

Report

Employee Payroll System

(Employee Payroll System Using C Programming)

Name : Sodeed ahmed

Roll Number : 68

Branch : CSE

Class : s4 CSE-B

Date : 15-07-2024

Introduction

Brief Overview of the Project

The Employee Payroll System is a C programming project designed to manage employee details, calculate salaries, handle allowances and deductions, and generate pay slips. The system also includes functionalities to save data to a file and read from it, ensuring data persistence.

Problem Statement

Manual payroll management is time-consuming and prone to errors. Automating the payroll process can significantly enhance accuracy and efficiency, reducing the workload on HR personnel.

Objective

To develop a C program that automates the payroll management process, including adding employee details, calculating salaries, generating pay slips, and saving/loading data from a file.

System Requirements

Hardware Requirements

- A computer with at least 1 GB of RAM
- A processor with a minimum speed of 1 GHz
- Sufficient storage to save employee data files

Software Requirements

- An operating system (Windows, Linux, or macOS)
- A C compiler (GCC, Clang, or similar)
- A text editor or IDE (Code::Blocks, Visual Studio Code, etc.)

Design and Development

Description of the Program Logic

The program uses a struct to store employee details, including ID, name, base salary, allowances, deductions, and net salary. The main menu offers options to add an employee, generate pay slips, display all employees, and save and exit. Data is stored in a binary file to ensure persistence.

Key Functions:

1. **addEmp:** Adds a new employee to the system.
2. **calcSal:** Calculates the net salary of an employee.
3. **genPaySlip:** Generates and displays a pay slip for a given employee.
4. **dispEmps:** Displays details of all employees.
5. **saveEmpsToFile:** Saves employee data to a file.
6. **loadEmpsFromFile:** Loads employee data from a file.

Flowchart or Pseudocode

Pseudocode:

START

Initialize employee array and count

Load employees from file

WHILE true

Display menu

Get user choice

```
IF choice == 1
    Add employee
ELSE IF choice == 2
    Generate pay slip for given employee ID
ELSE IF choice == 3
    Display all employees
ELSE IF choice == 4
    Save employees to file
    Exit program
ELSE
    Display invalid choice message
ENDIF

Clear input buffer
ENDWHILE

STOP
```

Testing and Results

Test Cases

- 1. Add Employee:**
 - **Input:** Employee ID = 1, Name = "Akash", Base Salary = 1000000000, Allowances = 1000000, Deductions = 1
 - **Expected Output:** Employee added successfully, net salary calculated correctly.
- 2. Generate Pay Slip:**
 - **Input:** Employee ID = 1
 - **Expected Output:** Pay slip displayed with correct details.
- 3. Display All Employees:**
 - **Input:** 3
 - **Expected Output:** List of all employees with correct details.
- 4. Save and Exit:**
 - **Input:** 4
 - **Expected Output:** Employee data saved to file, program exits.

Output Screenshots or Results

Add Employee:

```
Employee Payroll System
1. Add Employee
2. Generate Pay Slip
3. Display All Employees
4. Save and Exit
Enter your choice: 1
Enter employee ID: 1
Enter employee name: Akash
Enter base salary: 1000000000
Enter allowances: 1000000
Enter deductions: 1
```

Generate Pay Slip:

```
Employee Payroll System
1. Add Employee
2. Generate Pay Slip
3. Display All Employees
4. Save and Exit
Enter your choice: 2
Enter employee ID for pay slip: 1

Pay Slip for Employee ID: 1
Name: Akash
Base Salary: 1000000000.00
Allowances: 1000000.00
Allowances: 1000000.00
Deductions: 1.00
Net Salary: 1001000000.00
-----
```

Display All Employees:

```
Employee Payroll System
1. Add Employee
2. Generate Pay Slip
1. Add Employee
2. Generate Pay Slip
3. Display All Employees
4. Save and Exit
Enter your choice: 3
4. Save and Exit
Enter your choice: 3
Employee ID: 1
Name: Akash
Base Salary: 1000000000.00
Base Salary: 1000000000.00
Allowances: 1000000.00
Deductions: 1.00
Net Salary: 1001000000.00
-----
Employee ID: 2
Net Salary: 1001000000.00
-----
Employee ID: 2
Name: Adarsh
Base Salary: 1000000.00
Allowances: 100000.00
Deductions: 9900.00
Net Salary: 1090100.00
-----
```

Discussion of Results

The program performs all intended functionalities correctly. The employee data is accurately added, calculated, displayed, and saved. The file handling ensures that data is preserved between sessions, validating the persistence feature.

Conclusion

Summary of the Project

The Employee Payroll System effectively automates payroll management, reducing errors and improving efficiency. The project demonstrates the practical use of C programming for data management, file handling, and user interaction.

Future Enhancements

- 1. User Authentication: Adding login features to enhance security.**
- 2. GUI Interface: Developing a graphical user interface for better user experience.**
- 3. Database Integration: Using a database system for more robust data storage and management.**
- 4. Advanced Reporting: Generating comprehensive reports and analytics for HR management.**

References

- Study Materials by Prof Smitha Jacob**

Appendices

Code Listing:

```
C Final.c  X
Microproject > C Final.c > ...
1  #include <stdio.h>
2  #include <string.h>
3
4  #define MAX_EMP 100
5  #define EMP_FILE "empdata.dat"
6
7  typedef struct {
8      int id;
9      char name[50];
10     float base_sal;
11     float allow;
12     float deduct;
13     float net_sal;
14 } Emp;
15
16 void addEmp(Emp emps[], int *num_emp);
17 void calcSal(Emp *emp);
18 void genPaySlip(Emp emp);
19 void dispEmps(Emp emps[], int num_emp);
20 void saveEmpsToFile(Emp emps[], int num_emp);
21 int loadEmpsFromFile(Emp emps[]);
22
23 void addEmp(Emp emps[], int *num_emp) {
24     if (*num_emp >= MAX_EMP) {
25         printf("Max number of employees reached.\n");
26         return;
27     }
28
29     Emp new_emp;
30     printf("Enter employee ID: ");
31     if (scanf("%d", &new_emp.id) != 1) {
32         printf("Invalid input for employee ID.\n");
33         while (getchar() != '\n'); // Clear input buffer
34         return;
35     }
36
37     printf("Enter employee name: ");
```

```

38     if (scanf("%s", new_emp.name) != 1) {
39         printf("Invalid input for employee name.\n");
40         while (getchar() != '\n'); // Clear input buffer
41         return;
42     }
43
44     printf("Enter base salary: ");
45     if (scanf("%f", &new_emp.base_sal) != 1) {
46         printf("Invalid input for base salary.\n");
47         while (getchar() != '\n'); // Clear input buffer
48         return;
49     }
50
51     printf("Enter allowances: ");
52     if (scanf("%f", &new_emp.allow) != 1) {
53         printf("Invalid input for allowances.\n");
54         while (getchar() != '\n'); // Clear input buffer
55         return;
56     }
57
58     printf("Enter deductions: ");
59     if (scanf("%f", &new_emp.deduct) != 1) {
60         printf("Invalid input for deductions.\n");
61         while (getchar() != '\n'); // Clear input buffer
62         return;
63     }
64
65     calcSal(&new_emp);
66     emps[*num_emp] = new_emp;
67     (*num_emp)++;
68 }
69
70 void calcSal(Emp *emp) {
71     emp->net_sal = emp->base_sal + emp->allow - emp->deduct;
72 }
73

```

```

74 void genPaySlip(Emp emp) {
75     printf("\nPay Slip for Employee ID: %d\n", emp.id);
76     printf("Name: %s\n", emp.name);
77     printf("Base Salary: %.2f\n", emp.base_sal);
78     printf("Allowances: %.2f\n", emp.allow);
79     printf("Deductions: %.2f\n", emp.deduct);
80     printf("Net Salary: %.2f\n", emp.net_sal);
81     printf("-----\n");
82 }
83
84 void dispEmps(Emp emps[], int num_emp) {
85     for (int i = 0; i < num_emp; i++) {
86         printf("\n");
87         printf("Employee ID: %d\n", emps[i].id);
88         printf("Name: %s\n", emps[i].name);
89         printf("Base Salary: %.2f\n", emps[i].base_sal);
90         printf("Allowances: %.2f\n", emps[i].allow);
91         printf("Deductions: %.2f\n", emps[i].deduct);
92         printf("Net Salary: %.2f\n", emps[i].net_sal);
93         printf("-----\n");
94     }
95 }
96
97 void saveEmpsToFile(Emp emps[], int num_emp) {
98     FILE *file = fopen(EMP_FILE, "wb");
99     if (file == NULL) {
100         perror("Error opening file for writing");
101         return;
102     }
103
104     fwrite(&num_emp, sizeof(int), 1, file);
105     fwrite(emps, sizeof(Emp), num_emp, file);
106
107     fclose(file);
108 }
109
110 int loadEmpsFromFile(Emp emps[]) {

```



```

111     FILE *file = fopen(EMP_FILE, "rb");
112     if (file == NULL) {
113         // File doesn't exist, initialize to 0 employees
114         return 0;
115     }
116
117     int num_emp;
118     fread(&num_emp, sizeof(int), 1, file);
119     fread(emps, sizeof(Emp), num_emp, file);
120
121     fclose(file);
122     return num_emp;
123 }
124
125 int main() {
126     Emp emps[MAX_EMP];
127     int num_emp = loadEmpsFromFile(emps);
128     int choice;
129
130     while (1) {
131         printf("\nEmployee Payroll System\n");
132         printf("1. Add Employee\n");
133         printf("2. Generate Pay Slip\n");
134         printf("3. Display All Employees\n");
135         printf("4. Save and Exit\n");
136         printf("Enter your choice: ");
137
138         if (scanf("%d", &choice) != 1) {
139             printf("Invalid input. Please enter a number.\n");
140             while (getchar() != '\n'); // Clear input buffer
141             continue;
142         }
143
144         switch (choice) {
145             case 1:

```

```

146             addEmp(emps, &num_emp);
147             break;
148             case 2: {
149                 int emp_id;
150                 printf("Enter employee ID for pay slip: ");
151                 if (scanf("%d", &emp_id) != 1) {
152                     printf("Invalid input for employee ID.\n");
153                     while (getchar() != '\n'); // Clear input buffer
154                     continue;
155                 }
156                 int found = 0;
157                 for (int i = 0; i < num_emp; i++) {
158                     if (emps[i].id == emp_id) {
159                         genPaySlip(emps[i]);
160                         found = 1;
161                         break;
162                     }
163                 }
164                 if (!found) {
165                     printf("Employee not found.\n");
166                 }
167                 break;
168             }
169             case 3:
170                 dispEmps(emps, num_emp);
171                 break;
172             case 4:
173                 saveEmpsToFile(emps, num_emp);
174                 printf("Employee data saved. Exiting...\n");
175                 return 0;
176             default:
177                 printf("Invalid choice. Please try again.\n");
178                 while (getchar() != '\n'); // Clear input buffer
179         }
180     }
181 }

```

THANKYOU