

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 DBMS- Minor Work

COURSE CODE: 22UCSC501 COURSE TITLE: DBMS

SEMESTER: V DIVISION: A

COURSE TEACHER: Dr. U. P. Kulkarni



[Academic Year- 2023-24]

Date of Submission: 19/12/2024

Submitted
By

**Mr. Pratham M
USN: 2SD22CS063**

Table of Contents

Minor Work: -	3
Problem statement 1: - Write a C program to discuss to demonstrate all the functions related to system calls in Unix based OS.	3
Problem Statement 2: - Write a C program to demonstrate the indexing and its associated properties.	7
Problem Statement 3: - Write a Java program to access the given Excel file with known file format .	9

Minor Work: -

Problem statement 1: - Write a C program to discuss to demonstrate all the functions related to system calls in Unix based OS.

//A simple C program to showcase the System calls in Unix based OS systems

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

```
#include <fcntl.h>
```

```
#include <unistd.h>
```

```
#include <sys/stat.h>
```

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

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

```
#define SIZE 100
```

```
char* readFile(char *str);
```

```
void writeFile(char *str);
```

```
void appendFile(char *str);
```

```
int main(){
```

```
    int choice;
```

```
    char str[SIZE];
```

```
    printf("Enter (1-read) (2-write) (3-append) file\n");
```

```
    printf("Enter your choice\n");
```

```
    scanf("%d",&choice);
```

```
    getchar();
```

```
    switch (choice)
```

```
{
```

case 1:

```
printf("Enter the file name to read\n");  
scanf("%s",str);  
getchar(); //clr the I/P stream  
printf("The contents of the file are:\n%s\n",readFile(str));  
break;
```

case 2:

```
printf("Enter the file name to write\n");  
scanf("%s",str);  
getchar();  
writeFile(str);  
break;
```

case 3:

```
printf("Enter the file name to appended\n");  
scanf("%s",str);  
getchar();  
appendFile(str);  
break;
```

default:

```
break;
```

```
}
```

```
return 0;
```

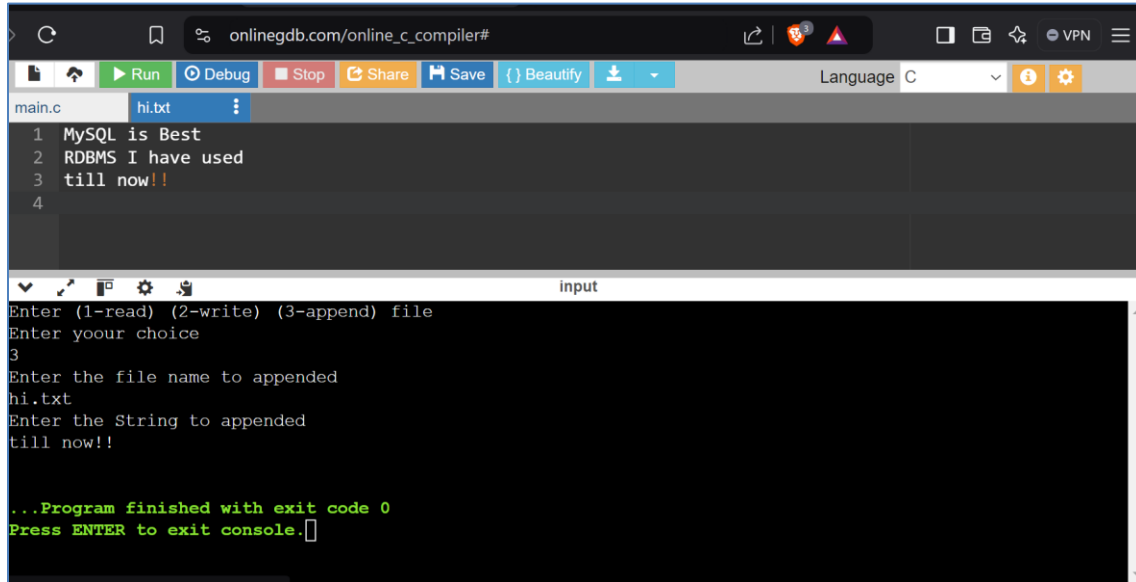
```
}
```

```
void writeFile(char *filename){  
    fflush(stdin);  
    int fd = open(filename, O_CREAT | O_WRONLY | O_TRUNC, S_IRUSR | S_IWUSR);  
    if(fd < 0){  
        perror("Error occurred while opening the file\n");  
        close(fd);  
        return;  
    }  
    char str[SIZE];  
    printf("Enter the string to be written\n");  
    fgets(str, SIZE, stdin);  
    write(fd, str, strlen(str));  
    close(fd);  
}
```

```
char *readFile(char *filename){  
    char *str = (char *)malloc(SIZE*sizeof(char));  
    int fd = open(filename, O_RDONLY);  
    if(fd < 0){  
        perror("Can't open this file\n");  
        free(str);  
        close(fd);  
        return NULL;  
    }  
    int bytesRead = read(fd, str, SIZE-1);
```

```
    str[bytesRead] = '\0';  
    close(fd);  
    return str;  
}  
  
void appendFile(char *filename){  
    int fd = open(filename, O_WRONLY | O_APPEND | O_CREAT, S_IRUSR | S_IWUSR);  
    if(fd < 0){  
        perror("Can't append to the file\n");  
        close(fd);  
        return;  
    }  
    char str[SIZE];  
    fflush(stdin);  
    printf("Enter the String to appended\n");  
    fgets(str, SIZE, stdin);  
    write(fd, str, strlen(str));  
}
```

O/P: -



The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler#`. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C. The code editor shows a file named `hi.txt` with the following content:

```
1 MySQL is Best
2 RDBMS I have used
3 till now!!
4
```

Below the code editor is an input/output console. The console shows the following interaction:

```
Enter (1-read) (2-write) (3-append) file
Enter your choice
3
Enter the file name to append
hi.txt
Enter the String to append
till now!!

...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 1.1 – Appending to the file

Problem Statement 2: - Write a C program to demonstrate the indexing and its associated properties.

//A simple C pgm to read and print the details of the employees

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

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

```
#define SIZE 30
```

```
#define ECOUNT 10
```

```
typedef struct Employee{
```

```
    int empID;
```

```
    char name[SIZE];
```

```
float salary;

}Employee;

void readEmployeeDetails(Employee *e, int n){

    for(int i=0; i<n; i++){

        printf("Enter the employee %d details\n", i+1);

        printf("Enter the employee id\n");

        scanf("%d",&e[i].empID);

        printf("Enter the employee name\n");

        scanf("%s",&e[i].name);

        printf("Enter the employee salary\n\n");

        scanf("%s",&e[i].salary);

    }

}
```

```
void printEmployeeDetails(Employee *e, int n){

    for(int i=0; i<n; i++){

        printf("EMPID : %d\n",e[i].empID);

        printf("EMP Name : %d\n",e[i].name);

        printf("EMP Salary : %d\n\n",e[i].salary);

    }

}
```

```
int main(){

    Employee e[ECOUNT];

    int n;
```



```

printf("Enter the no of employees\n");

scanf("%d",&n);

if(n > ECOUNT){

    printf("max employees = %d\n", ECOUNT);

    exit(0);

}

readEmployeeDetails(e, n);

printEmployeeDetails(e, n);

return 0;

}

```

Problem Statement 3: - Write a Java program to access the given Excel file with known file format

```

package DBMS_Demo.ExcelSheets;

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

//import java.io.FileInputStream;
//import java.io.IOException;
import java.io.*;
import java.util.*;

public class ReadWriteXExcel {
    public static void readExcelSheet(String filePath) {
        try (FileInputStream fis = new FileInputStream(filePath);
            Workbook workbook = new XSSFWorkbook(fis)) { //if you declare inside the try then no
            need to close the fis & workbook externally it will be automatically closed

```

```

Sheet sheet = workbook.getSheetAt(0); // Get the first sheet
for (Row row : sheet) {
    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;
            default:
                System.out.print("UNKNOWN\t");
                break;
        }
    }
    System.out.println();
}

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

}

//Appending the file
public static void appendFile(String filePath) {
    try (FileInputStream fis = new FileInputStream(filePath);
        Workbook workbook = new XSSFWorkbook(fis)) {

        // Access the existing sheet or create a new one if it doesn't exist
        Sheet sheet = workbook.getSheetAt(0); // Use the first sheet

        // Determine the next row index (one after the last row)
        int nextRowIndex = sheet.getLastRowNum() + 1;

        // Create a new row
        Row newRow = sheet.createRow(nextRowIndex);

        // Create new cells and set values
        // Cell cell1 = newRow.createCell(0);
        // Cell cell2 = newRow.createCell(1);

```

```
Scanner sc = new Scanner(System.in);
int columnsCount = sheet.getRow(1).getLastCellNum();
for(int i=0; i<columnsCount; i++) {
    Cell cell = newRow.createCell(i);
    System.out.println("Enter the " + (i+1) + " cell value");
    cell.setCellValue(sc.next());
}

// Write the changes to the file
try (FileOutputStream fos = new FileOutputStream(filePath)) {
    workbook.write(fos);
}

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

}

public static void main(String[] args) {
String filePath = "D:\\Github\\DBMS Assignment\\Book1.xlsx"; // Path to your Excel file
readExcelSheet(filePath);
appendFile(filePath);
}
}
```

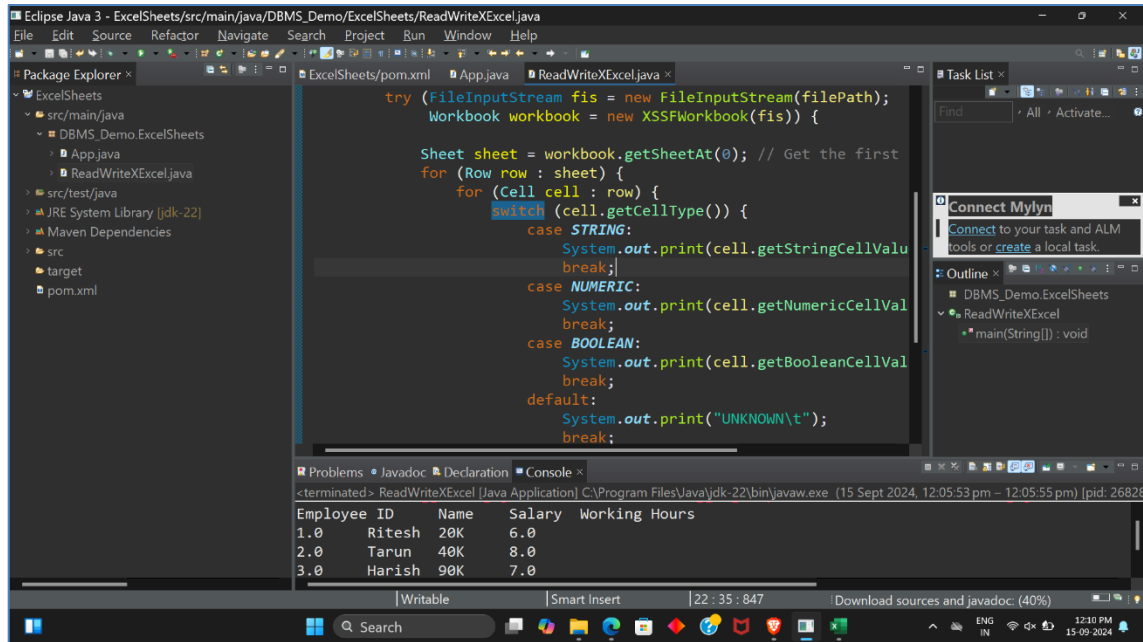


Figure 3.1 – R/W operations on Excel file

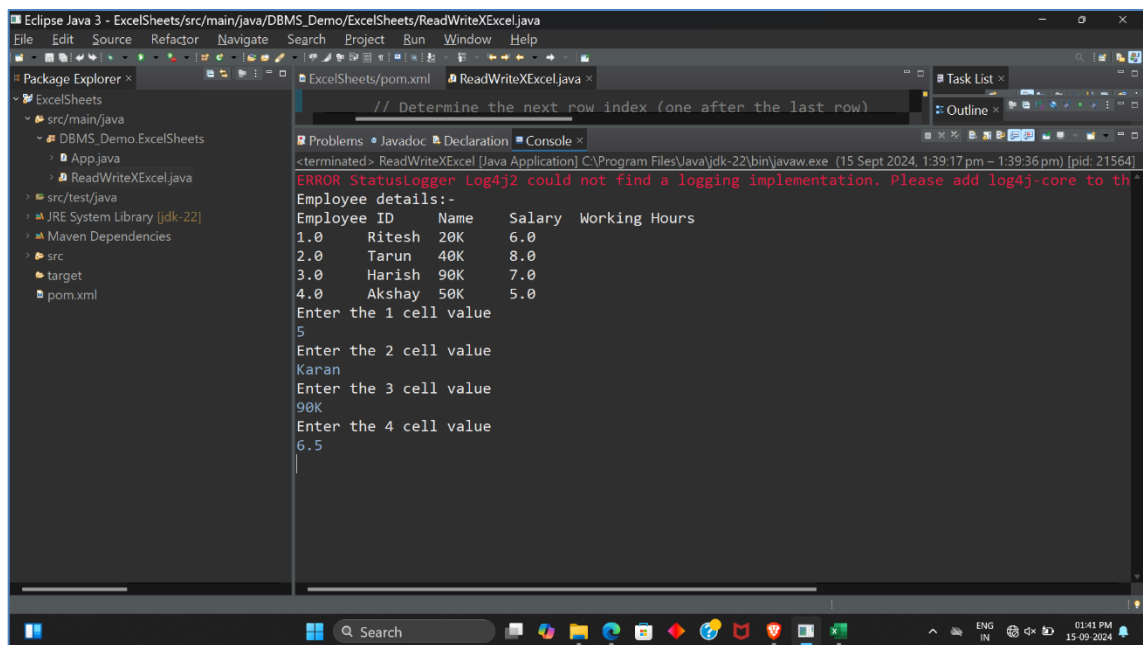
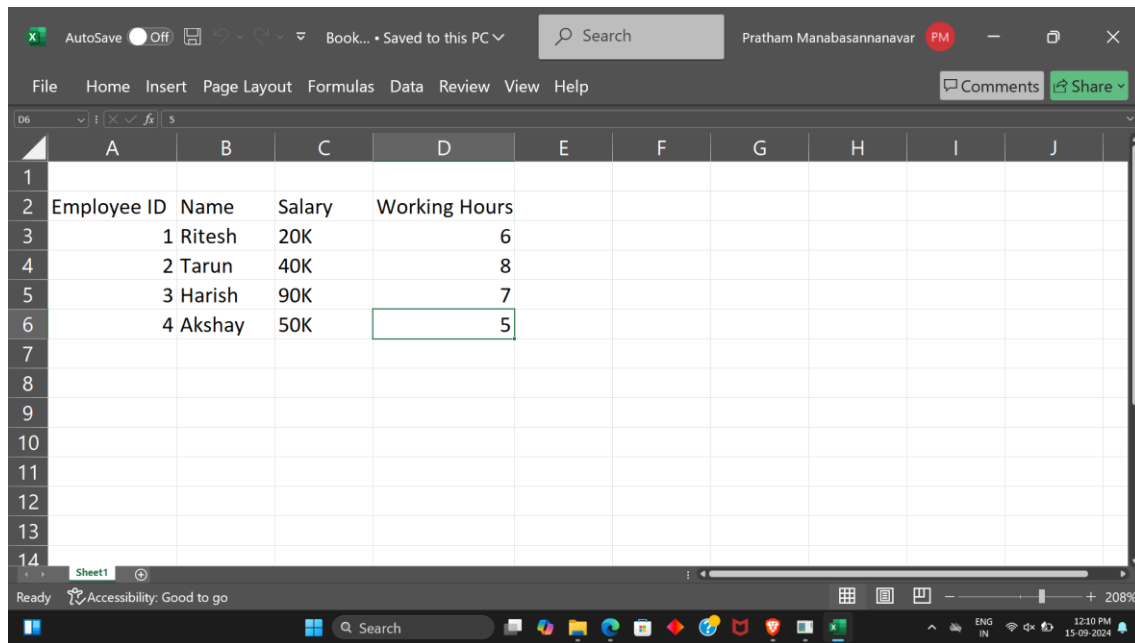


Figure 3.2 – Inserting values into Excel sheet



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J
1										
2	Employee ID	Name	Salary	Working Hours						
3		1 Ritesh	20K	6						
4		2 Tarun	40K	8						
5		3 Harish	90K	7						
6		4 Akshay	50K	5						
7										
8										
9										
10										
11										
12										
13										
14										

The spreadsheet is titled "Book1" and is saved to the PC. The user is Pratham Manabasanannavar. The status bar shows "Ready" and "Accessibility: Good to go". The taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system clock shows 12:10 PM on 15-09-2024.

Figure 3.3 – Excel sheet to perform R/W operations