ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ,
УПРАВЛЕНИЯ И ДИЗАЙНА

Курсовая работа

по курсу: *"Операционные системы"*

на тему: *"Разработка учебной файловой ОС"*

Выполнил студент группы ВТ-21: Потеренко А.Г.
Проверил преподаватель: Коноплянов А.В.

ДИМИТРОВГРАД 2005 Г.

ПОРЯДОК РАБОТЫ

1. Анализ задания и разработка способов представления объектов задачи в памяти, методов доступа к ним.
2. Разработка программы на языке ассемблер.
3. Тестирование программы.
4. Отладка программы.
5. Составление отчета.

СОДЕРЖАНИЕ ОТЧЕТА

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Разработать программу, загружающуюся с гибкого диска (1,4 Мб). Программа должна уметь работать с файлами и каталогами (создание).

ТЕОРИЯ, НЕОБХОДИМАЯ ДЛЯ СОЗДАНИЯ ПРОГРАММЫ

Бутсектор загружается БИОсом по адресу 0000:7C00h и занимает 512 байт. Признак того, что первый сектор загрузочный - два последних байта BOOT содержат код 55AAh. Если его затереть, то программа перестанет загружаться.

Файловая система требует того, чтобы весь диск был полностью отформатирован. Одна из программ занимается именно этим. В нашем распоряжении 2 стороны диска (0 и 1). На каждой стороне по 80 дорожек (0..79). Каждая дорожка содержит по 18 секторов (1..18). Каждый сектор содержит 512 байт. Этой информации достаточно, чтобы написать заданную программу.

АЛГОРИТМ РАБОТЫ ПРОГРАММЫ

Программа объединяет в себя 4 программы: DISK.EXE, FSNIL.EXE, UST.EXE, OS.COM, ZAGR.COM. Каждая программа имеет свое предназначение:

DISK.EXE – прописывает на диск загрузчик ZAGR.COM в первый сектор и OS.COM в 8-64 сектора. Программа выполняется из под WINDOWS.

FSNIL.EXE – форматирует дискету перед установкой ОС. Программа выполняется из под WINDOWS.

UST.EXE – занимает в таблице FAT первые 64 сектора. Создает корневой, пустой каталог. Программа выполняется из под WINDOWS.

OS.COM – собственно ядро. Загружается с дискеты.

ZAGR.COM – загрузчик программы. Загружает в память ядро и передает ей управление. Загрузчик загружает ядро в память по адресу: 0FFF:0410H, т.к. этот сегмент свободен.

Алгоритм работы OS.COM

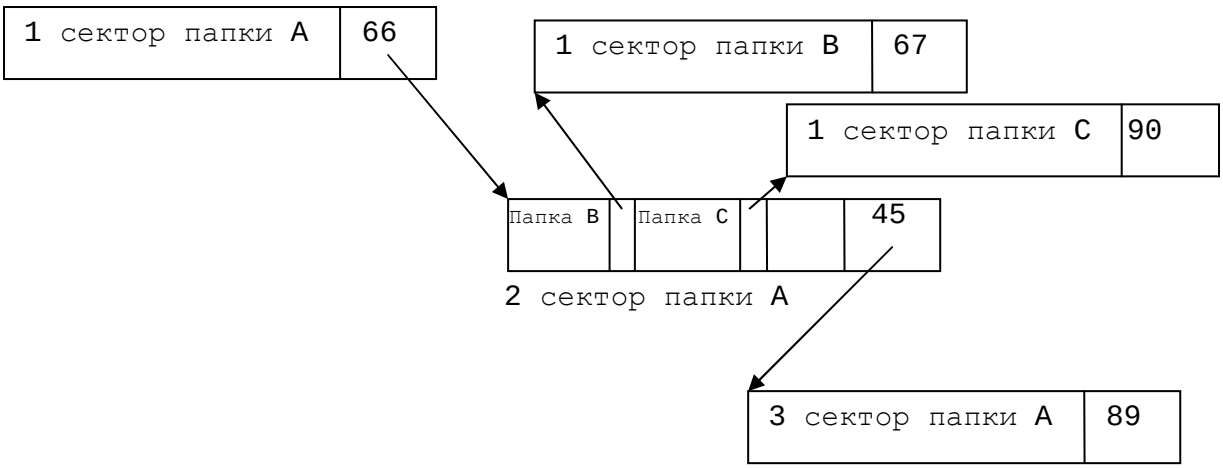
Всего у нас в самом начале занято 64 сектора: 1 сектор для загрузчика, 6 секторов для таблицы FAT и 57 секторов для ядра. Данная таблица FAT занимает 2880 байт, поэтому она занимает 3 Кб=6 секторов.

Формат файловой таблицы (основной).

Сторона диска																		
0 сторона										**1 сторона**								
Дорожки (0-79)										**Дорожки (0-79)**								
0	**1**	**2**	**3**	**..**	**..**	**..**	**..**	**..**	**79**	**0**	**1**	**..**	**..**	**..**	**..**	**..**	**..**	**79**
[1]=1	[1]=1	..	..	..	..	..	..	..	1	1	..	..	..	..	..	..	..	1
[2]=1	[2]=1	..	..	..	..	..	..	..	2	2	..	..	..	..	..	..	..	2
[3]=1	[3]=1	..	..	..	..	..	..	..	3	3	..	..	..	..	..	..	..	3
[4]=1	[4]=1	..	..	..	..	..	..	..	4	4	..	..	..	..	..	..	..	4
[5]=1	[5]=1	..	..	..	..	..	..	..	5	5	..	..	..	..	..	..	..	5
[6]=1	[6]=1	..	..	..	..	..	..	..	6	6	..	..	..	..	..	..	..	6
[7]=1	..	..	..	..	..	..	..	..	7	7	..	..	..	..	..	..	..	7
[8]=1	..	..	..	..	..	..	..	..	8	8	..	..	..	..	..	..	..	8
[9]=1	..	..	..	..	..	..	..	..	9	9	..	..	..	..	..	..	..	9
[10]=1	..	..	[64]=1	..	..	..	..	..	10	10	..	..	..	..	..	..	..	10
[11]=1	..	..	[65]=0	..	..	..	..	..	11	11	..	..	..	..	..	..	..	11
[12]=1	..	..	..	..	..	..	..	..	12	12	..	..	..	..	..	..	..	12
[13]=1	..	..	..	..	..	..	..	..	13	13	..	..	..	..	..	..	..	13
[14]=1	..	..	..	..	..	..	..	..	14	14	..	..	..	..	..	..	..	14
[15]=1	..	..	..	..	..	..	..	..	15	15	..	..	..	..	..	..	..	15
[16]=1	..	..	..	..	..	..	..	..	16	16	..	..	..	..	..	..	..	16
[17]=1	..	..	..	..	..	..	..	..	17	17	..	..	..	..	..	..	..	17
[18]=1	..	..	..	..	..	..	..	..	18	18	..	..	..	..	..	..	..	18

Корневая папка прописывается в 65 сектор и может занимать при необходимости несколько секторов.

Каждая папка занимает по несколько секторов информации (содержимое других папок). В примере ниже корневая папка занимает 3 сектора, то есть она может содержать в себе информацию о 51*3=153 папках. Каждая папка содержит информацию о себе в размере 10 байт = 8 (имя) + 2 указатель на папку (сектор, с которого надо начинать чтение этой папки). На примере корневой папки рассмотрим ее содержимое:



В конце каждого сектора должен быть указатель для продолжения чтения папки. Он занимает 2 байта (то есть число указателей равно 65535 штук). Если же эти 2 байта пусты, то дальше папку не читаем, то есть конец содержимого папки. Соответственно, задача ОС найти свободные сектора при запросе на продолжение папки. Таким образом, достигается большая вложенность папок - пока хватит секторов на дискете.

При чтении папки в ОС есть память, куда происходит чтение содержимого папки, а потом отображение пользователю на экран.

При открытии папки программа сохраняет указатель на сектор родительской папки, то есть пока открываем папки - записываем последовательно их указатели. Поэтому можно входить и выходить из папок.

ТЕКСТ ПРОГРАММЫ

Листинг ZAGR.COM

```
;-------------------------------------------------------------------------------
;        ПРОГРАММА, НАХОДЯЩАЯСЯ В ПЕРВОМ СЕКТОРЕ - ЗАГРУЗЧИК OSBT
;-------------------------------------------------------------------------------
.MODEL TINY
.CODE
;-------------------------------------------------------------------------------
    ORG 100H
START:
    ;-------------------------------------------------------------------------
    ;                     ЗАГРУЗКА ЯДРА В ПАМЯТЬ
    ;-------------------------------------------------------------------------
    JMP DATA
      J   DW 0
      JI  DB 0
      STROKA DB 'LOADING OF OS, PLEASE WAIT...' ;29 СИМВОЛОВ
    DATA:
    ;-------------------------------------------------------------------------
    ;                    ПОКАЗЫВАЕМ ПРОЦЕСС ЗАГРУЗКИ
    ;-------------------------------------------------------------------------
    MOV AH,00h
    MOV AL,02h
    INT 10h
    MOV AH,02H
    XOR AL,AL
    XOR BX,BX
    MOV DH,100
    MOV DL,0
    INT 10H
    ;-------------------ВЫВОД НАДПИСИ-----------------------------------------
    MOV AX,0B800h
    MOV ES,AX
    MOV CX,29
    MOV DI,1010
    MOV SI,7B00H+OFFSET STROKA
    PUSH DS
    XOR AX,AX
    MOV DS,AX
    CIL:
        MOV AH,7
        MOV AL,DS:[SI]
        MOV ES:[DI],AX
        ADD DI,2
        INC SI
    LOOP CIL
    POP DS
    ;-------------------------------------------------------------------------
    MOV AX,0FFFH  ;ЗАГРУЖАЕМ ВО 2 СЕГМЕНТ В ПАМЯТИ 0:10000H = 0FFFH:0010H
    MOV ES,AX
    MOV CX,11     ;НАША ОС ЗАНИМАЕТ 57 СЕКТОРОВ (ЯДРО + ПАМЯТЬ)
    MOV J,0
    MOV JI,8
    MOV DI,1140
    CIKL_UST:
          PUSH CX
            MOV AH,02H
            MOV AL,01 ;КОЛИЧЕСТВО ЧИТАЕМЫХ СЕКТОРОВ
            MOV BX,410H
            ADD BX,J
            ADD J,512
            MOV CH,00 ;ДОРОЖКА
            MOV CL,JI ;НАЧАЛЬНЫЙ СЕКТОР
            INC JI
            MOV DH,00 ;ГОЛОВКА
            MOV DL,00 ;ДИСКОВОД 0
            INT 13H
          ;-----------------PROGRESSBAR-----------------------------------
          PUSH ES
            MOV AX,0B800h
            MOV ES,AX
            MOV AH,31
            MOV AL,20H
            MOV ES:[DI],AX
            ADD DI,2
          POP ES
          ;-------------------------------------------------------------------
          POP CX
    LOOP CIKL_UST
    ;-------------------------------------------------------------------------
    MOV CX,18     ;НАША ОС ЗАНИМАЕТ 57 СЕКТОРОВ (ЯДРО + ПАМЯТЬ)
```

```
    MOV JI,1
    CIKL_UST1:
          PUSH CX
            MOV AH,02H
            MOV AL,01 ;КОЛИЧЕСТВО ЧИТАЕМЫХ СЕКТОРОВ
            MOV BX,410H
            ADD BX,J
            ADD J,512
            MOV CH,01 ;ДОРОЖКА
            MOV CL,JI ;НАЧАЛЬНЫЙ СЕКТОР
            INC JI
            MOV DH,00 ;ГОЛОВКА
            MOV DL,00 ;ДИСКОВОД 0
            INT 13H
          ;-----------------PROGRESSBAR-------------------------------------
          PUSH ES
            MOV AX,0B800h
            MOV ES,AX
            MOV AH,31
            MOV AL,20H
            MOV ES:[DI],AX
            ADD DI,2
          POP ES
          ;-------------------------------------------------------------------
          POP CX
    LOOP CIKL_UST1
    ;-----------------------------------------------------------------------
    MOV CX,18     ;НАША ОС ЗАНИМАЕТ 57 СЕКТОРОВ (ЯДРО + ПАМЯТЬ)
    MOV JI,1
    CIKL_UST2:
          PUSH CX
            MOV AH,02H
            MOV AL,01 ;КОЛИЧЕСТВО ЧИТАЕМЫХ СЕКТОРОВ
            MOV BX,410H
            ADD BX,J
            ADD J,512
            MOV CH,02 ;ДОРОЖКА
            MOV CL,JI ;НАЧАЛЬНЫЙ СЕКТОР
            INC JI
            MOV DH,00 ;ГОЛОВКА
            MOV DL,00 ;ДИСКОВОД 0
            INT 13H
          ;-----------------PROGRESSBAR-------------------------------------
          PUSH ES
            MOV AX,0B800h
            MOV ES,AX
            MOV AH,31
            MOV AL,20H
            MOV ES:[DI],AX
            ADD DI,2
          POP ES
          ;-------------------------------------------------------------------
          POP CX
    LOOP CIKL_UST2
    ;-----------------------------------------------------------------------
    MOV CX,10     ;НАША ОС ЗАНИМАЕТ 57 СЕКТОРОВ (ЯДРО + ПАМЯТЬ)
    MOV JI,1
    CIKL_UST3:
          PUSH CX
            MOV AH,02H
            MOV AL,01 ;КОЛИЧЕСТВО ЧИТАЕМЫХ СЕКТОРОВ
            MOV BX,410H
            ADD BX,J
            ADD J,512
            MOV CH,03 ;ДОРОЖКА
            MOV CL,JI ;НАЧАЛЬНЫЙ СЕКТОР
            INC JI
            MOV DH,00 ;ГОЛОВКА
            MOV DL,00 ;ДИСКОВОД 0
            INT 13H
          ;-----------------PROGRESSBAR-------------------------------------
          PUSH ES
            MOV AX,0B800h
            MOV ES,AX
            MOV AH,31
            MOV AL,20H
            MOV ES:[DI],AX
            ADD DI,2
          POP ES
          ;-------------------------------------------------------------------
          POP CX
    LOOP CIKL_UST3
    ;-----------------------------------------------------------------------
    ;                       ПЕРЕХОД НА ЗАГРУЗЧИК
```

```
    ;-------------------------------------------------------------------------
    DB 0EAh  ;МАШИННЫЙ КОД ДАЛЬНЕГО JMP
    DD 0FFF0410H
END START
```

Листинг UST.EXE

```
;-------------------------------------------------------------------------
;
;                              УСТАНОВКА ОС
;
;-------------------------------------------------------------------------
.MODEL TINY
.STACK 100H
.CODE
START:
;-------------------------------------------------------------------------
;
;                           НЕОБХОДИМЫЕ ПЕРЕМЕННЫЕ
;
;-------------------------------------------------------------------------
JMP DATA_USTANOVKA
                  FAT           DB 3072 DUP (0)
                  I             DB 0
                  J             DW 0
                  SECTOR_MAS    DW 0
                  DOR           DB 0
                  STOR          DB 0
                  SECT          DB 0
                  WI            DW 0
                  NEW_SECTOR    DW 0
DATA_USTANOVKA:
;-------------------------------------------------------------------------
;
;                           НЕОБХОДИМЫЕ ДЕЙСТВИЯ
;
;-------------------------------------------------------------------------
CALL FAT_PROPIS_SEVEN_BAIT
CALL FAT_PROC_WRITE_BEGIN_OS
;-----------------ЗАНИМАЕМ 8 СЕКТОР ПОД ГЛАВНУЮ ПАПКУ-------------------------
CALL CREATE_NEW_SECTOR
CALL FAT_WRITE
EXIT:
     MOV AX,4C00H
     INT 21H
;-------------------------------------------------------------------------
;
;                           НЕОБХОДИМЫЕ ПРОЦЕДУРЫ
;
;-------------------------------------------------------------------------
FAT_PROPIS_SEVEN_BAIT PROC
                          MOV SI,OFFSET CS:FAT
                          MOV CX,64
                          CIKL_FATE:
                                    MOV AL,1
                                    MOV CS:[SI],AL
                                    INC SI
                          LOOP CIKL_FATE
                          RET
FAT_PROPIS_SEVEN_BAIT ENDP
;-------------------------------------------------------------------------
FAT_PROC_WRITE_BEGIN_OS PROC
    ;---------------------------------------------------------------------
    MOV CS:I,2  ;НАЧАЛЬНЫЙ СЕКТОР №2 (1 СЕКТОР - ЗАГРУЗЧИК)
    MOV CS:J,0
    ;---------------------------------------------------------------------
    MOV AX,CS
    MOV ES,AX
    ;---------------------------------------------------------------------
    MOV CX,6
    CIKL_FAT:
             PUSH CX
             ;------------------------------------------------------------
             MOV AH,03H
             MOV AL,01
             LEA BX,CS:FAT
             ADD BX,CS:J
             MOV CH,0
             MOV CL,CS:I
             MOV DH,0
             MOV DL,0
             INT 13H
             ADD CS:J,512
             INC CS:I
```

```
            ;-------------------------------------------------------------
            POP CX
    LOOP CIKL_FAT
    ;-------------------------------------------------------------------------
    RET
FAT_PROC_WRITE_BEGIN_OS ENDP
;-------------------------------------------------------------------------------
;
;                          СОЗДАЕМ КОРНЕВУЮ ПАПКУ
;
;-------------------------------------------------------------------------------
CREATE_NEW_SECTOR PROC
          MOV AX,0
          MOV CS:WI,AX
          ;-------------------
          MOV SI,OFFSET CS:FAT
          MOV CX,2880
          MOV AL,0
          ;------------------------
          CIKL_NEW_MAS_SECTOR_MAS:
                 CMP CS:[SI],AL
                 JE OK
                 ;-------------------------------------------------------------
                 INC SI
                 INC CS:WI
          LOOP CIKL_NEW_MAS_SECTOR_MAS
          ;-------------------------------------------------------------------
OK:       MOV AX,CS:WI
          INC AX
          MOV CS:NEW_SECTOR,AX
          ;-------------------------------------------------------------------
          MOV SI,OFFSET CS:FAT
          DEC AX
          ADD SI,AX
          MOV AL,1
          MOV CS:[SI],AL
          RET
CREATE_NEW_SECTOR ENDP
;-------------------------------------------------------------------------------
;
;                       ВНОСИМ ИЗМЕНЕНИЯ В ТАБЛИЦУ
;
;-------------------------------------------------------------------------------
FAT_WRITE PROC
         XOR AX,AX
         MOV CS:J,AX
         ;--------------------------------------------------------------------
         MOV CX,6
         MOV AX,2      ;НАЧИНАЕМ СО ВТОРОГО СЕКТОРА
         MOV CS:SECTOR_MAS,AX
         CIKL_FAT_WRITE:
                        PUSH CX
                        ;------
                        CALL SECTOR
                        CALL WRITE_SECTOR
                        ADD CS:J,512
                        ;-------------
                        MOV AX,CS:SECTOR_MAS
                        INC AX
                        MOV CS:SECTOR_MAS,AX
                        ;-------------------
                        POP CX
         LOOP CIKL_FAT_WRITE
         RET
FAT_WRITE ENDP
;-------------------------------------------------------------------------------
;
;                 ПЕРЕВОД ИЗ ЛОГИЧЕСКОГО СЕКТОРА В ФИЗИЧЕСКИЙ
;
;-------------------------------------------------------------------------------
SECTOR PROC
      MOV AX,CS:SECTOR_MAS
      MOV BX,AX
      CMP AX,1440
      JG STOR_ONE
      ;--------------------------------------------------------------------
      MOV CS:STOR,0
      MOV CL,18
      DIV CL
      CMP AH,0
      JNE NET_OST
      ;--------------------------------------------------------------------
      DEC AX
      MOV CS:DOR,AL
```

```
      MOV DL,18
      MOV CS:SECT,DL
      JMP EXIT_SECTOR
      ;-------------------------------------------------------------------------
      NET_OST:
      MOV CS:DOR,AL
      MOV CS:SECT,AH
      JMP EXIT_SECTOR
      ;-------------------------------------------------------------------------
      STOR_ONE:
      MOV CS:STOR,1
      SUB AX,1440
      MOV CL,18
      DIV CL
      CMP AH,0
      JNE NET_OST1
      ;-------------------------------------------------------------------------
      DEC AX
      MOV CS:DOR,AL
      MOV DL,18
      MOV CS:SECT,DL
      JMP EXIT_SECTOR
      ;-------------------------------------------------------------------------
      NET_OST1:
      MOV CS:DOR,AL
      MOV CS:SECT,AH
      EXIT_SECTOR:RET
SECTOR ENDP
;-------------------------------------------------------------------------------
;
;                            ФИЗИЧЕСКАЯ ЗАПИСЬ НА ДИСК
;
;-------------------------------------------------------------------------------
WRITE_SECTOR PROC
       PUSH CS
       POP ES
       ;--------
       MOV AH,03
       MOV AL,01
       MOV DI,OFFSET CS:FAT
       ADD DI,CS:J
       MOV BX,DI
       MOV CH,CS:DOR
       MOV CL,CS:SECT
       MOV DH,CS:STOR
       MOV DL,00
       INT 13H
       RET
WRITE_SECTOR ENDP
;-------------------------------------------------------------------------------
END START
```

Листинг FSNIL.EXE

```
;-------------------------------------------------------------------------------
;                          ФОРМАТИРОВЩИК ДИСКЕТЫ
;-------------------------------------------------------------------------------
.MODEL TINY
.STACK 100H
.CODE
;-------------------------------------------------------------------------------
START:
     JMP ERROR
       i dw 0
       STOR DB 0
       DOR DB 0
       SECT DB 0
       sector_mas dw 0
       MAS DB 4096 DUP (0)
     ERROR:
MOV cs:i,1
CIKL_FORMAT:
     cmp cs:i,2881
     je exit
       mov ax,cs:i
       MOV CS:SECTOR_MAS,ax
       CALL SECTOR
       CALL WRITE
     inc cs:i
jmp CIKL_FORMAT
exit:
MOV AX,4C00H
INT 21H
```

```
;-------------------------------------------------------------------------
;/////////////////////////////////////////////////////////////////////////
;/////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------
WRITE PROC
       PUSH CS
       POP ES
       ;---------
       MOV AH,03H
       MOV AL,01
       LEA BX,CS:MAS
       MOV CH,CS:DOR
       MOV CL,CS:SECT
       MOV DH,CS:STOR
       MOV DL,00
       INT 13H
       RET
WRITE ENDP
;-------------------------------------------------------------------------
;/////////////////////////////////////////////////////////////////////////
;
;/////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------
SECTOR PROC
      MOV AX,CS:SECTOR_MAS
      MOV BX,AX
      CMP AX,1440
      JG STOR_ONE
      ;-----------------------------------------------------------------
      ;-----------------------------------------------------------------
      ;-----------------------------------------------------------------
      MOV CS:STOR,0
      MOV CL,18
      DIV CL
      ;-----
      CMP AH,0
      JNE NET_OST
      ;-----------------------------------------------------------------
      DEC AX
      MOV CS:DOR,AL
      MOV DL,18
      MOV CS:SECT,DL
      JMP EXIT_SECTOR
      ;-----------------------------------------------------------------
      NET_OST:
      MOV CS:DOR,AL
      MOV CS:SECT,AH
      JMP EXIT_SECTOR
      ;-----------------------------------------------------------------
      ;-----------------------------------------------------------------
      ;-----------------------------------------------------------------
      STOR_ONE:
      MOV CS:STOR,1
      SUB AX,1440
      ;-----------------------------------------------------------------
      MOV CL,18
      DIV CL
      ;-----
      CMP AH,0
      JNE NET_OST1
      ;-----------------------------------------------------------------
      DEC AX
      MOV CS:DOR,AL
      MOV DL,18
      MOV CS:SECT,DL
      JMP EXIT_SECTOR
      ;-----------------------------------------------------------------
      NET_OST1:
      MOV CS:DOR,AL
      MOV CS:SECT,AH
      JMP EXIT_SECTOR
      EXIT_SECTOR:RET
SECTOR ENDP
;-------------------------------------------------------------------------
END START
```

Листинг DISK.EXE

```
;-------------------------------------------------------------------------
;                  ПРОГРАММА, УСТАНАВЛИВАЮЩАЯ ОС НА ДИСК
;-------------------------------------------------------------------------
.MODEL SMALL
.STACK 100h
```

```
.CODE
;--------------------------------------------------------------------------
START:
    ;----------------------------------------------------------------------
    JMP NEAR:DATA
             BUFER  DB 512 DUP (0)
             BUFER2 DB 29184 DUP (0)
             NAMER1 DB 'D:\TASM\BIN\OS.COM',0
             NAMER2 DB 'D:\TASM\BIN\ZAGR.COM',0
             DECK EQU 29184
             DECK2 EQU 353
             JI DB 0
             J DW 0
             CXX DW 0
    DATA:
    ;----------------------------------------------------------------------
    MOV AX,@CODE
    MOV DS,AX
    ;-----------
    MOV AX,3D00H
    LEA DX,CS:NAMER2
    XOR CX,CX
    INT 21H
    PUSH AX
    POP BX
    ;-----------
    MOV AX,3F00H
    MOV CX,DECK2
    LEA DX,CS:BUFER  ;ЗАНЕСЛИ В БУФЕР РАЗМЕР COM-ФАЙЛА
    INT 21H
    ;-------------ПОКАЗАЛИ, ЧТО С ДИСКА НЕОБХОДИМО ГРУЗИТСЯ-------------------
    LEA SI,CS:BUFER
    ADD SI,510
    MOV AL,55h
    MOV DS:[SI],AL
    ;-----------
    LEA SI,CS:BUFER
    ADD SI,511
    MOV AL,0AAh
    MOV DS:[SI],AL
    ;----------------------------------------------------------------------
    ;                      ПРОПИСЫВАЕМ ПЕРВЫЙ СЕКТОР
    ;----------------------------------------------------------------------
    PUSH DS
    POP ES
    ;-----
    MOV AH,03H
    MOV AL,01
    LEA BX,CS:BUFER
    MOV CH,00
    MOV CL,01
    MOV DH,00
    MOV DL,00
    INT 13H
    ;----------------------------------------------------------------------
    ;                      ПРОПИСЫВАЕМ 8-64 СЕКТОРА
    ;----------------------------------------------------------------------
    MOV AX,3D00H
    LEA DX,CS:NAMER1
    XOR CX,CX
    INT 21H
    PUSH AX
    POP BX
    ;-----------
    MOV CS:JI,8
    MOV CS:J,0
    MOV AX,3F00H
    MOV CX,DECK
    LEA DX,CS:BUFER2  ;ЗАНЕСЛИ В БУФЕР РАЗМЕР COM-ФАЙЛА
    INT 21H
    MOV CX,11
    CIKL_UST:
          MOV CS:CXX,CX
            MOV AH,03H
            MOV AL,01
            LEA BX,CS:BUFER2
            ADD BX,CS:J
            MOV CH,00
            MOV CL,CS:JI
            INC CS:JI
            ADD CS:J,512
            MOV DH,00
            MOV DL,00
            INT 13H
```

```
          MOV CX,CS:CXX
    LOOP CIKL_UST
    ;-------------------------------------------------------------------------
    MOV CX,18     ;НАША ОС ЗАНИМАЕТ 57 СЕКТОРОВ (ЯДРО + ПАМЯТЬ)
    MOV JI,1
    CIKL_UST1:
          MOV CS:CXX,CX
            MOV AH,03H
            MOV AL,01
            LEA BX,CS:BUFER2
            ADD BX,CS:J
            MOV CH,01
            MOV CL,CS:JI
            INC CS:JI
            ADD CS:J,512
            MOV DH,00
            MOV DL,00
            INT 13H
          MOV CX,CS:CXX
    LOOP CIKL_UST1
    ;-------------------------------------------------------------------------
    MOV CX,18     ;НАША ОС ЗАНИМАЕТ 57 СЕКТОРОВ (ЯДРО + ПАМЯТЬ)
    MOV JI,1
    CIKL_UST2:
           MOV CS:CXX,CX
            MOV AH,03H
            MOV AL,01
            LEA BX,CS:BUFER2
            ADD BX,CS:J
            MOV CH,02
            MOV CL,CS:JI
            INC CS:JI
            ADD CS:J,512
            MOV DH,00
            MOV DL,00
            INT 13H
          MOV CX,CS:CXX
    LOOP CIKL_UST2
    ;-------------------------------------------------------------------------
    MOV CX,10     ;НАША ОС ЗАНИМАЕТ 57 СЕКТОРОВ (ЯДРО + ПАМЯТЬ)
    MOV JI,1
    CIKL_UST3:
          MOV CS:CXX,CX
            MOV AH,03H
            MOV AL,01
            LEA BX,CS:BUFER2
            ADD BX,CS:J
            MOV CH,03
            MOV CL,CS:JI
            INC CS:JI
            ADD CS:J,512
            MOV DH,00
            MOV DL,00
            INT 13H
          MOV CX,CS:CXX
    LOOP CIKL_UST3
;-------------------------------------------------------------------------------
EXIT:
    MOV AX,4C00H
    INT 21H
END START
```

Листинг OS.COM

```
.MODEL TINY
.CODE
     ORG 100H
START:
     JMP NEAR:DATA_OS
       ;-----------------------------------------------------------------------
       COMER DB 'COMMAND:'
       NAN DB 'OSBT'
       EDIT_I DW 0
       ;-----------------------------------------------------------------------
       DH_PIKI DB 0
       puti dw 0
       DH_KOREN_OR_NE_KOREN DB 0
       KAM DW 0
       ;---------------------FOLDER_CREATE-----------------------------------
       NAMER            DB 8 DUP (0)
       FOLDER_MAS       DW 100 DUP (0)
       I                DB 0
```

```
  FLAG_ONE          DB 0
  I_FOLD            DW 0
  TEC_UK            DW 0
  FLAG_PROVERKA_FOLDER DB 0
  UK                DW 0
  I_F               DW 0                              ;CHISLO BUKV V FAILE
  ;---------------------------MAIN-----------------------------------
  KORE              DW 0
  PAPKI             DW 100 DUP (0)
  PAP               DW 0
  UKI               DW 0
  AX_PAP            DW 0            ;УКАЗАТЕЛЬ НА ПАПКУ, НА КОТОРУЮ НАЖАЛИ
  DL1               DB 2
  DH_PIC            DB 2
  M                 DW 0
  M1                DW 0
  XAL               DB 0
  XAH               DB 0
  ;--------------------CREATE_NEW_SECTOR---------------------------------
  WI                DW 0
  FLAG_NEW_SECTOR   DB 0
  NEW_SECTOR        DW 0
  ;--------------------SECTOR_ONE-----------------------------------------
  JIK               DW 0
  SEC               DW 0
  SECTOR_MAS        DW 0
  ;--------------------FAT_PROC_WRITE_BEGIN_OS--------------------------
  J                 DW 0
  ;-----------------FAT_PROPIS_SEVEN_BAIT-------------------------------
  MAS               DB 10000 DUP (0)
  MAS_COPY          DB 10000 DUP (0)
  ;----------------------POISK_FAT-----------------------------------------
  FLAG_POISK_FAT    DB 0
  MOSK              DW 100 DUP (0)
  I_P               DB 0
  ONE_P             DB 0
  X                 DW 0
  FLAG_FAT          DB 0
  ;------------------------MAS_NUMBER-----------------------------------
  ROT               DW 0
  ;--------------------------READ_MEMORY-----------------------------------
  SLOBO             DW 0
  ;-------------------------PIKI_PROC-------------------------------------
  PIKI              DB 0
  ;----------------------NIL_FILE-------------------------------------------
  STROL             DB '...    '
  ;---------------------SECTOR-----------------------------------------------
  DOR               DB 0
  STOR              DB 0
  SECT              DB 0
  ;--------------------ANALIZ-----------------------------------------------
  UCASATEL_SECTOR   DW 0
  FLAG              DB 0
  ;---------------------ADD_2_BAIT-------------------------------------------
  ID                DB 0
  ;----------------------ADAS_PROC-------------------------------------------
  S6                DB 'A:\'
  S7                DB '...'
  ;-------------------------------------------------------------------------
  MEMORY_PUT DB 70 DUP (0)
  MP DW 0
  CONST_MP EQU 70
  ;-------------------------------------------------------------------------
  FLAG_PROVERKA_EDIT DB 0
  ;-----------------------------------ОС НОВАЯ-----------------------------
  BUFER_EDIT DB 1638 DUP (0)  ;БУФЕР ДЛЯ РЕДАКТОРА (1638 БАЙТ)
  BUFER_COM DB 60 DUP (0)
  FAT               DB 3072 DUP (0)
  UKBUL DW 0
DATA_OS:
;-------------------------------------------------------------------------
;-----------------УСТАНАВЛИВАЕМ ВИДЕОРЕЖИМ-----------------------------
;-------------------------------------------------------------------------
MOV AH,00H
MOV AL,02H
INT 10H
;-------------------------------------------------------------------------
;----------------------УБИРАЕМ КУРСОР С ЭКРАНА--------------------------
;-------------------------------------------------------------------------
MOV AH,02H
XOR BX,BX
MOV DH,26
MOV DL,0
INT 10H
```

```
;-------------------------------------------------------------------------
;---------------------ЧИТАЕМ В ПАМЯТЬ ТАБЛИЦУ FAT-----------------------
;-------------------------------------------------------------------------
CALL READ_FAT
;-------------------------------------------------------------------------
;-------------ЧИТАЕМ СОДЕРЖИМОЕ КОРНЕВОГО КАТАЛОГА В ПАМЯТЬ--------------
;-------------------------------------------------------------------------
MOV AX,65
;--------
MOV 310H+UK,AX
;-------------
MOV 310H+UKI,AX               ;ВНАЧАЛЕ PAPKI КОРНЕВОЙ КАТАЛОГ С 65 СЕКТОРА
CALL READ_MEMORY
;---------------
MOV 310H+KORE,1                              ;ДА, ВЫВОДИМ КОРНЕВУЮ ПАПКУ
MOV 310H+DH_KOREN_OR_NE_KOREN,2      ;ВЫВОДИМ С 2 СТРОКИ, Т.К. ЭТО КОРЕНЬ
MOV 310H+M1,1
CALL PAPKA
;-------------------------------------------------------------------------
;--------------------------------РИСУЕМ ИНТЕРФЕЙС-----------------------
;-------------------------------------------------------------------------
CALL NAM
CALL RAMA
CALL COMD
CALL DVET
;-------------------------------------------------------------------------
MOV 310H+DH_PIC,2
MOV 310H+PIKI ,2                             ;ВЫВОД ВО 2 СТРОКЕ УКАЗАТЕЛЯ
CALL PIKI_PROC
;-------------------------------------------------------------------------
;----------------------ГЛАВНЫЙ ЦИКЛ ПРОГРАММЫ---------------------------
;-------------------------------------------------------------------------
MAIN_CIKL:
         ;---------------------------------------------------------------
         ;///////////////////////////////////////////////////////////////
         ;                  +   ЖДЕМ НАЖАТИЯ КЛАВИШИ   +
         ;///////////////////////////////////////////////////////////////
         ;---------------------------------------------------------------
         MOV AH,11H
         INT 16H
         JNZ ES_KLAVA
         JMP NEAR:TIMER@
         ;---------------ДА, НАЖАЛИ КЛАВИШУ-------------------------------
         ES_KLAVA:
                  MOV AL,00
                  MOV AH,10H
                  INT 16H
                  MOV 310H+XAL,AL
                  MOV 310H+XAH,AH
         ;---------------------------------------------------------------
         ;///////////////////////////////////////////////////////////////
         ;                  +   ПРОВЕРКА ENTER   +
         ;///////////////////////////////////////////////////////////////
         ;---------------------------------------------------------------
         CMP AH,1CH
         JE N_ENTER
                   JMP NEAR:NET_ENTER
         N_ENTER:
         ;---------------------------------------------------------------
         ;///////////////////////////////////////////////////////////////
         ;                  +    НАЖАЛИ ENTER    +
         ;///////////////////////////////////////////////////////////////
         ;---------------------------------------------------------------
         ;----------НАЖАЛИ НА ENTER - СОЗДАТЬ ПАПКУ ИЛИ ВОЙТИ В ПАПКУ----
         ;-------ПРОВЕРКА НА СОЗДАНИЕ НОВОЙ ПАПКИ-------------------------
         CALL PROVERKA_CREATE_FOLDER
         CMP 310H+FLAG_PROVERKA_CREATE_FOLDER,1
         JNE NET_CREATE_FOLDER
            CALL OCHISTKA_MAS
            CALL READ_MEMORY                  ;UK - УЖЕ НАШЕЛСЯ ПРИ ВВОДЕ
            CALL FOLDER_CREATE                              ;СОЗДАЛИ ПАПКУ
            CALL PAPKA
            JMP MAIN_CIKL
         ;-------НЕТ, МЫ НЕ СОБИРАЕМСЯ СОЗДАВАТЬ НОВОЙ ПАПКИ-------------
         NET_CREATE_FOLDER:
                           MOV AX,310H+M1
                           ADD AX,310H+M
         ;-------------AX - ПАПКА, НА КОТОРУЮ НАЖАЛИ---------------------
         MOV 310H+AX_PAP,AX
         CMP AX,0
         JE EXIT_PAPKA@
         ;---------------------------------------------------------------
         ;---------------------МЫ ВХОДИМ В ПАПКУ-------------------------
         ;---------------------------------------------------------------
```

```
CALL PRED_UK_MEMORY     ;ЗАПОМИНАЕМ УКАЗАТЕЛЬ РОДИТЕЛЬСКОЙ ПАПКИ
CALL POISK_AX_UKAZATAL
MOV 310H+UK,AX
;---------------------
;КАКУЮ ПАПКУ ВЫВОДИТЬ?
;---------------------
CMP AX,65                       ;ЕСЛИ 8 СЕКТОР - ТО КОРНЕВУЮ ПАПКУ
 JNE EXI_KOREN@
   MOV 310H+DH_KOREN_OR_NE_KOREN,2
   MOV 310H+KORE,1               ;ДА, МЫ ВОШЛИ В ГЛАВНЫЙ КАТАЛОГ
   MOV 310H+M1,1
   JMP ES_EXI_KOREN@
 EXI_KOREN@:
   MOV 310H+DH_KOREN_OR_NE_KOREN,3
   MOV 310H+KORE,0
   MOV 310H+M1,0
 ES_EXI_KOREN@:
;---------------
CALL OCHISTKA_MAS
CALL READ_MEMORY                ;ЧИТАЕМ ПАПКУ ДОЧЕРНЮЮ В ПАМЯТЬ
CALL STIR_DISPLEI
MOV 310H+M,0
CALL PAPKA
;--------------
MOV 310H+DH_PIC,2
MOV 310H+PIKI,2
CALL PIKI_PROC
JMP MAIN_CIKL
;-----------------------------------------------------------------
;-------------------МЫ ВЫХОДИМ ИЗ ПАПКИ-------------------------
;-----------------------------------------------------------------
EXIT_PAPKA@:
CALL SLED_UK_MEMORY;ВОССТАНАВЛИВАЕМ УКАЗАТЕЛЬ РОДИТЕЛЬСКОЙ ПАПКИ
                        ;ВЫХОД ПАПКИ AX=УКАЗАТЕЛЬ НА РОД. ПАПКУ
MOV 310H+UK,AX
;-----------------------------------------------------------------
;-----------------КАКУЮ ПАПКУ ВЫВОДИТЬ?-------------------------
;-----------------------------------------------------------------
CMP AX,65
 JNE EXI_KOREN
   MOV 310H+DH_KOREN_OR_NE_KOREN,2
   MOV 310H+KORE,1                 ;ДА, МЫ ВОШЛИ В ГЛАВНЫЙ КАТАЛОГ
   MOV 310H+M1,1
   JMP ES_EXI_KOREN
 EXI_KOREN:
   MOV 310H+DH_KOREN_OR_NE_KOREN,3
   MOV 310H+KORE,0
   MOV 310H+M1,0
 ES_EXI_KOREN:
;-----------------------------------------------------------------
CALL READ_MEMORY                ;ЧИТАЕМ ПАПКУ ДОЧЕРНЮЮ В ПАМЯТЬ
CALL STIR_DISPLEI
MOV 310H+M,0
CALL PAPKA
;---------
MOV 310H+DH_PIC,2
MOV 310H+PIKI,2
CALL PIKI_PROC
JMP MAIN_CIKL
NET_ENTER:
;-----------------------------------------------------------------
;/////////////////////////////////////////////////////////////////
;              +    ПРОВЕРКА ВНИЗ ИЛИ ВВЕРХ    +
;/////////////////////////////////////////////////////////////////
;-----------------------------------------------------------------
CALL BN_OR_BB
;-----------------------------------------------------------------
;/////////////////////////////////////////////////////////////////
;               +    ВВОД В КОМАНДНУЮ СТРОКУ    +
;/////////////////////////////////////////////////////////////////
;-----------------------------------------------------------------
;----------ЗАЩИТА ОТ ПОСТРОННИХ КЛАВИШ--------------------------
;-----------------------------------------------------------------
CMP 310H+XAH,1DH
JNE NET_CTRL
             JMP NEAR:MAIN_CIKL
NET_CTRL:
;-----------------------------------------------------------------
CMP 310H+XAH,47H
JGE NET_@@
         JMP OK_BBOD
NET_@@:
       CMP 310H+XAH,53H
       JLE NET_@@@
```

```
                    JMP OK_BBOD
       NET_@@@:
                    JMP NEAR:MAIN_CIKL
;-----------------------------------------------------------
OK_BBOD:
CMP 310H+XAH,3BH
JGE NET_@@@@
            JMP OK_BBOD1
NET_@@@@:
        CMP 310H+XAH,44H
        JLE NET_@@@@@
                    JMP OK_BBOD1
        NET_@@@@@:
                    JMP NEAR:MAIN_CIKL
;-----------------------------------------------------------
OK_BBOD1:
CMP 310H+XAH,85H
JNE NET_F11
            JMP NEAR:MAIN_CIKL
NET_F11:
CMP 310H+XAH,86H
JNE NET_F12
            JMP NEAR:MAIN_CIKL
NET_F12:
;-----------------------------------------------------------
;///////////////////////////////////////////////////////////
;                +   ПРОВЕРКА НА НАЖАТИЕ BS  +
;///////////////////////////////////////////////////////////
;-----------------------------------------------------------
CMP 310H+XAH,0EH
JE ES_BS
;-----------------------------------------------------------
;///////////////////////////////////////////////////////////
;                 +    НЕ НАЖИМАЛИ BS   +
;///////////////////////////////////////////////////////////
;-----------------------------------------------------------
CMP 310H+UKBUL,54
JE NET_UKBIL
    MOV BX,0B800H
    MOV ES,BX
    ;------------------
    MOV SI,310H+OFFSET BUFER_COM
    ADD SI,310H+UKBUL
    MOV CS:[SI],AL
    ;-------------
    INC 310H+UKBUL
    MOV AH,7
    MOV CX,54
    MOV SI,310H+OFFSET BUFER_COM
    MOV DI,3856
    CIKL_CO:
        MOV AL,CS:[SI]
        MOV ES:[DI],AX
        INC SI
        ADD DI,2
    LOOP CIKL_CO
    ;-----------------------------------------------------------
NET_UKBIL:
JMP NEAR:MAIN_CIKL
;-----------------------------------------------------------
;///////////////////////////////////////////////////////////
;                  +     НАЖАЛИ BS     +
;///////////////////////////////////////////////////////////
;-----------------------------------------------------------
ES_BS:
    ;------------ПРОВЕРКА НА ЦЕЛОСТНОСТЬ-----------------------
    CMP 310H+UKBUL,0
    JNE ES11
       JMP NEAR:MAIN_CIKL
    ES11:
    ;-----------------------------------------------------------
    MOV BX,0B800H
    MOV ES,BX
    ;------------------
    MOV SI,310H+OFFSET BUFER_COM
    DEC 310H+UKBUL
    ADD SI,310H+UKBUL
    MOV AL,0
    MOV CS:[SI],AL
    ;-------------
    MOV AH,7
    MOV CX,54
    MOV SI,310H+OFFSET BUFER_COM
    MOV DI,3856
```

```
                CIKL_CO1:
                    MOV AL,CS:[SI]
                    MOV ES:[DI],AX
                    INC SI
                    ADD DI,2
                CIKL_CO1
                JMP MAIN_CIKL
                ;-------------------------------------------------------------
            ;-------------------------------------------------------------
            ;/////////////////////////////////////////////////////////////
            ;               +     ПОКАЗАЛИ ВРЕМЯ    +
            ;/////////////////////////////////////////////////////////////
            ;-------------------------------------------------------------
            TIMER@:
                  CALL TIMER
            NET_ES:
    JMP NEAR:MAIN_CIKL
;-------------------------------------------------------------------------
;/////////////////////////////////////////////////////////////////////////
;/////////////////////////////////////////////////////////////////////////
;                    ОСНОВНЫЕ ПРОЦЕДУРЫ ПРОГРАММЫ
;/////////////////////////////////////////////////////////////////////////
;/////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------
;                                ANALIZ
;-------------------------------------------------------------------------
ANALIZ PROC
       MOV SI,310H+OFFSET UCASATEL_SECTOR
       CMP 310H+SLOBO,0
       JE FIL
       ;----------------------------------------------------------------------
       MOV 310H+FLAG,1
       MOV AX,310H+SLOBO
       MOV WORD PTR CS:[SI],AX
       JMP EXIT_ANALIZ
       ;----------------------------------------------------------------------
       FIL:
       XOR AL,AL
       MOV 310H+FLAG,AL
       XOR AX,AX
       MOV 310H+UCASATEL_SECTOR,AX
       ;----------------------------------------------------------------------
       EXIT_ANALIZ:
                 RET
ANALIZ ENDP
;-------------------------------------------------------------------------
POISK_FAT PROC
         MOV AL,0
         MOV 310H+ONE_P,AL
         MOV 310H+I_P,AL
         MOV 310H+FLAG_FAT,AL
         ;-----------------
         MOV CX,310H+X
         MOV AX,0
         MOV SI,310H+OFFSET MOSK
         CIFER:
            MOV WORD PTR CS:[SI],AX
            ADD SI,2
         LOOP CIFER
         ;----------------------------------------------------------------------
         MOV DI,310H+OFFSET MOSK
         MOV SI,310H+OFFSET FAT
         MOV CX,2880
         CIKL_POISK_FAT:
               MOV AL,CS:[SI]
               CMP AL,1
               JE NET_MOSK
               ;-------------------------------------------------------------
               MOV CS:[DI],SI
               ADD DI,2
               ;-------
               DEC 310H+X
               XOR AX,AX
               CMP 310H+X,AX
               JE OK_FILE
               ;-------------------------------------------------------------
               NET_MOSK:
                    INC SI
         LOOP CIKL_POISK_FAT
         ;----------------------------------------------------------------------
         MOV 310H+FLAG_POISK_FAT,0
         JMP END_FAT
         ;----------------------------------------------------------------------
         OK_FILE:
```

```
        MOV AL,1
        MOV 310H+FLAG_POISK_FAT,AL
        ;-------------------------------------------------------------------
        END_FAT:
                RET
POISK_FAT ENDP
;-----------------------------------------------------------------------------
FAT_WRITE PROC
        XOR AX,AX
        MOV 310H+JIK,AX
        ;------------
        MOV CX,6
        MOV AX,2
        MOV 310H+SECTOR_MAS,AX
        CIKL_FAT_WRITE:
                        PUSH CX
                        ;------
                        CALL SECTOR
                        CALL WRITE_SECTOR
                        ADD 310H+JIK,512
                        ;-------------
                        MOV AX,310H+SECTOR_MAS
                        INC AX
                        MOV 310H+SECTOR_MAS,AX
                        ;-------------------
                        POP CX
        LOOP CIKL_FAT_WRITE
        RET
FAT_WRITE ENDP
;-----------------------------------------------------------------------------
CREATE_NEW_SECTOR PROC
         MOV AX,0
         MOV 310H+WI,AX
         ;-------------------
         MOV SI,310H+OFFSET FAT
         MOV CX,2880
         MOV AL,0
         ;------------------------
         CIKL_NEW_MAS_SECTOR_MAS1:
                CMP CS:[SI],AL
                JE OK
                ;-------------------------------------------------------------
                INC SI
                INC 310H+WI
         LOOP CIKL_NEW_MAS_SECTOR_MAS1
         ;---------------------------
         MOV 310H+FLAG_NEW_SECTOR,0
         JMP EXIT_NEW_SECTOR
         ;-------------------------------------------------------------------
         ;-------------------------------------------------------------------
         OK:
         MOV AX,310H+WI
         INC AX
         MOV 310H+NEW_SECTOR,AX
         ;-------------------------------------------------------------------
         MOV SI,310H+OFFSET FAT
         DEC AX
         ADD SI,AX
         MOV AL,1
         MOV CS:[SI],AL
         ;-------------
         MOV 310H+FLAG_NEW_SECTOR,1
         EXIT_NEW_SECTOR:RET
CREATE_NEW_SECTOR ENDP
;-----------------------------------------------------------------------------
;
;                        ЧИТАЕМ ТАБЛИЦУ FAT В ПАМЯТЬ
;
;-----------------------------------------------------------------------------
READ_FAT PROC
            MOV 310H+I,2
            MOV 310H+J,0
            ;-------------------------------------------------------------
            MOV AX,CS
            MOV ES,AX
            ;-------------------------------------------------------------
            MOV CX,6
            CIKL_FAT_READ:
                        PUSH CX
                        ;-----------------------------------------------
                        MOV AH,02H
                        MOV AL,01
                        MOV BX,310H+OFFSET FAT
                        ADD BX,310H+J
```

```
                           MOV CH,0
                           MOV CL,310H+I
                           MOV DH,0
                           MOV DL,0
                           INT 13H
                           ADD 310H+J,512
                           INC 310H+I
                           ;-------------------------------------------------
                           POP CX
            LOOP CIKL_FAT_READ
            RET
READ_FAT ENDP
;-------------------------------------------------------------------------------
;
;                             PROCEDURI
;
;-------------------------------------------------------------------------------
SECTOR PROC
      MOV AX,310H+SECTOR_MAS
      MOV BX,AX
      CMP AX,1440
      JG STOR_ONE
      ;-------------------------------------------------------------------------
      MOV 310H+STOR,0
      MOV CL,18
      DIV CL
      CMP AH,0
      JNE NET_OST
      ;-------------------------------------------------------------------------
      DEC AX
      MOV 310H+DOR,AL
      MOV DL,18
      MOV 310H+SECT,DL
      JMP EXIT_SECTOR
      ;-------------------------------------------------------------------------
      NET_OST:
      MOV 310H+DOR,AL
      MOV 310H+SECT,AH
      JMP EXIT_SECTOR
      ;-------------------------------------------------------------------------
      STOR_ONE:
      MOV 310H+STOR,1
      SUB AX,1440
      MOV CL,18
      DIV CL
      CMP AH,0
      JNE NET_OST1
      ;-------------------------------------------------------------------------
      DEC AX
      MOV 310H+DOR,AL
      MOV DL,18
      MOV 310H+SECT,DL
      JMP EXIT_SECTOR
      ;-------------------------------------------------------------------------
      NET_OST1:
      MOV 310H+DOR,AL
      MOV 310H+SECT,AH
      EXIT_SECTOR:RET
SECTOR ENDP
;-------------------------------------------------------------------------------
READ_MEMORY PROC
    XOR AX,AX
    MOV 310H+J,AX
    MOV AX,310H+UK
    CIKL_MEMORY:
       MOV  310H+SECTOR_MAS,AX
       CALL SECTOR
       CALL READ_SECTOR
       CALL ANALIZ
       CMP  310H+FLAG,1
       JNE EXIT_MEMORY
       ;----------------------------------------------------------------------
       MOV AX,310H+UCASATEL_SECTOR
       ADD 310H+J,512
    JMP CIKL_MEMORY
    EXIT_MEMORY:
    RET
READ_MEMORY ENDP
;-------------------------------------------------------------------------------
OCHISTKA_MAS PROC
            MOV SI,310H+OFFSET MAS
            MOV CX,10000
            XOR AL,AL
            CIKL_MAS_OCHISTKA:
```

```
                    MOV CS:[SI],AL
                    INC SI
            LOOP CIKL_MAS_OCHISTKA
            RET
OCHISTKA_MAS ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
WRITE_MAS_NEW_SECTORA PROC
                    MOV AX,310H+I_FOLD
                    MOV BL,2
                    DIV BL
                    ;------------------AX=CHISLO ZAPIS SECTOROV----------------
                    MOV CX,AX
                    ;--------
                    XOR AX,AX
                    MOV 310H+JIK,AX
                    ;-------------------
                    MOV SI,310H+OFFSET FOLDER_MAS
                    CIKL_WRITE_OLD:
                         PUSH CX
                         MOV AX,WORD PTR CS:[SI]
                         ADD SI,2
                         PUSH SI
                         ;-------------------
                         MOV 310H+SECTOR_MAS,AX
                         CALL SECTOR
                         CALL WRITE_SECTOR_OLD
                         ;--------------------
                         POP SI
                         POP CX
                    LOOP CIKL_WRITE_OLD
                    RET
WRITE_MAS_NEW_SECTORA ENDP
;-------------------------------------------------------------------------------
WRITE_SECTOR_OLD PROC
       PUSH CS
       POP ES
       ;--------
       MOV AH,03
       MOV AL,01
       MOV DI,310H+OFFSET MAS
       ADD DI,310H+JIK
       MOV BX,DI
       MOV CH,310H+DOR
       MOV CL,310H+SECT
       MOV DH,310H+STOR
       MOV DL,00
       INT 13H
       ;---------
       MOV AX,512
       ADD 310H+JIK,AX
       RET
WRITE_SECTOR_OLD ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                    СОЗДАНИЕ ПАПКИ В УКАЗАННОМ КАТАЛОГЕ
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
FOLDER_CREATE PROC
             XOR AX,AX
             MOV 310H+I_FOLD,AX
             MOV 310H+FLAG_PROVERKA_FOLDER,AL
             ;-----------------------------------------------------------------
             MOV SI,310H+OFFSET FOLDER_MAS
             MOV AX,310H+UK
             MOV WORD PTR CS:[SI],AX
             MOV AX,2
             ADD 310H+I_FOLD,AX
             ;-----------------------------------------------------------------
             MOV AL,0
             MOV 310H+I,AL
             MOV 310H+FLAG_ONE,AL
             MOV SI,310H+OFFSET MAS
             CIKL_FOLDER:
                     MOV AL,0
                     CMP CS:[SI],AL
                     JE OK_FOLDER
                     ;-----------------------------------------------------
                     MOV AL,1
                     MOV 310H+FLAG_ONE,AL
                     ;-----------------------------------------------------
```

```
        ADD SI,10
        INC 310H+I
        ;-------------------------------------------------------
        CALL PROVERKA
JMP CIKL_FOLDER
;---------------------------------------------------------------
;---------------------------------------------------------------
OK_FOLDER:
MOV AL,0
CMP 310H+I,AL
JNE NET_NEW_SECTOR_FOR_FOLDER
;---------------------------
CMP 310H+FLAG_ONE,AL
JE NET_NEW_SECTOR_FOR_FOLDER
;---------------------------------------------------------------
;---------------------------------------------------------------
;---------------------------------------------------------------
    MOV AL,1
    MOV 310H+FLAG_PROVERKA_FOLDER,AL
NET_NEW_SECTOR_FOR_FOLDER:
;---------------------------------------------------------------
;---------------------------------------------------------------
MOV CX,8
MOV DI,310H+OFFSET NAMER
CIKL_NAME_FOLDER:
          MOV AL,CS:[DI]
          MOV CS:[SI],AL
          INC SI
          INC DI
LOOP CIKL_NAME_FOLDER
;---------------------------------------------------------------
;---------------------------------------------------------------
;---------------------------------------------------------------
PUSH SI
  CALL CREATE_NEW_SECTOR
POP SI
MOV AL,1
CMP 310H+FLAG_NEW_SECTOR,AL
JNE NET_NEW_SECTOR
;---------------------------------------------------------------
;---------------------------------------------------------------
;---------------------------------------------------------------
MOV AX,310H+NEW_SECTOR
MOV WORD PTR CS:[SI],AX
;---------------------------------------------------------------
;---------------------------------------------------------------
;---------------------------------------------------------------
MOV AL,1
CMP 310H+FLAG_PROVERKA_FOLDER,AL
JNE NET_NEW_FOLDER
;---------------------------------------------------------------
;---------------------------------------------------------------
;---------------------------------------------------------------
PUSH SI
  CALL CREATE_NEW_SECTOR
POP SI
MOV AL,1
CMP 310H+FLAG_NEW_SECTOR,AL
JNE NET_NEW_SECTOR
;---------------------------------------------------------------
;---------------------------------------------------------------
;---------------------------------------------------------------
;СЕЙЧАС SI_TEC:   8 SECTOR:INFO(510) ? 1 PAPKA(8),SI_TEC(2)
; ? - УКАЗАТЕЛЬ ГЛАВНОЙ КОРНЕВОЙ ПАПКИ НА СЛЕД СЕКТОР
SUB SI,8
SUB SI,2  ;ТЕПЕРЬ МЫ СЮДА ПИХАЕМ НАШ АДРЕС
MOV AX,310H+NEW_SECTOR
MOV WORD PTR CS:[SI],AX
;---------------------------
MOV SI,310H+OFFSET FOLDER_MAS
MOV AX,2
SUB 310H+I_FOLD,AX
ADD SI,310H+I_FOLD
MOV AX,310H+NEW_SECTOR
MOV WORD PTR CS:[SI],AX
;----------------------
MOV AX,2
ADD 310H+I_FOLD,AX
;---------------------------------------------------------------
CALL WRITE_MAS_NEW_SECTORA
CALL FAT_WRITE
JMP NET_NEW_SECTOR
;---------------------------------------------------------------
;---------------------------------------------------------------
```

```
            ;-------------------------------------------------------------
            NET_NEW_FOLDER:
            CALL WRITE_MAS_NEW_SECTORA
            CALL FAT_WRITE
            ;-------------------------------------------------------------
            NET_NEW_SECTOR:RET
FOLDER_CREATE ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
PROVERKA PROC
          MOV AL,310H+I
          CMP AL,51
          JNE EXIT
             ;-------------------------------------------------------------
             MOV AX,WORD PTR CS:[SI]
             MOV 310H+TEC_UK,AX
             ;-------------------------------------------------------------
             XOR AL,AL
             MOV 310H+I,AL
             ADD SI,2
             ;-------------------------------------------------------------
             PUSH SI
               MOV SI,310H+OFFSET FOLDER_MAS
               ADD SI,310H+I_FOLD
               MOV AX,310H+TEC_UK
               MOV WORD PTR CS:[SI],AX
               MOV AX,2
               ADD 310H+I_FOLD,AX
             POP SI
          EXIT:
          RET
PROVERKA ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;            ЗАПИСЬ В ПАМЯТЬ ТЕКУЩЕГО УКАЗАТЕЛЯ НА ЛЮБУЮ ПАПКУ  (ВХОД UK)
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
PRED_UK_MEMORY PROC
            MOV SI,310H+OFFSET PAPKI
            ADD SI,310H+PAP                                    ;СЧЕТЧИК ПАПОК
            MOV AX,310H+UK                                    ;ВХОДНЫЕ ДАННЫЕ
            MOV WORD PTR CS:[SI],AX
            MOV AX,2
            ADD 310H+PAP,2              ;ПРОДВИГАЕМСЯ ДАЛЬШЕ ПО МАССИВУ PAPKI
            RET
PRED_UK_MEMORY ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;ПО НОМЕРУ В СПИСКЕ ВЫДАЕТ УКАЗАТЕЛЬ НА ПАПКУ, НА КОТОРУЮ НАЖАЛИ (ВЫХОД AX УК)
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
POISK_AX_UKAZATAL PROC
            XOR AL,AL
            MOV 310H+ID,AL
            ;---------------
            MOV AX,310H+AX_PAP
            MOV CX,AX
            DEC CX                      ;ТАК КАК МЫ ЧИТАЕМ НЕ N ПАПОК, А (N-1)
            ;-------------------------------------------------------------
            MOV SI,310H+OFFSET MAS
            CMP CX,0
            JE ALL_10_BAIT
            ;---------------НЕТ, В КОРНЕВОЙ ПАПКЕ БОЛЬШЕ ОДНОЙ ПАПКИ----------
            CIKL_POISK_AX_UKAZATAL:
                                PUSH CX
                                ADD SI,10
                                INC 310H+ID
                                CALL ADD_2_BAIT
                                POP CX
            LOOP CIKL_POISK_AX_UKAZATAL
            ;-------------------------------------------------------------
            ALL_10_BAIT:
            ADD SI,8                                  ;ПРОШЛИ ИМЯ ПАПКИ ИЛИ ФАЙЛА
            MOV AX,WORD PTR CS:[SI]
            ;--------------------ЗАПОМИНАЕМ UKI-------------------------------
            MOV 310H+UKI,AX
            RET
POISK_AX_UKAZATAL ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;         ВОССТАНАВЛИВАЕМ УКАЗАТЕЛЬ (ВЫХОД AX=УК. НА РОДИТЕЛЬСКУЮ ПАПКУ)
```

```
;////////////////////////////////////////////////////////////////////////////
;----------------------------------------------------------------------------
SLED_UK_MEMORY PROC
           MOV SI,310H+OFFSET PAPKI
           MOV AX,310H+PAP
           SUB AX,2
           MOV 310H+PAP,AX
           ;-------------------------------------------------------------------
           ADD SI,AX
           MOV AX,WORD PTR CS:[SI]
           RET
SLED_UK_MEMORY ENDP
;----------------------------------------------------------------------------
;////////////////////////////////////////////////////////////////////////////
;
;////////////////////////////////////////////////////////////////////////////
;----------------------------------------------------------------------------
WRITE_SECTOR PROC
       PUSH CS
       POP ES
       ;--------
       MOV AH,03
       MOV AL,01
       MOV DI,310H+OFFSET FAT
       ADD DI,310H+JIK
       MOV BX,DI
       MOV CH,310H+DOR
       MOV CL,310H+SECT
       MOV DH,310H+STOR
       MOV DL,00
       INT 13H
       RET
WRITE_SECTOR ENDP
;----------------------------------------------------------------------------
;////////////////////////////////////////////////////////////////////////////
;
;////////////////////////////////////////////////////////////////////////////
;----------------------------------------------------------------------------
ADD_2_BAIT PROC
           MOV AL,310H+ID
           CMP AL,51
           JNE EXIT_ADD_2_BAIT
           ;-------------------------------------------------------------------
           XOR AL,AL
           MOV 310H+ID,AL
           ADD SI,2
           ;---------------
           EXIT_ADD_2_BAIT:
           RET
ADD_2_BAIT ENDP
;----------------------------------------------------------------------------
;
;
;
;                            ОТЛАЖЕННЫЕ ПРОЦЕДУРЫ
;
;
;
;----------------------------------------------------------------------------
;////////////////////////////////////////////////////////////////////////////
;                  +    ПРОВЕРКА НА ВВЕРХ ИЛИ ВНИЗ    +
;////////////////////////////////////////////////////////////////////////////
;----------------------------------------------------------------------------
BN_OR_BB PROC
             ……
             RET
BN_OR_BB ENDP
;----------------------------------------------------------------------------
;////////////////////////////////////////////////////////////////////////////
;         +   ЧТЕНИЕ СЕКТОРА ПО ЗАДАННЫМ НАЧАЛЬНЫМ ПАРАМЕТРАМ    +
;////////////////////////////////////////////////////////////////////////////
;----------------------------------------------------------------------------
READ_SECTOR PROC
       MOV AX,CS
       MOV ES,AX
       ;---------
       MOV AH,02H
       MOV AL,01
       MOV BX,310H+OFFSET MAS
       ADD BX,310H+J
       MOV CH,310H+DOR
       MOV CL,310H+SECT
       MOV DH,310H+STOR
       MOV DL,00
```

```
       INT 13H
       ;-------------------------------------------------------------------
       MOV SI,310H+OFFSET MAS
       ADD SI,310H+J
       ADD SI,510
       MOV AX,WORD PTR CS:[SI]
       MOV 310H+SLOBO,AX
       RET
READ_SECTOR ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;              +     ОПРЕДЕЛЯЕМ КОЛИЧЕСТВО ФАЙЛОВ В ПАПКЕ     +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
MAS_NUMBER PROC
          ;-------------------------------------------------------------------
          XOR AX,AX
          MOV 310H+ROT,AX
          MOV AL,0
          MOV 310H+ID,AL
          ;-------------------------------------------------------------------
          MOV SI,310H+OFFSET MAS
          CIKL_GOK:
             MOV AL,CS:[SI]
             CMP AL,0
             JE EXIT_MAS_NUMBER
             ;----------------------------------------------------------------
             INC 310H+ROT
             ADD SI,10
             INC 310H+ID
             CALL ADD_2_BAIT
             ;----------------------------------------------------------------
          JMP CIKL_GOK
          EXIT_MAS_NUMBER:RET
MAS_NUMBER ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;        +   ПРОЦЕДУРА КОММАНДНОЙ СТРОКИ НА СОЗДАНИЕ НОВОЙ ПАПКИ   +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
PROVERKA_CREATE_FOLDER PROC
              JMP DATA1
                 CR_NEW_F DB 'NEWF '          ;КОММАНДА НА СОЗДАНИЕ НОВОЙ ПАПКИ
                 FLAG_PROVERKA_CREATE_FOLDER DB 0       ;ФЛАГ ДАННОЙ ПРОЦЕДУРЫ
              DATA1:
              ;----------------------------------------------------------------
              MOV DI,310H+OFFSET CR_NEW_F
              MOV SI,310H+OFFSET BUFER_COM
              MOV CX,5
              CIKL_SRAVN_NEW_F:
                        MOV AL,CS:[SI]
                        CMP AL,CS:[DI]
                        JNE NET_NEW_F
                        ;---------ПРОДОЛЖАЕМ СРАВНИВАТЬ----------------------
                        INC SI
                        INC DI
              LOOP CIKL_SRAVN_NEW_F
              ;----------------------------------------------------------------
              ;------ДА, МЫ СОЗДАЕМ ПАПКУ В УКАЗАННОЙ ДИРЕКТОРИИ--------------
              ;----------------------------------------------------------------
              MOV 310H+FLAG_PROVERKA_CREATE_FOLDER,1
              ;-----------ПЕРВЫЕ 8 БАЙТ БУДУТ НАЗВАНИЕМ ПАПКИ-----------------
              MOV CX,8
              MOV DI,310H+OFFSET NAMER
              CIKL_COPY_NAMER_NEW_F:
                      MOV AL,CS:[SI]
                      MOV CS:[DI],AL
                      INC SI
                      INC DI
              LOOP CIKL_COPY_NAMER_NEW_F
              JMP EXIT_PROVERKA_CREATE_FOLDER
              ;----------------------------------------------------------------
              ;-------МЫ НЕ СОЗДАЕМ ПАПКУ В УКАЗАННОЙ ДИРЕКТОРИИ--------------
              ;----------------------------------------------------------------
              NET_NEW_F:
                       MOV 310H+FLAG_PROVERKA_CREATE_FOLDER,0
                       EXIT_PROVERKA_CREATE_FOLDER:
              ;----------------------------------------------------------------
              ;-----------------ОЧИЩАЕМ БУФЕР COM-----------------------------
              ;----------------------------------------------------------------
              MOV CX,60
              MOV SI,310H+OFFSET BUFER_COM
              MOV 310H+UKBUL,0
              MOV AL,0
```

```
              CIKL_NAM@:
                      MOV CS:[SI],AL
                      INC SI
              LOOP CIKL_NAM@
              ;-----------------------------------------------------------
              ;---------ОЧИЩАЕМ КОММАНДНУЮ СТРОКУ-----------------------------
              ;-----------------------------------------------------------
              MOV CX,60
              MOV DH,24
              MOV DL,8
              MOV SI,310H+OFFSET BUFER_COM
              CIKL_COMM_DEL:
                        PUSH CX
                          MOV AL,0
                          MOV CS:[SI],AL
                          MOV AL,20H
                          MOV AH,7
                          MOV CX,1
                          CALL SIMBOL
                          INC SI
                          INC DL
                        POP CX
              LOOP CIKL_COMM_DEL
              RET
PROVERKA_CREATE_FOLDER ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                  +       ВЫВОД ТРОЕТОЧИЯ НА ЭКРАН       +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
NIL_FILE PROC
      ……
      RET
NIL_FILE ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;            +     ВЫВОД В СТРОКЕ 310H+PIKI УКАЗАТЕЛЯ НА ФАЙЛ     +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
PIKI_PROC PROC
         ……
         RET
PIKI_PROC ENDP
;-------------------------------------------------------------------------------
PIKI_PROC_STIR PROC
         ……
         RET
PIKI_PROC_STIR ENDP
;-------------------------------------------------------------------------------
PIC_UP PROC
    ……
    RET
PIC_UP ENDP
STIR_PIC_UP PROC
    ……
    RET
STIR_PIC_UP ENDP
;-------------------------------------------------------------------------------
PIC_DOWN PROC
    ……
    RET
PIC_DOWN ENDP
STIR_PIC_DOWN PROC
    ……
    RET
STIR_PIC_DOWN ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                       +      ВЫВОД 22 ПАПОК     +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
USER_31_FOLDER PROC
        ……
        RET
USER_31_FOLDER ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                     +       ВЫВОД 21 ПАПКИ      +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
USER_30_FOLDER PROC
      ……
      RET
USER_30_FOLDER ENDP
```

```
;-------------------------------------------------------------------------
;/////////////////////////////////////////////////////////////////////////
;    +    ВХОД 310H+KORE - КОРНЕВАЯ ПАПКА ИЛИ НЕТ  (1-ЗНАЧИТ КОРНЕВАЯ)  +
;/////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------
PAPKA PROC
      CALL MAS_NUMBER  ;=>ROT - ЧИСЛО ФАЙЛОВ В ПАПКЕ
      CMP 310H+ROT,22
      JGE CHISLO_FAILOV_BOLSHE_21
      ;-----------------------------------------------------------------
      ;---------------ПРОВЕРКА НА КОРНЕВУЮ ПАПКУ-------------------------
      ;-----------------------------------------------------------------
      CMP 310H+KORE,1
      JE ES_KOREN_PAPKA@
               CALL NIL_FILE      ;НЕТ, ЭТО НЕ КОРНЕВАЯ ПАПКА, 3 ТОЧКИ СТАВИТЬ
               CALL USER_30_FOLDER
               JMP EXIT_PAPKA
      ES_KOREN_PAPKA@:
      ;-----------------------------------------------------------------
      ;----------ВЫВЕЛИ 22 ПАПКИ В КОРНЕВОМ КАТАЛОГЕ----------------------
      ;-----------------------------------------------------------------
      CALL USER_31_FOLDER
      JMP EXIT_PAPKA
      ;-----------------------------------------------------------------
      ;--------------ЧИСЛО СОЗДАННЫХ ФАЙЛОВ БОЛЬШЕ 21-----------------------
      ;-----------------------------------------------------------------
      CHISLO_FAILOV_BOLSHE_21:
              ;---------------------------------------------------------
              ;-----------ПРОВЕРКА НА КОРНЕВУЮ ПАПКУ------------------------
              ;---------------------------------------------------------
              CMP 310H+KORE,1
              JE ES_KOREN_PAPKA
                ;------------НЕТ, ЭТО НЕ КОРНЕВАЯ ПАПКА-------------------
                CMP 310H+M,0
                JNE NETO
                   CALL NIL_FILE
                   CALL USER_30_FOLDER
                   JMP EXIT_PAPKA
                 NETO:
                   CALL USER_31_FOLDER
                   JMP EXIT_PAPKA
              ES_KOREN_PAPKA:
              ;---------ДА, ЭТО КОРНЕВАЯ ПАПКА-----------------------------
              CALL USER_31_FOLDER
      EXIT_PAPKA:RET
PAPKA ENDP
;-------------------------------------------------------------------------
;/////////////////////////////////////////////////////////////////////////
;                        +    ОЧИСТКА ТЕКУЩЕГО ОКНА   +
;/////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------
STIR_DISPLEI PROC
        MOV AX,0B800H
        MOV ES,AX
        ;---------------------------------------------------------------
        MOV CX,21
        MOV DI,322
        CIKL_VERT:
                 PUSH CX
                 ;-----------------------------------------------------
                 MOV CX,78
                 CIKL_GOR:
                     MOV AH,7
                     MOV AL,0
                     MOV ES:[DI],AX
                     INC SI
                     ADD DI,2
                 LOOP CIKL_GOR
                 ;-----------------------------------------------------
                 POP CX
                 ADD DI,4
        LOOP CIKL_VERT
        RET
STIR_DISPLEI ENDP
;-------------------------------------------------------------------------
;/////////////////////////////////////////////////////////////////////////
;                           +        NAM       +
;/////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------
NAM PROC
    ……
    RET
NAM ENDP
;-------------------------------------------------------------------------
```

```
;////////////////////////////////////////////////////////////////////////////
;                        +          COMD          +
;////////////////////////////////////////////////////////////////////////////
;----------------------------------------------------------------------------
COMD PROC
    ……
    RET
COMD ENDP
;----------------------------------------------------------------------------
;////////////////////////////////////////////////////////////////////////////
;                        +          EDIT        +
;////////////////////////////////////////////////////////////////////////////
;----------------------------------------------------------------------------
EDIT_PROC PROC
         JMP EDIT_DATA
           UK_GOR DB 0                ;ГОРИЗОНТАЛЬНОЕ ПОЛОЖЕНИЕ КУРСОРА
           UK_VER DB 0                ;ВЕРТИКАЛЬНОЕ ПОЛОЖЕНИЕ КУРСОРА
           UK_ED DW 0                 ;ОБЩИЙ СЧЕТЧИК В ПАМЯТИ НА ФАЙЛ
         EDIT_DATA:
         ;-------------------------------------------------------------------
         MOV DH,310H+UK_VER
         MOV DL,310H+UK_GOR
         ADD DH,2
         INC DL
         MOV CX,1
         MOV AL,95
         MOV AH,7
         CALL SIMBOL
         ;-------------------------------------------------------------------
         CIKL_EDIT:
              MOV AH,01H
              INT 16H
              JNZ ES_SIMBOL
                JMP NEAR:NET_SIMBOL
              ES_SIMBOL:
              ;----------ДА, МЫ НАЖАЛИ СИМВОЛ---------------------------------
              MOV AH,0
              MOV AL,0
              INT 16H
              ;---------------ПРОВЕРКА НА ВЫХОД-------------------------------
              CMP AL,27
              JNE NET_ESC
                JMP EXIT_EDIT
              NET_ESC:
              ;---------------ПРОВЕРКА НА BS----------------------------------
              CMP AL,8
              JNE NET_BS
                 ;-------------ПРОВЕРКА НА ЦЕЛОСТНОСТЬ-----------------------
                 CMP 310H+UK_ED,0
                 JNE ES1
                   JMP NEAR:CIKL_EDIT
                 ES1:
                 ;-----------------------------------------------------------
                 MOV CX,310H+UK_ED
                 DEC CX
                 MOV 310H+UK_ED,CX
                 LEA DI,310H+BUFER_EDIT
                 ADD DI,CX
                 XOR AL,AL
                 MOV 310H+[DI],AL
                 ;-----------------------------------------------------------
                 CMP 310H+UK_GOR,0
                 JNE NET_GOR1
                    MOV 310H+UK_GOR,78
                    DEC 310H+UK_VER
                 NET_GOR1:
                 DEC 310H+UK_GOR
                 JMP CURSOR
              NET_BS:
              ;---------------ПРОВЕРКА НА ЛЕВО--------------------------------
              CMP AH,4BH
              JNE NET_LEV
                 ;-------------ПРОВЕРКА НА ЦЕЛОСТНОСТЬ-----------------------
                 CMP 310H+UK_GOR,0
                 JNE ES2
                   JMP NEAR:CIKL_EDIT
                 ES2:
                 ;-----------------------------------------------------------
                 DEC 310H+UK_ED
                 DEC 310H+UK_GOR
                 JMP CURSOR
              NET_LEV:
              ;---------------ПРОВЕРКА НА ПРАВО-------------------------------
              CMP AH,4DH
```

```
              JNE NET_PRAV
                 ;-------------ПРОВЕРКА НА ЦЕЛОСТНОСТЬ-----------------------
                 CMP 310H+UK_GOR,77
                 JNE ES3
                   JMP NEAR:CIKL_EDIT
                 ES3:
                 ;-------------------------------------------------------------
                 INC 310H+UK_ED
                 INC 310H+UK_GOR
                 JMP CURSOR
              NET_PRAV:
              ;---------------ПРОВЕРКА НА BNIZ-------------------------------
              CMP AH,50H
              JNE NET_BNIZ
                 ;-------------ПРОВЕРКА НА ЦЕЛОСТНОСТЬ-----------------------
                 CMP 310H+UK_VER,20
                 JNE ES4
                   JMP NEAR:CIKL_EDIT
                 ES4:
                 ;-------------------------------------------------------------
                 ADD 310H+UK_ED,78
                 INC 310H+UK_VER
                 JMP CURSOR
              NET_BNIZ:
              ;---------------ПРОВЕРКА НА BBERH------------------------------
              CMP AH,48H
              JNE NET_BBERH
                 ;-------------ПРОВЕРКА НА ЦЕЛОСТНОСТЬ-----------------------
                 CMP 310H+UK_VER,0
                 JNE ES5
                   JMP NEAR:CIKL_EDIT
                 ES5:
                 ;-------------------------------------------------------------
                 SUB 310H+UK_ED,78
                 DEC 310H+UK_VER
                 JMP CURSOR
              NET_BBERH:
              ;-------------ПРОПИСЫВАЕМ СИМВОЛ В ПАМЯТЬ-----------------------
              ;-------------ПРОВЕРКА НА ЦЕЛОСТНОСТЬ---------------------------
              CMP 310H+UK_ED,1637
              JNE ES6
                   JMP NEAR:CIKL_EDIT
              ES6:
              ;-------------------------------------------------------------
              MOV CX,310H+UK_ED
              LEA DI,310H+BUFER_EDIT
              ADD DI,CX
              MOV 310H+[DI],AL
              ;-------------
              INC 310H+UK_GOR
              CMP 310H+UK_GOR,78
              JNE NET_GOR
                MOV 310H+UK_GOR,0
                INC 310H+UK_VER
              NET_GOR:
              ;-----------
              MOV CX,310H+UK_ED
              INC CX
              MOV 310H+UK_ED,CX
              ;-------------------------------------------------------------
              CURSOR:
                     CALL DISPLEI
                     MOV DH,310H+UK_VER
                     MOV DL,310H+UK_GOR
                     ADD DH,2
                     INC DL
                     MOV CX,1
                     MOV AL,95
                     MOV AH,7
                     CALL SIMBOL
              ;-------------------------------------------------------------
              NET_SIMBOL:
         JMP CIKL_EDIT
         EXIT_EDIT:
         RET
EDIT_PROC ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                      +         DISPLEI         +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
DISPLEI PROC
        ;-------------------------------------------------------------
        MOV AX,0B800H
```

```
        MOV ES,AX
        ;-------------------------------------------------------------------
        MOV SI,310H+OFFSET BUFER_EDIT
        MOV CX,21
        MOV DI,322
        CIKL_VERT@:
                PUSH CX
                ;-----------------------------------------------------------
                MOV CX,78
                CIKL_GOR@:
                    MOV AH,7
                    MOV AL,CS:[SI]
                    MOV ES:[DI],AX
                    INC SI
                    ADD DI,2
                LOOP CIKL_GOR@
                ;-----------------------------------------------------------
                POP CX
                ADD DI,4
        LOOP CIKL_VERT@
        RET
DISPLEI ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                     +              SIMBOL           +
;НА ВХОД:
;        DH - СТРОКА
;        DL - СТОЛБЕЦ
;        CX - ЧИСЛО ПОВТОРЕНИЙ СИМВОЛА
;        AL - КОД СИМВОЛА
;        AH - ЦВЕТ
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
SIMBOL PROC
       PUSH AX
        MOV AX,0B800H
        MOV ES,AX
       POP AX
       ;-------------------------------------------------------------------
       PUSH AX
        MOV AL,160
        MUL DH
        MOV BX,AX
        MOV AL,2
        MUL DL
        ADD BX,AX
        MOV DI,BX
       POP AX
       ;-------------------------------------------------------------------
       CMP SIMBOL_FLAG,1
       JNE NET_SIMBOL_FLAG
       CIKL1:
             MOV ES:[DI],AX
             ADD DI,2
       LOOP CIKL1
       JMP EXIT_SIMBOL
       NET_SIMBOL_FLAG:
       ;-------------------------------------------------------------------
       CIKL2:
             MOV ES:[DI],AX
             ADD DI,160
       LOOP CIKL2
       ;-------------------------------------------------------------------
       EXIT_SIMBOL:RET
SIMBOL ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                            +        RAMA       +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
RAMA PROC
     ……
     RET
RAMA ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                            +        TIMER      +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
TIMER PROC
        MOV AL,00H
        MOV 310H+DL1,78
        CALL TIME    ;СЕКУНДА
        ;--------
```

```
        MOV AL,02H   ;МИНУТА
        MOV 310H+DL1,75
        CALL TIME
        ;--------
        MOV 310H+DL1,72
        MOV AL,04H   ;ЧАС
        CALL TIME
        RET
TIMER ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                            +        TIMER       +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
TIME PROC
       OUT 70H,AL
       IN AL,71H
       ;-----------------------ВЫВОДИМ РАЗРЯД ДЕСЯТОК-------------------------
       PUSH AX
         SHR AL,4
         ADD AL,30H
         ;---------
         MOV DH,24
         MOV DL,310H+DL1
         MOV AH,7
         MOV CX,1
         CALL SIMBOL
       POP AX
       ;-----------------------ВЫВОДИМ РАЗРЯД ЕДИНИЦ--------------------------
         SHL AL,4
         SHR AL,4
         ADD AL,30H
         ;---------
         MOV DH,24
         MOV DL,310H+DL1
         INC DL
         MOV AH,7
         MOV CX,1
         CALL SIMBOL
     RET
TIME ENDP
;-------------------------------------------------------------------------------
;///////////////////////////////////////////////////////////////////////////////
;                            +        DVET       +
;///////////////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------------------
DVET PROC
     ……
     RET
DVET ENDP
;-------------------------------------------------------------------------------
END START
```

ТЕСТИРОВАНИЕ ПРОГРАММЫ

Тестирование данной программы прошло успешно: программа загрузилась с дискеты в память и показала свою работоспособность - можно было создавать каталоги - входить в них и выходить обратно.

Создадим на диске несколько папок:

```
1----------->
            3------->4--------->5
            7                   6
                                7
2
12345678
DOOM3------->set1
DOOM4
```

Для проверки работоспособности файловой системы перезагрузим компьютер. При загрузке мы видим тот же набор файлов, который был до перезагрузки, т.о. файловая системы работает.