

```
1  /*
2  * DBMS Implementation
3  */
4
5  #ifndef RECORDPTR_H
6  #define RECORDPTR_H
7
8  #include <sys/types.h>
9  #include "dbmsproj.h"
10
11 // location of a specific record in the buffer
12 // block is the index number of the block the record is in
13 // record is the index number of the record in the entries table
14
15 typedef struct {
16     unsigned int block;
17     unsigned int record;
18 } recordPtr;
19
20 // struct for making linked lists of recordPtrs
21
22 struct linkedRecordPtr {
23     recordPtr ptr;
24     linkedRecordPtr *next;
25 };
26
27 //overloading operators for use with struct recordPtr
28 inline bool operator==(const recordPtr &ptr1, const recordPtr &ptr2) {
29     if (ptr1.block==ptr2.block && ptr1.record==ptr2.record){
30         return true;
31     }
32     else return false;
33 };
34 inline bool operator<(const recordPtr &ptr1, const recordPtr &ptr2) {
35     if (ptr1.block == ptr2.block) {
36         if (ptr1.record < ptr2.record)
37         {
38             return true;
39         }
40         else
41         {
42             return false;
43         }
44     }
45     else
46     {
47         if (ptr1.block < ptr2.block)
48         {
49             return true;
50         }
51         else
52         {
53             return false;
54         }
55     }
56 }
57
58 inline bool operator<=(const recordPtr &ptr1, const recordPtr &ptr2)
59 {
60     if (ptr1 < ptr2 || ptr1 == ptr2) { return true; }
```

```
61     else return false;
62 }
63
64 inline recordPtr newPtr(recordPtr ptr,unsigned int offset,unsigned int size) {
65     recordPtr result;
66     result.block = ptr.block + offset / size;
67     int rest = offset % size;
68
69     if (ptr.record + rest >= size) {
70         result.block += 1;
71         result.record = ptr.record + rest - size;
72     }
73     else {
74         result.record = ptr.record + rest;
75     }
76     return result;
77 }
78
79 inline recordPtr newPtr(unsigned int offset,unsigned int size) {
80     recordPtr zero;
81     zero.block = 0;
82     zero.record = 0;
83     return newPtr(zero,offset,size);
84 }
85
86 // increases the ptr so that it points to the next record
87 // if pointing at the end of a block, moves to the start of the next
88
89 inline void incr(recordPtr &ptr,unsigned int size) {
90     if (ptr.record < size - 1) {
91         ptr.record += 1;
92     } else {
93         ptr.record = 0;
94         ptr.block += 1;
95     }
96 }
97
98 // decreases the ptr so that it points to the next record
99 // if pointing at the start of a block, moves to the end of the previous
100
101 inline void decr(recordPtr &ptr,unsigned int size) {
102     if (ptr.record > 0) {
103         ptr.record -= 1;
104     } else if (ptr.block > 0) {
105         ptr.record = size - 1;
106         ptr.block -= 1;
107     }
108 }
109
110 #endif
```