Open Ended Lab -2

Spring 2025

Course Title: Data Structures Lab

Course Code: CSE1302

Section: 4

Submitted by:

Student Name: Md. Tazminur Rahman Tanim

ID: 242014124

Department of CSE

University of Liberal Arts Bangladesh (ULAB)

Payroll Management System (Using Singly Linked List)

This open-ended lab report presents the design and implementation of an **Employee Payroll Management System** using the **C programming language**. The core data structure used is a **singly linked list**, which allows dynamic handling of employee records such as insertion, deletion, searching, and sorted display based on employee ID. The goal of the project is to demonstrate how basic data structures can be applied to a real-world problem scenario — managing employee salary records in a structured, efficient, and scalable way.

Objective of the System

The payroll system is designed to perform the following operations:

- Add a new employee's salary details.
- Remove an employee using their Employee ID.
- Search for an employee using their Employee ID.
- Display all employees sorted by Employee ID.

These functionalities are integrated into a **menu-driven program** that allows continuous interaction with the system until the user chooses to exit.

Use of Structures in the Program

To represent each employee, a `struct` is used with the following fields:

- `int id`: A unique integer identifier for each employee.
- `char name[50]`: A character array to store the employee's name.
- `int salary`: An integer to hold the employee's salary amount.
- `struct Employee* next`: A pointer to the next employee node.

This structure provides a convenient and organized way to encapsulate employee information while enabling dynamic chaining of nodes in a linked list.

Operations Implemented Using Linked List

The payroll system is functionally broken down into the following modular components:

- **main()**
  Manages user interaction via a menu. Based on the user's selection, it calls the corresponding function to perform the required operation.

- **addEmployee(int id, char name[ ], int salary)**
  Creates a new node dynamically using malloc(), assigns the employee's ID, name, and salary, and inserts the node at the end of the linked list. If the list is empty, it sets the new node as the head.
- **removeEmployee(int id)**
  Traverses the list to find the employee by ID. If found, it adjusts the link of the previous node to skip the current node, deallocates the memory, and removes the node. If not found, an appropriate message is displayed.
- **searchEmployee(int id)**
  Traverses the entire list from the head and prints the employee's details if the ID matches. If not found, a message is displayed.
- **displaySortedEmployees( )**
  Collects all nodes into an array, sorts the array based on employee ID using bubble sort, and prints the sorted list with each employee's ID, name, and salary.

Advantages of Using Linked List

1. **Dynamic Memory Management** – No need to declare a fixed size; memory is allocated as needed.
2. **Efficient Insertion/Deletion** – Nodes can be easily added or removed without shifting other elements.
3. **Scalability** – Suitable for applications where the number of records is large and variable.

Conclusion

This Payroll Management System is a fundamental yet practical implementation of the **singly linked list** data structure. It covers essential operations such as insertion, deletion, search, and sorting of employee records dynamically. The project highlights the importance of choosing the right data structure for effective problem-solving. It also serves as an excellent example of how foundational data structure knowledge can be applied in real-world software systems like payroll or HR management solutions.

The complete source code for the system is attached below.