192311035

M.RAVI SAI VINAY

EXPERIMENT-16

Develop a C program for implementing random access file for processing the employee details.

AIM:-

To develop a C program for implementing random access file processing for managing employee details, including adding, displaying, searching, updating, and deleting records.

ALGORITHM:-

1. Start the Program.
2. Create a structure to store employee details (ID, name, and salary).
3. Implement the following functionalities:
    - Add Employee: Append new employee details to the file.
    - Display Employees: Read all records sequentially from the file and display them.
    - Search Employee: Find a specific employee record by ID.
    - Update Employee: Modify an existing employee record using fseek for random access.
    - Delete Employee: Remove a record by copying all except the matching record to a temporary file and replacing the original file.
4. Provide a menu-driven interface for the user to perform these operations.
5. Repeat until the user chooses to exit.
6. End the Program.

CODE:-

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>

#define FILE_NAME "employee.dat"

struct Employee {
    int id;
    char name[50];
    float salary;
};

void addEmployee() {
    FILE *file = fopen(FILE_NAME, "ab");
    struct Employee emp;
    printf("Enter Employee ID: ");
    scanf("%d", &emp.id);
    printf("Enter Employee Name: ");
    scanf("%s", emp.name);
    printf("Enter Employee Salary: ");
    scanf("%f", &emp.salary);
    fwrite(&emp, sizeof(emp), 1, file);
    fclose(file);
}
```

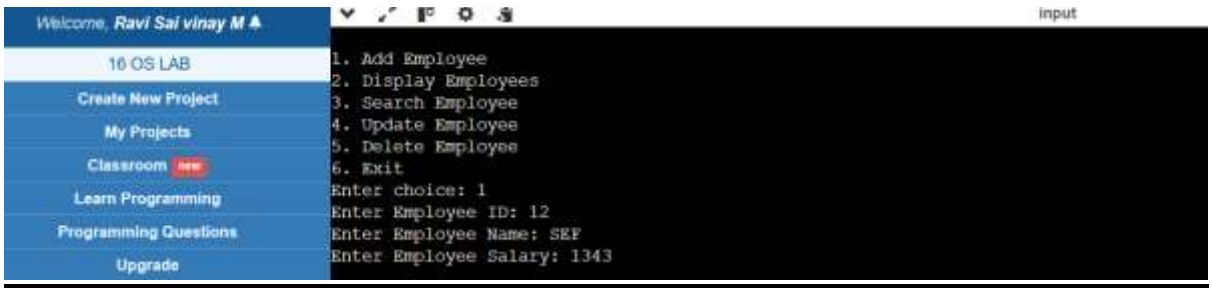
```
void displayEmployees() {
    FILE *file = fopen(FILE_NAME, "rb");
    struct Employee emp;
    printf("\nEmployee Details:\n");
    while (fread(&emp, sizeof(emp), 1, file)) {
        printf("ID: %d, Name: %s, Salary: %.2f\n", emp.id, emp.name, emp.salary);
    }
    fclose(file);
}

void searchEmployee() {
    FILE *file = fopen(FILE_NAME, "rb");
    int searchId, found = 0;
    struct Employee emp;
    printf("Enter Employee ID to search: ");
    scanf("%d", &searchId);
    while (fread(&emp, sizeof(emp), 1, file)) {
        if (emp.id == searchId) {
            printf("ID: %d, Name: %s, Salary: %.2f\n", emp.id, emp.name, emp.salary);
            found = 1;
            break;
        }
    }
    if (!found) printf("Employee not found.\n");
```

```
    fclose(file);
}

void updateEmployee() {
    FILE *file = fopen(FILE_NAME, "rb+");
    int searchId, found = 0;
    struct Employee emp;
    printf("Enter Employee ID to update: ");
    scanf("%d", &searchId);
    while (fread(&emp, sizeof(emp), 1, file)) {
        if (emp.id == searchId) {
            printf("Enter new Name: ");
            scanf("%s", emp.name);
            printf("Enter new Salary: ");
            scanf("%f", &emp.salary);
            fseek(file, -sizeof(emp), SEEK_CUR);
            fwrite(&emp, sizeof(emp), 1, file);
            found = 1;
            break;
        }
    }
    if (!found) printf("Employee not found.\n");
    fclose(file);
}
```

```
void deleteEmployee() {
    FILE *file = fopen(FILE_NAME, "rb");
    FILE *tempFile = fopen("temp.dat", "wb");
    int deleteId, found = 0;
    struct Employee emp;
    printf("Enter Employee ID to delete: ");
    scanf("%d", &deleteId);
    while (fread(&emp, sizeof(emp), 1, file)) {
        if (emp.id == deleteId) {
            found = 1;
        } else {
            fwrite(&emp, sizeof(emp), 1, tempFile);
        }
    }
    fclose(file);
    fclose(tempFile);
    remove(FILE_NAME);
    rename("temp.dat", FILE_NAME);
    if (found) printf("Employee deleted successfully.\n");
    else printf("Employee not found.\n");
}

int main() {
```

```
    int choice;
    do {
        printf("\n1. Add Employee\n2. Display Employees\n3. Search Employee\n4. Update Employee\n5. Delete Employee\n6. Exit\nEnter choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1: addEmployee(); break;
            case 2: displayEmployees(); break;
            case 3: searchEmployee(); break;
            case 4: updateEmployee(); break;
            case 5: deleteEmployee(); break;
            case 6: printf("Exiting.\n"); break;
            default: printf("Invalid choice.\n");
        }
    } while (choice != 6);
    return 0;
}
```

OUTPUT:-



RESULT:-

The program was successfully implemented to process employee records using a random access file. Operations such as adding, displaying, searching, updating, and deleting employee details worked correctly, demonstrating the effectiveness of random access file processing in C.