CHAPTER 1

INTRODUCTION

1.1 Introduction to File Structure

File Structures is the organization of data in secondary storage device in such a way that minimize the access time and the storage space.

1.1.1 File Structure:

It is a combination of

- Representation for data in files and
- The operations for accessing the data

It allows applications to read, write and modify data search. An improvement in file structure design makes an application hundred times faster. To access the data faster from the storage disk we use file structure. Disks are slow which are used to pack Thousands of megabytes of data.

1.1.2 Record and its types:

Records are organized collections of information that represent a single entity or item. Each record typically consists of multiple fields or attributes, which store specific pieces of data related to that entity. The type of record and its structure can vary depending on the database management system and the specific application.

Common types of records:

- Flat Records
- Hierarchical Records
- Relational Records
- Object-oriented Records
- Key-Value Records
- XML Records

1.1.3 Why we need File Structure

As we know without proper structure of organizing are ordinary system, it will generate some types of problems. So, avoiding this kind of problems we go for File Structure.

The File structure means it tells us about how the system will access the record from memory and also its tells us how the disk are preforms its tasks and also how to speed up the execution of transferring data.

Data processing from a computer science prospective:
-storage of data
-organization of data
-access to data
This will be built on your knowledge of Data Structures. Some of the importantconcepts used:
Field Structures:

- Fixed Length Fields: The method to organize fields is by limiting the maximum size of each field. The advantage in this method is that since the size of each field is fixed, the entire field can be read at once.
- Length Indicator Fields: The length of each field is specified as a prefix to actual data
- Delimited Fields: Any special character which is not a part of actual data can beused as separator.

1.2 Variable Length Record

A record has at least one variable-length field. The length of the entire record, therefore, varies according to what data is placed in the variable -length field.

Variable-length records arise in database systems in several ways:

- Storage of multiple record types in a file.
- Record types that allow variable lengths for one or more fields such asstrings (varchar).

- Record types that allow repeating fields (used in some older data models). Attributes are stored in sorted order. Variable length attributes represented by fixedsize (offset, length) with actual data stored after all fixed length attribute. Null values are represented by null-value bitmap.

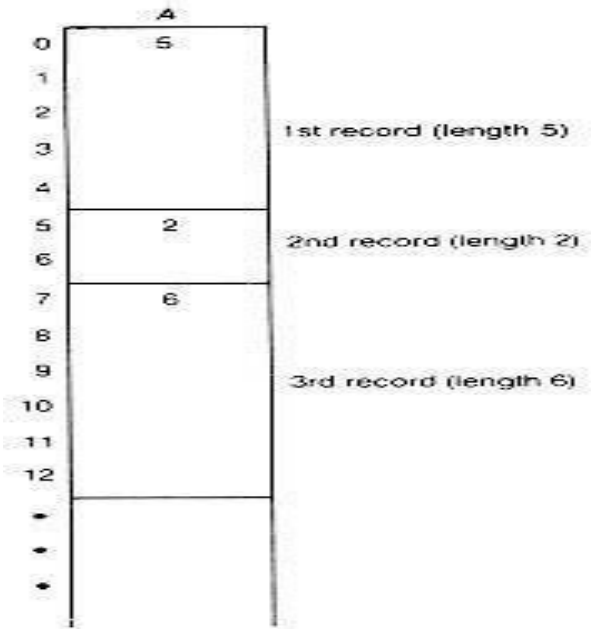

Figure 1.1: Shows the Variable Length Record

CHAPTER 2

ALGORITHM

Variable Length Record are the records that will be having exactly the same length. Variable Length Record can contain the Pack, Unpack, Search and Modify Methods.

1. PACK: The Pack Method copies the Characters of its Argument to the Bufferand the adds The Delimiter Character. We have used pack function to create a record.
2. UNPACK: Unpack function retrieves data from file and converts in to bepresented to the user. We have used this function to display record.
3. SEARCH: The search function helps us search for the record based on anyspecified parameter.
4. MODIFY: Modify function helps us to modify the existing Record.

PACK ( ) FUNCTION:

1) Erase the buffer garbage value
2) Store the fields in the buffer along with delimiter between them. End of therecord is denoted by „ $‟ symbol
3) End

UNPACK ( ) FUNCTION:

1) Declare and initialize i to 0
2) while buffer[i]!=‟ |‟ do

   strcat(field1,buffer[i])
3) End while
4) +while buffer[i]!=‟|”do

   strcat(field2,buffer[i]
5) End while
6) while buffer[i]!=‟ |‟do

   strcat(field3,buffer[i])
7) End End while
8) while buffer[i]!=‟ $‟ do

strcat(field4,buffer[i])

9) End while

10) End

SEARCH ( ) FUNCTION:

1) Declare fstream variable
2) Initialize recno ,flag and pos to 0
3) Open file in the input mode
4) While not end of the file do
    a) erase buffer from in putline
    b) call unpack()
    c) iffild1==rregno
    d) display the postion at which record found
    e) setflag=1
    f) return pos
    g) end if
5) End while
6) Close the file
7) if flag is not equalto 1 Print record notfound
8) return pos
9) End

MODIFY ( ) FUNCTION:

1) if del(pname) return true
    a) read fields to modify
    b) erase buffer contents
    c) call pack()
    d) call write_file()
2) end if
3) end

DELETE ( ) FUNCTION:

1) declare fstream variable
2) declare mark and initialize it to '*'
3) declare pos and initialize flag=0
4) call search()
5) if(pos)
    a. open the file in input and output mod
    b. pos=pos-2
    c. t= fp.get()
    d. while t is not equal to $ and pos! =0
    e. decrement the pos
    f. move the file
    g. c.t=fp.get()
    h. end while
6) if pos is not equal to 0
        Move the file pointer to (pos+2)th location
7) else
        Move file pointer to pos
8) put mark(*)on that pos
9) Set flag==1
10) Close the file
11) if flag=0
        Return 0
12) else
        Return 1
13) end if
14) end

CHAPTER 3

IMPLEMENTATION

3.1 Overview of Online Voting System.

Online voting is often seen as a tool for storing process more efficient and for increasing trust in its management. Properly implemented, storage of files can increase the security of the ballot, speed up the processing of results and make voting easier. However, the challenges are considerable. If not carefully planned and designed, voting can undermine the confidence in the whole electoral process. This policy paper outlines contextual factors that can influence the success of storage of files and highlights the importance of taking these fully into account before choosing to introduce voting information storage technologies.

3.2 Problem Statement

As information technology evolves over time, the need for a better, faster, more convenient and secure voting is essential requirement. The storage of informationis one of the main concerns, such as storing in secondary device, access of information.

3.3 Solution

The program is developed for managing the online voting activities and stores the details of voter, candidate, candidate's party and the ward in computer. This mini project will first display the menu for the user and allows the user to select the optionand use the menu accordingly. The user can make his/her own party. Then it will be stored in file.

3.4 Objectives

Our project introduces storing the information of voters and candidates with an objective to make the storage of information is more efficient, easier and fast. This project explores how computer technology can be used to solve the problem of users.

The main objectives provided by this software are as follows-

- User can select the option by looking into the menu.
- Can create his/her password.
- Can create own party and own party symbol.
- Can display the information about voters, candidates
- Can search the information
- Can modify the information using specific key.
- Can delete the information using specific key.

CHAPTER 4

SNAPSHOTS

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
1

VOTER TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
1
Enter details[voter id, voter name, date of birth, gender, voter password, booth number]
1 gagan 6/5/2001 male 122 12

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
1
Enter details[voter id, voter name, date of birth, gender, voter password, booth number]
2 disha 5/6/2001 female 123 12

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
2

Voter Id: 1
Voter Name:gagan
Date of Birth: 6/5/2001
Gender: male
Voter Password: 122
Booth number: 12

Voter Id: 2
Voter Name:disha
Date of Birth: 5/6/2001
Gender: female
Voter Password: 123
Booth number: 12

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
```

Figure 4.1: Shows the Snapshots of Pack and Unpack Operation of the Voter Table.

```
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
1

VOTER TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
3
Enter voter id to search
4

Record is 4|manu|5/6/2001|male|123|15$
Record Found at Position 3
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
4
Enter voter id to search
4

Record is 4|manu|5/6/2001|male|123|15$
Record Found at Position 3
Enter details[voter id, voter name, date of birth, gender, voter password, booth number] to modify
5 raju 5/6/2002 male 123 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
2

Voter Id: *
Voter Name:gagan
Date of Birth: 6/5/2001
Gender: male
Voter Password: 122
Booth number: 12

Voter Id: *
Voter Name:disha
Date of Birth: 5/6/2001
Gender: female
Voter Password: 123
Booth number: 12

Voter Id: *
Voter Name:manu
Date of Birth: 5/6/2001
Gender: male
Voter Password: 123
Booth number: 15

Voter Id: 5
Voter Name:raju
Date of Birth: 5/6/2002
Gender: male
Voter Password: 123
Booth number: 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
```

Figure 4.2: Shows the Snapshots of Search and Modify Operation of the Voter Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
1

VOTER TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu

5
Enter voter id of the record to delete
2

Record is 2|disha|5/6/2001|female|123|12$
Record Found at Position 2
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
2

Voter Id: *
Voter Name:gagan
Date of Birth: 6/5/2001
Gender: male
Voter Password: 122
Booth number: 12

Voter Id: *
Voter Name:disha
Date of Birth: 5/6/2001
Gender: female
Voter Password: 123
Booth number: 12

Voter Id: 4
Voter Name:manu
Date of Birth: 5/6/2001
Gender: male
Voter Password: 123
Booth number: 15

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.Main Menu
```

Figure 4.3: Shows the Snapshot of Delete Operation of the Voter Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
2

CANDIDATE TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[Candidate id, candidate name, gender, party name, ward number]
1 manu male bjp 12

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[Candidate id, candidate name, gender, party name, ward number]
2 raju male jds 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Candidate Id: 1
Candidate Name:manu
Gender: male
Party Name: bjp
Ward number: 12

Candidate Id: 2
Candidate Name:raju
Gender: male
Party Name: jds
Ward number: 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.4: Shows the Snapshot of Pack and Unpack Operation of the Candidate Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
2

CANDIDATE TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
3
Enter candidate id to search
6

Record is 6|basava|male|cgp|45$
Record Found at Position 3
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
4
Enter candidate id to search
6

Record is 6|basava|male|cgp|45$
Record Found at Position 3Enter details[Candidate id, candidate name, gender, party name, ward number]
5 mamu male bjp 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Candidate Id: *
Candidate Name:manu
Gender: male
Party Name: bjp
Ward number: 12

Candidate Id: *
Candidate Name:raju
Gender: male
Party Name: jds
Ward number: 45

Candidate Id: *
Candidate Name:basava
Gender: male
Party Name: cgp
Ward number: 45

Candidate Id: 5
Candidate Name:mamu
Gender: male
Party Name: bjp
Ward number: 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.5: Shows the Snapshots of Search and Modify Operation of the Candidate Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
2

CANDIDATE TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
5
Enter candidate id of the record to delete
2

Record is 2|raju|male|jds|45$
Record Found at Position 2
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.6: Shows the Snapshot of Delete Operation of the Candidate Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
3

PARTY TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[party name party symbol]
bjp flower

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[party name party symbol]
cgp hand

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Party Name: bjp
Party Symbol:flower

Party Name: cgp
Party Symbol:hand

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.7: Shows the Snapshot of Pack and Unpack Operation of the Party Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
3

PARTY TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
3
Enter party name to search
bjp

Record is bjp|flower$
Record Found at Position 1
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
4
Enter party name to search
bjp

Record is bjp|flower$
Record Found at Position 1Enter details[party name party symbol]
jds fruits

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Party Name: *jp
Party Symbol:flower

Party Name: cgp
Party Symbol:hand

Party Name: jds
Party Symbol:fruits

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.8: Shows the Snapshots of Search and Modify Operation of the Party Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
3

PARTY TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
5
Enter party name of the record to delete
cgp

Record is cgp|hand$
Record Found at Position 2
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Party Name: *jp
Party Symbol:flower

Party Name: *gp
Party Symbol:hand

Party Name: jds
Party Symbol:fruits

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.9: Shows the Snapshot of Delete Operation of the Party Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
4

WARD TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[ward number, ward area, booth count]
12 goodluck 13

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[ward number, ward area, booth count]
13 2circel 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Ward Number: 12
Ward Area:goodluck
Booth count: 13

Ward Number: 13
Ward Area:2circel
Booth count: 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.10: Shows the Snapshot of Pack and Unpack Operation of the Ward Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
4

WARD TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
3
Enter ward number to search
12

Record is 12|goodluck|13$
Record Found at Position 1
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
4
Enter ward number to search
12

Record is 12|goodluck|13$
Record Found at Position 1Enter details[ward number, ward area, booth count]
1 goodcircel 56

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Ward Number: *2
Ward Area:goodluck
Booth count: 13

Ward Number: 13
Ward Area:2circel
Booth count: 45

Ward Number: 1
Ward Area:goodcircel
Booth count: 56

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.11: Shows the Snapshots of Search and Modify Operation of the Ward Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
4

WARD TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
5
Enter ward number of the record to delete
1

Record is 1|goodcircel|56$
Record Found at Position 3
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

Ward Number: *2
Ward Area:goodluck
Booth count: 13

Ward Number: 13
Ward Area:2circel
Booth count: 45

Ward Number: *
Ward Area:goodcircel
Booth count: 56

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.12: Shows the Snapshot of Delete Operation of the Ward Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
5

Boothofficer TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[boothofficer id, boothofficer name, ward area, ward no]
1 disha good1cicel 12

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
1
Enter details[boothofficer id, boothofficer name, ward area, ward no]
2 jai nagar 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

boothofficer id: 1
boothofficer name:disha
ward_area: good1cicel
Ward number: 12

boothofficer id: 2
boothofficer name:jai
ward_area: nagar
Ward number: 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.13: Shows the Snapshot of Pack and Unpack Operation of the Booth Officer Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
5

Boothofficer TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
3
Enter candidate id to search
6

Record is 6|manu|raninagar|12$
Record Found at Position 3
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
4
Enter candidate id to search
6

Record is 6|manu|raninagar|12$
Record Found at Position 3Enter details[boothofficer id ,boothofficer name, ward area,ward no]
5 manu good2circel 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

boothofficer id: *
boothofficer name:disha
ward_area: good1cicel
Ward number: 12

boothofficer id: *
boothofficer name:jai
ward_area: nagar
Ward number: 45

boothofficer id: *
boothofficer name:manu
ward_area: raninagar
Ward number: 12

boothofficer id: 5
boothofficer name:manu
ward_area: good2circel
Ward number: 45

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.14: Shows the Snapshots of Search and Modify Operation of the Booth Officer Table.

```
gagan@gagan-virtual-machine:~/gagan$ ./a.out proj1.cpp

VARIABLE LENGTH RECORDS
1:VOTER 2:CANDIDATE 3.PARTY 4.WARD 5.Booth Officer
5

Boothofficer TABLE

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
2

boothofficer id: *
boothofficer name:disha
ward_area: good1cicel
Ward number: 12

boothofficer id: 2
boothofficer name:jai
ward_area: nagar
Ward number: 45

boothofficer id: 6
boothofficer name:manu
ward_area: raninagar
Ward number: 12

1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
5
Enter candidate id of the record to delete
2

Record is 2|jai|nagar|45$
Record Found at Position 2
1:Pack 2:Unpack 3:Search 4:Modify 5.Delete 6.main Menu
```

Figure 4.15: Shows the Snapshot of Delete Operation of the Booth Officer Table.

CHAPTER 5

CONCLUSION

- Provides the facilities like creation, deletion, modification and searching of the details for both voters and candidates.
- One can search the voter, candidate, party or ward by using unique identification.
- It stores all the information of users.
- Candidate can make his/her own party.
- Candidate can select the ward for the election.

REFERENCES

- How to store variable length records using C++.
- Title-File structures – An object-oriented approach with C++ - 3rd. Edition.
- Publisher- Michael J. Folk, Bill Zoellick, Greg Riccardi, Year-1998
- w3 schools.com