



## **LAB 11**

### **FILE PROCESSING**

**Faculty of Electronic Engineering Technology**  
**Universiti Malaysia Perlis**

## 1. OBJECTIVES:

- 1.1 To be able to create, read, write and update files.
- 1.2 To become familiar with sequential access file processing.
- 1.3 To become familiar with random-access file processing.

## 2. INTRODUCTION:

A file is a group of related records and it is used to store large amounts of data for permanent retention. A group of related files is called a database. Computers store files on secondary storage devices, especially disk storage devices. Two ways of organizing, accessing and processing records in a file; sequential-access file and random-access file.

## 2.1 Declaring File Pointer Variables

A file pointer variable has to be declared and initialized before a file can be assessed and used. The following statements declare the file pointer variables **infilep** and **outfilep**:

```
FILE *infilep;
FILE *outfilep;
```

Notice that the data type of `infilep` and `outfilep` is `FILE*`—a pointer to `FILE`. `FILE` is a structure type defined by the `stdio` library with components to hold information needed for file access.

## 2.2 Opening and Closing a File

general format to open a file is given as:

```
internal filename = fopen (external filename, mode);
```

For example:

```
infilep = fopen ("data.txt", "r");  
outfilep = fopen ("results.txt", "w");
```

**fclose** function is used to close a file. The general format to close a file is given as:

```
fclose (internal filename);
```

For example:

```
fclose (infilep); fclose
(outfilep);
```

**feof** function can be used to determine whether the **EOF**\* has been reached.

\***EOF** or end-of-file is an indicator that marks the end of the entire file. For computers running the

Linux/UNIX OS, **<control-d>** is used as **EOF**. **2.3 Writing to and Reading from a File** **fprintf**

and **fwrite** functions are used to write data to a file :

```
For example: fprintf (outfile, "%d", number); fwrite (&client,
               sizeof(struct clientData), 1, outfile);
```

address                      number of bytes to copy to file      file pointer      number of values to be written to file

**fscanf** and **fread** functions are used to read data from a file:

```
For example: fscanf(infile, "%d", &number);
             fread(&client, sizeof(struct clientData), 1, infile);
```

### 3. TASKS:

3.1 Answer the following questions:

- a. Declare a variable pointer type of **FILE** named **cfPtr**.

```
FILE *cfPtr;
```

- b. Write a statement that opens a file name "input.txt" for reading and assigns the returned file pointer to **cf1Ptr**.

```
cf1Ptr = fopen ("input.txt","r");
```

- c. Write a statement that opens a file name "output.txt" for writing (and creation) and assigns the returned file pointer to **cf2Ptr**.

```
cf2Ptr = fopen ("output.txt","w");
```

- d. Write a statement that reads a record from the file "input.txt". The record consists of **long integer MatrixID**, **string name** and **floating point cgpa**.

```
fscanf (cf1Ptr,"%ld %s %f",&MatrixId,&name,&cgpa);
```

- e. Write a statement that writes a record to the file "output.txt". The record consists of **integer month** and **floating point avg\_temp**.

```
fprintf (cf1Ptr,"%d %f", month, avg_temp);
```

- f. State the usage of the following functions:

<b>fgetc</b> -	To get the next character from the specified stream
<b>fgets</b> -	To get the line from the specified stream
<b>fread</b> -	To reads data from the given stream
<b>fseek</b> -	To set the files position of the given stream

### 3.2 Sequential Access File:

The following program creates a sequential file named "student.txt". The program will get information from keyboard and write the information into the file "student.txt".

```
#include<stdio.h>

int main ( )
{
    long int MatrixID;
    char name[30];
    float cgpa;

    FILE *cfPtr;                                //file pointer declaration

    if ((cfPtr = fopen("student.txt", "w")) == NULL) //create and open file for writing
        printf("\nFile can't be opened");
    else
    {
        printf("Enter Matrix No, Name and CGPA\n");
        printf("Enter EOF to end input\n");
        printf(": ");
        scanf("%ld %s %f", &MatrixID, name, &cgpa);

        while (!feof(stdin))                    //check for end of file (EOF)
        {
            fprintf(cfPtr, "%ld %s %.2f\n", MatrixID, name, cgpa);    //write into file
            printf(": ");
            scanf("%ld %s %f", &MatrixID, name, &cgpa);              //get data from
                                                                        //keyboard
        }

        fclose(cfPtr);                            //close file
    } return
0;
}

"stdin"      - is a standard input stream. A system file pointer for keyboard's input stream. - is
"stdout"     a standard output stream. A system file pointer for screen's output stream. - is an
"stderr"     error output stream.
"student.txt" - is a text file that is created and saved in the hard disk. Find the file student.txt in
              the same directory of your program run. Open the file and you will see the data
              that you have keyed in.

Check your output in the file "student.txt".
```

```
90812345 Ahmad 3.50
90854321 Salina 3.90
90823456 Lee 3.75
90865432 Jason 3.30
```

### 3.3 Sequential Access File:

The following program reads data from a sequential file and displays the data to the monitor / screen. For example, we have a file name "student.txt", and we want to read the contents of the file and display the contents on the screen. #include <stdio.h>

```
int main ( )
{
    long int MatrixID;
    char name[30];
    float cgpa;

    FILE *cfPtr;                                //file pointer declaration

    if ((cfPtr = fopen("student.txt", "r")) == NULL)    //open file for
        reading printf("\nFile can't be opened"); else
    {
        printf("%-13s%6s%16s\n", "Matrix No", "Name", "CGPA"); //write a title
                                                                or //header

        //read contents from file and load them to variable MatrixID, name, and
        cgpa fscanf(cfPtr, "%d %s %f", &MatrixID, name, &cgpa);

        while(!feof(cfPtr))                            //check for end of file (EOF)
        {
            printf("%-15d%-13s%7.2f\n", MatrixID, name, cgpa);    //display data
                                                                //to screen
            fscanf(cfPtr, "%d %s %f", &MatrixID, name, &cgpa);    //read data
                                                                //from file fclose(cfPtr);
        }
    }
    return
0;
}
```

- a. Check whether the data displayed are the same as in the input file. Write down the

output Let say the contents of the file are:

```
90812345 Ahmad 3.5
90854321 Salina 3.9
90823456 Lee 3.75
90865432 Jason 3.3
```

The output will be displayed as:

Matrix No	Name	CGPA
90812345	Ahmad	3.5
90854321	Salina	3.9
90823456	Lee	3.75
90865432	Jason	3.3

Matrix No	Name	CGPA
90854321	Salina	3.90
90823456	Lee	3.75
90865432	Jason	3.30
90865432	Jason	3.30

Yes, the output is the same as the file data.

### 3.4 Random Access File:

- a. The following program creates a random access file. This program creates 10 blank records to a file named "credit.dat". Type, compile and run the following program.

```
#include <stdio.h>

struct clientData                                //clientData structure definition
{
    int acctNum; char
    lastName[15]; char
    firstName[15];
    float balance;
};

int main( )
{ int i;

    FILE *cfPtr;                                //file pointer declaration

    //initialize all the structure members with 0 and blanks struct
    clientData client = {0, "", "", 0.0};

    if((cfPtr = fopen("credit.dat", "w")) == NULL)
        printf("\nFile can't be opened"); else
    {
        //write 10 blanks records to file "credit.dat" for (i = 1;
        i<=10; i++) fwrite(&client, sizeof(struct clientData), 1,
        cfPtr);
    } fclose(cfPtr);
    return 0;
}
```

Open the file "credit.dat". What is the content of "credit.dat"?

Blank file.

- b. The previous program in (a) illustrates how to write data sequentially to a random access file, now, the following programs in (b) and (c) show how to write data randomly to a random access file and how to read data from the file. Type, compile and run both programs.

```
#include <stdio.h>

struct clientData
{
    int acctNum; char
    lastName[15]; char
    firstName[15]; float
    balance;
};

int main ( )
{
    FILE *cfPtr; struct
    clientData client;

    if((cfPtr = fopen("credit.dat", "r+")) == NULL)
        printf("File can't be opened");
    else
    {
        printf("\nEnter account number(1 to 100, 0 to end input):
        "); scanf("%d", &client.acctNum);

        //enter data to be copied into file
        while (client.acctNum != 0)
        {
            printf("Enter client's last name, first name and balance: "); fscanf(stdin, "%s
            %s %f" , client.lastName, client.firstName, &client.balance);

            //seek position in file to user-specified record fseek(cfPtr,
            (client.acctNum - 1) * sizeof(struct clientData), SEEK_SET);

            //write user-specified information in file fwrite(&client,
            sizeof(struct clientData), 1, cfPtr); printf("\nEnter account
            number(1 to 100, 0 to end input: "); scanf("%d",
            &client.acctNum); }

        }
    fclose(cfPtr); return
    0;
}
```

- c. The following program shows how to read data from a random access

```
file. #include <stdio.h>

struct clientData
{
    int acctNum; char
    lastName[15]; char
    firstName[15];
    float balance;
};

int main ( )
{
```



```

FILE *cfPtr; struct
clientData client;

if((cfPtr = fopen("credit.dat", "r")) == NULL)
    printf("File can't be opened");
else
{
    printf("%-6s%16s%15s%10s\n", "Account No", "Last Name", "First
Name", "Balance");

    //read all records from file (until eof)
    while (!feof(cfPtr))
    {
        fread(&client, sizeof(struct clientData), 1, cfPtr);
        //display records on screen
        if (client.acctNum != 0)
            printf("%-6d%16s%15s%14.2f\n", client.acctNum,
client.lastName, client.firstName, client.balance);
    }
} fclose (cfPtr);
return 0;
}

```

Compare the data on the screen. Is the data displayed same as your input data?

Yes the output of the (c) program are same as the input of the (b) program.

### 3.5 Problem Solving:

As an engineer for air traffic controller, you are required to develop a system that will alert the air traffic controller of incoming aircraft. You will have to write two programs:-

- a program to store the coordinates of incoming aircrafts into a file (input file) and
- another program to read and use the data in the file to calculate the distance of each aircraft from the control center. The distance is then saved and stored in another file (output file).

If the distance calculated is less than 20 km, write an alarm message to the screen that will inform the number of the incoming aircraft. Distance formula =  $\sqrt{x^2 + y^2}$ . (Note: use **#include <math.h>** in your program, and use : **gcc -lm 'programfilename.c'** to compile).

For example:

Input File:

Aircraft	Coordinate (km)	
	x	y
A	25.0	20.0
B	10.0	16.0
C	8.0	16.0
D	136.0	85.0

Output File:

Aircraft	Distance from Control Center (km)
A	32.02
B	18.87
C	17.89
D	160.38

Output on screen:

Incoming Aircraft(s): 2

## PROGRAM 1

```
#include <stdio.h>
#include <stdlib.h>

//Struct
typedef struct iCoordinate
{
    float axisX;
    float axisY;
}iCoordinate;

// Main body
int main()
{
    int i;

    FILE *fcor;
    iCoordinate CoXY [4];

    if ((fcor = fopen("flight.dat","w"))== NULL)
    {
        printf("Error Opening File");
        exit (-1);
    }

    for (i=0;i<4;i++)
    {
        printf ("Input X[%d] :",i);
        scanf ("%f",&CoXY[i].axisX);
        fprintf(fcor,"%0.1f\t",CoXY[i].axisX);
        printf ("Input Y[%d] :",i);
        scanf ("%f",&CoXY[i].axisY);
        fprintf(fcor,"%0.1f\n",CoXY[i].axisY);
    }
    fclose (fcor);

    return (0);
}
```

## PROGRAM 2

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

// Main body
int main()
{
    int i,iCount=0;
    float axisX[4],axisY[4],iDis[4];
    char Var[4] = {'A','B','C','D'};

    FILE *fcor;
    FILE *fdis;

    if ((fcor = fopen("flight.dat","r"))== NULL)
    {
        printf("Error Opening File");
        exit (-1);
    }

    // Reading Data
    while (!feof(fcor))
    {
        for (i=0;i<4;i++)
        {
            fscanf (fcor,"%f %f",&axisX[i],&axisY[i]);
        }

    }
    fclose (fcor);

    if ((fdis = fopen("flightdis.dat","w"))== NULL)
    {
        printf("Error Opening File");
        exit (-1);
    }

    // Calculation Function
    for (i=0;i<4;i++)
    {
        iDis[i] = sqrt ((pow(axisX[i],2))+ (pow (axisY[i],2)));

        if (iDis[i] > 20.00)
        {
            iCount++;
        }
    }
    printf ("Incoming Aircraft(s): %d\n",iCount);

    // Distance Output display
    fprintf(fdis,"Aircraft      Distance from Control Center\n");
```

```
do
{
    for (i=0;i<4;i++)
    {
        fprintf(fdis,"%c\t\t\t%.2f\n",Var[i],iDis[i]);
    }
}
while (feof(fdis));

fclose (fdis);

return (0);
}
```