

Lecture 20

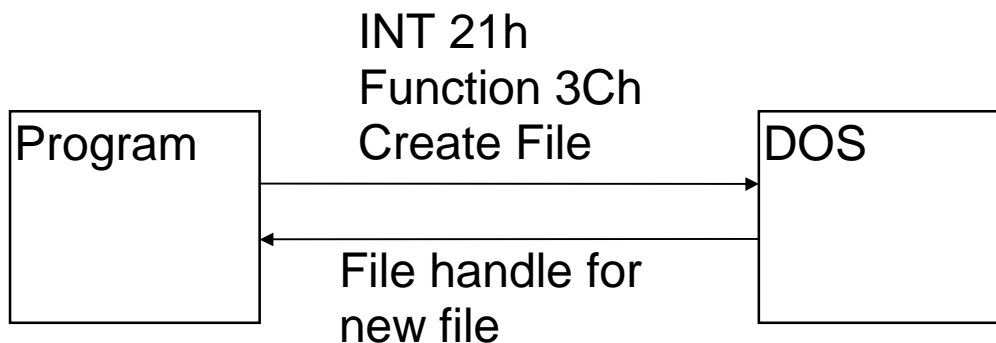
Disk Processing: Reading and Writing Files

Text: Chapter 17 (4th and 5th editions)

The file system is under control of DOS.

File operations, such as creating or opening an existing file, are done through the DOS INT 21h request.

Programs refer to a file using a **file handle**, not the name of a file.



The program will save the file handle in a word and use it for all references to the file.

DOS can also send back return codes in the AX register and the carry flag to report problems, such as

- 02 File not found
- 05 Access denied
- 19 Write protected disk
- 21 Drive not ready

(see figure 17.1 in the text for a complete list)

Example: Create a disk file

```
STACKSG SEGMENT PARA STACK 'Stack'
        DW      32
STACKSG ENDS
DATASG  SEGMENT PARA 'Data'
message db 'computer science 201'
pathname db 'A:\newfile.dat',00h
handle  dw  ?
CreateERRMSG db '** Error in file creation'
WriteERRMSG  db '** Error writing to file '
DiskFullERRMSG db '** Error: Disk is full '
DATASG  ENDS
CODESG  SEGMENT PARA 'Code'
BEGIN   PROC FAR
        ASSUME SS:STACKSG,DS:DATASG,CS:CODESG
        MOV     AX,DATASG
        MOV     DS,AX
        call    createfile
        call    writerec
        call    closefile
        MOV     AX,4C00H
        INT     21H
BEGIN   ENDP

createfile proc near
        mov     ah,3Ch ;function create
        mov     cx,00 ; normal attribute
        lea    dx,pathname ;name of file
        int    21h
        jc     create_error ; carry set -> error
        mov    handle,ax ;save the handle
        ret
create_error:
        lea    dx,CreateERRMSG
        call   PrintError
        ret
createfile endp
```

```
;
; write the 20-character record to the file.
; if carry flag is set there was a problem.
; AX returns with actual number of chars written

writerec proc near
    mov     ah,40h      ;function write
    mov     bx,handle  ;using this handle
    mov     cx,20      ;20 characters
    lea     dx,message ;in this message
    int     21h        ;do the write
    jc      WriteErr   ;carry means problem
    cmp     ax,20      ;all chars written?
    jne     DiskFull   ;if not, problem!
    ret                               ;else we're done
;   There was an error in writing to the file.
;   Print an error message
WriteErr:
    lea     dx,WriteERRMSG
    call    PrintError
    ret
;   Couldn't write... disk is full
;   Print an error message
DiskFull:
    lea     dx,DiskFullERRMSG
    call    PrintError
    ret
writerec endp
; All done writing .. close the file
closefile proc near
    mov     ah,3Eh     ;function close
    mov     bx,handle ;this file handle
    int     21h
    ret
closefile endp
CODESEG     ENDS
            END       BEGIN
```

Reading from an existing disk file

1. Open the file (give a name, get a handle) function 3Dh
2. Read records (function 3Fh)
3. Close the file (function 3Eh)

Function 3Dh: Open a File

DX: Address of ASCIIZ string with path name

AL: Access code (R/O,W/O,R/W,sharing...)

AX: Gets the file handle

Function 3Fh: Read a record

BX: File handle

CX: Number of bytes to read

DX: Address of the input area to receive bytes

AX: Gets number of bytes actually read
(0 means past end of file)

```

TITLE   P17HANRD  Read disk records sequentially
        .MODEL   SMALL           ; .EXE
        .STACK   64

; -----
        .DATA
ENDCDE  DB        00             ;End process indicator
HANDLE  DW        ?
IOAREA  DB        32 DUP(' ')
OPENMSG DB        '*** Open error ***', 0DH, 0AH
PATHNAM DB        'D:\NAMEFILE.SRT',0
READMSG DB        '*** Read error ***', 0DH, 0AH
ROW     DB        00

; -----
        .CODE
BEGIN   PROC      FAR
        MOV       AX,@data      ;Initialize
        MOV       DS,AX        ; segment
        MOV       ES,AX        ; registers
        MOV       AX,0600H
        CALL      Q10SCR       ;Clear screen
        CALL      Q20CURS     ;Set cursor
        CALL      E10OPEN     ;Open file, set DTA
        CMP       ENDCDE,00    ;Valid open?
        JNZ       A90         ; no - exit
A20LOOP:
        CALL      F10READ     ;Read disk record
        CMP       ENDCDE,00    ;Normal read?
        JNZ       A90         ; no - exit
        CALL      G10DISP     ; yes - display name,
        JMP       A20LOOP     ; continue
A90:
        MOV       AX,4C00H     ;Exit to DOS
        INT      21H
BEGIN   ENDP

```

```
;      Open file:
;      -----
E10OPEN PROC    NEAR
    MOV     AH,3DH      ;Request open
    MOV     AL,00      ;Normal file
    LEA     DX,PATHNAM
    INT     21H
    JC      E20        ;Error?
    MOV     HANDLE,AX  ; no - save handle
    RET
E20:
    MOV     ENDCDE,01  ; yes --
    LEA     DX,OPENMSG ; display
    CALL    X10ERR    ; error message
    RET
E10OPEN ENDP
;      Read disk record:
;      -----
F10READ PROC    NEAR
    MOV     AH,3FH      ;Request read
    MOV     BX,HANDLE
    MOV     CX,32      ;30 for name, 2 for CR/LF
    LEA     DX,IOAREA
    INT     21H
    JC      F20        ;Error on read?
    CMP     AX,00      ;End of file?
    JE      F30
    CMP     IOAREA,1AH ;EOF marker?
    JE      F30        ; yes - exit
    JP      F90
F20:      ; no --
    LEA     DX,READMSG ; invalid read
    CALL    X10ERR
F30:
    MOV     ENDCDE,01  ;Force end
F90:      RET
F10READ ENDP
```

```

;      Display name:
;      -----
G10DISP PROC    NEAR
    MOV     AH,40H      ;Request display
    MOV     BX,01      ;Set handle
    MOV     CX,32      ;  and length
    LEA     DX,IOAREA
    INT     21H
    CMP     ROW,20     ;Bottom of screen?
    JAE     G80        ;  yes - bypass
    INC     ROW        ;  no - increment row
    JMP     G90

G80:
    MOV     AX,0601H
    CALL    Q10SCR     ;Scroll
    CALL    Q20CURS    ;Set cursor

G90:  RET
G10DISP ENDP
;      Scroll screen:
;      -----
Q10SCR PROC     NEAR      ;AX set on entry
    MOV     BH,1EH      ;Set color
    MOV     CX,0000
    MOV     DX,184FH    ;Request scroll
    INT     10H
    RET

Q10SCR ENDP
;      Set cursor:
;      -----
Q20CURS PROC    NEAR
    MOV     AH,02H      ;Request set
    MOV     BH,00      ;  cursor
    MOV     DH,ROW     ;  row
    MOV     DL,00      ;  column
    INT     10H
    RET
Q20CURS ENDP

```



```
;      Display disk error message:
;      -----
X10ERR PROC    NEAR
        MOV     AH,40H      ;DX contains address
        MOV     BX,01      ;Handle
        MOV     CX,20      ;Length
        INT     21H        ;  of message
        RET
X10ERR ENDP
        END     BEGIN
```

Exercises - Lecture 20

1. Write a program that will prompt a user for a last name, and put all users whose names begin with A-L in a file called A2L.dat, and users with last names from M-Z in a file called M2Z.dat.
2. Write a program that will remove all TAB characters from a file, creating a new TAB-free file. Let the user type in the names of the input and output files.