PICT, PUNE

Assignment 10

Name : Bhavana Bafna
Class : SE09
Batch : E09
Roll no. : 23107

1) Title: File Handling.

2) Aim: Implementation of sequential file.

3) Problem Statement: Department maintain students database. The file contains roll no., name, div, address. Write a program to create a sequential file to store and maintain student data. It should allow user to:
a) Create a student database
b) Add record
c) Delete record
d) Search and display record.

4) Theory:

1) File Data Structure:

File structure is the organisation of data in secondary storage device in such a way that it maintain access time and storage space.

2) Need:

- File can preserve your data even if program terminates.
- We can easily access contents of file using some commands.
- We can easily move data from one file to another.

3) Types of File:

a) Data & code File:
- A data file is computer file which stores data to be used by a computer application or system including input and output data.
- It usually does not contain instruction a code to be executed.

b) Variable Fixed length file:
- File length records: All records in file having same size.
- Variable length record: Different records in file have different size.

c) Text vs Binary file:
- Text file contain textual information in form of alphabets digits & special characters or symbols.
- Binary files contain bytes or a compiled version of text file.

d) Based of data organization:
- Sequential file: Contains & stores data in chronological order
- Index sequential file: Records are stored in the order that they are within to the disk.
- Direct access file: All records are stored in direct access storage device (DASD) such as hard disk, randomly throughout the file.

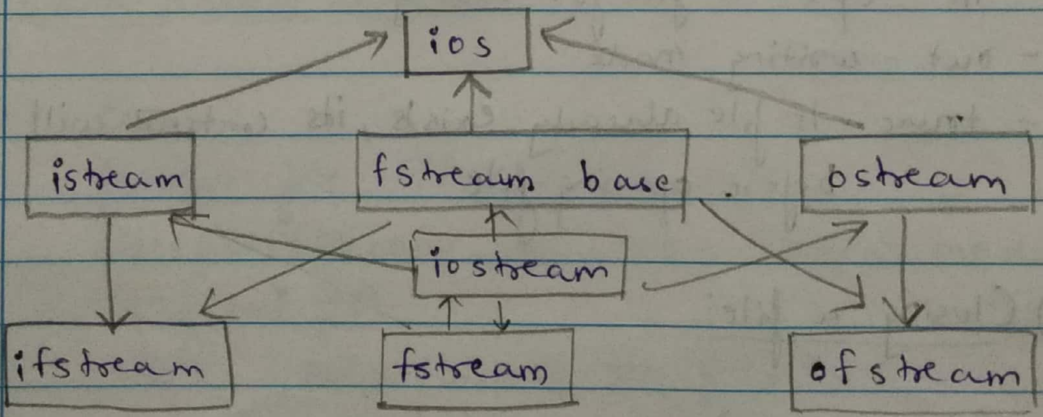
4) File Application:

Read, Store, update data on file.

5) List down various operation on file:

- Create
- Append
- Delete
- open
- Close
- Read
- Write
- Seek
- Get attribute
- Set attribute
- Rename

6) File basic : class hierarchy:



- iostream library:

<iostream.h> : Contains cin, cout, cerr & clog objects
<iomanip.h> : Contains parameterized stream manipulators
<fstream.h> : Contains information important to user controlled file processing operation.

7) File mode Syntax:

i) Opening a file:

file.open (filename, ios::openmode)

Modes

- app - append mode
- ate - Open file for output & move read/write control to end of file.
- in - open a file for reading
- out - writing mode
- trunc - If file already exists, its contents will be truncated before opening file.

ii) Closing a file:

file.close()

iii) Write in a file:

outfile.write ((char*) & data, size of (data))

8) isopen, eof meaning:

- isoopen : Return boolean value and check if file is current open or stream is currently associated to file.
- eof : This method checks if the stream has raised any EOF (End of file) error. It

9) File pointer functions:

- tell g(): used to know where the pointer is in the file.
- tell p(): get position in output sentence.
- seek(): used to move pointer to desired location w.r.t reference pointer.

Algorithm:

1) Create Student Database:

```
create()
fstream file  //create file object
file.open (filename, ios::out) //output mode
if (!out file)
        print (cannot open file)
        exit(1)
print ("Enter data for input")
cin >> data
file << data     // writing data into file.
file.close()
```

2) Append data:

```
void add()
out file . open (filename, ios : app)   //Append Mode
out file << "This data will be added".
outfile . close ()   //close file.
```

3) Delete record:

```
fstream file(filename , ios::in)         //open in input mode
fstream temp ("Temp.txt", ios::out)   // write mode
   cout << Enter roll no to be deleted
   cin >> roll
  while (file.read ((char*)& s, size of(s)))
         if (roll != s.roll)
             temp.write((char*) & s , size of(S))
         else
             Continue
 // end while
   file.close()
   temp.close()
   remove ("Original.txt")
   rename ("Temp.txt", "Original.txt").
```

In this function we copy all data from original file to temp and skip the data to be deleted. At the end we delete original file and rename it to original file and rename it to original file name.

PICT, PUNE

4) Search record:

```
search ( ) {
    fstream file (filename, ios::in)   // Read mode
    cin >> key // take input to be searched.
while (file. read (char*) &s , size of(s)) {

        if (s. roll = key)
            print ("Record found")
            // display the details
            return  // get out of function
      } //end while

    print ("Record not found")
    return.
  }
```

Test case / Validations:

- Limit validation for n records.
- input data validation for respective database
- file open validation

Scanned with CamScanner

PICT, PUNE

Conclusion:

Sequential file are suitable for application that require sequential of entire data. It is simple to program and easy to design operation like searching deletion updating consumes more time. It is not possible to add a record in middle easy. It to access a record it takes more time than direct access file.
When the order in which you keep the records is not important sequential file is best choice.