“Basics of microcontroller”

Lecture 11.

Selecting addresses in memory and organizing microcomputer memory

Lecturer: Onon Otgonbaatar, MD.

ӨРГӨТГӨСӨН DOS-ЫН ФУНКЦҮҮДИЙГ АШИГЛАН ФАЙЛТАЙ АЖИЛЛАХ

ASCIIZ тэмдэгт

Өргөтгөсөн DOS-ын олон функцүүд ASCIIZ тэмдэгтүүдийг ашигладаг. ASCIIZ тэмдэгтэд файлыг агуулж буй дискний төхөөрөмж, католог, файлын нэр, өргөтгөл, 0 утга гэх мэт файлын байрлал агуулагдана. Жишээ нь:

```
PATH1    DB   'B:\TEST.ASM', 0
PATH2    DB   'C:\TASM\EXAMPLE.EXE', 0
```

0 утга нь ASCIIZ тэмдэгтүүдийн төгсгөлийг заана.
LEA DX, PATH1 гэх мэтээр DX регистрт ASCIIZ тэмдэгтийн хаягийг ачаалж, тасалдлуудад хэрэглэнэ.

Файлын мэдээ ба алдааны кодууд

Файлд хандахдаа (файлын нээх болон үүсгэх) AX регистрт байгаа 1 word утгаар илэрхийлэгдэх файлд хандах мэдээгээр хандана. Зарим стандарт төхөөрөмжүүдэд хандахдаа тэднийг нээхгүйгээр шууд хандаж болно.

0 – оролт, голдуу гарнаас (CON)
1 – гаралт, голдуу дэлгэц (CON)
2 – алдааны гаралт, голдуу дэлгэц (CON)
3 – туслах төхөөрөмж (AUX)
4 – принтер (LPT1 ба PRN)

Дискэнд хандахдаа болон файл үүсгэх, файл нээх зэрэгт ASCIIZ тэмдэгтүүдийг хэрэглэх ба DOS-ийн 3CH, 3DH зэрэг функцүүдийг ашиглана. Амжилттай хйигдвэл, CF=0, AX регистрт утга буцаана. Амжилтгүй хйигдвэл, CF=1, AX регистрт алдааны кодыг буцаана.
Алдааны кодууд:

01 – буруу функц
02 – файл олдоогүй
03 – зам нь олдоогүй
04 – хэтэрхий олон файл нээлттэй
05 – хандалтыг татгалзсан
06 – буруу хандалт
07 – санах ойн удирдлагын блок (MCB) устгагдсан
08 – санах ой хүрэлцээгүй
09 – буруу санах ойн блокийн хаяг
10 – буруу орчин
11 – буруу формат
12 – буруу хандалтын код
13 – буруу өгөгдөл
15 – буруу драйвер заагдсан
16 – алсын католог уруу хандсан
17 – төхөөрөмж өөрчлөгдсөн
18 - файл байхгүй.

Файлыг үүсгэх: 3CH

DX регистрт ASCIIZ тэмдэгтийг ачаалан, CX регистрт шаардагдах атрибутыг хийнэ. Өмнө бид атрибут байтыг үзсэн. Энгийн функцын хувьд атрибут нь 0 байна.

```
MOV     AH, 3CH             ; файл үүсгэх хүсэлт
MOV     CX, 00              ; энгийн файл
LEA     DX, PATHNM1         ; ASCIIZ тэмдэгт
INT     21H                 ; DOS-ыг дуудах
JC      ERROR               ; алдаа гарвал программаас гарна.
MOV     HANDLE1, AX         ; файлын тухай мэдээг хадгална.
```

Уг үйлдлийн үр дүнд алдаа гарвал AX регистрт алдааны код, алдаа гараагүй бол AX регистрт файлын тухай мэдээг хадгална.

Файлд бичих: 40H

Файлын мэдээг BX регистрт, бичих байтуудын тоог CX регистрт, DX регистрт бичих хаягийг хийнэ. Доох жишээнд OUTREC –ээс 256 байт бичлэгийг бичнэ.

```
HANDLE  DW  ?
OUTREC  DB  256 DUP(' ')
        MOV     AH, 40H             ; файлд бичих хүсэлт
        MOV     BX, HANDLE1         ; файлын мэдээ
        MOV     CX, 256             ; бичлэгийн урт
        LEA     DX, OUTREC          ; гаргах талбарын хаяг
        INT     21H                 ; DOS-ыг дуудах
        JC      ERROR               ; алдаа гарвал ERROR уруу
        CMP     AX, 256             ; бүх байтууд бичигдсэн үү?
        JNE     ERROR               ; үгүй бол алдаа
```

Зөв хийгдвэл санах ойгоос бичлэгийг бичээд, CF-ийг цэвэрлэн, AX регистрт бичигдсэн байтуудын тоог хийнэ.
Уг үйлдлийн үр дүнд алдаа гарвал AX регистрт алдааны код, CF –д 1-ийг хадгална.

Файлыг хаах: 3EH

Файлд бичээд дууссан бол файлыг доорх функцээр хааж болох ба санах ойд үлдсэн бичлэгүүдийг хуулаад, FAT болон санг шинэчилнэ.

```
        MOV     AH, 3EH             ; файлыг хаах хүсэлт
        MOV     BX, HANDLE1         ; файлын мэдээ
        INT     21H                 ; DOS-ыг дуудах
```

Гарч болох цорын ганц алдаа нь AX регистрт 06 (invalid handle) юм.

Файлд хандах мэдээг ашиглан файлыг үүсгэх жишээ программ

Гарнаас оруулсан нэрсээр файл үүсгэнэ. Үндсэн процедуруудыг талбарлавал:
CREA 3CH функцыг ашиглан, файлыг үүсгэх ба HANDLE гэж нэрлэгдсэн хувьсагчид файлд хандах мэдээг хадгална.
ACCS Гарнаас мэдээг хүлээж авах ба нэрний төгсгөлийн хэсгээс оролтын талбарын төгсгөл хүртэлх хэсгийг хоосолно.
WRIT 40h функцыг ашиглан бичлэгүүдийг файлд бичнэ.
CLSE 3EH функцыг ашиглан программын төгсгөлд сангийн өгөгдлүүдийг үүсгэхийн тулд файлыг хаана.
Оролтын талбар нь 30 байт бөгөөд carriage return(0DH), line feed(0AH) гэсэн мөрний төгсгөл, дараагийн мөрний эхлэлд аваачих 2 тэмдэгтийн хамт нийт 32 байт байна. Иймд уг программ 32 байтын тогтмол урттай бичлэгүүдийг файлд бичнэ.

```
PAGE    60, 132
TITLE   HANDLE  Create disk file of names
.model small
.stack 100h
.data
        NAMEPAR         LABEL   BYTE        ; нэрний параметр лист
        MAXLEN          DB      30
        NAMLEN          DB      ?
        NAMREC          DB      30    DUP(' '), 0DH, 0AH     ; орсон нэр ба шинэ мөрөнд
                                              эхлүүлэх CR / LF

        ERRCDE          DB      00
        HANDLE          DW      ?
        PATHNAM         DB      'C:\NAMEFILE.DAT', 0
        PROMPT          DB      'NAME? '
        ROW             DB      01
        OPNMSG          DB      '---OPEN ERROR---', 0DH, 0AH
        WRTMSG          DB      '---WRITE ERROR---', 0DH, 0AH
;--------------------------------
.code
BEGIN   PROC    FAR
        MOV     AX, @DATA
        MOV     DS, AX
        MOV     ES, AX

        CALL    CLRSCR          ; дэлгэц цэвэрлэх
        CALL    SETCUR          ; курсорыг байрлуулах
        CALL    CREA            ; файлыг үүсгэх ба DTA-г тавих
        CMP     ERRCDE, 00      ; алдаагүй юу?
        JZ      ALOOP           ; тийм бол үргэлжлүүл
        RET                     ; алдаатай бол программаас гарна.
ALOOP:
        CALL    ACCS            ; оролт авна.
        CMP     NAMLEN, 00      ; оролтын төгсгөл үү?
        JNE     ALOOP           ; үгүй бол үргэлжлүүл
        CALL    CLSE            ; тийм бол дуусга.

        MOV     AX, 4C00H       ; программаас гарах
        INT     21H
        RET
BEGIN   ENDP
;-----------Дискэнд файл үүсгэх----------------
CREA    PROC    NEAR
        MOV     AH, 3CH         ; файл үүсгэх хүсэлт
        MOV     CX, 00          ; энгийн файл
        LEA     DX, PATHNAM
        INT     21H
        JC      C20             ; алдаа гарсан уу?
        MOV     HANDLE, AX      ; үгүй бол мэдээг хадгална.
        RET
C20:
        LEA     DX, OPNMSG      ; алдааны мэдээг хэвлэх
```

```
        CALL    ERROR
        RET
CREA    ENDP
;--------Оролтыг гарнаас хүлээж авах -----------
ACCS    PROC    NEAR
        MOV     AH, 40H         ; дэлгэцэнд хэвлэх хүсэлт
        MOV     BX, 01          ; файлд хандах мэдээ
        MOV     CX, 06          ; промптын урт
        LEA     DX, PROMPT      ; хэвлэх промпт
        INT     21H

        MOV     AH, 0AH         ; оруулах хүсэлт
        LEA     DX, NAMEPAR     ; нэрийг хүлээж авах хэсэг
        INT     21H
        CMP     NAMLEN, 00      ; нэр орсон уу?
        JNE     D20             ; тийм бол D20 уруу үсэрнэ.
        RET                     ; үгүй бол гарна.
D20:
        MOV     AL, 20H         ; хоосон тэмдэгтийг хадгална.
        SUB     CH, CH
        MOV     CL, NAMLEN      ; нэрийн урт
        LEA     DI, NAMREC
        ADD     DI, CX          ; хаяг+урт
        NEG     CX              ; үлдсэн зайг
        ADD     CX, 30          ;тооцох
        REP     STOSB           ; хоосон тэмдэгтийг хуулна.
D90:
        CALL    WRIT            ; дискний файлд бичнэ.
        CALL    SCRL            ; дэлгэц дүүрсэн бол хуудсыг дээш
                                ;       гүйлгэнэ.
        RET
ACCS    ENDP
;----------Дэлгэцийг шалгах----------
SCRL    PROC    NEAR
        CMP     ROW, 18         ; дэлгэцийн төгсгөл мөн үү?
        JAE     E10             ; тийм бол гүйлгэнэ.
        INC     ROW             ; үгүй бол нэг төр нэмнэ.
        JMP     E90
E10:
        MOV     AX, 0601H       ; нэг мөрөөр дээш ахиулах
        CALL    CLRSCR          ; дэлгэц цэвэрлэнэ.
E90:
         CALL   SETCUR          ; курсорыг байрлуулна.
         RET
SCRL    ENDP
;-------------Файлд бичлэгийг бичих-------------
WRIT    PROC    NEAR
        MOV     AH, 40H         ; бичих хүсэлт
        MOV     BX, HANDLE
        MOV     CX, 32          ; нэр 30 тэмдэгт +CR /LF 2 тэмдэгт
        LEA     DX, NAMREC
        INT     21H
        JNC     DX, WRTMSG      ; үгүй бол алдааны
```

```
        CALL    ERROR           ; мэдээг хэвлэнэ.
        MOV     NAMLEN, 00
F20:
        RET
WRIT    ENDP
;-------------Файлыг хаах------------
CLSE    PROC    NAER
        MOV     NAMREC, 1AH  ; файлын төгсгөлийн тэмдэг
        CALL    WRIT
        MOV     AH, 3EH         ; хаах хүсэлт
        MOV     BX, HANDLE
        INT     21H
        RET
CLSE    ENDP
;-----------------Дэлгэц цэвэрлэх---------------
CLRSCR  PROC    NEAR    ; AX регистрт утга ирсэн
        MOV     BH, 1EH         ; хөх дээр шар
        MOV     CX, 0000
        MOV     DX, 184FH
        INT     10H             ; дэлгэц цэвэрлэнэ.
        RET
CLRSCR  ENDP
;----------------Курсорыг байрлуулах------------
SETCUR  PROC    NEAR
        MOV     AH, 02
        MOV     BH, 00
        MOV     DH, ROW         ; курсор тавина.
        MOV     DL, 00
        INT     10H
        RET
SETCUR  ENDP
;------------------Алдааны процедур--------------
ERROR   PROC    NEAR
        ; DX регистрт алдааны мэдээний хаяг байна.
        MOV     AH, 40H
        MOV     BX, 01
        MOV     CX, 21          ; урт
        INT     21H
        MOV     ERRCDE, 01      ; алдааны кодыг тавина.
        RET
ERROR   ENDP
        END BEGIN
```

Файлыг нээх: 3DH

Программаар файлаас унших гэж байгаа бол эхлээд файлыг 3DH функцээр нээх хэрэгтэй. Энэ үйлдэл нь файлын нэрийг шалган, файлыг олно. Дараа нь DX регистрт ASCIIZ тэмдэгтийг нь ачаалан, AL регистрт хандах кодыг нь тавина. AL=0 файлд зөвхөн өгөгдөл оруулахын тулд нээнэ.
AL=1 файлаас зөвхөн өгөгдөл уншихын тулд нээнэ.
AL=2 файлд өгөгдөл оруулах ба гаргахын тулд нээнэ.

Бусад битүүдийнх нь зориулалт нь файлыг хамтран эзэмшихэд DOS 3.0 үед ашиглана. Файлд бичихдээ 3CH буюу үүсгэх функцыг ашиглана. Файлыг уншихын тулд нээхэд хэрэглэх жишээ:

```
MOV     AH, 3DH          ; нээх хүсэлт
MOV     AL, 00           ; зөвхөн уншихаар
LEA     DX, PATHNM1      ; ASCIIZ string
INT     21H              ; DOS-ыг дуудах
JC      ERROR            ; алдаа гарвал ERROR уруу
MOV     HANDLE1, AX      ; мэдээг хадгалах
```

Хэрэв өгөгдсөн нэртэй файл олдвол бичлэгийн хэмжээг 1 гэж тавин, CF битийг цэвэрлэн, AX регистрт файлын мэдээг тавина.
Хэрэв файл олдоогүй бол CF=1 болох ба AX регистрт алдааны кодыг тавина. Эхлээд CF төлөвийн битийг шалгах хэрэгтэй, тэгэхгүй бол AX регистрт 0005 гэсэн файлын мэдээ нь алдааны 05 гэсэн кодтой андуурагдах боломжтой.

Файлыг унших: 3FH

Бичлэгийг уншихдаа 3FH функцыг ашиглана. BX регистрт файлын мэдээг, CX регистрт унших байтынхаа тоог, DX регистрт оролтын талбарын хаягийг тавина. Дараах байдлаар 512 байт бичлэгийг уншиж болно.

```
HANDLE  DW  ?
INREC   DB  512  DUP(‘ ’)
        MOV     AH, 3FH          ; унших хүсэлт
        MOV     BX, HANDLE1
        MOV     CX, 512          ; бичлэгийн урт
        LEA     DX, INPREC       ; оролтын талбарын хаяг
        INT     21H              ; DOS-ыг дуудах
        CMP     AX, 00           ; алдааг шалгах
        JE      ENDFILE
```

Уг үйлдэл нь санах ойд бичлэгийг уншиж аваад, CF-ийг цэвэрлэн, AX регистрт уншигдсан байтуудын тоог тавина. AX регистрт 0 гэсэн утга байвал файлын төгсгөлөөс унших гэж оролдсон гэсэн үг. Уншилтын алдаа нь CF-ийг тавих ба AX регистрт алдааны код 05(хандалтыг татгалзсан) эсвэл 06(буруу хандалт)-г буцаана.

Бусад нэмэлт функцүүд

- Дискний сул зайг авах: 36h
  Энэ функц нь дискний төхөөрөмж дээрх зайны талаарх мэдээг илгээнэ. Төхөөрөмжийг дугаарыг DL регистрт тавиад функцээ дуудна.
  0-идэвхитэй байгаа төхөөрөмж, 1-A, 2-B, г.м

  ```
  MOV     AH, 36H
  MOV     DL, 0
  INT     21H
  ```

  Дискний төхөөрөмжийг дугаарыг буруу өгсөн бол AX регистрт FFFFH гэсэн утга буцаах ба зөв хийгдсэн бол доорх утгууд буцаагдана.
  AX регистрт Нэг кластер дахь секторуудын тоо
  BX регистрт Хандах боломжтой кластеруудын дугаар
  CX регистрт Нэг сектор дахь байтуудын тоо
  DX регистрт Уг төхөөрөмж дахь кластеруудын нийт тоо
  DOS 2.0-оос өмнөх хувилбаруудад 1BH функцыг ашиглан, FAT-ын мэдээг авч болно.

- Файлыг устгах: 41h
  Зөвхөн унших боломжтой файлаас бусад файлуудыг энэ функцээр устгаж болно. Файлын нэр болон уг файлын байрлаж байгаа дискний төхөөрөмж дээрх замыг агуулсан ASCIIZ тэмдэгтийн хаягийг зааж өгөх хэрэгтэй.

```
MOV     AH, 41H          ; устгах хүсэлт
LEA     DL, PATHNAME     ; ASCIIZ тэмдэгт
INT     21H              ; DOS–ыг дуудна.
```

  AX регистрт алдааны кодыг буцаах ба 02 байвал файл олдоогүй, 05 бол хандалтыг татгалзсан гэсэн үг.

- Файлын заагчийг шилжүүлэх: 42h
  DOS нь файлын заагчийг үүсгэх ба анхны утгыг нь 0 гэж тавиад, дарааллуулан бичлэгүүдийг унших буюу бичих бүрт нэмэгдүүлнэ. Энэ файлын заагчийг шилжүүлэх функцыг файл доторх дурын байрлалд байгаа бичлэгийг унших буюу бичихэд хэрэглэнэ.
  BX регистрт файлд хандах мэдээг тавиад, CX:DX регистрийн хосд шаардлагатай шилжлэг хаягийг тавина. 65.535-аас дээш байтын зайд шилжих тохиолдолд CX регистрт 0, DX регистрт утга нь өгнө. Мөн, AL регистрт шилжлэг хаягийг авах аргын дугаарыг өгөх хэрэгтэй. Үүнд:

  0 0-р арга: Шилжлэг хаягийг файлын эхлэлээс авна.
  1 1-р арга: Шилжлэг хаягийг файлын заагчийн тухайн байрлалаас авна.
  2 2-р арга: Шилжлэг хаягийг файлын төгсгөлөөс авна. Файлын хэмжээг CX:DX регистрийн хосд 0 утга олгон, 2-р аргыг хэрэглэн авч болно.

  Доорх жишээгээр файлын эхлэлээс 1024 байт заагчийг шилжүүлнэ.

```
MOV     AH, 42H          ; заагчийг шилжүүлэх хүсэлт
MOV     AL, 0            ; файлын эхлэлээс
LEA     BX, HANDLE1      ; файлд хандах мэдээг тавина.
MOV     CX, 0
MOV     DX, 1024         ; 1024 байт шилжлэг хаяг
INT     21H              ; DOS-ыг дуудна.
JC      ERROR
```

  Үйлдэл алдаагүй хийгдвэл CF-ийг цэвэрлэх ба DX:AX регистрийн хосд шинэ заагчийн байрлалыг хийнэ.
  Алдаатай хийгдсэн бол CF-ийн тавих ба AX регистрт 01(буруу аргын дугаар), 16(буруу хандалт) гэх мэт алдааны кодыг буцаана.

- Атрибутыг шалгах буюу өөрчлөх: 43h
  Сан доторх файлын атрибутыг шалгах болон өөрчлөхөд энэ функцыг хэрэглэнэ. DX регистрт ASCIIZ тэмдэгтийн хаягийг олгоод, AL регистрт файлын атрибутыг шалгах гэж байгаа бол 00 гэсэн утга өгнө. Хэрэв өөрчлөх гэж байгаа бол AL регистрт 01, Cx регистрт шинэ атрибутыг өгнө. Доорх байдлаар энгийн атрибутыг тавьж болно.

```
MOV     AH, 43H          ; атрибут тавих хүсэлт
MOV     AL, 01           ; энгийн
MOV     CX, 00           ;атрибут
LEA     DX, PATHNAME     ; ASCIIZ тэмдэгт
INT     21H              ; DOS-ыг дуудна.
```

  Атрибутыг шалгахад CX регистрт идэвхитэй байгаа атрибут буцна. Өөрчлөх үед CX регистрт байгаа атрибутад сангийн өгөгдлийг хийнэ. Алдаатай үйлдэл хийгдвэл AX регистрт 02, 03, 05 гэх мэт алдааны кодуудын нэгийг буцаана.

- Идэвхитэй байгаа санг унших: 47h
  Энэ функцээр дискний төхөөрөмжийн одоо идэвхитэй байгаа санг уншина. Хамгийн урт замын өгөгдлийг хадгалахад тохиромжтой хэмжээгээр зайг нөөцлөөд, DX

регистрт үүний нэрийг ачаална. DL регистрт дискний драйвийн нэрийг өгнө. 1=A, 2=B гэх мэт. Уг функц нь дискний драйвын нэрийг агуулахгүйгээр, замыг нь буцаана. Жишээ нь:
ASSEMBLE\EXAMPLES0
Сүүлийн 0 гэсэн утга нь замын нэрийн төгсгөлийг заана. Хэрэв хүссэн сан нь boot байх юм бол буцаагдах замын нэр нь 00 байна.
Энэ функцээр дурын файлд хандахдаа замыг нь авч болно.

- Тохирох файлыг олох: 4EH, 4FH
  Энэ функцүүд нь үндсэн DOS-ын 11h, 12h функцүүдтэй ижил үүрэгтэй. 4EH функцыг сан дотор хайлт эхлэхдээ, 4FH функцыг хайлтыг үргэлжлүүлэхдээ хэрэглэнэ. Хайлт эхлэхдээ DX регистрт замыг агуулж буй ASCIIZ тэмдэгтийн хаягийг өгнө. (?, * тэмдэгүүдийг агуулсан байж болно) CX регистрт файлын атрибутыг олгоно. Энгийн, сан, нууцлагдсан, системийн гэх мэт битүүдийн нэгдлээр атрибутыг тодорхойлно.

```
MOV     AH, 4EH             ; эхний хайлт хийх хүсэлт
MOV     CX, 00              ; энгийн атрибут
LEA     DX, PATHNAME        ; ASCIIZ тэмдэгт
INT     21H                 ; DOS-ыг дуудна
```

  Энэ үйлдэл нь тохирох файлыг олоод, FCB дахь одоогийн DTA-д доорх утгуудыг бичнэ.

```
00      DOS-д нөөцлөгдсөн.
21      Файлын атрибут
22      Файлын хугацаа
24      Файлын огноо
26      Файлын хэмжээ: бага wordь дараа нь ахлах word
30      Нэр ба өргөтгөл: 13 байт ASCIIZ тэмдэгт, ардаа 00-тэй
```

  AX регистрт буцаасан алдааны код нь 02(файл олдоогүй), 18(файл байхгүй) байж болно. 4FH –г ашиглан цааш хайхдаа DTA-г хэвээр нь үлдээнэ.

```
MOV     AH, 4FH     ; дараагийн хайлт
INT     21H         ; DOS-ыг дуудна
```

  Алдаа гарсан тохиолдолд 18(файл байхгүй) гэсэн код буцна. CF-ийг өөрчлөхгүй.

- Файлын нэрийг өөрчлөх: 56h
  Программаар файлын нэрийг өөрчлөхдөө энэ функцыг хэрэглэнэ. DX регистрт хуучин диск драйв, зам, файлын нэр зэргийг агуулсан ASCIIZ тэмдэгтийн хаягийг өгнө. ES:DI регистрт хосоор шинэ диск драйв, зам, файлын нэр зэргийг агуулсан ASCIIZ тэмдэгтийн хаягийг өгнө. Дискний драйвын дугаар ижил байх хэрэгтэй, харин сан, зам, нэр өөр байж болно. Уг үйлдэл нь файлын нэрийг өөрчлөн, өөр санд шилжүүлнэ.

```
MOV     AH, 56H             ; файлын нэрийг өөрчлөх хүсэлт
LEA     DX, хуучин_утга     ; DS:DX
LEA     DI, шинэ_утга       ; ES:DI
INT     21H                 ; DOS-ыг дуудна
```

  Алдаа гарвал AX регистрт 03(зам олдоогүй), 05(хандалт татгалзагдсан), 17(дискний төхөөрөмж өөрчлөгдсөн) гэсэн алдааны кодуудын нэг буцаагдана.

Бусад файлтай хамааралтай DOS-ын функцүүдийг дурдвал:

```
39H     Сан үүсгэх
3AH     Сангийн мэдээг устгах
3BH     Одоогийн санг өөрчлөх
44H     Дискний төхөөрөмжийн оролт гаралтыг удирдах
45H, 46H  Файлд хандах мэдээг хуулбарлах
54H     Шалгах төлөвийг унших
```

Файлыг үүсгэх ба нээх функцүүд нь файлтай цааш хийх үйлдэлд хэрэглэх файлын хандалтын мэдээг буцаадаг.
Файлд бичилт хийхдээ файлыг үүсгэх функцыг, файлаас уншилт хийхдээ файлыг нээх функцыг ашиглах дүрэмтэй. Санд өөрчлөлтийг оруулахын тулд бичилт хийсний дараа файлыг заавал хаах хэрэгтэй. Файлтай хийх үйлдлүүдийн алдааны ихэнх тохиолдолд CF төлөвийн бит тавигдах (1 болно) ба AX регистрт алдааны кодыг оруулдаг.
.

Textbook

1. Peter Abel, “IBM PC Assembler language and programming”, USA, 1987
2. Jim Mischel, "Macro magic with Turbo Assember ”, USA, 1993
3. William C.Runnion, “Structured programming in Assembly Language for the IBM PC”, Boston, 1988
4. Thomas A.Wadlow, “Memory resident programming on the IBM PC”, USA, 1987
5. Robert S.Lai, “Writing MS-DOS device drivers”, USA, 1987
6. E. Majigsuren, “IBM assembly language”, MGL, 2003
7. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay, “The 8051 microcontroller and Embedded Systems Using Assembly and C”,USA, 2007