Session - 4

File Handling

VSAM

Different types of VSAM Datasets used in CICS are :

- ESDS Entry Sequenced Dataset
- KSDS Key Sequenced Dataset
- RRDS Relative Record Dataset

Services provided by CICS

- Basic Operations required for a file are

  Adding a Record.
  Modifying an Existing Record.
  Deleting an Existing Record.
  Browsing One or Selected or All Records.

- In Addition, CICS Provides

  Exclusive Control. (Record Level Locking).
  Data Independence.
  Journaling.
  Opening and closing Files.

Defining a file to CICS

- Files should be defined in FCT (File Control Table).
- FCT will contain all the Information about a File. (like dataset name, access methods, permissible file service request, etc.)
- Defining Files can be done either by CEDA Transaction or DFHFCT Macro.

Syntax of DFHFCT Macro

```
DFHFCT   TYPE=FILE,ACCMETH=VSAM,
                DATASETNAME=NAME,
                SERVRQ=(ADD,BROWSE,DELETE,READ,UPDATE),
               FILSTAT=(ENABLED,OPENED)
```

File Handling in Programs

- Files should not be defined in the Program.
- Program should not open or close a File.
- Records can be written in any order. A number of records can be added at a time.
- Records can be inserted, updated or deleted.

Important keywords

- Dataset/File :- Name in the FCT.
- Into/From (WS-Rec) :- Working-Storage Area defined in the program where the CICS Puts/Gets the Data.
- RIDFLD :- Contains the Record Key.
- RESP :- Contains the return code of the executed command.
- LENGTH :- Length of the Record to be Retrieved or Written.

Random Read

Move 1010 to rec-key

EXEC CICS **READ**

File(ozags1f)

[Into(ws05-rec)]

RIDFLD(Rec-Key)

generic

GTEQ

END-EXEC.

Condition: DISABLED, NOTOPEN, NOTFND, LENGERR, DUPKEY, IOERR.

Example for Random Read

```
EXEC CICS READ
                File( ‘MAPLE41F ')
                Into(WS-EMP-REC)
                Length(WS-EMP-LEN)
                RIdfld('7135950602') | RIdfld(WS-
                EMP-KEY)
END-EXEC.
```

Sequential Read

- Sequential Read is done by Browse Oper.
- Establish the pointer to the First Record to be Read Using StartBr.
- Next and Previous Records can be Read as required Using ReadNext and ReadPrev.
- End the Browse Operation at last.
- Browse can be re-positioned.
- During Browse Operation, Records cannot be Updated.

Syntax for STARTBR

EXEC CICS **STARTBR**

FILE(filename)

RIDFLD(data-area)

END-EXEC.

Condition : DISABLED, IOERR, NOTFND, NOTOPEN.

Reading the Record after STARTBR

- Sequentially the Next or Previous Record can be read by a READNEXT or READPREV.

- The first READNEXT or READPREV will read the Record where the STARTBR has positioned the File Pointer.

Syntax of READNEXT/ READPREV

EXEC CICS **READNext | READPrev**

FILE(name)

INTO(data-area)

RIDFLD(data-area)

resp(ws-resp)

END-EXEC.

If ws-resp= dfhresp(endfile)

Condition : DUPKEY, ENDFILE, IOERR, LENGERR, NOTFND.

ENDBRowse

- ENDBRowse terminates a Previously issued STARTBR.

- SYNTAX :-

  EXEC CICS **ENDBR**

  FILE(filename)

  END-EXEC.

  Condition: INVREQ

RESETBR

- Its effect is the same as ENDBR and then giving another STARTBR.

- Syntax :

  EXEC CICS **RESETBR**

  FILE(filename)

  RIDFLD(data-area)

  END-EXEC.

Condition: IOERR, NOTFND.

WRITE command

- Adds a new record into the File.
- For ESDS, RIDFLD is not used but after write execution, RBA value is returned and Record will be written at the end of the File.
- For KSDS, RIDFLD should be the Record Key. The record will be written depending on the Key.
- MASSINSERTion must be done in ascending order of the Key.

Syntax for WRITE

```
EXEC CICS  WRITE
                    FILE(filename)
                    FROM(data-area)
                    RIDFLD(data-area)
END-EXEC.
```

Condition: DISABLED, DUPREC, IOERR, LENGERR, NOSPACE, NOTOPEN.

REWRITE Command

- Updates a Record which is Previously Read with UPDATE Option.
- REWRITE automatically UNLOCKs the Record after execution.

Syntax for REWRITE

```
EXEC CICS REWRITE
                    FILE(filename)
                    FROM(data-area)
END-EXEC.
```

Condition: DUPREC, IOERR, LENGERR, NOSPACE.

DELETE Command

- Deletes a Record from a dataset.

- Record can be deleted in two ways,

  RIDFLD with the full key in it. And

  the record read with READ with UPDATE will be deleted.

- Multiple Records Delete is possible using Generic Option.

Syntax of DELETE

```
EXEC CICS DELETE
                    FILE(filename)
                    RIDFLD(data-area)          Optional
END-EXEC.
```

Condition: DISABLED, DUPKEY, IOERR, NOTFND, NOTOPEN.

UNLOCK

- To Release the Record which has been locked by READ with UPDATE Command.

- Syntax :

  EXEC CICS UNLOCK

  FILE(filename)

  :

  [other options]

  END-EXEC.

Condition: DISABLED, IOERR, NOTOPEN.

General Exceptions

- The following Exceptions usually will occur for **ALL** CICS File Handling Commands.

  FILENOTFOUND,

  NOTAUTH,

  SYSIDERR,

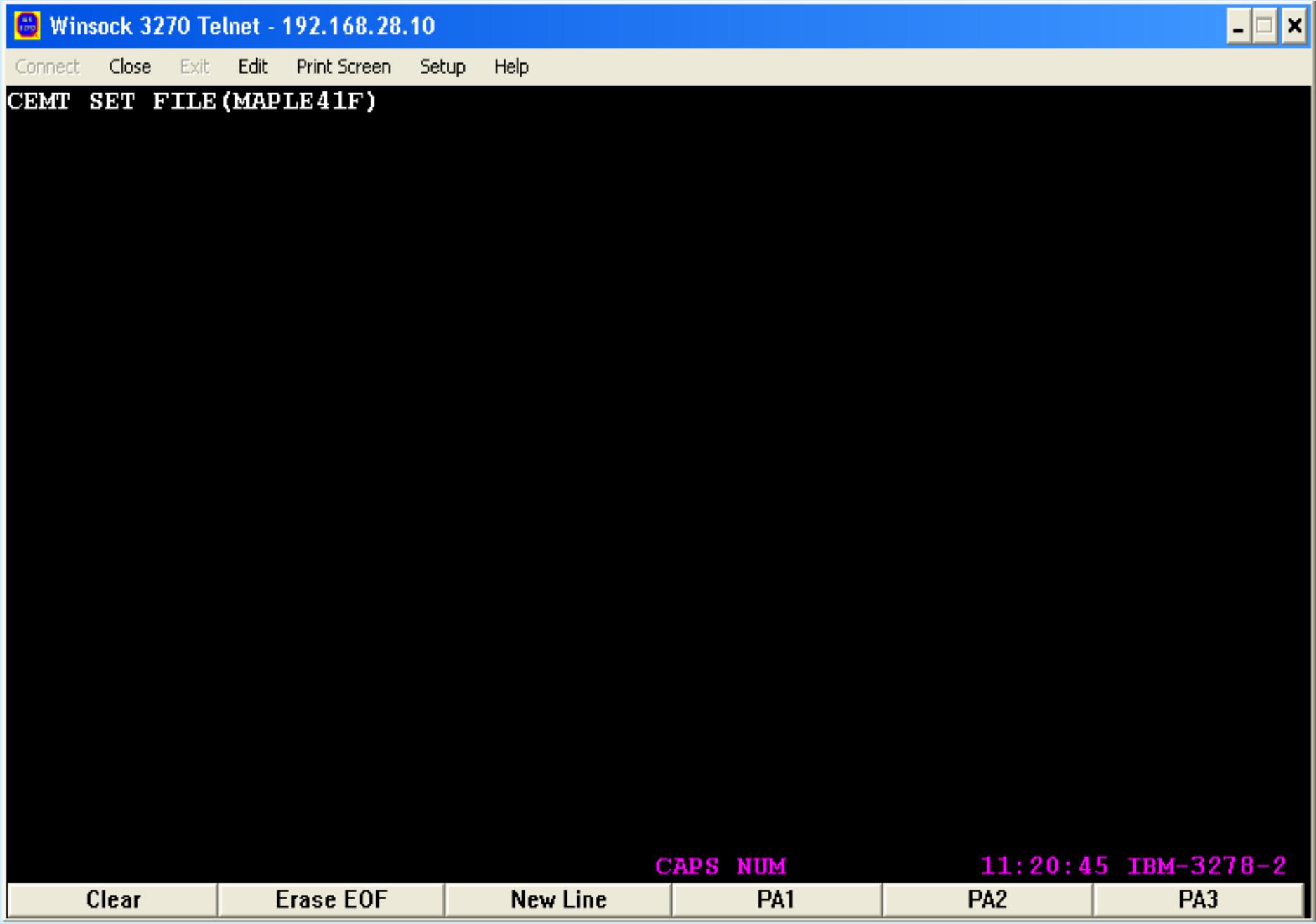
  INVREQ

In Addition to the above, Exceptions shown along the systax will occur.

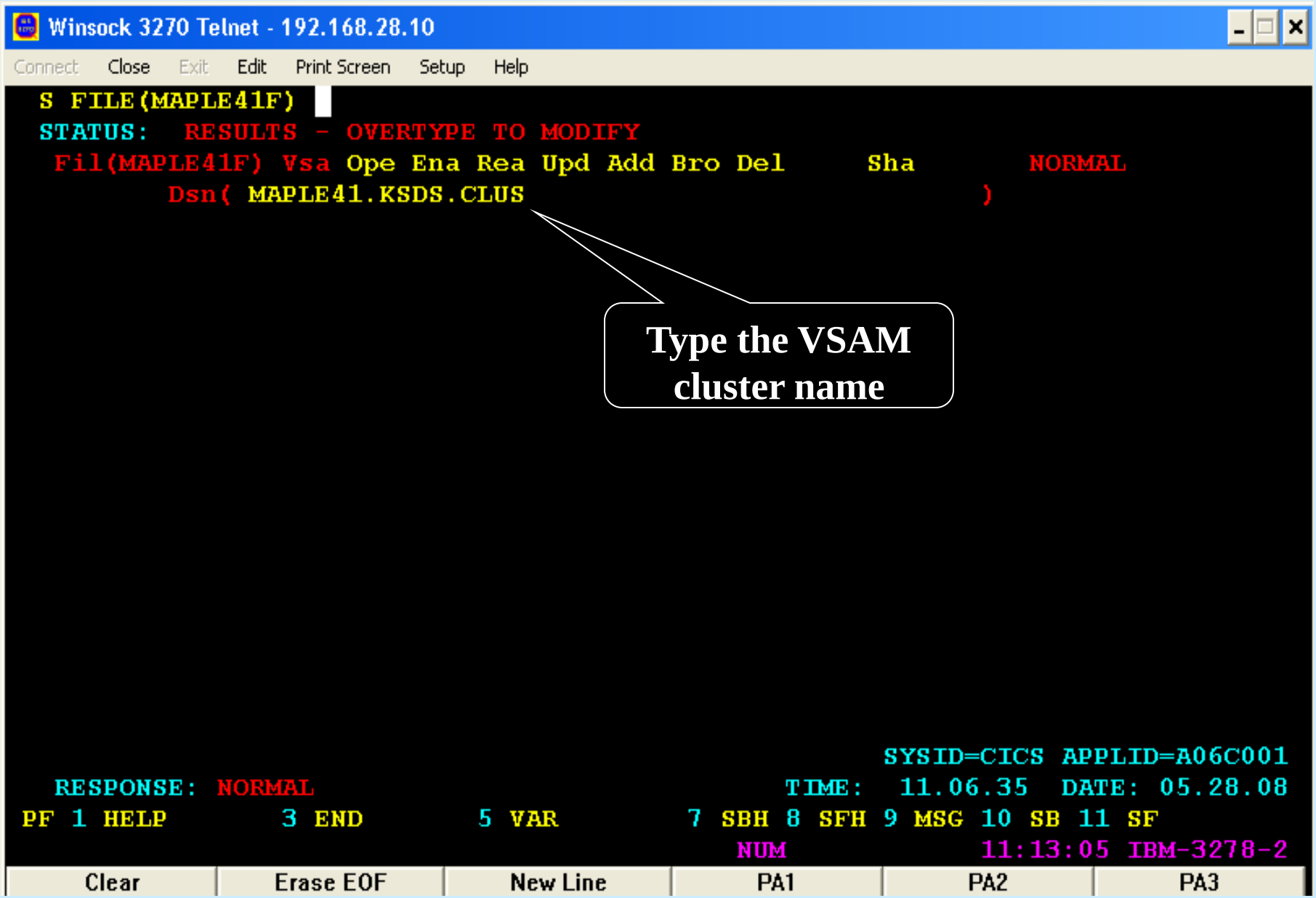
Recap

- What is VSAM?
- What are the types of VSAM files?
- What are the operations that can be done on file?
- Name some of the file commands of CICS?
- How is random access done in KSDS file?
- What are the commands for sequential access?
- How is sequential access of ESDS done?

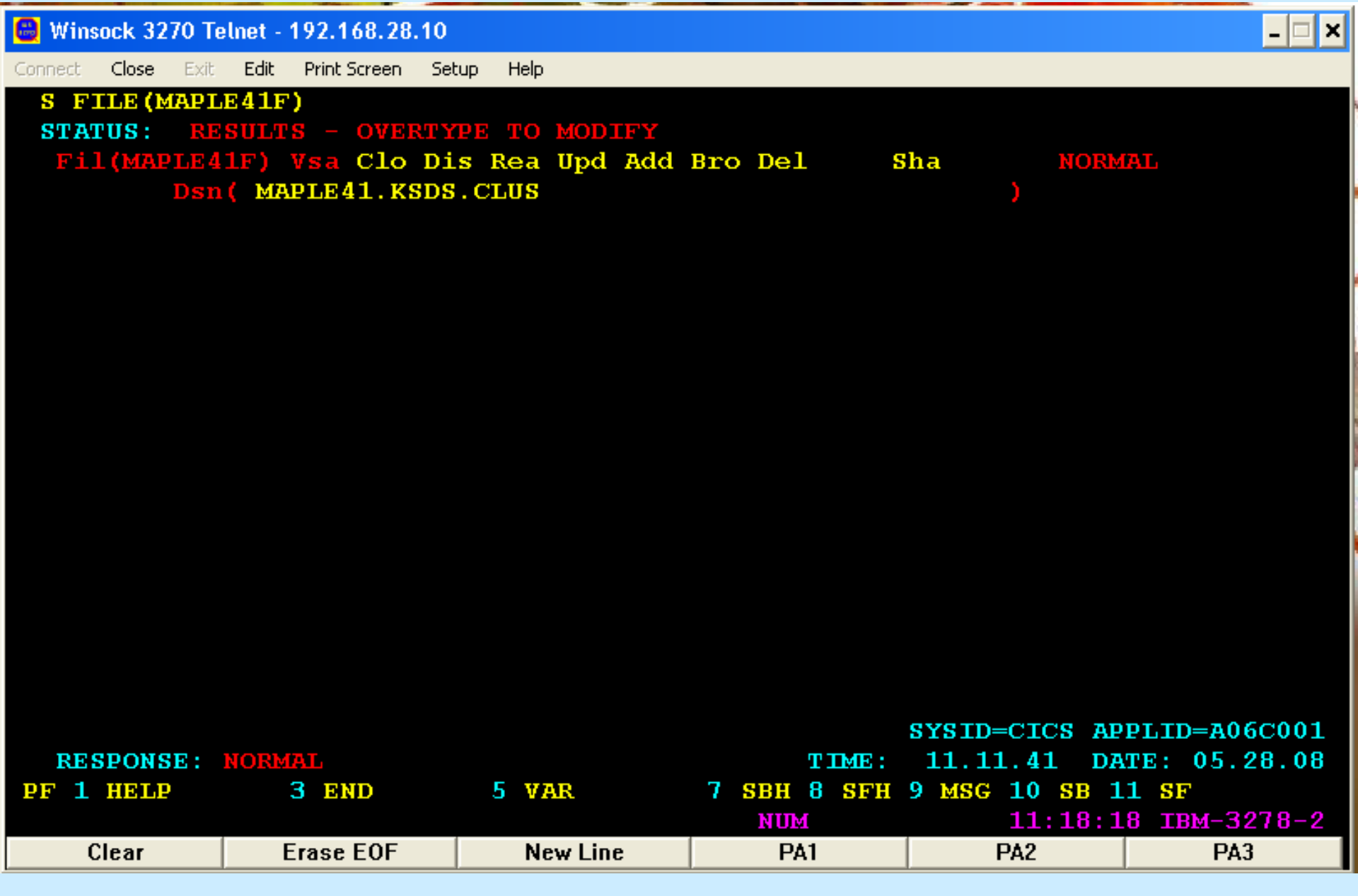
Lab Session


Winsock 3270 Telnet - 192.168.28.10
Connect
Close
Exit
Edit
Print Screen
Setup
Help
CEMT SET FILE(MAPLE41F)
CAPS NUM
11:20:45 IBM-3278-2
Clear
Erase EOF
New Line
PA1
PA2
PA3

Open & Enable the File



Close & Disable the File



Read the record from File

```
000001       IDENTIFICATION DIVISION.
000002       PROGRAM-ID. TESTNG.
000003       ENVIRONMENT DIVISION.
000004       DATA DIVISION.
000005       WORKING-STORAGE SECTION.
000006            COPY EMAP.
000007       01 REC1.
000008          02 ENO         PIC X(4).
000009          02 FILLER    PIC X.
000010          02 ENAME    PIC X(15).
000011          02 FILLER    PIC X(60).
000012       PROCEDURE DIVISION.
000013       PARA1.
000014            MOVE '1111' TO ENO.
```

```
000015          EXEC CICS READ
000016             FILE('MAPLE41F')
000017             RIDFLD(ENO)
000018             INTO(REC1)
000019          END-EXEC.
000020          MOVE ENO TO ENOO.
000021          MOVE ENAME TO ENAMEO.
000022          EXEC CICS SEND
000023             MAP('EMP') MAPSET('MAPLE41')
000024          END-EXEC.
000025          EXEC CICS
000026             RETURN
000027          END-EXEC.
000028          STOP RUN.
```

Read all the records in the file using START BROWSE and READNEXT

```
000001        IDENTIFICATION DIVISION.
 000002        PROGRAM-ID. TTSG.
 000003        ENVIRONMENT DIVISION.
 000004        DATA DIVISION.
 000005        WORKING-STORAGE SECTION.
 000006             COPY EMAP.
 000007        01 REC1.
 000008           02 ENO PIC X(4).
 000009           02     PIC X.
 000010           02 ENAME PIC X(15).
 000011           02     PIC X(60).
 000012        77 WS-RESP PIC S9(8) COMP.
 000013        PROCEDURE DIVISION.
 000014        PARA1.
 000015             MOVE "1111" TO ENO.
 000016             EXEC CICS STARTBR
 000017                FILE('MAPLE41F')
 000018                RIDFLD(ENO)
 000019                GTEQ
 000020             END-EXEC.
```

```
000021       PARA2.
000022           EXEC CICS READNEXT
000023              FILE('MAPLE41F')
000024              RIDFLD(ENO)
000025              INTO(REC1)
000026              RESP(WS-RESP)
000027           END-EXEC.
000028           IF WS-RESP = DFHRESP(ENDFILE)
000029              GO TO PARA3
000030           ELSE
000031              MOVE ENO TO ENOO
000032              MOVE ENAME TO ENAMEO
000033              EXEC CICS SEND
000034                   MAP('EMP') MAPSET('MAPLE41')
000035              END-EXEC
000036              EXEC CICS DELAY
000037                INTERVAL(000005)
000038              END-EXEC
000039              GO TO PARA2.
000040       PARA3.
000041           EXEC CICS ENDBR
000042             FILE('MAPLE41F')
000043           END-EXEC.
000044           EXEC CICS
000045               RETURN
000046           END-EXEC.
000047           STOP RUN.
```

Input Data in Map and write the record to File

```
000001        IDENTIFICATION DIVISION.
000002         PROGRAM-ID. TSTNG.
000003         ENVIRONMENT DIVISION.
000004         DATA DIVISION.
000005         WORKING-STORAGE SECTION.
000006              COPY EMAP.
000007         01 REC1.
000008            02 ENO PIC X(4).
000009            02 F PIC X.
000010            02 ENAME PIC X(15).
000011            02 F PIC X(60).
000012         PROCEDURE DIVISION.
000013         PARA1.
000014              MOVE LOW-VALUES TO EMPI, EMPO.
000015              MOVE LOW-VALUES TO REC1.
000016              EXEC CICS SEND
000017                 MAP('EMP') MAPSET('MAPLE41')
000018              END-EXEC.
```

```
000019          EXEC CICS RECEIVE
000020            MAP('EMP') MAPSET('MAPLE41')
000021          END-EXEC.
000022          MOVE ENOI TO ENO.
000023          MOVE ENAMEI TO ENAME.
000024          EXEC CICS WRITE
000025            FILE('MAPLE41F')
000026            FROM(REC1)
000027            LENGTH(LENGTH OF REC1)
000028            RIDFLD(ENO)
000029          END-EXEC.
000030          EXEC CICS RETURN END-EXEC.
000031          STOP RUN.
```

Rewrite the record in KSDS File with new values from map.

```
000001        IDENTIFICATION DIVISION.
 000002        PROGRAM-ID. TESTN.
 000003        ENVIRONMENT DIVISION.
 000004        DATA DIVISION.
 000005        WORKING-STORAGE SECTION.
 000006             COPY EMAP.
 000007        01 REC1.
 000008           02 ENO PIC X(4).
 000009           02     PIC X.
 000010           02 ENAME PIC X(15).
 000011           02     PIC X(60).
 000012        PROCEDURE DIVISION.
 000013        PARA1.
 000014             MOVE 'E002' TO ENO.
 000015             EXEC CICS READ
 000016               FILE('MAPLE41F')
 000017               INTO(REC1)
 000018               RIDFLD(ENO)
 000019               UPDATE
 000020             END-EXEC.
```

```
000021          MOVE ENO TO ENOO.
000022          MOVE ENAME TO ENAMEO.
000023          EXEC CICS SEND
000024            MAP('EMP') MAPSET('MAPLE41')
000025          END-EXEC.
000026          EXEC CICS RECEIVE
000027            MAP('EMP') MAPSET('MAPLE41')
000028          END-EXEC.
000029          MOVE ENOI TO ENO.
000030          MOVE ENAMEI TO ENAME.
000031          EXEC CICS REWRITE
000032             FILE('MAPLE41F')
000033             FROM(REC1)
000034             LENGTH(LENGTH OF REC1)
000035          END-EXEC.
000036          EXEC CICS
000037             RETURN
000038          END-EXEC.
000039          STOP RUN.
```

Delete the Record in KSDS file

```
000001       IDENTIFICATION DIVISION.
000002       PROGRAM-ID. TSTN.
000003       ENVIRONMENT DIVISION.
000004       DATA DIVISION.
000005       WORKING-STORAGE SECTION.
000006            COPY EMAP.
000007       01 REC1.
000008          02 ENO PIC X(4).
000009          02     PIC X.
000010          02 ENAME PIC X(15).
000011          02     PIC X(60).
000012       PROCEDURE DIVISION.
000013       PARA1.
000014            MOVE LOW-VALUES TO EMPI, EMPO.
000015            MOVE 'E009' TO ENO.
```

```
000016          EXEC CICS READ
000017              FILE('MAPLE41F')
000018              INTO(REC1)
000019              RIDFLD(ENO)
000020              UPDATE
000021          END-EXEC.
000022          EXEC CICS SEND
000023              MAP('EMP') MAPSET('MAPLE41')
000024          END-EXEC.
000025          EXEC CICS DELETE
000026             FILE('MAPLE41F')
000027             RIDFLD(ENO)
000028          END-EXEC.
000029          EXEC CICS RETURN END-EXEC.
000030          STOP RUN.
```

Try Yourself !

- Write programs to write records into KSDS file.The record may have student details.(try with ESDS and RRDS as well)
- Write a program to access the record at random from KSDS file. (ESDS,RRDS as well)
- Write a program to access all the records sequentially from the KSDS file(ESDS, RRDS also)