EMPLOYEE MANAGEMENT SYSTEM

Subhajit Nath
Computer Science and Engineering, 3Rd Year
09/2022

Introduction

Day by day we want to work faster and also see that our working time and money are spent less. For these reasons we use different technologies. By using this employee management system, we can get more profit in less time with less employees. It is nothing but a software written on C program.

Objective

The main objective of this project is to digitally manage employee data. If we store the data digitally in the computer, we can find it very easily in less time. It is very easy and quick to operate, due to which we need a smaller number of workers to do this work resulting in more profit.

Benefits

- *It is completely digital.*
- *No document needs to be written in the ledger.*
- *It is possible to do data entry in very less time.*
- *Data can be found very easily.*
- *An entire industry's employee data can be managed by just one person.*
- *It is also possible to know when a worker joined the work at any time.*

Required

Hardware: *Only One Computer.*

Coding: *The step is to design the software. Software programming using the IDE (Integrated Development Environment) before starting programming C compiler settings in IDE through the tools menu and adjusting. The program code below is specific to the employee management system.*

```c
#include <assert.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>

struct Employee_Details
{
  int Id;
  char FirstName[40];
  char LastName[30];
  char DOB[12];
  char Mobile[15];
  char Mail[60];
  char Aadhar_No[14];
  char Joining[12];
  int Pin;
} ED;

struct Attendance_Employee
{

  int Id;
  char Attendance_Date[64];
  char Date[12];
} AE;

FILE *File_Pointer;

void New_Employee_Data_Entry();
void All_Employee_list();
void Remove_Employee();
void Employee_All_Details();
void Attendance_Employee();
void Employee_Attendance_List();

int main(int argc, char const *argv[])
{
  int Ch;
```

```c
while (1)
{
  system("cls");
  printf("<====  Employee Managemnt System  ====>\n\n");
  printf("1. New Employee Data Entry.\n");
  printf("2. All Employee list.\n");
  printf("3. Remove Employee Using ID.\n");
  printf("4. Search Employee Full Details Using ID.\n");
  printf("5. Employee Attendance. \n");
  printf("6. All Attendance Date & Time. \n");
  printf("0. Exit. \n");

  printf("Enter your choice: ");
  scanf("%d", &Ch);

  switch (Ch)
  {
  case 0:
    exit(0);
  case 1:
    New_Employee_Data_Entry();
    break;

  case 2:
    All_Employee_list();
    break;

  case 3:
    Remove_Employee();
    break;
  case 4:
    Employee_All_Details();
    break;
  case 5:
    Attendance_Employee();
    break;
  case 6:
    Employee_Attendance_List();
    break;
  default:
```

```
      break;
    }
    printf("Press Any Key To Continue!.....");
    getch();
  }

  return 0;
}

void New_Employee_Data_Entry()
{
  int id;
  system("cls");
table:
  printf("<==    New Entry    ==>\n\n");

  printf("\n Create Employee Id: ");
  scanf("%d", &id);

  File_Pointer = fopen("Employee_Details.txt", "rb");

  while (fread(&ED, sizeof(ED), 1, File_Pointer) == 1)
  {
    if (ED.Id == id)
    {
      printf("Id Is Allrady Exiet \n");

      goto table;
    }
  }

  fclose(File_Pointer);

  File_Pointer = fopen("Employee_Details.txt", "ab");
  ED.Id = id;
  char myDate[12];
  time_t t = time(NULL);
  struct tm tm = *localtime(&t);
  sprintf(myDate, "%02d/%02d/%d", tm.tm_mday, tm.tm_mon + 1,
tm.tm_year + 1900);
```

```
  strcpy(ED.Joining, myDate);

  printf("Full Name: ");
  scanf("%s %s", ED.FirstName, ED.LastName);

  printf("DOB(dd/mm/yyyy): ");
  scanf("%s", ED.DOB);

  printf("Mobile Number: ");
  scanf("%s", ED.Mobile);

  printf("Email Id: ");
  scanf("%s", ED.Mail);

  printf("Aadhar Number(12 Digit): ");
  scanf("%s", ED.Aadhar_No);

  printf("Pin: ");
  scanf("%d", &ED.Pin);

  fwrite(&ED, sizeof(ED), 1, File_Pointer);
  fclose(File_Pointer);

  printf("<===    Id Added Successfully    ===> \n\n");
}

void All_Employee_list()
{
  system("cls");
  printf("<==  All Employee  ==>\n\n");
  File_Pointer = fopen("Employee_Details.txt", "rb");
printf("%-30s %-40s %s\n", "ID", "Name", "Mobile Number");
  while (fread(&ED, sizeof(ED), 1, File_Pointer) == 1)
  {
    printf(" %-30d%s %-30s  %s \n", ED.Id, ED.FirstName, ED.LastName,
ED.Mobile);
  }

  fclose(File_Pointer);
}
```

```c
void Remove_Employee()
{
  system("cls");
  printf("<==  Remove Employee Details  ==>\n\n");
  int id, f = 0;
  system("cls");
  printf("\n Id: ");
  scanf("%d", &id);
  FILE *ft;
  File_Pointer = fopen("Employee_Details.txt", "rb");
  ft = fopen("temp.txt", "wb");

  while (fread(&ED, sizeof(ED), 1, File_Pointer) == 1)
  {
    if (ED.Id == id)
    {
      f = 1;
    }
    else
    {
      fwrite(&ED, sizeof(ED), 1, ft);
    }
  }
  fclose(File_Pointer);
  fclose(ft);

  remove("Employee_Details.txt");
  rename("temp.txt", "Employee_Details.txt");

  if (f == 1)
  {
    printf("\n\n <===   Deleted Successfully.   ===> \n\n");
  }
  else
  {
    printf("\n\n<===   Record Not Found!   ===>\n\n");
  }
}
```

```c
void Employee_All_Details()
{
  int id, a = 0;

  system("cls");
  printf("<== Employee All Details ==>\n\n");
  printf("\n  Id: ");
  scanf("%d", &id);
  system("cls");
  File_Pointer = fopen("Employee_Details.txt", "rb");

  while (fread(&ED, sizeof(ED), 1, File_Pointer) == 1)
  {
    if (ED.Id == id)
    {
      printf("Id: %d \n", ED.Id);

      printf("Full Name:  %s %s \n", ED.FirstName, ED.LastName);

      printf("DOB(dd/mm/yyyy):  %s \n", ED.DOB);

      printf("Mobile Number:  %s \n", ED.Mobile);

      printf("Email Id:  %s \n", ED.Mail);

      printf("Aadhar Number(10 Digit):  %s \n", ED.Aadhar_No);

      printf("Pin: %d \n", ED.Pin);

      a = 1;
    }
  }

  fclose(File_Pointer);

  if (a == 0)
  {
    printf("\n\n<===   Record Not Found!   ===>\n\n");
  }
}
```

```
void Attendance_Employee()
{
  system("cls");
  int id, f;
  int result;

  time_t tt = time(NULL);
  struct tmm *tmm = localtime(&tt);
  char s[64];
  size_t ret = strftime(s, sizeof(s), "%c", tmm);
  assert(ret);

  AE.Id = id;
  char myDate[12];
  time_t t = time(NULL);
  struct tm tm = *localtime(&t);
  sprintf(myDate, "%02d/%02d/%d", tm.tm_mday, tm.tm_mon + 1,
tm.tm_year + 1900);

  system("cls");
  printf("<===   ATTENDANCE   ===>\n");
  f = 0;
  printf("Enter Id: ");
  scanf("%d", &id);

  File_Pointer = fopen("Employee_Details.txt", "rb");
  while (fread(&ED, sizeof(ED), 1, File_Pointer) == 1)
  {

    if (id == ED.Id)
    {
      printf("Id: %d\n", ED.Id);
      printf("Name: %s %s\n", ED.FirstName, ED.LastName);
      printf("DOB (dd/mm/yyyy): %s\n", ED.DOB);
      f = 1;
    }
  }

  fclose(File_Pointer);
```

```
File_Pointer = fopen("Employee_Attendance.txt", "rb");

while (fread(&AE, sizeof(AE), 1, File_Pointer) == 1)
{
  if (id == AE.Id)
  {
    result = strcmp(myDate, AE.Date);
    if (result == 0)
    {
      f = 2;
    }
  }
}
fclose(File_Pointer);

if (f == 2)
{
  printf("\n<===   You Are Allrady Attend.   ===> \n\n\n");
}

if (f == 0)
{
  printf("\n\n<===   Record Not Found!   ===>\n\n");
}

if (f == 1)
{

  File_Pointer = fopen("Employee_Attendance.txt", "ab");

  AE.Id = id;
  strcpy(AE.Date, myDate);
  strcpy(AE.Attendance_Date, s);
  printf("Attend Date: %s\n", AE.Attendance_Date);

  fwrite(&AE, sizeof(AE), 1, File_Pointer);
  fclose(File_Pointer);
}
```

```
  printf("<===   ATTENDANCE Successfull   ===>\n");
}

void Employee_Attendance_List()
{

  int id, f = 0;
  // Top:
  system("cls");
  printf("<== Employee Details ==>\n\n");
  printf("Enter Employee Id: ");
  scanf("%d", &id);

  File_Pointer = fopen("Employee_Details.txt", "rb");

  while (fread(&ED, sizeof(ED), 1, File_Pointer) == 1)
  {
    if (id == ED.Id)
    {
      printf("Id: %d\n", ED.Id);
      printf("Name: %s %s\n", ED.FirstName, ED.LastName);
      printf("DOB (dd/mm/yyyy): %s\n", ED.DOB);
      f = 1;
    }
  }

  if (f == 0)
  {
    printf("\n\n<===   Record Not Found!   ===>\n\n");
  }

  fclose(File_Pointer);

  if (f == 1)
  {
    File_Pointer = fopen("Employee_Attendance.txt", "rb");
    while (fread(&AE, sizeof(AE), 1, File_Pointer) == 1)
    {
      if (id == AE.Id)
      {
```

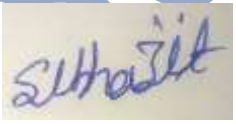
```
      printf("Attend Date: %-40s \n", AE.Attendance_Date);
    }
   }

   fclose(File_Pointer);
  }
}
```

Methodology

*After setting up the IDE, copy and paste the C program code. After pasting save with '.c' extension. Compile the program after save the code. '**.exe**' file will be generated after compiling. With this, process of creating your employee management system is complete.*

Conclusion

Time keeping of industrial employees is possible with this system at very low cost and with less manpower and by using this system we do not need to record any information ledger. Any small or large company needs it. It is managed by computer, it is less likely to make mistakes than humans.



Signature