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

ASSIGNMENT-1

COURSE CODE: 22UCSC501 COURSE TITLE: Database Management System

SEMESTER:5 DIVISION:A

COURSE TEACHER: DR.U.P.KULKARNI

[Academic Year- 2024-25]

Date of Submission: 24-10-2024

Submitted

By

Ms. Smital S K USN:2SD23CS409

1. C program to study operations related system calls supported by unix OS and C for the operations.

Code:

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

int main() {
  int fd;
  char buffer[100];
  ssize_t bytesRead, bytesWritten;

  // Open the file (create if it doesn't exist) in append mode
  fd = open("simple_output.txt", O_RDWR | O_CREAT | O_APPEND, 0644);
  if (fd < 0) {
    perror("Failed to open file");
    exit(EXIT_FAILURE);
  }

  // Write to the file
  const char *text =
      "We truly cannot understand the real worth of the environment. But we "
      "can estimate some of its importance that can help us understand its "
      "importance. It plays a vital role in keeping living things healthy in "
      "the environment.\n";
  bytesWritten = write(fd, text, strlen(text));
  if (bytesWritten < 0) {
    perror("Failed to write to file");
    close(fd);
    exit(EXIT_FAILURE);
  }
  printf("Written %zd bytes to file.\n", bytesWritten);

  // Move file pointer to the start
  lseek(fd, 0, SEEK_SET);

  // Read from the file
  bytesRead = read(fd, buffer, sizeof(buffer) - 1);
  if (bytesRead < 0) {
    perror("Failed to read from file");
    close(fd);
    exit(EXIT_FAILURE);
  }
```

```
 buffer[bytesRead] = '\0'; // Null-terminate the string
 printf("File content:\n%s\n", buffer);

 // Close the file
 if (close(fd) < 0) {
  perror("Failed to close file");
  exit(EXIT_FAILURE);
 }

 return 0;
}
```

Output:

```
Written 224 bytes to file.
File content:
This is some unique content!
We truly cannot understand the real worth of the environment. But we c
```

2. C program to demonstrate indexing and associated operations.

Code:

```
#include <stdio.h>

int main() {
   int numbers[5];  // Array with a fixed size of 5
   int size = 5;    // Total number of elements
   int i, index, value;

   // Input array elements from the user
   printf("Enter 5 elements:\n");
   for (i = 0; i < size; i++) {
      scanf("%d", &numbers[i]);
   }

   // Display the elements with their indexes
   printf("Initial array elements with indexes:\n");
   for (i = 0; i < size; i++) {
      printf("Index %d: %d\n", i, numbers[i]);
   }

   // Modify an element at a specific index
   printf("\nEnter the index to modify (0 to 4): ");
   scanf("%d", &index);
   if (index >= 0 && index < size) {
      printf("Enter the new value: ");
```

```
        scanf("%d", &value);
        numbers[index] = value;

        // Display the modified array
        printf("\nArray after modification:\n");
        for (i = 0; i < size; i++) {
            printf("Index %d: %d\n", i, numbers[i]);
        }
    } else {
        printf("Invalid index!\n");
    }

    return 0;
}
```

Output:

```
Enter 5 elements:
2 33 10 5 6
Initial array elements with indexes:
Index 0: 2
Index 1: 33
Index 2: 10
Index 3: 5
Index 4: 6

Enter the index to modify (0 to 4): 3
Enter the new value: 25

Array after modification:
Index 0: 2
Index 1: 33
Index 2: 10
Index 3: 25
Index 4: 6
```

3. Java Program to access the given excel file with known format.

   Used spring initializer for the context of fetching and accessing the data of the excel file present.
   These are the steps followed:

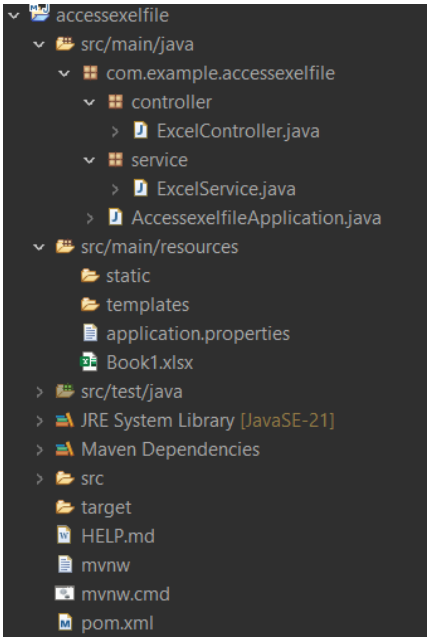
   - Set Up Spring Boot Project: You can create a Spring Boot project using Spring Initializer. Include the following dependencies: **Spring Web**: For creating web applications. **Apache POI**: For reading Excel files.
     <dependency>

     <groupId>org.apache.poi</groupId>

     <artifactId>poi-ooxml</artifactId>

     <version>5.2.3</version>

     </dependency>
   - All this below listed files are save in an extracted file of a spring-boot which navigate and provide service between the packages. The excel file is also added in the folder itself as “Book1” in resource folder.



   - Running the Application: AccessexcelfileApplication.java

Code:

ExcelService.java

```
package com.example.accessexelfile.service;

import org.apache.poi.ss.usermodel.*;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.springframework.stereotype.Service;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

@Service
public class ExcelService {
   public List<List<String>> readExcel() throws IOException {
      List<List<String>> data = new ArrayList<>();

      // Load the Excel file from resources
      try (InputStream is =
getClass().getClassLoader().getResourceAsStream("Book1.xlsx");
           Workbook workbook = new XSSFWorkbook(is)) {
         Sheet sheet = workbook.getSheetAt(0);
         for (Row row : sheet) {
            List<String> rowData = new ArrayList<>();
            for (Cell cell : row) {
               switch (cell.getCellType()) {
                  case STRING:
                     rowData.add(cell.getStringCellValue());
                     break;
                  case NUMERIC:
                     rowData.add(String.valueOf(cell.getNumericCellValue()));
                     break;
                  case BOOLEAN:
                     rowData.add(String.valueOf(cell.getBooleanCellValue()));
                     break;
                  default:
                     rowData.add("Unknown Type");
               }
            }
            data.add(rowData);
         }
      }
      return data;
   }
}
```

ExcelController.java

```
package com.example.accessexelfile.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

import com.example.accessexelfile.service.ExcelService;

import java.io.IOException;

import java.util.List;


@RestController

public class ExcelController {

    @Autowired

    private ExcelService excelService;

    @GetMapping("/read-excel")

    public List<List<String>> readExcel() {

        try {

            return excelService.readExcel();

        } catch (IOException e) {

            e.printStackTrace();

            return null; // You may want to handle this error more gracefully

        }

    }

}
```

To get output:

Go to chrome(or other) and type http://localhost:8080/read-excel.

Output:

[["ID","Name","Age","City"],["1.0","Alice","30.0","New York"],["2.0","Bob","25.0","Los Angeles"],["3.0","Charlie","28.0","Chicago"],["4.0","David","35.0","Houston"],["5.0","Eva","22.0","Miami"]]