

Министерство образования и науки Украины
Севастопольский национальный технический университет



РАБОТА С ДИСКОВЫМИ ФАЙЛАМИ В ОС MS DOS

Методические указания
к выполнению лабораторной работы №5
по дисциплине «Архитектура ЭВМ»
для студентов специальности 7.080401
«Информационные управляющие системы
и технологии»
всех форм обучения

Севастополь
2007



Методические указания к выполнению лабораторной работы №5 по дисциплине «Архитектура ЭВМ» / Сост. А. Ю. Дрозин, В. С. Чернега – Севастополь: Изд-во СевНТУ, 2007. – 12 с.

Целью методических указаний является помощь студентом в выполнении лабораторных работ. Излагаются теория к выполнению данной лабораторной работы, требования к содержанию отчета.

Методические указания предназначены для студентов дневной и заочной формы обучения специальности 7.080401 "Информационные управляющие системы и технологии"

Методические указания рассмотрены и утверждены на заседании кафедры Информационных систем, протокол №16 от 27.06.2007 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензенты: Е. В. Шишкевич, канд. техн. наук, доцент кафедры кибернетики и вычислительной техники.

СОДЕЖАНИЕ

1. Цель работы.....	4
2. Программа работы.....	4
3. Основные положения.....	5
4. Содержание отчета	9
5. Контрольные вопросы.....	9
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	11
ПРИЛОЖЕНИЕ А	12

1. ЦЕЛЬ РАБОТЫ

Исследовать процедуры работы с дисковыми файлами в ОС MS DOS. Изучить основные приемы обращения к файлам: открытие, чтение, запись. Приобрести практические навыки написания и отладки ассемблерных программ для работы с файлами.

2. ПРОГРАММА РАБОТЫ

- 1) Изучить основные команды для работы с дисковыми файлами в DOS, а также повторить команды пересылки данных и логических операций (выполняется в процессе домашней подготовки к лабораторной работе).
- 2) Исследовать, что происходит с существующим файлом, при создании нового файла с таким же именем. Реализовать ассемблерную программу, демонстрирующую это на примере с любым заранее подготовленным файлом.
- 3) Исследовать реакцию DOS при попытке создания нового файла если указан:
 - несуществующий каталог;
 - несуществующий файл;
 - неполный путь к файлу.
 Реализовать ассемблерную программу, демонстрирующую данные ситуации на примере.
- 4) Исследовать реакцию DOS при попытке открытия файла для записи, при наличии у файла установленного атрибута только для чтения. Реализовать ассемблерную программу, демонстрирующую это на примере с любым заранее подготовленным файлом.
- 5) Исследовать реакцию DOS при попытке произвести запись в файл, который был предварительно создан с атрибутом только для чтения. Реализовать ассемблерную программу, демонстрирующую это на примере с любым заранее подготовленным файлом.
- 6) Подготовить в текстовом редакторе исходный файл Ф1. Файл Ф1 может содержать произвольный набор символов. Реализовать ассемблерную программу, которая будет выполнять следующие действия: открывать файл Ф1 для чтения, программно создавать новый файл Ф2, затем копировать содержимое файла Ф1 в Ф2 в прямом и обратном порядке.
- 7) Подготовить в текстовом редакторе два входных файла Ф1 и Ф2. Файл Ф1 должен содержать произвольный набор символов латинского алфавита в различных регистрах, файл Ф2 пустой. Реализовать ассемблерную программу, которая будет выполнять следующие действия: открывать файл Ф1 для чтения, открывать файл Ф2 для записи, затем, читая символы из файла Ф1 и записывая их в файл Ф2, преобразовать входную последовательность символов к нижнему регистру вариант 1 либо к верхнему вариант 2 (задается преподавателем).
- 8) Подготовить в текстовом редакторе два входных файла Ф1 и Ф2, содержащих произвольные целочисленные данные, представленные в матричном виде, показанном ниже. Подготовить в текстовом редакторе пустой файл Ф3.

Файл Ф1 $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ матрица А	Файл Ф2 $\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$ матрица В
---	---

Реализовать ассемблерную программу, которая будет выполнять следующие действия: открывать файлы Ф1, Ф2 для чтения, открывать файл Ф3 для записи, прочитать данные из файлов Ф1 и Ф2, преобразовать из ASCII формата к цифровому виду. Программа также должна в соответствии с вариантом задания произвести вычисления результирующей матрицы С, выполнить преобразование результатов вычислений в ASCII формат и записать результирующую матрицу в файл Ф3.

Варианты заданий и пример выполнения приведены в приложении А.

3. ОСНОВНЫЕ ПОЛОЖЕНИЯ

Для обращения к дисковым файлам в DOS используются дескрипторы файлов, они представляют собой 16-битовую без знаковую целую величину и позволяют однозначно указывать на требуемый дисковый файл. Дескрипторы файлов упрощают операции записи, чтения файлов и программный доступ к ним. После создания нового файла или открытия существующего в дальнейшем для выполнения операцию любой сложности и типа можно оперировать только с файловым дескриптором.

Основные принципы, используемые при работе с файловым вводом-выводом:

- 1) Операцию открытия файла необходимо выполнить до начала чтения или записи данных. Операция открытия не уничтожает уже существующих данных в файле.
- 2) При создании нового файла он автоматически открывается для чтения и записи, а если файл с таким именем уже существует в системе, то вся информация в нем стирается.
- 3) Запись в файл происходит не сразу. Операция записи в DOS сначала помещает данные в буферы памяти, только потом они переносятся непосредственно в файл.
- 4) При закрытии файла все данные из буфера записываются на диск.

Для открытия дискового файла необходимо поместить адрес переменной, содержащей имя файла в регистр DS:DX. Переменная, содержащая адрес файла – строка, оканчивающаяся нулем. Номер функции следует занести в регистр AH. Функция открытия файла – 03Dh. Установив все необходимые параметры, следует передать управление в DOS, вызвав прерывание 21h. Пример процедуры:

```

DATASG SEGMENT PARA      'Data'
    filename    db    'c:\test.dat', 0
    handle      dw    ?
CODESG  SEGMENT PARA      'Code'
    mov  dx,offset filename ;поместить адрес Filename в ds:dx
    mov  ah,03Dh           ;функция 03Dh - открытие файла
  
```

```

mov  al,0           ;0 - доступ только на чтение
int   21h           ;прерывание 21h – передача управления DOS
jc    Error         ;переход на подпрограмму обработки ошибок
mov   handle,ax     ;сохранение дескриптора файла

```

Имя файла может содержать в себе название диска или каталога, в котором находится файл. Адресация на имя файла при вызове функции DOS 03Dh осуществляется содержанием регистра DS:DX. В регистр AL помещается код типа открытия файла. Код может иметь одно из трех значений:

- al=0 – только чтение;
- al=1 – только запись;
- al=2 – чтение и запись.

После вызова функции DOS, флаг переноса CF показывает, была ли операция успешной. Таким образом, используя условный переход JC можно сразу перейти на процедуру обработки ошибок, при этом в регистре AX содержится код ошибки (приведены в таблице 3.1). Если ошибки не произошло, то в регистре AX содержится дескриптор файла, который для удобства дальнейшего использования можно сохранить в переменной handle.

Таблица 3.1 – Коды ошибок

Cod e	Значение
01	Разделение файла не допускается
02	Файл не существует
03	Путь или файл не существует
04	Нет больше свободных дескрипторов
05	Доступ запрещен (ошибочный атрибут файла)
0Ch	Неверное значение для доступа в регистре al

Для закрытия файла необходимо поместить в регистр BX дескриптор любого открытого файла. Номер функции следует занести в регистр AH. Функция закрытия файла – 03Eh. При закрытии файла в DOS будет произведена запись на диск всех данных, содержащихся в буфере памяти, будут установлены значения даты и времени создания, размера файла. Для закрытия файла можно использовать следующий фрагмент кода:

```

mov  bx, handle     ;поместить дескриптор в bx
mov  ah, 03Eh       ;функция 03Eh – закрытие файла
int   21h           ;прерывание 21h – передача управления DOS
jc    Error         ;переход на подпрограмму обработки ошибок

```

Создание нового файла происходит аналогично открытию существующего. Для этого необходимо сохранить его имя в виде строки ASCIIZ в регистрах DS:DX и атрибут файла в регистр CX. Создание файла происходит при вызове функции 03Ch – создание файла:

```

DATASG SEGMENT PARA 'Data'
      filename db 'c:\test.txt',0
      handle   dw ?
DATASG ENDS

```

CODESG SEGMENT PARA 'Code'

```

...
mov  dx, offset filename; Адрес строки в ds:dx
mov  ax, 03Ch           ;функция 03Ch – создание файла
xor   cx, cx            ;cx=0, установить тип файла – обычный
int   21h              ;прерывание 21h – передача управления DOS
jc    Error            ;переход на подпрограмму обработки ошибок
mov   handle, ax        ;сохранение дескриптора
...

```

Таблица 3.2 – Атрибуты создаваемых файлов

Атрибу т	Значение
00	Обычный файл (большинство файлов данных)
01	Только чтение (операции записи не выполняются)
02	Скрытый (невидимый для команды DIR)
04	Системный файл

Важно помнить, что DOS не проверяет при создании файла, есть ли уже файл с таким именем. Если создается файл с уже существующим именем, то содержимое старого файла уничтожается. Для недопущения этого следует осуществлять проверку на наличие файла в системе до начала операции создания.

Для чтения данных из файла посимвольно (побайтно) необходимо вызвать функцию DOS 3Fh, поместить в регистр BX дескриптор входного файла, в регистре CX указать количество байтов для чтения, результат операции будет помещен по адресу в памяти DS:DX. Пример процедуры:

DATASG SEGMENT PARA 'Data'

```

    handle    dw    0
    value     db    0

```

DATASG ENDS

CODESG SEGMENT PARA 'Code'

```

...
mov  ah, 3Fh           ;функция 03Fh – чтение файла
mov  bx, handle        ;запись в bx дескриптора входного файла
mov  cx, 1             ;количество байт для считывания
mov  dx, offset value  ;адрес переменной в ds:dx
int   21h              ;прерывание 21h – передача управления DOS
jc    Error            ;переход на подпрограмму обработки ошибок
...

```

Для записи данных в файл посимвольно (побайтно) необходимо вызвать функцию DOS 40h, в регистр BX поместить дескриптор выходного файла, а в регистре CX указать количество байт для записи, причем значение для записи должно быть определено в регистрах DS:DX. Пример процедуры:

DATASG SEGMENT PARA 'Data'

```

    handle    dw    0
    value     db    0

```



```
DATASG ENDS
CODESG SEGMENT PARA 'Code'
```

```
...
mov ah, 40h          ;функция 40h – запись файла
mov bx, handle       ;запись в bx дескриптора выходного файла
mov cx, 1            ;количество байт для записи
mov dx, offset value ;адрес переменной в ds:dx
int 21h              ;прерывание 21h – передача управления DOS
jc Error             ;переход на подпрограмму обработки ошибок
...
```

Существует большая разница между посимвольным (побайтным) чтением и записью файла и обработкой файлов, содержащих многобайтные записи. В некоторых случаях необходимо осуществлять чтение или запись файла в произвольном порядке, то есть осуществлять произвольный доступ к данным. Необходимо следует учитывать, что:

- операция поиска позиционирует внутренний указатель позиции на первом байте записи в файле;
- чтение и запись определенного количества байт после операции поиска влияет только на одну запись в файле, оставляя другие неизменными.

Операция поиска позиционирует внутренний указатель на место в файле, откуда будет производиться чтение, или осуществляться запись.

Важно помнить и то, что DOS всегда ищет позицию байта в файле, независимо от того, сколько байтов занимает каждая запись. Поэтому нужно определить размер данных в файле данных и переместить указатель на необходимое число записей. Пример фрагмента кода поиска требуемой позиции в файле:

```
mov ax, number      ; ax – номер искомой записи
mov cx, recsize      ; cx – размер записи в байтах
mul cx              ; ax:dx – позиция в файле
mov cx, dx           ; старшее слово в cx
mov dx, ax           ; младшее слово в dx
mov ah, 042h         ; функция 42h – поиск в файле
mov al, 0            ; al=0 – поиск от начала файла
mov bx, handle       ; дескриптор файла в bx
int 21h              ; прерывание 21h – передача управления DOS
jc Error             ; переход на подпрограмму обработки ошибок
...
```

В результате выполнения команда `mul` в регистр `DX` находится старшая, а в регистр `AX` – младшая часть 32-битного результата, по требованиям `042h` функции прерывания `21h` (DOS) – функции поиска, необходимо записать старшую часть в регистр `CX`, а младшую в регистр `DX`.

При `AL=0` значение позиции байта в регистрах `CX:DX` рассматривается как абсолютное, т.е. относительно начала файла. Используя `AL=1` можно осуществлять поиск с текущего положения указателя, тогда в регистрах `CX:DX` будет содержаться смещение относительно текущей позиции. При `AL=2` поиск выполняется от конца файла, это позволяет легко позиционировать указатель на конец файла для добавления новых данных.

Следует обратить внимание что, после позиционирования можно производить операции чтения или записи над данными в файле. Операцию чтения можно выполнить следующим образом:

```
mov ah, 03Fh      ; функция 03Fh – чтение файла
mov bx, handle     ; дескриптор файла в bx
mov cx, recsiz     ; количество считываемых байтов в cx
mov dx, offset buffer ; адрес назначения – переменная buffer
int 21h           ; прерывание 21h – передача управления DOS
jc Error          ; переход на подпрограмму обработки ошибок
```

...

При чтении или записи указатель файла переходит на следующую запись, поэтому для последовательного чтения (записи) данных нет необходимости выполнять операцию позиционирования. Также, можно отметить, что в регистре ВХ после операции поиска уже содержит дескриптор файла, что позволяет не перегружать снова дескриптор. Запись в файл осуществляется аналогично чтению, но вместо функции 03Fh вызывается функция 040h.

4. СОДЕРЖАНИЕ ОТЧЕТА

- 1) Цель и программа работы.
- 2) Текст и листинг ассемблерной программы для заданного варианта.
- 3) Результаты проведенных исследований.
- 4) Выводы по работе с анализом результатов выполненных исследований.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) В каких режимах можно открывать файл?
- 2) Какие ошибки могут возникнуть при открытии файлов, как их отследить?
- 3) Какие атрибуты создаваемых файлов можно задать? Приведите пример.
- 4) Как, используя абсолютное смещение, произвести позиционирование указателя в файле на пятую запись, если длина записи равна трем байтам? Приведите текст соответствующей программы.
- 5) Как, используя относительное смещение, произвести позиционирование указателя в файле на четвертую запись, если длина записи равна двум байтам? Приведите текст соответствующей программы.
- 6) Расскажите об особенностях архитектуры микропроцессоров четвертого поколения.
- 7) Что такое пакетный режим передачи данных? Опишите этапы передачи данных в пакетном режиме. За счет чего при пакетная передача данных осуществляется быстрее чем в стандартном режиме?
- 8) Расскажите о назначении и особенностях работы буферов отложенной памяти в микропроцессорах четвертого поколения.
- 9) Объясните различие между суперскалярными и VLIW-процессорами.
- 10) Для чего и как в суперскалярных процессорах используется механизм предсказания переходов?
- 11) Опишите разновидности зависимостей по данным. Объясните их особенности на примерах.

- 12) Расскажите об отличительных особенностях архитектуры процессоров класса Pentium.
- 13) Начертите структурную схему процессора семейства Pentium, объясните назначение каждого из блоков.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сван Т. Освоение Turbo Assembler / Т. Сван. – К.: “Диалектика”, 1996. - 544 с.
2. Абель П. Язык Ассемблера для IBM PC и программирования / П. Абель; Пер. с англ. Ю. В. Сальникова. – М.: Высш. шк., 1992. - 447 с.
3. Брэдли Д. Программирование на языке Ассемблера ПЭВМ фирмы IBM / Д. Бредли. – М.: Радио и связь, 1988. - 442 с.
4. Микропроцессорный комплект K1810: Структура, программирование, применение / Под ред. Ю. М. Казаринова. - М.: Высш. шк., 1990.
5. Самофалов К.Г. Микропроцессоры /К. Г. Самофалов, О. В. Викторов. – К.: Техника, 1989. - 462 с.
6. Гук М. Процессоры от Intel 8086 до Pentium II / М. Гук. – СПб: Питер, 1998. - 224с.
7. Финогенов К.Г. Самоучитель по системным функциям MS-DOS / К. Г. Финогенов. – М.: Малип, 1993. - 262 с.

ПРИЛОЖЕНИЕ А

В задании 8 номер варианта выбирается в соответствии с последней цифрой зачетки студента, i - целое число равное предпоследней цифре номера зачетки студента.

Варианты к заданию 8 приведены в таблице А.1.

Таблица А.1 – Формулы для расчета результирующей матрицы.

Вариант №	Формула для расчета результирующей матрицы
0	$C=(A*i)-B$
1	$C=A+(B*i)$
2	$C=(A-i)+B$
3	$C=(A*i)+(B*i)$
4	$C=A+B+i$
5	$C=(A-i)+B$
6	$C=A+B+2i$
7	$C=A+B-i$
8	$C=(A*i)-(B*i)$
9	$C=(B*i)-A$

Примеры результатов выполнения заданий приведены в таблице А.2.

Таблица А.2 – Примеры результатов выполнения заданий.

Задание	Файл	Состояние файлов до выполнения программы	Состояние файлов после выполнения программы
6	Ф1	abcde4f	abcde4f
	Ф2	не существует	abcde4ff4edcba
7	Ф1	aAbCedF	aAbCedF
	Вариант1: Ф2	пустой	AABCEDF
	Вариант 2: Ф2	пустой	aabcedf
8	Ф1, матрица А	2 2 2	2 2 2
		2 2 2	2 2 2
		2 2 2	2 2 2
	Ф2, матрица В	3 3 3	3 3 3
		3 3 3	3 3 3
		3 3 3	3 3 3
	Ф3 Вариант 4: $C=A+B+i$, $i=4$.	пустой	9 9 9 9 9 9 9 9 9

Заказ № _____ от «___» _____ 200__ г. Тираж _____ экз.
Изд-во СевНТУ