/\* Implementation of a direct access file -Insertion and deletion of a record from a direct access file \*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define size 200

struct emp

{

int id;

char \*name;

}\*emp1, \*emp3;

void display();

void create();

void update();

char \* arrr = "/home/ubuntu/suzume/StuRecord.txt";

FILE \*fp, \*fp1;

int count = 0;

int main()

{

int i, n, ch;

printf("1] Create a Record\n");

printf("2] Display Records\n");

printf("3] Update Records\n");

printf("4] Exit");

while (1)

{

printf("\nEnter your choice : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

fp = fopen(arrr, "a");

create();

break;

case 2:

fp1 = fopen(arrr,"rb");

display();

break;

case 3:

fp1 = fopen(arrr, "r+");

update();

break;

case 4:

exit(0);

}

}

}

/\* To create an employee record \*/

void create()

{

int i;

char \*p;

emp1 = (struct emp \*)malloc(sizeof(struct emp));

emp1->name = (char \*)malloc((size)\*(sizeof(char)));

printf("Enter name of employee : ");

scanf(" %[^\n]s", emp1->name);

printf("Enter emp id : ");

scanf(" %d", &emp1->id);

fwrite(&emp1->id, sizeof(emp1->id), 1, fp);

fwrite(emp1->name, size, 1, fp);

count++; // count to number of entries of records

fclose(fp);

}

/\* Display the records in the file \*/

void display()

{

emp3=(struct emp \*)malloc(1\*sizeof(struct emp));

emp3->name=(char \*)malloc(size\*sizeof(char));

int i = 1;

if (fp1 == NULL)

printf("\nFile not opened for reading");

while (i <= count)

{

fread(&emp3->id, sizeof(emp3->id), 1, fp1);

fread(emp3->name, size, 1, fp1);

printf("\n%d %s",emp3->id,emp3->name);

i++;

}

fclose(fp1);

free(emp3->name);

free(emp3);

}

void update()

{

int id, flag = 0, i = 1;

char s[size];

if (fp1 == NULL)

{

printf("File cant be opened");

return;

}

printf("Enter employee id to update : ");

scanf("%d", &id);

emp3 = (struct emp \*)malloc(1\*sizeof(struct emp));

emp3->name=(char \*)malloc(size\*sizeof(char));

while(i<=count)

{

fread(&emp3->id, sizeof(emp3->id), 1, fp1);

fread(emp3->name,size,1,fp1);

if (id == emp3->id)

{

printf("Enter new name of emplyee to update : ");

scanf(" %[^\n]s", s);

fseek(fp1, -204L, SEEK\_CUR);

fwrite(&emp3->id, sizeof(emp3->id), 1, fp1);

fwrite(s, size, 1, fp1);

flag = 1;

break;

}

i++;

}

if (flag != 1)

{

printf("No employee record found");

flag = 0;

}

fclose(fp1);

free(emp3->name); /\* to free allocated memory \*/

free(emp3);

}

/\*

1] Create a Record

2] Display Records

3] Update Records

4] Exit

Enter your choice : 1

Enter name of employee : abc

Enter emp id : 101

Enter your choice : 2

101 abc

Enter your choice : 3

Enter employee id to update : 101

Enter new name of emplyee to update : qqq

Enter your choice : 2

101 qqq

Enter your choice : 4 \*/

MADHUR CODE:

/\*

Madhur Jaripatke

SE A S3

Roll No. 55

\*/

/\*

Implementation of a direct access file -Insertion and deletion of a record from a direct access file

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define size 200

struct emp

{

int id;

char \*name;

}\*emp1, \*emp3;

void display();

void create();

void update();

char \* arrr = "/home/madhur/StudentRecord.txt";

FILE \*fp, \*fp1;

int count = 0;

int main()

{

int i, n, ch;

printf("1] Create a Record\n");

printf("2] Display Records\n");

printf("3] Update Records\n");

printf("4] Exit");

while (1)

{

printf("\nEnter your choice : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

fp = fopen(arrr, "a");

create();

break;

case 2:

fp1 = fopen(arrr,"rb");

display();

break;

case 3:

fp1 = fopen(arrr, "r+");

update();

break;

case 4:

exit(0);

}

}

}

/\* To create an employee record \*/

void create()

{

int i;

char \*p;

emp1 = (struct emp \*)malloc(sizeof(struct emp));

emp1->name = (char \*)malloc((size)\*(sizeof(char)));

printf("Enter name of employee : ");

scanf(" %[^\n]s", emp1->name);

printf("Enter emp id : ");

scanf(" %d", &emp1->id);

fwrite(&emp1->id, sizeof(emp1->id), 1, fp);

fwrite(emp1->name, size, 1, fp);

count++; // count to number of entries of records

fclose(fp);

}

/\* Display the records in the file \*/

void display()

{

emp3=(struct emp \*)malloc(1\*sizeof(struct emp));

emp3->name=(char \*)malloc(size\*sizeof(char));

int i = 1;

if (fp1 == NULL)

printf("\nFile not opened for reading");

while (i <= count)

{

fread(&emp3->id, sizeof(emp3->id), 1, fp1);

fread(emp3->name, size, 1, fp1);

printf("\n%d %s",emp3->id,emp3->name);

i++;

}

fclose(fp1);

free(emp3->name);

free(emp3);

}

void update()

{

int id, flag = 0, i = 1;

char s[size];

if (fp1 == NULL)

{

printf("File cant be opened");

return;

}

printf("Enter employee id to update : ");

scanf("%d", &id);

emp3 = (struct emp \*)malloc(1\*sizeof(struct emp));

emp3->name=(char \*)malloc(size\*sizeof(char));

while(i<=count)

{

fread(&emp3->id, sizeof(emp3->id), 1, fp1);

fread(emp3->name,size,1,fp1);

if (id == emp3->id)

{

printf("Enter new name of emplyee to update : ");

scanf(" %[^\n]s", s);

fseek(fp1, -204L, SEEK\_CUR);

fwrite(&emp3->id, sizeof(emp3->id), 1, fp1);

fwrite(s, size, 1, fp1);

flag = 1;

break;

}

i++;

}

if (flag != 1)

{

printf("No employee record found");

flag = 0;

}

fclose(fp1);

free(emp3->name);

free(emp3);

}