Real Time Systems, January 2024

Implementation of a typical program involving file related system calls

Opened: Wednesday, 6 March 2024, 12:00 AM
Due: Wednesday, 13 March 2024, 10:59 PM

✔ Done

Management of Student data in terms of "data" file and "index file"

In this assignment you are required to manage (add, delete, search, modify, compact, etc., as explained later) student data stored in files. Information about a single student can be accommodated in the following C language structure

```
struct student {
        int roll; /* storing the roll number of the student */
        char *fname; /* storing the first name of the student */
        char *mname; /* storing the middle name of the student */
        char *sname; /* storing the surname of the student */
        char *desc /* storing description of the student */
};
```

storing a student record having following attributes (RDBMS style):

1. ***roll: integer***
2. ***fname: varchar[50]***
3. ***mname: varchar[50]***
4. ***sname: varchar[50]***
5. ***desc: blob***

Let there be a file "**student.data**" which contains records of a number of students as shown below.

student.data file

File Contents	Remarks
<number of records, say **n**>	***sizeof(int)*** bytes
Student Record 1	some bytes
Student Record 2	some bytes
. . .	. . .
Student Record **n**	some bytes

Please note that different records (having different students' complete information) would require different amount of space (bytes) in this file.

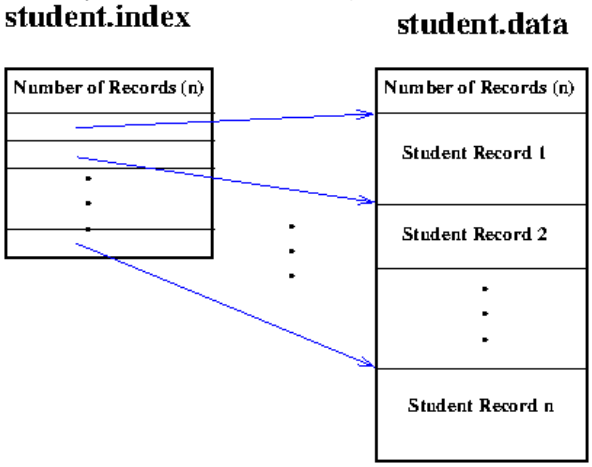
Another file "**student.index**" stores the position of the student records as they are present in "**student**.data" file. Contents of **"student.index"** file are depicted in the following table.

student.index file

File Contents	Remarks
<number of records, say **n**>	***sizeof(int)*** bytes

File Contents	Remarks
Position (i.e. offset) of Student Record 1 (as it appears in "**student.data**" file.)	***sizeof(off_t)*** bytes [Note that the type of the variables storing ***offset*** is specified to be "off_t". Please refer to functions like "**off_t** lseek(**int** fd, **off_t** offset, **int** ref)" which repositions the offset of the open file associated with the file descriptor fd.]
offset of Student Record 2	***sizeof(off_t)*** bytes
. . .	. . .
offset of Student Record **n**	***sizeof(off_t)*** bytes

Pictorially the whole scheme can be presented as shown in the following figure.



Please note that the records in "**student.index**" file are of same size (each of ***sizeof(off_t)*** bytes).

Hence the offset of the i^{th} student record (in "**student.data**" file) is available at **sizeof(int)** + (***i***-1)****sizeof(off_t)*** position (offset) in "**student.index**" file.

You have to write functions to support operations like

1. Addition of a student record to
2. Deletion of a student record
3. Searching of a student record
4. Modifying a student record
5. Compaction of student.data and student.index files. [Please note that deletions of student records may produces holes in these two files. Compaction removes those holes.

Please make your own assumptions wrt the signature (name, parameters, return type) of these functions and how exactly they should perform their jobs.

Write a main function to demonstrate that your functions are working as desied.

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading
Grading status	Not graded
Time remaining	Assignment was submitted 7 days late
Last modified	Wednesday, 20 March 2024, 11:52 PM

File submissions

Readme.txt	20 March 2024, 11:52 PM
student.c	20 March 2024, 11:52 PM
student.data	20 March 2024, 11:52 PM
student.index	20 March 2024, 11:52 PM

Submission comments

Comments (0)

◄ Implementation of a program involving file and directory related system calls

Jump to...

Filesystem Fundamentals ►