

## **“Basics of microcontroller”**

*Lecture 10.*

*Memory, memory matrix, operating principle, RAM,  
ROM control signals*

**Lecturer: Onon Otgonbaatar, MD.**

## ҮНДСЭН DOS-ЫН ФУНКЦҮҮДИЙГ АШИГЛАН ФАЙЛТАЙ АЖИЛЛАХ

Дисктэй ажиллах нь файл ба бичлэгүүдийг тодорхойлох FCB (file control block) буюу файл удирдах блоктой холбоотой юм. Үүний хаягийг дисктэй ажиллах бүх үйлдлүүдэд хангаж өгөх ёстой. Файл бичих нь файлыг үүсгэх бөгөөд DOS нь санд өөрчлөлт оруулна. Бүх бичлэгүүд орсны дараа программ нь файлыг хаах ёстой бөгөөд ингэсний дараа DOS сантай ажиллаж дуусна. Файлыг уншихдаа файлын байгаа үгүйг шалгаж нээнэ.

Дискэнд хандахдаа DOS INT 21H тасалдлыг ашиглана. Дараагийн нам түвшний хандалт нь DOS INT 25H, 26H тасалдлууд байх ба дискний секторуудтай хамааралтай абсолют хаяглалтыг хэрэглэнэ. Хамгийн доод түвшний хандалт нь BIOS INT 13H тасалдал бөгөөд энэ нь зам ба секторын дугаараар ханддаг.

### FCB - Файл удирдах блок

Дисктэй ажиллахдаа программист өгөгдлийн талбарт FCB-ийг тодорхойлно. FCB нь файлын замыг агуулахгүй, тухайн сан дотроос файлтай ажиллана. FCB нь файл ба түүний бичлэгүүдийн талаарх тодорхойлсон доорх мэдээллүүдийг агуулна. Программист 0-15, 32-36 дугаар байтуудын утгыг тавих ба 16-31 дугаар байтуудын утгыг DOS тавина.

Байт	Зориулалт
0	Дискний төхөөрөмж 01-A төхөөрөмж, 02-B төхөөрөмж гэх мэт
1-8	Файлын нэр Файлын нэр, мөн нөөц нэр байж болно. Жишээлбэл: LPT1 гэвэл принтер
9-11	Файлын өргөтгөл Жишээ нь: ASM, DTA гэх мэт. DOS нь файлын нэр ба өргөтгөлийг санд хадгалдаг.
12-13	Одоогийн блокийн дугаар. Блок нь 128 бичлэгээс бүрдэнэ. Унших, бичих үйлдлүүд нь тухайн блокийн дугаар ба 32 дугаар байт дахь бичлэгийн дугаар зэргийг ашиглан бичлэгийг олохдоо ашиглана. Блокийн дугаар 0 эхлэнэ.
14-15	Бичлэгийн логик хэмжээ Файлыг дөнгөж нээхэд энэ талбар нь 128 буюу 80H гэсэн утгыг агуулна. Унших бичих үйлдлүүдийн явцад бичлэгийн уртыг программист өөрөө тодорхойлж өгнө.
16-19	Файлын хэмжээ (бичлэгийн хэмжээ*бичлэгийн тоо) Файлыг үүсгэхэд DOS түүний хэмжээг байтаар тодорхойлдог бөгөөд энэ утгаа катологт мөн бичнэ. Файлыг нээх үед каталог дотроос файлын хэмжээг уншиж аваад энэ талбарыг бөглөнө.
20-21	Огноо DOS нь файлыг хаах үедээ огноог өөрчилж, катологт бичдэг. Файлыг нээх үед DOS нь катологоос авч, энэ талбарыг бөглөнө.
22-31	DOS-д нөөцлөгдөнө.
32	Одоогийн бичлэгийн дугаар (0-127 гэсэн тоо байна) Энэ дугаарыг блок бичлэгийг файлд байрлуулахад хэрэглэнэ. 12, 13 байтуудтай хамт хэрэглэгдэнэ.
33-36	Бичлэгийн харьцангуй дугаар Бичлэгт шууд хандах үед энэ талбар нь тухайн бичлэгийн харьцангуй дугаарыг агуулж байх ёстой. Энэ дугаарыг DOS нь блок болон бичлэгийн дугаарт хөрвүүлэн тухайн бичлэгт хандана. Жишээ нь: 25 (19h) бичлэгийг уншихад энэ талбарт 19000000 гэж бичигдэнэ. Файлын хамгийн их хэмжээ нь 1.073.741.824

байт байх боломжтой. Бичлэгийн хэмжээ 64-өөс их бол 36-р байт байнга 00 гэсэн утгатай байна.

### Дискний файлыг үүсгэхдээ FCB-ийг ашиглах нь:

Дискэн дээрх файльтай харьцахдаа зөв тодорхойлсон FCB-ийг программ агуулна. Дисктэй хийх оролт гаралтын үйлдүүдэд DX регистрт FCB-ын хаягийг хийх ба DS:DX регистрүүдээр FCB доторх талбаруудад хандана. Шинэ файл үүсгэхдээ DOS INT 21H тасалдлын 16H функцыг ашиглана.

```
MOV  AH, 16H      ; файл үүсгэх
LEA  DX, FCBнэр
INT 21H
```

DOS нь FCB-д байгаа өгөгдлийг ашиглан файлын нэр байгаа санг хайна. Олбол, сан дахь зайг дахин хэрэглэх ба олоогүй бол дискэнд сул зайг хайна. Дараа нь файлын хэмжээг 0 болгох ба файлыг нээнэ. Нээх алхам нь дискний зайг шалгах ба AH регистрт доорх утгын аль нэгийг тавина. 00-дискэнд зай байна. FF-диск зайгүй.

Нээх үйлдлээр FCB-ийн дугаарыг 0 болгон, FCB-ийн бичлэгийн хэмжээг 128 (80h) байт болгоно. Бичлэгийг бичихийн өмнө эдгээр утгуудыг өөрийн бичлэгийн хэмжээг тавин өөрчилж болно.

Файлыг нээн, түүнтэй ажиллахын тулд өгөгдлийн мужын эхлэлийн хаягийг зааж өгөх хэрэгтэй. Энэ мужыг Disk Transfer Area (DTA) гэдэг. FCB нь бичлэгийн хэмжээг агуулах тул DTA –ийн төгсгөлийг заасан тэмдэгт хэрэггүй. Бичилт хийхийн өмнө 1AH функцээр DTA-ийн хаягийг DOS-д өгнө.

```
MOV  AH, 1AH      ; DTA-ийн хаягийг тавина.
LEA  DX, DTAнэр
INT 21H
```

Хэрэв программ нэг файльтай ажиллавал DTA-ийн анхны утгыг нэг л удаа тавьж өгнө. Харин нэгээс олон файльтай ажиллавал унших болон бичилт хийх бүрийн өмнө зөв DTA-ийг тодорхойлж өгөа болно.

Дараалсан бичлэгүүдийг бичихдээ 15h функцыг ашиглана.

```
MOV  AH, 15H      ; дараалсан бичлэгүүдийг бичнэ
LEA  DX, FCBнэр
INT 21H
```

Бичих үйлдэл нь FCB дахь мэдээллүүд, одоогийн DTA-ийн хаяг зэргийг хэрэглэнэ. Хэрэв бичлэг нь нэг секторын хэмжээтэй бол шууд бичнэ. Үгүй бол, секторын урттай буферийг дүүртэл бичлэгүүдийг бичнэ. Жишээлбэл: бичлэг бүр нь 128 байтын урттай бол 4 бичлэгээр буферийг дүүргээд  $4 * 128 = 512$  байт болсны дараа буферийн мэдээг 1 бүтэн секторт бичнэ. Бичилт амжилттай болбол, DOS нь FCB-ийн файлын хэмжээ бичдэг талбарын утгыг нэмэгдүүлж бичих ба одоогийн бичлэгийн дугаарыг 1-ээр нэмэгдүүлнэ. Үдоогийн бичлэгийн дугаар нь 128-аас давбол үүнийг 0 болгон, FCB-ийн одоогийг блокийн дугаарыг нэгээр нэмэгдүүлнэ. Дараа нь AL регистрт код буцаана. 00-амжилттай хийгдсэн, 02-DTA зайгүй гэсэн утгатай байна.

Бичлэгүүдийг бичиж дуусахад файлын төгсгөл гэсэн (1AH) тэмдэгтийг оруулаад, 10h функцыг ашиглан файлыг хаана.

```
MOV  AH, 10H      ; файлыг хаана
LEA  DX, FCBнэр
INT 21H
```

Энэ үйлдэл хийгдэхэд DOS нь буферт байгаа үлдсэн өгөгдлүүдийг дискэнд бичин, огноо, файлын хэмжээ зэргийг сан дотор өөрчилнө. AL регистрт код буцаана. 00-амжилттай хаасан, FFH-файл нь сан дотор байгаагүй, хэрэглэгч дискээ сольсон бйаж магадгүй гэсэн утга илэрхийлнэ.

## FCB-ийг ашиглан дискний файлыг зохион байгуулах нь:

Гарнаас хэрэглэгчийн оруулсан нэрсээр файл үүсгэн дискэнд хадгалах жишээ программыг тайлбарлая.

Программ дотор FCB-ийг доорх байдлаар зарлана. FCBREC нэртэй FCB нь доорх өгөгдлүүдийг хадгална.

FCBREC	LABEL	BYTE	
FCBDRIV	DB	04	;файл байрлуулах дискний төхөөрөмж
FCBNAME	DB	'NAMEFILE'	; файлын нэр
FCBEXT	DB	'DAT'	; файлын өргөтгөл
FCBBLK	DW	0000	; блокийн дугаар 0
FCBRCSZ	DW	?	; бичлэгийн хэмжээ
FCBFLSZ	DD	?	; файлын хэмжээ
FCBDATA	DW	?	
	DT	?	
FCBSQRC	DB	00	; одоогийн бичлэгийн дугаар
FCBRNRC	DD	?	; бичлэгийн харьцангуй дугаар

FCBREC LABEL BYTE нь 37 байтыг хамарсан, өөрөө зай эзлэхгүй, хөрвүүлэлтэд хэрэглэгддэг директив юм.

Манай программд доорх процедурууд агуулагдана.

BEGIN: Сегмент регистрүүдийн анхны утгыг тавих ба OPENF дэд процедур нь файлыг үүсгэх ба DOS-д DTA-ийн хаягийг тавина. Мөн PROCF дэд процедурыг өгөгдөл оруулахын тулд дуудна. Илүү оролт байхгүй бол CLOSEF дэд процедурыг дуудан, программыг дуусгана.

OPENF: Сан доторх файлын өгөгдлүүдийг үүсгэх ба бичлэгийн хэмжээг 32 (20h) гэж тавина. DTA-ийн хаягийн анхны утгыг олгоно.

PROCF: Гарнаас оролт болох нэрсийг хүлээн авч, бичих ба WRIT дэд процедурыг дуудан, оруулсан нэрсийг дискэнд бичнэ.

DISP: Дэлгэц цэвэрлэх, курсор байрлуулах зэрэг үйлдлүүдийг хийнэ.

WRITEF: Дискний файл уруу нэрсийг бичнэ.

CLOSEF: Файлын төгсгөлийн тэмдгийг бичин, фхйлыг хаана.

ERROR: Файл үүсгэх болон файлд бичих үед алдаа гарвал алдааны мэдээг дэлгэцэнд хэвлэнэ.

PAGE 60, 132

TITLE FCBCREATE FCB-ийг ашиглан файл үүсгэх

.model small

.stack 100h

.data

RECLEN EQU 32

NAMEPAR LABEL BYTE

MAXNLEN DB RECLEN

NAMELEN DB ?

NAMEDTA DB RECLEN DUP('')

FCBREC LABEL BYTE

FCBDRIV DB 03 ;файлыг байрлуулах дискний төхөөрөмж

FCBNAME DB 'examfile' ; файлын нэр

FCBEXT DB 'DAT' ; файлын өргөтгөл

FCBBLK	DW	0000	; блокийн дугаар 0
FCBRCSZ	DW	?	; бичлэгийн хэмжээ
FCBFLSZ	DD	?	; файлын хэмжээ
	DW	?	
	DT	?	
FCBSQRC	DB	00	; одоогийн бичлэгийн дугаар
	DD	?	; бичлэгийн харьцангуй дугаар

CRLF	DB	13, 10, '\$'
ERRCDE	DB	00
PROMPT	DB	'NAME? ', '\$'
ROW	DB	01
OPNMSG	DB	'*** OPEN ERROR ***', '\$'
WRTMSG	DB	'*** WRITE ERROR ***', '\$'

;-----

.code

```
BEGIN PROC FAR
    MOV     AX, @DATA
    MOV     DS, AX
    MOV     ES, AX

    MOV     AX, 0600H
    CALL    SETSCR           ; дэлгэцийн горимыг тавих
    CALL    SETCUR          ; курсорыг байрлуулах
    CALL    OPENF           ; файлыг нээн, DTA-г тавих
    CMP     ERRCDE, 00     ; файл үүсгэх зай байна уу?
    JZ      ALOOP          ; тийм бол үргэлжлүүл
    RET                                           ; үгүй бол DOS уруу буцна.
```

```
ALoop:
    CALL    PROCF
    CMP     NAMELEN, 00    ; оролтын төгсгөл үү?
    JNE     ALOOP         ; үгүй бол үргэлжлүүл
    CALL    CLOSEF        ; тийм бол файлыг хаана.

    MOV     AX, 4C00H
    INT     21H
    RET                                           ; DOS уруу буцна.
```

BEGIN ENDP

; open disk file and set DTA

;-----

```
OPENF PROC NEAR
    MOV     AH, 16H        ; файлыг үүсгэх хүсэлт
    LEA     DX, FCBREC
    INT     21H
    CMP     AL, 00        ; файл үүсгэх зай байна: уу?
    JNZ     C20           ; үгүй бол алдааны мэдээ гаргах

    MOV     FCBRCSZ, RECLen ; бичлэгийн хэмжээ
    LEA     DX, NAMEDTA    ; DTA-ийн хаягийг тавих
    MOV     AH, 1AH
    INT     21H
    RET
```

```

C20:
    LEA    DX, OPNMSG                ; алдааны мэдээ
    CALL  error
RET
OPENF    ENDP
;   accept input
;   -----
PROCF PROC NEAR
    MOV    AH, 09                    ; дэлгэцэнд хэвлэх хүсэлт
    LEA    DX, PROMPT                ; промптыг хэвлэх
    INT    21H

    MOV    AH, 0AH                   ; гарнаас оруулах хүсэлт
    LEA    DX, NAMEPAR                ; нэрийг авна.
    INT    21H
    CALL  DISP                        ; дэлгэц цэвэрлэн, курсор тавих
    CMP    NAMELEN, 00                ; нэр орсон уу?
    JNE D20                           ; тийм бол D20 уруу үсэр
    RET                                ; үгүй бол гарна.
D20:
    MOV    BH, 00                    ; return тэмдэгтийг солих
    MOV    BL, NAMELEN
    MOV    NAMEDTA[BX], ''           ; хоосон тэмдэгтийг оруулах
    CALL  WRITF                       ; бичих ажиллагаа

    CLD
    LEA    DI, NAMEDTA                ; нэрний
    MOV    CX, RECLen/2                ; талбарыг
    MOV    AX, 2020H                  ; цэвэрлэнэ.
    REP    STOSW
    RET
PROCF ENDP
;   scroll and set cursor
;   -----
DISP    PROC NEAR
    MOV    AH, 09                    ; дэлгэцэнд хэвлэх
    LEA    DX, CRLF                    ; carriage ret / line feed
    INT    21H                          ; DOS-ыг дуудна
    CMP    ROW, 18                      ; дэлгэцийн төгсгөл мөн үү?
    JAE    E20                          ; тийм бол E20 уруу үсэр
    INC    ROW                          ; үгүй бол мөрийг нэгээр нэмэгдүүл
    RET
E20:
    MOV    AX, 0601H                  ; нэг мөрийг цэвэрлэх
    CALL  SETSCR                       ; дэлгэцийн горим тавих
    CALL  SETCUR                       ; курсорыг байрлуулах
    RET
DISP    ENDP
;   write disk record
;   -----
WRITF PROC NEAR
    MOV    AH, 15H                    ; бичих хүсэлт

```

```

    LEA    DX, FCBREC
    INT    21H
    CMP    AL, 00                ; бичигдсэн үү?
    JZ     F20                  ; тийм бол F20 уруу үсэр
    LEA    DX, WRTMSG          ; үгүй бол
    CALL   error                ; алдааны процедурыг дуудна.
    MOV    NAMELEN, 0
F20:   RET
WRITEF ENDP
;   close disk file
;   -----
CLOSEF  PROC  NEAR
    MOV    NAMEDTA, 1AH        ; EOF тэмдэгтийг тавина.
    CALL   WRITEF
    MOV    AH, 10H            ; файлыг хаах хүсэлт
    LEA    DX, FCBREC
    INT    21H
    RET
CLOSEF  ENDP
;   scroll screen
;   -----
SETSCR  PROC  NEAR            ; АХ-д утгаа авсан байгаа
    MOV    BH, 1EH            ; хөх дээр шар
    MOV    CX, 0000
    MOV    DX, 184FH
    INT    10H
    RET
SETSCR  ENDP
;   set cursor
;   -----
SETCUR  PROC  NEAR
    MOV    AH, 02
    MOV    BH, 00
    MOV    DL, 00
    MOV    DH, ROW            ; курсор тавина
    INT    10H
    RET
SETCUR  ENDP
;   disk error routine
;   -----
error   PROC  NEAR
    MOV    AH, 09            ; DX -д хэвлэх мэдээний хаяг
    INT    21H                ; мэдээ хэвлэнэ.
    MOV    ERRCODE, 01        ; алдааны кодыг тавина.
    RET
error   ENDP
END     BEGIN

```

Бичих үйлдэл бүр FCBSQRC (одоогийн бичлэгийн дугаар)-ыг нэгээр нэмэгдүүлэх ба FCBFLSZ-д бичлэгийн хэмжээ 20h-ийг нэмнэ. Бичлэг бүр 32 байт урттай учир буферийг 16 бичлэгээр дүүргэсний дараа буферийн мэдээг дискний нэг секторт бичнэ. DTA ба буферийн байдлыг дүрсэлбэл:

DTA | одоогийн бичлэг |  
Буфер | бичлэг 00 | бичлэг 01 | бичлэг 02 | бичлэг 03 | ...  
... | бичлэг 15 |

Хэрэв хэрэглэгч 25 нэр оруулбал, бичлэгийн тоолуур 1-25 (19h) хүртэл нэмэгдэх ба файлын хэмжээ:

$25 * 32\text{байт} = 800\text{ байт (320h)}$  болно.

Хаах үйлдэл нь буферт үлдсэн 9 бичлэгийг 2 дахь секторт бичээд, сан доторх огноо, файлын хэмжээг өөрчилнө. Хэмжээ нь 20 03 00 00 гэж урвуугаараа бичигдэнэ. Сүүлийн буферийг дүрсэлбэл:

Буфер | бичлэг 16 | бичлэг 17 | ... | бичлэг 24 | 1АН | ...

Хэрэв бичлэг нь тэмдэгтүүд биш, 2-тын тоонууд бол файлын төгсгөлийн тэмдэгтийг хэрэглэхгүй, 2-тын тоо нь 1АН-ийг агуулна.

### Дараалан хандах аргаар файлыг унших

Дискэнд байгаа файлыг үндсэн DOS-ын функцүүдийг ашиглан уншихдаа FCB-ыг бичиж өгнө. Дараа нь 0FH функцыг ашиглан файлыг нээнэ.

MOV AH, 0FH ; файлыг нээнэ.

LEA DX, FCBNAME

INT 21H

Нээх үйлдлийн үр дүнд AL регистрт үр дүнгээ буцаана. AL=FFh бол дискэнд уг файл байхгүй гэсэн үг. Файлыг олсон бол AL=0 болох ба FCB-д файлын хэмжээ, одоогийн блокийн дугаарт 0, FCB-ийн бичлэгийн хэмжээг 80 гэж тавина. Нээсний дараа эдгээр утгуудыг өөрчилж болно.

Оруулах бичлэгүүдийн эхлэл нь DTA бөгөөд 1АН функцыг (EOF тэмдгийн 1АН –тэй андуурч болохгүй) ашиглан DTA –ийн хаягийг файлыг үүсгэхдээ хийдгийн адилаар тавина.

MOV AH, 1АН

LEA DX, DTANAME

INT 21H

Файлын бичлэгийг уншихдаа 14H функцыг ашиглана.

MOV AH, 14H ; бичлэгийг дараалан уншина.

LEA DX, FCBNAME

INT 21H

Унших үйлдэл нь FCB дахь мэдээллийг дискэн дээрх бичлэгийг DTA –ийн хаяг уруу дамжуулахдаа хэргэлэнэ. AL регистрт доорх утгуудын нэгийг буцаана.

00 Амжилттай уншсан.

01 Файлын төгсгөл, өөр өгөгдөл байхгүй.

02 DTA-д нэг бичлэгийг унших хангалттай зай байхгүй.

03 Файлын төгсгөл, 0-ээс бүтсэн бичлэгийг уншсан.

Эхний уншилтаар нэг бүхэл секторын өгөгдлийг DOS-ын буферт авна. FCB-оос бичлэгийн хэмжээг олоод, DTA-уруу буферээс өгөгдлийг дамжуулна. Ийнхүү буферт өгөгдөлгүй болтол дамжуулаад, дараагийн секторын хаягийг тодорхойлоод, түүний өгөгдлийг буферт авна.

Амжилттай уншилтын дараа FCB дахь одоогийн бичлэгийн дугаар нь нэгээр нэмэгдэнэ. Файлын төгсгөлийг шалгаж явах хэрэгтэй бөгөөд уншилт хийж дууссан бол файлыг хаах нь олон файл зэрэг хэрэглэгдэх үед зөв алхам болно.

### FCB-ийг ашиглан дискэн дэх файлыг унших нь:

BEGIN: Сегмент региструудийн анхны утгыг тавих ба OPENF дэд процедурыг дуудан, файлыг нээн, DTA-г тавина. Мөн READF дэд процедурыг бичлэг уншихын тулд

дуудна. Файлын төгсгөл хүрсэн бол программаас гарч, хүрээгүй бол DISP дэд процедурыг дуудна.

OPENF: Файлын нээн, бичлэгийн хэмжээг 32 (20h) гэж тавина. DTA-ийн хаягийн анхны утгыг олгоно.

READF: Дискэн дэх бичлэгүүдийг дараалуулан уншина. FCB-ийн одоогийн бичлэгийн дугаарыг автоматаар нэмэгдүүлнэ.

DISP: Дэлгэцэнд бичлэгүүдийг хэвлэнэ.

ERROR: Файл нээх болон файлаас унших үед алдаа гарвал алдааны мэдээг дэлгэцэнд хэвлэнэ.

PAGE 60, 132

TITLE FCBREAD FCB-ийг ашиглан файл унших

.model small

.stack 100h

.data

RECLEN EQU 32 ; бичлэгийн хэмжээ

FCBREC	LABEL	BYTE	
FCBDRIV	DB	03	; файлыг байрлуулах дискний төхөөрөмж
FCBNAME	DB	'examfile'	; файлын нэр
FCBEXT	DB	'DAT'	; файлын өргөтгөл
FCBBLK	DW	0000	; блокийн дугаар 0
FCBRCSZ	DW	?	; бичлэгийн хэмжээ
FCBFLSZ	DD	?	; файлын хэмжээ
	DW	?	
	DT	?	
FCBSQRC	DB	00	; одоогийн бичлэгийн дугаар
	DD	?	; бичлэгийн харьцангуй дугаар
NAMEFLD	DB	RECLEN DUP (' '), 13, 10, '\$'	
ENDCDE	DB	00	
ROW	DB	00	
OPENMSG	DB	'*** OPEN ERROR ***', '\$'	
READMSG	DB	'*** READ ERROR ***', '\$'	

;-----

.code

BEGIN PROC FAR

MOV AX, @DATA

MOV DS, AX

MOV ES, AX

MOV AX, 0600H

CALL SETSCR ; дэлгэцийн горимыг тавих

CALL SETCUR ; курсорыг байрлуулах

CALL OPENF ; файлыг нээн, DTA-г тавих

CMP ENDCDE, 00 ; файл нээгдсэн үү?

JNZENDING ; үгүй бол зогсооно.

ALoop:

CALL READF ; бичлэгийг уншина.

CMP ENDCDE, 00 ; хэвийн уншилт уу?

JNZENDING ; үгүй бол зогсооно.

CALL DISP ; дэлгэцэнд хэвлэнэ.

```

    JMP    ALOOP                ; үргэлжлүүлнэ
ENDING:                            ; төгсгөл
    MOV    AX, 4C00H
    INT    21H
    RET                            ; DOS уруу буцна.
BEGIN ENDP
;   open disk file and set DTA
;   -----
OPENF PROC NEAR
    MOV    AH, 0FH                ; файлыг нээх хүсэлт
    LEA    DX, FCBREC
    INT    21H
    CMP    AL, 00                ; файл олдсон уу?
    JNZ    C20                  ; үгүй бол алдааны мэдээ гаргах

    MOV    FCBRCSZ, RECLEN ; бичлэгийн хэмжээ
    LEA    DX, NAMEFLD          ; DTA-ийн хаягийг тавих
    MOV    AH, 1AH
    INT    21H
    RET

C20:
    MOV    ENDCDE, 01
    LEA    DX, OPENMSG          ; алдааны мэдээ
    CALL   ERROR
    RET

OPENF    ENDP
;   дискний бичлэгийг унших
;   -----
PROCF PROC NEAR
    MOV    AH, 14h                ; унших хүсэлт
    LEA    DX, FCBREC
    INT    21H
    CMP    NAMEFLD, 1AH          ; файлын төгсгөл үү?
    JNED   20                    ; үгүй бол D20 уруу үсэр
    MOV    ENDCDE, 01            ; тийм
    JMP    D90

D20:
    CMP    AL, 00                ; энгийн уншилт уу?
    JZ     D90                    ; тийм бол гарна
    MOV    ENDCDE, 01            ; үгүй бол:
    CMP    AL, 01                ; файлын төгсгөл үү?
    JZ     D90                    ; тийм бол гарна.
    LEA    DI, READMSG           ; буруу уншилт
    CALL   ERROR

D90:
    RET
PROCF ENDP
;   бичлэгийг дэлгэцэнд хэвлэх
;   -----
DISP PROC NEAR
    MOV    AH, 09                ; дэлгэцэнд хэвлэх
    LEA    DX, NAMEFLD

```

```

INT    21H          ; DOS-ыг дуудна
CMP    ROW, 20     ; дэлгэцийн төгсгөл мөн үү?
JAE    E20        ; үгүй бол E20 уруу үсэр
INC    ROW        ; тийм бол мөрийг нэгээр нэмэгдүүл
JMP    E90

E20:
MOV    AX, 0601H
CALL   SETSCR     ; дэлгэцийн горим тавих
CALL   SETCUR     ; курсорыг байрлуулах
E90:
RET

DISP   ENDP
;   дэлгэцийн горим тавих
;   -----
SETSCR PROC NEAR   ; AX-д утгаа авсан байгаа
MOV    BH, 1EH    ; хөх дээр шар
MOV    CX, 0000
MOV    DX, 184FH
INT    10H
RET
SETSCR ENDP
;   курсорыг байрлуулах
;   -----
SETCUR PROC NEAR
MOV    AH, 02
MOV    BH, 00
MOV    DL, 00
MOV    DH, ROW    ; курсор тавина
INT    10H
RET
SETCUR ENDP
;   disk error routine
;   -----
ERROR PROC NEAR
MOV    AH, 09     ; DX -д хэвлэх мэдээний хаяг
INT    21H       ; мэдээ хэвлэнэ.
RET
ERROR ENDP
END    BEGIN

```

Файлыг нээх үйлдлээр санд уг файлыг нэр болон өргөтгөлөөр нь хайна. Олмогцоо FCB дотор файлын хэмжээ, огноо, бичлэгийн хэмжээ зэргийн утгуудыг автоматаар тавина. Уншилтаар бичлэг бүрийг буферээс DTA уруу дамжуулна. Хамгийн сүүлийн уншилтаар, файлын төгсгөл болохыг заан, AL регистрт 01 утгыг тавина.

Файлыг дараалуулан унших болон бичих аргаас гадна файл дахь дурын бичлэгэд хандах доорх функцүүд байдаг.

- Дурын бичлэгийг уншихдаа шаардлагатай бичлэгийнхээ дугаарыг FCB доторх бичлэгийн харьцангуй дугаар гэсэн 33-36 дугаар байтуудад хадгалаад, DOS-ын 21H функцыг дуудна.  
MOV AH, 21h ; дурын бичлэг унших хүсэлт

LEA DX, FCBнэр  
INT 21H

Энэ унших үйлдэл нь бичлэгийн харьцангуй дугаарыг одоогийн блокийн болон бичлэгийн дугаар уруу хөрвүүлээд, шаардагдах бичлэгээ олж, DTA уруу дамжуулна. Мөн AL регистрт үйлдлийн үр дүнг илэрхийлэх утга буцаана.

00 Амжилттай уншсан.

01 Унших өгөгдөл байхгүй.

02 DTA-д хангалттай зай байхгүй тул дамжуулалт тасарсан.

03 0-ээс бүтсэн бичлэгийг уншсан.

Энд файлын төгсгөл гэсэн утга буцаагдахгүй.

- Дурын бичлэгийг бичихдээ мөн шаардлагатай бичлэгийнхээ дугаарыг FCB доторх бичлэгийн харьцангуй дугаар гэсэн 33-36 дугаар байтуудад хадгалаад, DOS-ын 21H функцийг дуудна.

MOV AH, 22h ; дурын бичлэг бичих хүсэлт

LEA DX, FCBREC

INT 21H

00 Амжилттай бичсэн.

01 Диск дүүрэн.

02 DTA-д хангалттай зай байхгүй тул дамжуулалт дууссан.

---

Эдгээр аргуудаас гадна кискэнд блокоор буюу багц бичлэгээр хандах боломжтой. Энэ үед файлыг бүхлээр нь DTA-аас дискэнд бичиж болно. Эсвэл файлыг бүхлээр нь DTA уруу уншиж болно. Үүний тулд файлыг нээх ба DTA-г тодорхойлох шаардлагатай. Блок бичлэг хийх тохиолдолд CX регистр нийт бичлэгийн тоог агуулах бөгөөд FCB доторх бичлэгийн харьцангуй дугаар нь дискэн дээр бичлэгийн эхлэлийг заана. Блок бичлэгийг бичихдээ:

MOV AH, 28H ; дурын блокийг бичих хүсэлт

MOV CX, бичлэгүүдийн-тоо

LEA DX, FCBнэр

INT 21H ; DOS-ыг дуудна.

Үр дүнг AL регистрт буцаана. 00-бүх бичлэгүүдийг амжилттай бичсэн, 01-дискэнд зай байхгүй учир бичилт хийгдээгүй.

Блок бичлэгийг уншихдаа:

MOV AH, 27H ; дурын блокийг унших хүсэлт

MOV CX, бичлэгүүдийн-тоо

LEA DX, FCBнэр

INT 21H ; DOS-ыг дуудна.

Үр дүнг AL регистрт буцаана. 00-бүх бичлэгүүдийг амжилттай уншсан, 01-файлын төгсгөлд хүрээгүй, сүүлийн бичлэг бүрэн, 02-DTA дахь боломжтой бүх бичлэгүүдийг уншсан, 03-файлын төгсгөлд хүрээгүй, сүүлийн бичлэг бүрэн бус.

CX регистрт уншсан бичлэгүүдийн тоо хадгалагдана.

Дискнээс доод түвшний хандалтаар шууд уншихдаа DOS INT 25H, INT 26H тасалдлуудыг ашиглана. Энэ тохиолдолд блок, католог зэргийг ашигладаг давуу талууд алга болно. Энэ үйлдэл нь бичлэгүүдийг сектор сектороор авч, хандана. Хаяг нь бичлэгийн логик дугаар байна.

Зам Сектор Бичлэгийн логик дугаар

0 1 0

0 2 1

1 1 9  
1 9 17  
2 9 26

Олох томъёо нь:

Бичлэгийн логик дугаар = (Зам\*9) + (Сектор-1)

Жишээ нь: зам 2, сектор 9 бол  $(2*9)+(9-1)=18+8=26$

MOV AL, дискний төхөөрөмж

MOV BX, хаяг

MOV CX, секторын тоо

MOV DX, эхлэх бичлэгийн логик дугаар

INT 25H ; уншихдаа, INT 26H- бичихдээ

CF –д амжилттай болсон бол 0, амжилтгүй болсон бол 1 гэсэн утга буцаана. Амжилтгүй болсон бол AL регистрт алдааны мэдээ тавина.

1000 0000 төхөөрөмж хариу өгөөгүй

0100 0000 төхөөрөмжийг хайхад алдаа гарсан

0010 0000 удирдлагын хэсгийн алдаа

0001 0000 мэдээллийн алдаа

0000 1000 DMA –ийн ачаалал ихэдсэн

0000 0100 хүссэн секторыг олоогүй.

0000 0011 хамгаалалттай дискэнд бичиж болохгүй.

0000 0010 хаяг олдоогүй

#### Дисктэй ажиллах бусад үйлдлүүд

- Дискний төхөөрөмжийг анхны төлөвт оруулах: 0DH

MOV AH, 0DH

INT 21H

Үүний дараа нээлттэй байгаа бүх файлуудыг хаан, DOS-ын буферууд цэвэрлэгдэнэ.

- Дискний төхөөрөмжийг сонгох: 0EH

MOV AH, 0EH

MOV DL, 02 ; C төхөөрөмж

INT 21H

0-A төхөөрөмж, 1-B төхөөрөмж гэх мэт

- Сангийн мэдээг хайх: 11H, 12H

Файлын нэрийг өөрчлөх эсвэл файлыг устгах зэрэгт эдгээр функцүүдийг хэрэглэнэ.

MOV AH, 11H ; эхний сангийн мэдээг хүснэ.

MOV DX, FCBнэр; нээгээгүй FCB

INT 21H

MOV AH, 12H ; дараагийн сангийн мэдээг хүснэ.

MOV DX, FCBнэр; нээгээгүй FCB

INT 21H

Үйлдлийн үр дүн AL регистрт орох ба 00-олдсон, FF-олдоогүй гэсэн үг.

Энэ функцүүдийн онцлог нь:

FCB-ийн тодорхойлсон файлыг хаяж байгаа боловч түүнийг нээхгүй, файлаа олоод FCB-ийн талбарт л өөрчлөлт хийнэ.

- Файлыг устгах: 13H

MOV AH, 13H ; файлыг устгах хүсэлт

MOV DX, FCBнэр; нээгээгүй FCB

INT 21H

AL регистрт 00 бол амжилттай, FF бол алдаатай гэсэн утга үйлдлийн дараа буцаана.

- Файлын нэрийг өөрчлөх: 17H

Хуучин нэрийг дарж, FCB-д шинэ нэрийг хадгална.

MOV AH, 17H

MOV DX, FCB<sub>нэр</sub>; FCB-н хаяг

INT 21H

AL регистрт 00 бол амжилттай, FF бол алдаатай гэсэн утга үйлдлийн дараа буцаана.

- Идэвхитэй байгаа дискний төхөөрөмжийн дугаарыг авах: 19H

MOV AH, 19H

INT 21H

0-A төхөөрөмж, 1-B төхөөрөмж, 2-C төхөөрөмж гэх мэт утгуудыг AL регистрт буцаана.

### **Textbook**

1. Peter Abel, "IBM PC Assembler language and programming", USA, 1987
2. Jim Mischel, "Macro magic with Turbo Assembler ", 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. Majjasuren, "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