количество значений, у которых количество установленных бит в старшем байте больше количества установленных бит в младшем байте. Имя исходного файла вводит пользователь.

28	Дан файл, содержащий целые 16-ти разрядные числа. Вычислить сумму нечетных отрицательных значений и записать ее в текстовом виде в новый файл. Имя исходного файла вводит пользователь. Имя нового файла – res.txt.
29	Дан файл, содержащий целые 8-ми разрядные числа. Записать в файл 1.dat все четные значения, а в файл 2.dat – все нечетные значения из исходного файла. Имя исходного файла вводит пользователь.
30	Дан файл, содержащий текст. Определить наиболее часто встречающийся в нем знак препинания. Полученное значение вывести на экран. Имя исходного файла вводит пользователь.

Задание 2

Разработать программу согласно варианту задания. При обработке допускается использовать динамически выделенную память. Все файлы содержат информацию в бинарном виде (если другое не оговорено в условии задания). Имена файлов вводятся с клавиатуры или кодируются жестко, если другое не оговорено в условии задания. Варианты заданий:

1	Дан текстовый файл. Удалить из файла все знаки пунктуации. Имя исходного файла вводит пользователь.
2	Дан файл, содержащий целые 16-ти разрядные числа. Упорядочить все значения в файле по возрастанию. Имя исходного файла вводит пользователь.
3	Дан текстовый файл. Удалить из файла все двойные пробелы. Имя исходного файла вводит пользователь.
4	Дан файл, содержащий целые 16-ти разрядные числа. Удалить из файла все четные значения. Имя исходного файла вводит пользователь.
5	Дан текстовый файл. Сделать первую букву каждого слова в файле заглавной. Имя исходного файла вводит пользователь.
6	Дан файл, содержащий целые 16-ти разрядные числа. Вставить перед каждым отрицательным значением в файле нулевое значение. Имя исходного файла вводит пользователь.
7	Дан текстовый файл. Заменить в файле все символы табуляции двумя пробелами. Имя исходного файла вводит пользователь.
8	Дан файл, содержащий целые 16-ти разрядные числа. Удалить каждое третье четное значение в файле. Имя исходного файла вводит пользователь.
9	Дан текстовый файл. Заменить в файле все символы табуляции двумя пробелами. Имя исходного файла вводит пользователь.
10	Дан файл, содержащий целые 16-ти разрядные числа. Вставить после каждого положительного элемента нулевое значение. Имя исходного файла вводит пользователь.
11	Дан текстовый файл. Упорядочить все символы латинского алфавита в обратном алфавитном порядке, а остальные символы удалить. Имя исходного файла вводит пользователь.
12	Дан файл, содержащий целые 16-ти разрядные числа. Переписать в обратном порядке все значения в файле, расположенные между максимальным и минимальным значением. Имя исходного файла вводит пользователь.
13	Дан текстовый файл. Заменить все цифры в файле символом '#'. Имя исходного файла вводит пользователь.
14	Дан файл, содержащий целые 16-ти разрядные числа. Удалить из файла все значения, расположенные после последнего отрицательного значения. Имя исходного файла вводит пользователь.
15	Дан текстовый файл. Если букв латинского алфавита в файле меньше чем цифр, то заменить их символом '~'. В противном случае заменить все цифры символом '~'. Имя исходного файла вводит пользователь.

16	Дан текстовый файл. Заменить все символы пунктуации в файле символом пробела. Имя исходного файла вводит пользователь.
17	Дан файл, содержащий целые 16-ти разрядные числа. Упорядочить все значения в файле по убыванию. Имя исходного файла вводит пользователь.
18	Дан текстовый файл. Вставить после каждой цифры в файле символ '!'. Имя исходного файла вводит пользователь.
19	Дан файл, содержащий целые 16-ти разрядные числа. Удалить из файла все значения кратные K. Число K жестко кодируется в программе. Имя исходного файла вводит пользователь.
20	Дан текстовый файл. Удалить из файла все переводы строк, заменив их символом табуляции. Имя исходного файла вводит пользователь.
21	Дан файл, содержащий целые 16-ти разрядные числа. Удалить из файла все значения меньшие среднего значения всего файла. Имя исходного файла вводит пользователь.
22	Дан текстовый файл. Удалить из файла все заглавные буквы латинского алфавита. Имя исходного файла вводит пользователь.
23	Дан файл, содержащий целые 16-ти разрядные числа. Перед каждым четвертым отрицательным значением вставить максимальное значение в файле. Имя исходного файла вводит пользователь.
24	Дан текстовый файл. Вставить перед каждой цифрой символ пробела. Имя исходного файла вводит пользователь.
25	Дан файл, содержащий целые 16-ти разрядные числа. Удалить все положительные числа, являющиеся значением 2 в целой степени. Имя исходного файла вводит пользователь.
26	Дан текстовый файл. Переставить символы в файле так, чтобы сначала были буквы латинского алфавита, затем цифры и потом все остальные символы. Имя исходного файла вводит пользователь.
27	Дан файл, содержащий целые 16-ти разрядные числа. Удалить из файла все значения, расположенные между максимальным и минимальным значением. Имя исходного файла вводит пользователь.
28	Дан текстовый файл. Заменить все символы C1 на C2, а C2 на C1. Значения C1 и C2 жестко кодируются в программе. Имя исходного файла вводит пользователь.
29	Дан файл, содержащий целые 16-ти разрядные числа. Переписать в обратном порядке все значения, расположенные до первого отрицательного значения. Имя исходного файла вводит пользователь.
30	Дан текстовый файл. Удалить из файла все символы, не являющиеся буквой латинского алфавита (кроме пробелов). Имя исходного файла вводит пользователь.

Контрольные вопросы

1. Что такое прерывание?
2. Какие виды прерываний бывают?
3. Что такое аппаратное прерывание?
4. Какими командами осуществляется управление обработкой аппаратных прерываний?
5. Что такое исключение?
6. Что такое программное прерывание?
7. Какой командой вызывается программное прерывание?
8. Что такое таблица векторов прерываний и где она находится?
9. В каком прерывании реализован видео сервис?
10. Перечислите основные прерывания BIOS.
11. Перечислите основные прерывания MS DOS.
12. В каком прерывании реализован сервис MS DOS?