Apply the knowledge

ASSIGNMENT 12

Aim: Implement direct access file using hashing (chaining w/o replacement) and perform following operations on it.

a. Create Database
b. Display Database
c. Add record
d. Search record
e. Modify record.

Theory: Direct access files are files that have been designed to make direct record retrieval as easy and efficient as possible is known as direct access file.

Advantages:
- Rapid access to records in a direct fashion
- Does not use large index tables and dictionaries

Characteristics:

- Records are stored sequentially but accessed randomly
- Faster access method
- Addition overhead to maintain index.

Algorithm:

① add record()

Ⓘ int i, cntr, addr, addr1 = 0, prev
② fp.open ("file.dat", ios::in | ios::out)

③ accept input
④ addr = hash_key (data)
⑤ fp.seekg (addr * sizeof (data), ios::beg);
fp.read ((char *) & temp_data);
⑤ if (data.rollno == -1)
{

accept data
temp_data.chain = NULL
fp.seekp (addr * sizeof (temp_data), ios::beg);
fp.write ((char *) & temp_data);
go to ⑨

}
else

addr1 = hash_key (temp1_data.rollno)
⑥ if (addr1 == addr)
i = addr;
call append
go to ⑨
⑦ i = addr, cnt = 0
while (temp_data.roll = addr)
i = (i+1) % size
fp.seekg (i * sizeof (temp_data), ios::beg);
fp.read (char *) & temp_data);
cnt++;

⑧ if (hash_key (temp_data.roll) = addr)
call append_chain
goto ⑨
else if (temp_data.rollno <= 0)
{
accept temp_data.name, mark1, mark2, mark3;
temp → data.chain = NULL;
fp.seekp (i * sizeof (temp_data), ios::beg);
fp.write ((char *) & temp_data), size of (temp_data)
}
else
display ("Table overflow")
⑨ Accept reply
⑩ if (reply = 'Y' || reply = 'y')
go to ③
⑪ fp.close()

end procedure.

⑫ append_chain()

i = addr;
while (temp_data.rollno != temp_data.rollno) and
temp_data.chain != NULL)
{
i = temp_data.chain;
fp.seekg (i * sizof (temp_data), ios::beg);
fp.read (char *) & temp_data, sizof (temp_data));
}

```
if (tmp_data.rollno == tmp_data.rollno)
{
    display ("Record exists in direct access file");
    goto *(2)
}
prev_syn_pos = i;
while (temp1 -> data.rollno > 0)
{
    i = (i+1) % size;
    fp.seekg (i * sizeof (temp_data), ios::beg);
    fp.read ((char*) & temp_data, sizeof (temp_data));
}
accept marks1, marks2, marks3
temp_data.chain = NULL;
fseek (fp, i * sizeof (stud), 0);
fwrite (& temp_data, sizeof (stud), 1, fp);
fseek (fp, prev_syn_pos * sizeof (stud), 0);
fread (& temp_data, sizeof (stud), 1, fp)
temp_data.chain = 1;
return.

end procedure.
```

(iii) Display chain

```
while (tmp_data.roll != tmp1_data.roll)
        i = tmp_data.chain;
        fp.seekg (i * sizeof (tmp_data));
        fp.read (char (char *) & tmp_data);
if (tmp_data.roll == tmp1_data.roll)
        display record
else
        display "Does not exist"
return;
end procedure.
```

(iv) Updat chain

```
While (tmp1_data.roll != temp_data.roll)
        i = tmp1_data.chain
        fp.seekg (i * sizeof (tmp_data));
        fp.read ((char *) & tmp_data);

if (temp1 - data.roll = tmp_data.roll)
        display;
        accept new data
        tmp_data.chain = tmp1_data.chain
        fp.seekg (i * sizeof (tmp_data));
        fp.write ((char *) & tmp_data);
else
        Display " Record does not exist "
return
end procedure.
```

Conclusion: In Direct file organization records can be added randomly and access to any record in a file requires constant time. It is useful for ~~inde~~ interactive and on-line applications.