Exercise 11: Files

Nanda H Krishna

12 April 2018

1 Read from file

Problem description:

1. We have `telephone.in`, a file. It is a sequence of lines. Each line has two fields, separated by `|` character.
   (a) Number
   (b) Name
2. Define `Entry` as a structure composed of `number` and `name`. Define an array of pointers to `Entry` structures.
3. Write a function `read_telephones()` that reads the file and converts each line to an `Entry` structure, and stores them in the array of pointers to `Entry` structures.
4. Write a function `print_entries()` to display the entries in the array to `stdout`.

Program:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAXLEN 100
#define N 100
struct entry {
  int number;
  char name[MAXLEN];
};
typedef struct entry Entry;
int read_entries (Entry* e[]);
void print_entries(Entry* e[], int n);
int main ()
{
```

```
  Entry* telephones[N];
  int n;
  n = read_entries (telephones);
  print_entries(telephones,n);
  return 0;
}
void print_entries(Entry* e[], int n)
{
  for(int i=0;i<n;i++)
      printf("%s\t%d\n",e[i]->name,e[i]->number);
}
int read_entries (Entry* e[])
{
  FILE* fp;
  int i = 0;
  char line[MAXLEN];
  char* name;
  char* number;
  fp = fopen("telephone1.in", "r");
  for (i = 0; fgets(line, MAXLEN, fp) != NULL; i++) {
    name = strtok(line, "|");
    number = strtok(NULL, "|");
    e[i] = (Entry*)malloc(sizeof(Entry));
    e[i]->number = atoi(number);
    strcpy(e[i]->name,name);
  }
  fclose(fp);
  return i;
}
```

Output:

```
JYOTHISHMATHI C V            217104066
KAILASHWAR N                 217104067
KANDAVEL A                   217104068
KANISHQ S                    217104069
KARAN D                      217104070
KARTHIKEYAN R                217104071
KARTHIK VISWANATH S          217104072
KAVITHA A                    217104073
```

KAVYA J	217104074
KEERTHIVASAN RAJAVADIVEL	217104075
KEVIN J THELLY	217104076
KISHORE S M	217104077
KRIJESHAN G	217104078
KRISHNAKANTH E	217104079
KUMAR H	217104080
LAKSHMI NARASIMHAN R	217104081
LOKESH S	217104082
MALAVIKA T	217104083
MANISHA L	217104084
MANO BALAJE S	217104085
MITHUMARY C M	217104086
MOHAMED MUSARAF P M	217104087
MONIKA N	217104088
MOURIESH S K	217104089
MUSUNURU YASASWI	217104090
NACHIAPPAN N N	217104091
NAKUL KRISHNAN	217104092
NANDA H KRISHNA	217104093
NANDHINI R	217104094
NARESH KUMAR R	217104095
NAVEENA M	217104096
NAVEEN NARAYANAN	217104097
NIMISH S	217104098
NITIN NIKAMANTH A B	217104099
PAVILA V	217104100
PAVITHRA N	217104101
PAVYA S	217104102
POOJA S (29.12.1999)	217104103
POOJA S (11.06.2000)	217104104
PRADEEP KUMAR B	217104105
PRAGATHEESHWARI JAYASANKER	217104106
PRAGNA REDDY N	217104107
PRANATHY M S	217104108
PRANAVI SHEKHAR	217104109
PRANAV RAVEENDRAN	217104110
PRANAV VIJAY	217104111
PRATHEEP S	217104112

```
PRATHISH E                    217104113
PRAVEEN KUMAR R               217104114
PREETHI S (04.11.1999)        217104115
PREETHI S (25.11.1999)        217104116
PRIYA J                       217104117
PRIYADHARSHINI N              217104118
RAGHUL P                      217104119
RAHUL V                       217104120
RAJESH R                      217104121
RAJESWARA RAJAN M             217104122
RAKESH M                      217104123
RAKSHANAA R                   217104124
RAMKAUSHIK R                  217104125
RAMYA NIVASINI U S            217104126
RANJANA S                     217104127
REENU RITA P S                217104128
RESHMA RAMESH BABU            217104129
RIYA RAJU                     217104130
```

2 Search for an entry

2.1 Number-wise

Problem description: Define a function `search_number()` that searches for a given number and prints the number and the name.

Function:

```
int search_number(int p,Entry* e[],int n)
{
  for(int i = 0; i < n; i++) {
      if(e[i]->number==p)
{
  return i;
}
  }
  return n;
}
```

2.2 Name-wise

Problem description: Define a function `search_name()` that searches for a given name and prints the number and the name. You can search for a substring using the library function `strstr()`.

Function:

```
int search_name(char p[],Entry* e[],int n)
{
  for(int i = 0; i < n; i++) {
      if(strcmpi(e[i]->name == p) == 0)
  return i;
  }
 return n;
}
```

3 Insert an entry

Problem description: Write a function `insert_entry()` that reads a name and number from the user and adds it to the array. If the number already exists, it should not be inserted.

Function:

```
int insert(Entry p, Entry* e[], int* n)
{
  int pos = search_number(p.number,e,*n);
  if(pos<*n) {
      e[*n]=(Entry*)malloc(sizeof(Entry));
      for(int i=(*n)-1;i>=pos;i--)
  e[i+1]=e[i];
      e[pos]->number = p.number;
      strcpy(e[i]->name,p.name);
      (*n)++;
      return 1;
  }
  return 0;
}
```

4 Delete an entry

Problem description: Write a function `delete_entry()` that reads a number from the user and deletes it from the array. When you delete an entry, pack the array by moving the subsequent entries up. **Function:**

```
int delete(int p,Entry* e[],int* n)
{
  int pos = search_number(p, e, *n);
  if(pos < (*n)) {
      for(int i=pos;i<n-1;i++)
{
  e[i]=e[i+1];
}
      (*n)--;
      return 1;
  }
  return 0;
}
```

5 Interactive loop

Problem description: Write a loop that interacts with the user: It reads one of the options from the user and performs the function.

```
q quit
s number (search for a number)
f name (search for a name)
i number name (insert an entry (number, name))
d number (delete the entry with the number)
```

When the program quits, the array of `Entry` structures should be written to the `telephone.in` file. **Program:**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAXLEN 100
#define N      100
struct entry {
  int number;
```

```
  char name[MAXLEN];
};
typedef struct entry Entry;
int read_entries (Entry* e[]);
Entry* get_node (char* name, char* number);
void print_entries (Entry* e[], int n);
int write_entries (Entry* e[], int n);
void print_entry(Entry* e[],int p);
int search_number(int p,Entry* e[],int n);
int insert(Entry p,Entry* e[],int* n);
int delete(int p,Entry* e[],int* n);

int main ()
{
  Entry* telephones[N];
  int n;
  char line[MAXLEN];
  char choice;
  char name[MAXLEN];
  int number,res;
  n = read_entries (telephones);
  while(1) {
      printf("? ");
      fgets (line, MAXLEN, stdin);
      choice = line[0];
      switch (choice) {
case 's':
  sscanf (line+1, "%d", &number);
  int pos=search_number(number,telephones,n);
  if(pos<n)
     print_entry(telephones,pos);
  else
     printf("No such record\n");
  break;
case 'i':
  sscanf (line+1, "%d%s", &number, name);
  Entry p;
  p.number=number;
  strcpy(p.name,name);
```

```
  res= insert(p,telephones,&n);
  if(res==0)
       printf("Record already exists\n");
  break;
case 'd':
  sscanf (line+1, "%d", &number);
  res=delete(number,telephones,&n);
  break;
case 'q':
  write_entries(telephones, n);
  print_entries(telephones,n);
  return 0;
       default:
  printf("Invalid choice\n");
    }
  }
return 0;
}
void print_entry(Entry* e[],int p)
{
  printf("%s\t %d\n",e[p]->name,e[p]->number);
}
void print_entries (Entry* e[], int n)
{
  for (int i = 0; i < n; i++)
    printf ("%d,%s,%d\n", i, e[i]->name, e[i]->number);
}
int read_entries (Entry* e[])
{
  FILE* fp;
  int i;
  char line[MAXLEN];
  char* name;
  char* number;
  fp = fopen ("telephone.in", "r");
  for (i = 0; fgets(line, MAXLEN, fp) != NULL; i++)
    {
      name = strtok (line, "|");
      number = strtok (NULL, "|");
```

```
      e[i] = get_node (name, number);
    }
  fclose(fp);
  return i;
}
Entry* get_node (char* name, char* number)
{
  Entry* t = (Entry*) malloc (sizeof(Entry));
  strcpy(t->name, name);
  t->number = atoi(number);
  return t;
}
int write_entries (Entry* e[], int n)
{
  int i;
  FILE* fp;

  fp = fopen ("telephone.in", "w");
  for (i = 0; i < n; i++)
    fprintf(fp, "|%s|%d|\n", e[i]->name, e[i]->number);
  fclose (fp);
}
int search_number(int p,Entry* e[],int n)
{
  for(int i=0;i<n;i++)
    {
      if(e[i]->number==p)
{
  return i;
}
    }
  return n;
}
int insert(Entry p,Entry* e[],int* n)
  {
    int pos=search_number(p.number,e,*n);
    if(pos==*n)
      {
e[*n]=(Entry*)malloc(sizeof(Entry));
```

```
e[pos]->number=p.number;
strcpy(e[pos]->name,p.name);
(*n)++;
return 1;
      }
    return 0;
  }
 int delete(int p,Entry* e[],int* n)
 {
     int pos=search_number(p,e,*n);
     if(pos<(*n))
     {
for(int i=pos;i<(*n)-1;i++)
{
   e[i]=e[i+1];
}
(*n)--;
return 1;
     }
     return 0;
 }
```

Output:

```
? s 217104114
PRAVEEN KUMAR R  217104114
? s 217104131
No such record
? i 217104131 JAYARAMAN
? i 217104093 NANDA
Record already exists
? s 217104131
JAYARAMAN  217104131
? d 217104131
? s 217104131
No such record
? q
0,JYOTHISHMATHI C V,217104066
1,KAILASHWAR N,217104067
2,KANDAVEL A,217104068
```

3,KANISHQ S,217104069
4,KARAN D,217104070
5,KARTHIKEYAN R,217104071
6,KARTHIK VISWANATH S,217104072
7,KAVITHA A,217104073
8,KAVYA J,217104074
9,KEERTHIVASAN RAJAVADIVEL,217104075
10,KEVIN J THELLY,217104076
11,KISHORE S M,217104077
12,KRIJESHAN G,217104078
13,KRISHNAKANTH E,217104079
14,KUMAR H,217104080
15,LAKSHMI NARASIMHAN R,217104081
16,LOKESH S,217104082
17,MALAVIKA T,217104083
18,MANISHA L,217104084
19,MANO BALAJE S,217104085
20,MITHUMARY C M,217104086
21,MOHAMED MUSARAF P M,217104087
22,MONIKA N,217104088
23,MOURIESH S K,217104089
24,MUSUNURU YASASWI,217104090
25,NACHIAPPAN N N,217104091
26,NAKUL KRISHNAN,217104092
27,NANDA H KRISHNA,217104093
28,NANDHINI R,217104094
29,NARESH KUMAR R,217104095
30,NAVEENA M,217104096
31,NAVEEN NARAYANAN,217104097
32,NIMISH S,217104098
33,NITIN NIKAMANTH A B,217104099
34,PAVILA V,217104100
35,PAVITHRA N,217104101
36,PAVYA S,217104102
37,POOJA S (29.12.1999),217104103
38,POOJA S (11.06.2000),217104104
39,PRADEEP KUMAR B,217104105
40,PRAGATHEESHWARI JAYASANKER,217104106
41,PRAGNA REDDY N,217104107

42,PRANATHY M S,217104108
43,PRANAVI SHEKHAR,217104109
44,PRANAV RAVEENDRAN,217104110
45,PRANAV VIJAY,217104111
46,PRATHEEP S,217104112
47,PRATHISH E,217104113
48,PRAVEEN KUMAR R,217104114
49,PREETHI S (04.11.1999),217104115
50,PREETHI S (25.11.1999),217104116
51,PRIYA J,217104117
52,PRIYADHARSHINI N,217104118
53,RAGHUL P,217104119
54,RAHUL V,217104120
55,RAJESH R,217104121
56,RAJESWARA RAJAN M,217104122
57,RAKESH M,217104123
58,RAKSHANAA R,217104124
59,RAMKAUSHIK R,217104125
60,RAMYA NIVASINI U S,217104126
61,RANJANA S,217104127
62,REENU RITA P S,217104128
63,RESHMA RAMESH BABU,217104129
64,RIYA RAJU,217104130
65,JAYARAMAN,204104131