SDM COLLEGE OF ENGINEERING AND TECHNOLOGY

Dhavalagiri, Dharwad-580002, Karnataka State, India.

Email: cse.sdmcet@gmail.com

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Minor Work

COURSE CODE: 22UCS501 COURSE TITLE: Database Management System
SEMESTER:V DIVISION:A
COURSE TEACHER: Dr. U.P Kulkarni

[Academic Year- 2024-25]

Date of Submission: 22-10-2024

Submitted
By

Ms.Anusha Hegde USN: 2SD22CS015

Table of Contents

A1..3

A2..8

A3..4

A1. Write a C program to study all file operations related SYSTEM CALLS supported by UNIX OS and C libraries for file operations.

```
#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <sys/stat.h>


#define FILENAME "Operations.txt"

#define BUFFER  1024


void create_file() {

        int fd = open(FILENAME, O_CREAT | O_WRONLY | O_TRUNC, S_IRUSR
| S_IWUSR);
```

```
    if (fd == -1) {
    perror("Error creating file");
    exit(EXIT_FAILURE);
    }
    close(fd);
    perror("Error writing to file");
    close(fd);
    exit(EXIT_FAILURE);
    }

    printf("Wrote %zd bytes to %s\n", bytes_written, FILENAME);
    close(fd);
}

void read_file() {
    char buffer[BUFFER];
    int fd = open(FILENAME, O_RDONLY);
    if (fd == -1) {
    perror("Error opening file for reading");
    exit(EXIT_FAILURE);
    }
```

```
ssize_t bytes_read;
printf("Contents of %s:\n", FILENAME);
while ((bytes_read = read(fd, buffer, sizeof(buffer) - 1)) > 0) {
buffer[bytes_read] = '\0'; // Null terminate the string
        printf("%s", buffer);}


void write_to_file() {
        int fd = open(FILENAME, O_WRONLY | O_APPEND);
        if (fd == -1) {
        perror("Error opening file for writing");
        exit(EXIT_FAILURE);
        }


char *text = "Hello, I am Anusha";
ssize_t bytes_written = write(fd, text, strlen(text));
if (bytes_written == -1) {


}


if (bytes_read == -1) {
        perror("Error reading from file");
        }

        close(fd);
}

void delete_file() {
        if (unlink(FILENAME) == 0) {
        printf("Deleted file %s successfully.\n", FILENAME);
        } else {
        perror("Error deleting file");
        }
}

int main() {
        create_file();
        write_to_file();
        read_file();
        delete_file();

        return 0;
```

```
            }
```

OUTPUT:

```
Wrote 20 bytes to Operations.txt
Contents of Operations.txt:
Hello, I am Anusha
Deleted file Operations.txt successfully.
```

A2. Write a C program to demonstrate indexing and associated operations.

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


#define MAX_STUDENTS 100

#define LENGTH 50


typedef struct {

        int id;

        char name[LENGTH];

} Student;
```

```
typedef struct {
    Student students[MAX_STUDENTS];
    int count;
} StudentIndex;

void add_student(StudentIndex *index, int id, const char *name) {
    if (index->count < MAX_STUDENTS) {
    index->students[index->count].id = id;
    strncpy(index->students[index->count].name, name, LENGTH);
    index->count++;
    } else {
    printf("Index is full, cannot add more students.\n");
    }
}

void display_students( StudentIndex *index) {
    printf("Student Records:\n");
    for (int i = 0; i < index->count; i++) {
    printf("ID: %d, Name: %s\n", index->students[i].id, index->students[i].name);
    }
}
```

```
int search_student(const StudentIndex *index, int id) {
    for (int i = 0; i < index->count; i++) {
    if (index->students[i].id == id) {
    return i; // Return index of found student
    }
    }
    return -1; // Not found
}

int main() {
    StudentIndex index = { .count = 0 };
    int choice, id;
    char name[LENGTH];

    while (1) {
    printf("\nMenu:\n");
    printf("1. Add Student\n");
    printf("2. Display Students\n");
    printf("3. Search Student by ID\n");
    printf("4. Exit\n");
    printf("Choose an option: ");
    scanf("%d", &choice);
```

```
switch (choice) {
case 1:
        printf("Enter student ID: ");
        scanf("%d", &id);
        printf("Enter student name: ");
        scanf(" %[^\n]", name); // Read string with spaces
        add_student(&index, id, name);
        break;
case 2:
        display_students(&index);
        break;
case 3:
        printf("Enter student ID to search: ");
        scanf("%d", &id);
        int found_index = search_student(&index, id);
        if (found_index != -1) {
        printf("Found Student-ID:%d, Name:%s\n",
index.students[found_index].id, index.students[found_index].name);
        } else {
        printf("Student with ID %d not found.\n", id);
        }
        break;
case 4:
```

```
                    exit(0);

            default:

                    printf("Invalid choice. Please try again.\n");

            }

            }


    return 0;

}
```

OUTPUT:

```
Menu:
1. Add Student
2. Display Students
3. Search Student by ID
4. Exit

Choose an option: 1
Enter student ID: 1
Enter student name: Samy Pal

Menu:
1. Add Student
2. Display Students
3. Search Student by ID
4. Exit
Choose an option: 1
Enter student ID: 2
Enter student name: Mani Singh

Menu:
1. Add Student
2. Display Students
3. Search Student by ID
```

```
4. Exit
Choose an option: 2
Student Records:
ID: 1, Name: Samy Pal
ID: 2, Name: Mani Singh

Menu:
1. Add Student
2. Display Students
3. Search Student by ID
4. Exit
Choose an option: 3
Enter student ID to search: 2
Found Student-ID:2, Name:Mani Singh

Menu:
1. Add Student
2. Display Students
3. Search Student by ID
4. Exit
Choose an option: 4
```

A3. Write a Java program to access the given excel file with known file format.

```
import org.apache.poi.ss.usermodel.*;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.FileInputStream;
import java.io.IOException;

public class ExcelReader {
    public static void main(String[] args) {
        String excelFilePath = "path/to/your/excelfile.xlsx"; // Update this path

        try (FileInputStream fis = new FileInputStream(excelFilePath);
             Workbook workbook = new XSSFWorkbook(fis)) {

            Sheet sheet = workbook.getSheetAt(0); // Get the first sheet
            // Loop through each row in the sheet
            for (Row row : sheet) {
                // Loop through each cell in the row
                for (Cell cell : row) {
                    switch (cell.getCellType()) {
                        case STRING:
                            System.out.print(cell.getStringCellValue() + "\t");
                            break;
                        case NUMERIC:
                            System.out.print(cell.getNumericCellValue() + "\t");
                            break;
                        case BOOLEAN:
                            System.out.print(cell.getBooleanCellValue() + "\t");
                            break;
                        case FORMULA:
                            System.out.print(cell.getCellFormula() + "\t");
                            break;
                        default:
                            System.out.print("Unknown Type\t");
                    }
                }
                System.out.println(); // Move to the next line after each row
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

OUTPUT: