

SDM COLLEGE OF ENGINEERING AND TECHNOLOGY

Dhavalagiri, Dharwad-580002, Karnataka State, India.

Email: cse.sdmcet@gmail.com

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Report
on

CTA Assignment

COURSE CODE: 22UCSC501 COURSE TITLE: DATABASE MANAGEMNET SYSTEM

SEMESTER: 5 DIVISION: A

COURSE TEACHER: Dr. U. P. Kulkarni



[Academic Year- 2023-24]

Date of Submission: 24-10-2024

Submitted
By

Miss. Neha Vithob Nayak USN: 2SD22CS129



Table of Contents

Write a C program to study all file operations related SYSTEM CALLS supported by UNIX OS and C libraries for file operations	3
Write a C program to demonstrate indexing and associated operations.....	5
Write a Java program to access the given excel file with known file format	8



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

This C program demonstrates file operations using both **system calls** and the **C standard library**:

1. **System Calls:** It creates (open), writes (write), reads (read), and closes (close) a file (system_file.txt).
2. **C Standard Library:** It performs the same operations (create, write, read, close) using fopen, fwrite, fread, and fclose on another file (lib_file.txt).

Code:

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>

int main() {
    const char sysFileName = "system.txt";
    const char CFileName = "clib.txt";

    char writeBuffer[] = "C Library and System Operations";
    char readBuffer[100];

    printf("System Call File Operations:\n");

    // Open file (create if it doesn't exist) for writing using system call
    int sysFile = open(sysFileName, O_CREAT | O_WRONLY, 0644);
    if (sysFile < 0) {
        perror("Error opening file using system call");
        exit(1);
    }
    write(sysFile, writeBuffer, sizeof(writeBuffer) - 1);

    // Close file using system call
    close(sysFile);

    printf("\nC Library File Operations:\n");
```



```
// Open file for writing using C library
FILE *libFile = fopen(CFileName, "w");
if (libFile == NULL) {
    perror("Error opening file using C library");
    exit(1);
}
fwrite(writeBuffer, sizeof(char), sizeof(writeBuffer) - 1, CFileName);
fclose(libFile);

// Open file for reading using C library
libFile = fopen(fileLib, "r");
if (libFile == NULL) {
    perror("Error opening file for reading using C library");
    exit(1);
}
fread(readBuffer, sizeof(char), sizeof(readBuffer), CFileName);
printf("Content read from C library file: %s\n", readBuffer);

// Close file using C library
fclose(libFile);

return 0;
}
```

Output:

```
System Call File Operations:
Content read from system call file: C Library and System Operationsvämä

C Library File Operations:
Content read from C library file: C Library and System Operationsvämä
(.venv)
nehan@LAPTOP-9IT1UHHS MINGW64 /d/Codes/DBMS
```



Write a C program to demonstrate indexing and associated operations.

This C program demonstrates basic array operations, including insertion, deletion, and searching. It maintains an integer array with a maximum size of 100. The user can:

1. **Display the Array:** Shows current elements in the array.
2. **Insert an Element:** Adds a new element at a specified index, shifting subsequent elements as needed.
3. **Delete an Element:** Removes an element at a given index and shifts remaining elements.
4. **Search for an Element:** Checks if a specified element exists and returns its index.

Code:

```
#include <stdio.h>

#define MAX_SIZE 100

// Function to display the array
void displayArray(int arr[], int size) {
    printf("Array elements: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

// Function to insert an element at a specific index
void insertElement(int arr[], int size, int element, int index) {
    if (size >= MAX_SIZE) {
        printf("Array is full! Cannot insert element.\n");
        return;
    }
    if (index < 0 || index > size) {
        printf("Index out of bounds! Cannot insert element.\n");
        return;
    }
    for (int i = size; i > index; i--) {
        arr[i] = arr[i - 1];
    }
    arr[index] = element;
}

// Function to delete an element at a specific index
void deleteElement(int arr[], int size, int index) {
```



```

if (size == 0) {
    printf("Array is empty! Cannot delete element.\n");
    return;
}
if (index < 0 || index >= size) {
    printf("Index out of bounds! Cannot delete element.\n");
    return;
}
for (int i = index; i < size - 1; i++) {
    arr[i] = arr[i + 1];
}
}

// Function to search for an element
int searchElement(int arr[], int size, int element) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == element) {
            return i;
        }
    }
    return -1;
}

// Main function
int main() {
    int arr[MAX_SIZE] = {10, 20, 30, 40, 50};
    int size = 5;
    // Display the initial array
    displayArray(arr, size);

    // Inserting an element
    printf("Inserting 25 at index 2...\n");
    insertElement(arr, size, 25, 2); // Insert 25 at index 2
    size++; // Increment the size after insertion
    displayArray(arr, size);

    // Deleting an element
    printf("Deleting element at index 3...\n");
    deleteElement(arr, size, 3); // Delete element at index 3
    size--; // Decrement the size after deletion
    displayArray(arr, size);

    // Searching for an element
}

```



```
int elementToSearch = 30;
int index = searchElement(arr, size, elementToSearch);
if (index != -1) {
    printf("Element %d found at index %d.\n", elementToSearch, index);
} else {
    printf("Element %d not found in the array.\n", elementToSearch);
}

return 0;
}
```

Output:

```
nehan@LAPTOP-9IT1UHHS MINGW64 /d/Codes/DBMS
● $ cd /d/Codes/DBMS/ && gcc Indexing.c -o Indexing && ./Indexing
Array elements: 10 20 30 40 50
Inserting 25 at index 2...
Array elements: 10 20 25 30 40 50
Deleting element at index 3...
Array elements: 10 20 25 40 50
Element 30 not found in the array.
(.venv)
```



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

This Java program uses the Apache POI library to read an Excel file named data.xlsx. It opens the file, accesses the first sheet, and iterates through each row and cell, extracting their values based on cell types (numeric, boolean, string) using the getCellValue method. The extracted values are printed to the console. Finally, the program ensures that resources are closed properly by closing the workbook and the file input stream after reading.

Code:

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;

import org.apache.poi.xssf.usermodel.XSSFCell;
import org.apache.poi.xssf.usermodel.XSSFRow;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ReadXlFile {

    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub
        File file = new File("D:\\ENGI 🎓 \\Notes 5 sem\\DBMS\\data.xlsx");
        FileInputStream fis = new FileInputStream(file);
        XSSFWorbook workbook = new XSSFWorbook(fis);
        XSSFSheet sheet = workbook.getSheetAt(0);

        int rowCount = sheet.getPhysicalNumberOfRows();

        for(int i = 0; i < rowCount; i++) {
            XSSFRow row = sheet.getRow(i);
            int cellCount = row.getPhysicalNumberOfCells();
            for(int j = 0; j < cellCount; j++) {
                XSSFCel cell = row.getCell(j);
                String cellValue = getCellValue(cell);
                System.out.println("|| "+cellValue);
            }
            System.out.println();
        }

        workbook.close();
        fis.close();
    }

    public static String getCellValue(XSSFCell cell) {
```



```
switch(cell.getCellTypeEnum()) {  
    case NUMERIC:  
        return String.valueOf(cell.getNumericCellValue());  
    case BOOLEAN:  
        return String.valueOf(cell.getBooleanCellValue());  
    case STRING:  
        return cell.getStringCellValue();  
    default:  
        return cell.getStringCellValue();  
}  
}  
  
}
```

Output:

```
||101||Deepa||1001||12||1200  
||102||Pushpa||1002||10||200  
||103||Varsha||1003||23||850  
||104||Arti||1004||10||3000  
||105||Anita||1005||45||5500
```

