МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«_15_» _12_ 2017 г.

ПРОГРАММИРОВАНИЕ РАБОТЫ С ФАЙЛАМИ НА ЯЗЫКЕ ASSEMBLER

Методические указания для проведения лабораторных занятий и выполнения самостоятельной внеаудиторной работы по дисциплине «Системное программное обеспечение» для студентов направления подготовки 09.03.04 «Программная инженерия»

Курск 2017

УДК 681.3

Составитель: А.В. Малышев

Рецензент

Кандидат технических наук, начальник отдела информатизации
ГУ КРО ФСС РФ А.Ф. Рубанов

Программирование работы с файлами на языке Assembler: методические указания для проведения лабораторных занятий и выполнения самостоятельной внеаудиторной работы по дисциплине «Системное программное обеспечение» / Юго-Зап. гос. ун-т; сост. А.В. Малышев. Курск, 2017. 17 с.: ил. 3. Библиогр.: с. 16

Содержат сведения по вопросам низкоуровневого создания и обработки файлов в операционной среде Windows. Приведены соответствующие примеры программного кода на языке Assembler, а также граф-схемы алгоритмов.

Предназначены для студентов направления подготовки 09.03.04.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16
Усл. печ. л. . Уч.-изд. л. . Тираж экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

1.1 Цель лабораторной работы

Изучение основ строения файловой системы, понятий файл и каталог; изучение 32-байтного формата элемента главного корневого каталога; назначение SFT, блока описания файла и дескриптора файла; приобретение навыков системного программирования файловых функций DOS и WINDOWS.

1.2 Задание

1. В соответствии с вариантом задания составьте граф-схему решения задачи и на ее основе разработайте программу .
2. Получите файл листинга программы .LST.

1.3 Содержание отчёта

- титульный лист;
- задание;
- описание формата элемента корневого каталога;
- граф-схема решения задачи;
- описание используемых в программе файловых функций;
- текст программы в виде распечатанного .LST файла с комментариями;
- результаты работы программы

2. Основные понятия

2.1 Элемент корневого каталога, дескриптор

В Ассемблере обращение к файлу распадается на следующие операции [1]:

- создание или открытие файла по ASCIIZ-спецификации;
- запись или чтение файла поэлементно;
- закрытие файла.

Операция открытия файла связана с выделением свободного элемента [2], называемого блок описания файлов, в System File Table (SFT), ее размер задается в CONFIG.SYS командой FILES = n, где n – число открытых файлов.

Часть информации в элементе SFT определяется полями элемента главного корневого каталога, часть – операционной системой. Например, элемент SFT содержит указатель файла (32 бита) для прямого доступа к файлу с позиции указателя. Структура элемента главного корневого каталога приведена в таблице1.

Таблица 1.

Смещение	Размер	Описание
1	2	3
0	8	Имя файла
8	3	Расширение

Продолжение табл.1

1	2	3
11	1	Атрибуты
12	10	Резерв
22	2	Дата
24	2	Время
26	2	Начальный кластер
28	4	Размер файла

Значения атрибутов файла приведены в таблице 2.

Таблица 2.

Атрибут	Назначение
01h	файл только для чтения; запрет модификации или удаления
02h	скрытый файл
04h	системный файл
08h	метка тома; существует только в корневом каталоге
10h	файл представляет собой каталог
20h	архивный файл

В первом байте имени файла могут находиться 3 специальных кода:

1. 00h – код неиспользовавшегося элемента каталога;
2. E5h – код удаленного элемента каталога;
3. 2Eh – (точка) код подкаталога

Ссылка на выделенный блок описания файла возвращается в программу в виде дескриптора (порядковый номер). Обращение к открытому файлу осуществляется по присвоенному дескриптору..

Некоторые значения дескрипторов зарезервированы для стандартных устройств ввода/вывода:

0 – стандартный ввод (клавиатура);

1 – стандартный вывод (монитор);

2 – стандартная ошибка (вывод);

3 – стандартный вспомогательный порт;

4 – стандартный принтер.

2.2 Файловые функции DOS

Для работы с обычными файлами, значения дескрипторов которых больше 4, используются следующие группы операций (рис. 1).

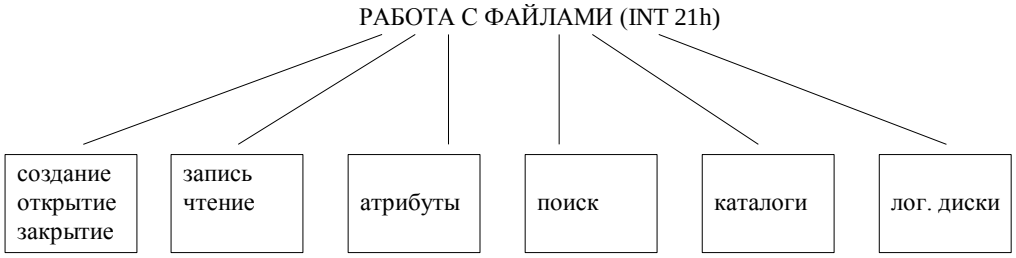

Рис. 1. Значения дескрипторов для работы с файлами

2.2.1 Операции создания, открытия, закрытия файлов.

Вход: AH = 3Ch – создание файла;

DS:DX – ASCIIZ – спецификация файла;

CX – атрибуты создаваемого файла:

$$CX = \begin{cases} 01h\ только\ чтение \\ 02h\ скрытый \\ 04h\ системный \\ 08h\ метка\ тома \\ 10h\ каталог \\ 20h\ неархивный \end{cases}.$$

$$Выход:\ CF = \begin{cases} 0 - нет\ ошибки,\ AX - дескриптор\ создаваемого\ файла \\ 1 - ошибка,\ AX = \begin{cases} 2 - ошибка\ имени\ файла \\ 3 - ошибка\ имени\ каталога \end{cases} \end{cases}.$$

Вход: AH=5Bh – создание нового файла;

DS:DX – ASCIIZ – спецификация файла;

CX – атрибуты создаваемого файла:

$$CX = \begin{cases} 01h\ только\ чтение \\ 02h\ скрытый \\ 04h\ системный \\ 08h\ метка\ тома \\ 10h\ каталог \\ 20h\ неархивный \end{cases}.$$

$$Выход:\ CF = \begin{cases} 0 - нет\ ошибки,\ AX - дескриптор\ создаваемого\ файла \\ 1 - ошибка,\ AX = \begin{cases} 2 - ошибка\ имени\ файла \\ 3 - ошибка\ имени\ каталога \end{cases} \end{cases}.$$

Различие 3Ch и 5Bh: функция 3Fh всегда создает новый файл без проверки. Функция 5Bh не может создать файл с уже имеющимся именем и описывает эту нештатную ситуацию как CF=1, AX=2.

Общим свойством функций 3Ch, 5Bh является позиционирование указателя на начальный (нулевой) байт.

Вход: AH=3Dh –открытие ***существующего*** *файла;*

DS:DX – ASCIIZ- спецификация открываемого файла;

AL-режим доступа к файлу:

$$AL = \begin{cases} 0 - чтение \\ 1 - запись \\ 2 - чтение / запись \end{cases}.$$

$$Выход:\ CF = \begin{cases} 0 - нет\ ошибки,\ AX - дескриптор\ открытого\ файла \\ 1 - ошибка,\ AX - код\ ошибки \end{cases}.$$

Операция открытия существующего файла устанавливает указатель файла на начальный (нулевой) байт.

Вход: AH=3Eh – закрытие файла;
BX – дескриптор закрываемого файла.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

Вход: AH=41h – удаление файла;
DS:DX – ASCIIZ – спецификация файла.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

2.2.2 Операции чтения/записи файлов

Вход: AH=42h – установка указателя файла;
BX - дескриптор файла;
AL - положение указателя:

$$AL = \begin{cases} 0 - \text{начало файла} \\ 1 - \text{смещение от текущего положения указателя в } CX:DX \\ 2 - \text{конец файла} \end{cases}.$$

Выход: CX:DX – значение установленного указателя

Особенность функции 42h – смещение знаковое в **дополнительном коде.**

Вход: AH=3Fh – чтение из файла;
BX – дескриптор файла;
CX- количество читаемых байт;
DS:DX – адрес буфера.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки, } AX - \text{реальное количество прочитанных байт} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

Данная функция пересылает данные из файла в буфер программы и модифицирует указатель. При чтении в режиме ASCII читается строка указанной длины или до символа CR, если он встретился раньше.

Вход: AH = 40h – запись в файл;
BX – дескриптор файла;
CX- количество записываемых байт;
DS:DX – адрес буфера программы.

Выход

$$CF = \begin{cases} 0 - \text{нет ошибки, } AX - \text{реальное количество записанных байт} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}.$$

2.2.3 Операции с атрибутами файла

Вход: AH = 43h - работа с атрибутом;
AL=1 – запись атрибута файла:

$$CX = \begin{cases} 01h\ только\ чтение \\ 02h\ скрытый \\ 04h\ системный \\ 08h\ метка\ тома \\ 10h\ каталог \\ 20h\ неархивный \end{cases};$$

DS:DX – ASCIIZ – спецификация файла или каталога;

AL=0 – чтение атрибута файла:

Выход: $CF = \begin{cases} 0 - нет\ ошибки,\ CX - считанные\ атрибуты\ для\ AL = 0 \\ 1 - ошибка,\ AX - код\ ошибки \end{cases}$

Вход AH=56h – переименование файла;

DS:DX – ASCIIZ –старая спецификация файла;

ES:DI – ASCIIZ – новая спецификация файла.

Выход: $CF = \begin{cases} 0 - нет\ ошибки \\ 1 - ошибка,\ AX - код\ ошибки \end{cases}$.

2.2.4 Операции поиска.

Вход: AH=4Eh – найти первый файл;

CX – байт атрибутов;

DS:DX – ASCIIZ – спецификация файла.

Выход: $CF = \begin{cases} 0 - нет\ ошибки,\ имя\ и\ расширение\ в\ DTA \\ 1 - ошибка,\ AX - код\ ошибки \end{cases}$

Вход: AH = 4Fh – найти следующий файл;

CX – байт атрибутов;

DS:DX – ASCIIZ – спецификация файла поиска.

Выход: $CF = \begin{cases} 0 - нет\ ошибки,\ имя\ и\ расширение\ в\ DTA \\ 1 - ошибка,\ AX - код\ ошибки \end{cases}$.

Перед использованием функций поиска необходимо организовать область передачи данных (DTA - Disk Transfer Area) размером не менее 43 байт. Выделение массива под DTA осуществляется функцией 1Ah int 21h, при этом в DS:DX хранится указатель на массив. Состав DTA описан в таблице3.

Таблица 3.

Состав полей DTA.

Смещение	Размер, байт	Назначение
1	2	3
+00h	1	биты 0÷6 – ASCII-код буквы диска; бит 7 – диск сетевой
+01h	11	Маска поиска (без пути)
+0Ch	1	Атрибуты поиска
+0Dh	2	Порядковый номер файла в директории

Продолжение табл.3

1	2	3
+0Fh	2	Номер кластера начала внешней директории
+11h	4	Резерв
+15h	1	Атрибут найденного файла
+16h	2	Время создания файла в формате DOS
+18h	2	Дата создания файла в формате DOS
+1Ah	4	Размер файла
+1Eh	13	ASCIIZ-имя найденного файла с расширением

2.2.5 Операции над каталогами и логическими дисками

Вход: AH=39h – создание каталога;
DS:DX – ASCIIZ – спецификация создаваемого каталога.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$

Вход: AH=3Ah – удаление каталога;
DS:DX – ASCIIZ – спецификация удаляемого каталога.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$

Вход: AH=3Bh – смена текущего каталога;
DS:DX – ASCIIZ – спецификация нового каталога.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$

Вход: AH=47h – определить текущий каталог;
DL=номер диска (0 – текущий, 1- A, 2-B, 3-C и т.д.);
DS:SI – 64-байтный буфер (ASCIIZ – спецификация нового каталога без имени диска, первого и последнего символов «\».

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

Вход: AH=0Eh – выбор диска;
AL-код диска (0=A, 1=B, C=2, D=3 и т.д.).
Выход: AL – общее число логических дисков.
Вход: AH=19h – код текущего диска.
Выход: AL-код диска (0=A, 1=B, C=2, D=3 и т.д.).
Вход: AH=36h – получение информации о диске;
DL- код диска (0=A, 1=B, C=2, D=3 и т.д.).
Выход: AX – число секторов в кластере;
BX – число свободных кластеров;
CX – размер сектора в байтах;
DX – общее число кластеров на диске.

В случае ошибки AX=FFFFh.

Вход: AH=68h – сброс файловых буферов DOS на диск;

BX- идентификатор.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

2.3 Файловые функции Windows

Для работы с длинными именами файлов используются следующие функции LFN (Long File Name).

Создать или открыть файл с длинным именем

Вход: AX=716Ch;

BX – режимы доступа (рис. 2);

CX – атрибуты файла;

DX – действие:

0000h – открыть файл;

0001h – заменить файл;

0010h – создать файл;

DS:SI - ASCIIZ – спецификация файла;

DI – запись DI в конец короткого имени файла.

Выход: *AX=идентификатор файла;*

CX=1, если файл открыт;

CX=2, если файл создан;

CX=3, если файл заменен;

$CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

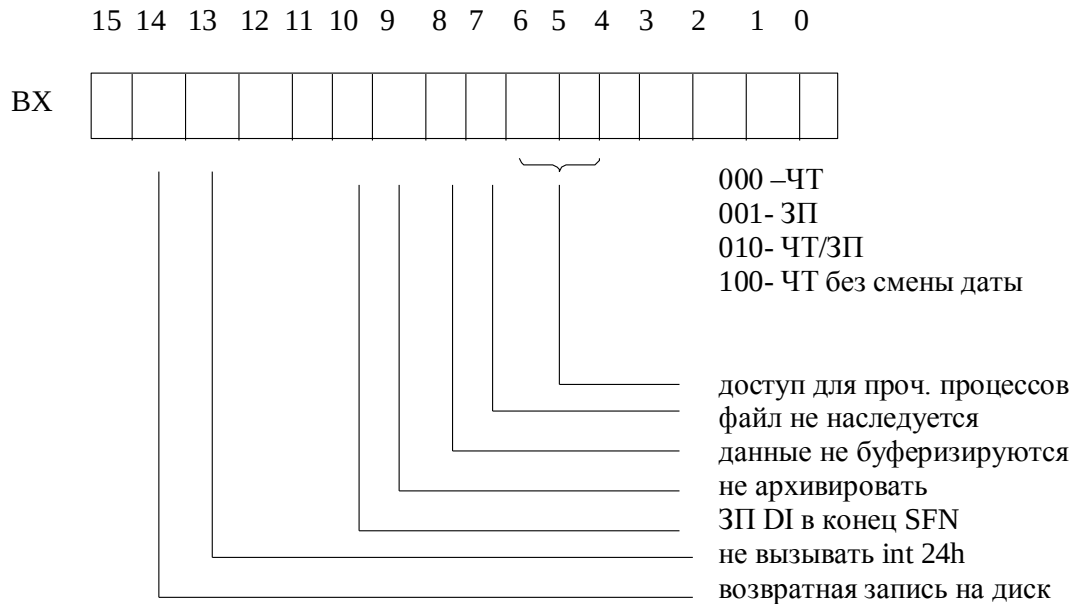

Рис. 2. Режимы доступа

Изменить максимальное число идентификаторов(20-65535)

Вход: *AX=67h;*

BX – новое число идентификаторов (20-65535).

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

Удалить файл с длинным именем

Вход: *AX=7141h;*

DS:SI - ASCIIZ – спецификация файла;

SI=0000h – маски и атрибуты игнорируются.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

При работе с длинными именами используются свои 3 операции: найти первый файл, найти следующий файл, прекратить поиск.

Найти первый файл с длинным именем

Вход: *AX=714Eh;*

CX- атрибуты;

SI=0 – Win95 формат даты/времени;

SI=1 – DOS формат даты/времени;

DS:DX - ASCIIZ – спецификация файла с маской;

ES:DI – адрес 318-байтного буфера для инф. о файле.

Выход: *AX – идентификатор файла;*

CX – Unicode-флаг;

$CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибк} \end{cases}$.

В 318-байтном буфере по смещению 2Ch располагается - ASCIIZ – длинная спецификация файла размером в 260 байт, а по смещению 130h - ASCIIZ - короткая спецификация файла размером в 14 байт.

Найти следующий файл с длинным именем

Вход: *AX=714Fh;*

BX- поисковый идентификатор;

SI=0 – Win95 формат даты/времени;

SI=1 – DOS формат даты/времени;

ES:DI – адрес 318-байтного буфера для инф. о файле.

Выход: *CF=0 следующий файл найден; CX – Unicode-флаг;*

CF=1 – ошибка, AX – код ошибки.

Закончить поиск файла

Вход: *AX=71A1h;*

BX- поисковый идентификатор.

Выход: *CF=0 операция выполнена;*

CF=1 – ошибка, AX – код ошибки.

Операции над каталогами с длинным именем

Вход: AX=7139h – создание каталога LFN;

DS:DX – ASCIIZ – спецификация создаваемого каталога;

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

Вход: AX=713Ah – удаление каталога LFN;
DS:DX – ASCIIZ – спецификация удаляемого каталога.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

Вход: AH=713Bh – смена текущего каталога LFN;
DS:DX – ASCIIZ – спецификация нового каталога.

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$.

Вход: AX=7147h – определить текущий каталог LFN;
DL=номер диска (0 – текущий, 1- A, 2-B, 3-C и т.д.);
DS:SI – 64-байтный буфер (ASCIIZ – спецификация нового каталога без имени диска, первого и последнего символов «\».

Выход: $CF = \begin{cases} 0 - \text{нет ошибки} \\ 1 - \text{ошибка, } AX - \text{код ошибки} \end{cases}$

Примечание: Перед использованием функций LFN необходимо:
1. один раз вызвать подфункцию A0h для определения размера буферов;
2. при любом вызове LFN устанавливать CF=1.

Получить информацию о разделе файловой системы

Вход: *AX=71A0h;*
DS:DX – адрес ASCIIZ – спецификации с именем раздела (например, db «C:\»,0);
ES:DI – адрес буфера для имени файловой системы (FAT, NTFS, CDFS);
CX=размер буфера в ES:DI (достаточно 32 байта).

Выход: *BX=флаги файловой системы:*
Бит 0 – учет регистра символов;
Бит 1 – учет регистра символов для каталогов;
Бит 2 – использование символов Unicode;
Бит 14 – поддержка функций LFN;
Бит 15 – включено сжатие (DBLSpase);
CX=max длина имени файла (≤255 символов)
DX=max длина пути (≤260 символов)
CF=1 – ошибка, AX – код ошибки.

Задача. В каталоге C:\TEMP\ создать файл file1.txt и записать в него символьную строку длиной не менее 25 символов. Скопировать из file1.txt 15 символов, начиная с 10-ого символа, и сохранить их по спецификации C:\TEMP\1\file2.txt . Переименовать file2.txt в text.doc и вывести его на экран.

Решение.

Для работы с файлами необходимо в сегменте данных задать ASCIIZ-спецификации создаваемых файлов и ASCIIZ-спецификацию пути создания каталога 1.

```
filename1 db'C:\temp\file1.txt',0
filename2 db'C:\temp\1\file2.txt',0
newname  db'C:\temp\1\text.doc',0
directory db 'C:\temp\1',0
```

Кроме того, выполнение файловых операций чтения/записи связано с использованием двух промежуточных буферов **buffer** и **copy**, которые также должны быть объявлены в сегменте данных.

```
buffer  db 80  ; максимальный размер буфера
buf_len db ?   ; фактическая длина буфера
cont    db 78 dup (?)  ; содержимое буфера
copy db 15 dup (?)
```

Граф-схема решения задачи имеет следующий вид (рис.3)

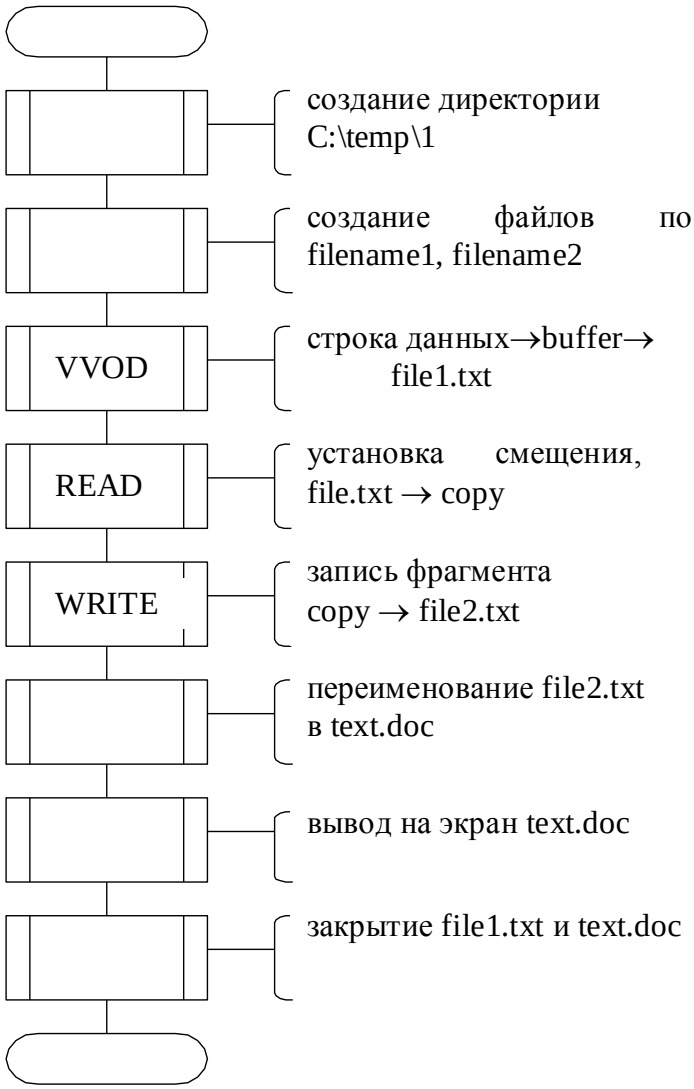

Рис. 3. Пример граф-схемы решения задачи работы с файлами

Текст программы

```
.model small
.stack 512
.data
filename1 db'C:\temp\file1.txt',0
filename2 db'C:\temp\1\file2.txt',0
newname   db'C:\temp\1\text.doc',0
handle1 dw ?
handle2 dw ?
buffer  db 80          ; максимальный размер буфера
buf_len db ?    ; фактическая длина буфера
cont    db 78 dup (?); содержимое буфера
len1=25        ; минимальная длина строки
len2=15        ; длина копируемого фрагмента
msg db 'введите строку в file1.txt не более 25 символов$',10,13
copy db 15 dup (?)
directory db 'C:\temp\1',0
.code
mov ax,@data
mov ds,ax
; создание директории C:\temp\1
mov ah,39h
mov dx,offset directory
int 21h
;создание файлов по спецификациям filename1, ;filename2
mov AH,3Ch
mov CX, 0 ;  без атрибутов
mov DX, offset  filename1
int 21h
mov handle1, ax     ; дескриптор file1 в handle1
mov ax,3c00h
mov dx, offset filename2
int 21h
mov handle2, AX  ; дескриптор file2 в handle2
; ввод символьной строки в file1.txt
vvod: mov ah,09h
mov dx,offset msg
int 21h
mov AH,0Ch    ; очистка буфера
mov AL,0Ah    ; и ввод строки
mov DX, offset buffer
int 21h   ; в buf_len - фактическая длина строки
mov AL,buf_len
cmp al, len1
```

```
        jb vvod
; запись из буфера в FILE1.TXT
        mov AH, 40h
        mov BX, handle1
        xor cx,cx
        mov Cl, buf_len
        mov DX, offset cont        ; содержимое buffer
        int 21h
; установка указателя в FILE1.TXT
        mov AH, 42h
        mov  BX, handle1
        mov AL, 0  ; указатель относительно начала файла
        mov CX, 0
        mov DX, 10 ;        CX:DX - адрес смещения
        int 21h
; чтение из file1.txt фрагмента 15 символов
        mov ah,3fh
        mov bx,handle1     ; идентификатор file1.txt
        mov cx, len2       ; длина копируемого фрагмента
        mov dx, offset copy
        int 21h
; запись copy в file2.txt
        mov ah,40h
        mov bx,handle2     ; идентификатор file2.txt
        xor cx,cx
        mov cl,buf_len
        mov dx, offset copy
        int 21h
; переименование file2.txt в text.doc
        push DS
        pop ES
        mov AH, 56h
        mov DX, offset filename2
        mov DI, offset newname
        int 21h
; вывод text.doc на экран
        mov AH, 40h
        mov BX,1    ; дескриптор монитора
        mov CX, len2
        mov dx, offset copy
        int 21h
; закрытие file1.txt
        mov AH, 3Eh
        mov BX, handle1
```

```
        int 21h
; закрытие text.doc
        mov bx,handle2
        int 21h
        mov ax,4c00h
        int 21h
        end
```

3 Варианты заданий

1. В каталоге C:\USERS\GUEST создать файл с произвольным именем (не более 8 символов) и записать в него символьную строку с клавиатуры. Инвертировать полученный файл и вывести его на экран.
2. В каталоге C:\TEMP создать файл с заданным именем, записать в него данные с входного файла (не менее 25 символов). Добавить в созданный файл заданную символьную строку. Полученный файл вывести на экран.
3. .В каталоге C:\TEMP создать файл с произвольным именем, записать в него заданную символьную строку. Добавить в созданный файл произвольную символьную строку с 5-ой позиции. Полученный файл переименовать.
4. В каталоге C:\USERS\GUEST создать файл с произвольным именем, записать в него данные с входного файла, подсчитать длину файла и вывести ее на экран. Добавить в начало файла произвольную символьную строку. Осушествить вывод полученного файла в выходной файл OUT.DOC по данному пути, а также скопировать полученный файл по спецификации C:\TEMP.
5. *В каталоге C:\TEMP создать временный файл, записать в него данные с клавиатуры, установить атрибут “чтение”, выполнить попытку записи в конец файла с подачей звукового сигнала. Переименовать файл (имя определяется программистом) и вывести его на экран.
6. *В каталоге C:\USERS\GUEST создать временный файл, записать в него данные с входного файла. Вывести на экран дату и время создания файла.
7. В каталоге C:\TEMP создать временный файл, записать в него данные с входного файла. Инвертировать поле файла с 5-ого по 10-ый символы. Полученный файл переименовать (имя определяется программистом) и переместить по спецификации C:\USERS\GUEST.
8. *В каталоге C:\ создать временный файл, записать в него символьную строку с клавиатуры. Исходный файл скопировать по спецификациям C:\TEMP, C:\TIT (подразумевается, что каталог TEMP существует, а TIT еще не создан). Исходный файл удалить.
9. В каталоге C:\ создать временный файл, записать в него данные с входного файла. Удалить в исходном файле первые 10 символов и подсчитать длину измененного файла. Поставить файлу атрибут

скрытый. Вывести файл на экран.

10. В каталоге C:\ создать файл, записать в начало файла символьную строку с клавиатуры. Исходный файл переместить по спецификации C:\TEMP, D:\TEMP (подразумевается, что каталог TEMP существует на обоих дисках).
11. Найти в корневом каталоге все файлы, имеющие расширения DOC и TXT. Вывести количество найденных файлов по расширениям.
12. Найти в каталогахC:\ D:\ все файлы маской C*.DOC имеющие расширения DOC и TXT. Первый найденный файл вывести на экран.
13. Найти в корневом каталоге все файлы, имеющие расширения DOC и TXT. Вывести количество найденных файлов по расширениям.
14. *В текущем каталоге найти первые два файла LFN с расширением .DOC. Выполнить поиск вхождений первого слова из первого найденного файла во втором файле. Сохранить количество вхождений в файл found.txt.
15. Выполнить замену русской буквы «Н» на латинскую букву «H» во всех файлах с расширением .TXT в текущем каталоге.

4. Контрольные вопросы

1. Как изменить формат записи каталога для использования длинных имен файлов и как при этом сохранить совместимость.
2. Укажите на различия между функциями “создать файл”, “создать новый файл”, “создать временной файл”. Что такое ASCIIZ-строка.
3. *Объясните, почему открытие, удаление и переименование файла выполняется через ASCIIZ-спецификацию, а не через дескриптор файла.
4. Какие предварительные действия необходимо выполнить для использования файловых функций LFN.
5. *Как удалить содержимое файла.
6. Укажите различия между элементом корневого каталога и дескриптором файла.
7. Укажите количество и назначение элементов главного корневого каталога для создания каталога.
8. Как реализовать функцию APPEND (добавление данных) в файл, как определить размер файла.
9. Назначение DTA.
10. Назначение стандартных дескрипторов файлов.

Библиографический список

1. Калуцкий, Игорь Владимирович. Системное программное обеспечение [Текст] : учебное пособие / И.В. Калуцкий, Е.А. Титенко. - Курск: ЮЗГУ, 2014. - 231 с.

2. Синицын, Сергей Владимирович. Операционные системы [Текст] : учебник / С.В. Синицын, А.В. Батаев, Н.Ю. Налютин. - М.: Академия, 2012. - 304 с.