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 Systems
SEMESTER:V DIVISION: A
COURSE TEACHER: Prof.Dr.U.P.Kulkarni

[Academic Year- 2024-25]

Date of Submission: 24-10-2024

Submitted
By

NAME:PUNARVASU USN: 2SD23CS407

Q1.write a C program to demonstrate all file operations related to system calls supported by unix operating system and c libraries

openReadoperation:

```
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#define BUF_SIZE 1024
int main(int argc,char *argv[]){
int fd=open(argv[1],O_RDONLY);
printf("%d\n",fd);
char buf[BUF_SIZE];
if(fd!=-1)
{
ssize_t numread=read(fd,buf,BUF_SIZE-1);
while(numread>0){
   buf[numread]='\0';
   printf("%s\n",buf);
   numread=read(fd,buf,BUF_SIZE-1);
}
}
else{
   printf("error in openning file\n");
   return -1;
}
return 0;
}
```

Output:

```
gcc openRead.c -o openRead

./openRead passwordmysql.txt

3

alter user 'root'@'localhost' identified by '23cs407'
```

createWrite:

```
#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <sys/stat.h>

int main(int argc,char *argv[]){

    int fd=open("E:/c prog/new.txt",O_CREAT | O_WRONLY,S_IRUSR |
S_IWUSR | S_IXUSR);

    if(fd>-1){

        write(fd,"test data\n",11);

    }else{

        printf("failed to create file ");

    }

    return 0;

}
```

Output:

```
gcc createWrite.c -o createWrite

./createWrite

Ls

Mode                LastWriteTime         Length Name

----                -------------         ------ ----

-a----        24-10-2024     08:40             12 new.txt
```

<u>LseekClose</u>:

```
#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#define BUF_SIZE 11

int main(int argc,char *argv[]){

    int fd=open(argv[1],O_RDONLY);

    char buf[BUF_SIZE];

    if(fd != -1){

        int seek=lseek(fd,(BUF_SIZE-1)*2,SEEK_SET);

        if(seek != -1){

        ssize_t numread=read(fd,buf,BUF_SIZE-1);

         printf("%s",buf);

        }

       else{

        printf("error in seeking file!\n");

        return -1;

    }

    close(fd);

    }else{

        printf("error in opening file!\n");

        return -1;

    }}
```

Output:

```
cat fileopiseek.txt

003,121,123

121,323,334

232,232,343

232,433,343

./IseekClose fileopiseek.txt

232,232,343
```

FILEHNADLING

Write

```
#include <stdio.h>

#include <stdbool.h>

int main(void)

{

   FILE *fh_output ;

   fh_output=fopen("io.txt","w");

    fputs("abc",fh_output);

   fputs("123\n",fh_output);

   fputs("A string\n",fh_output);

   int data=5;

   fprintf(fh_output,"data:%d\n",data);

    fclose(fh_output);

   return 0;}
```

Output:

```
gcc FILEHANDLING2.c -o FILEHANDLING2

./FILEHANDLING2

abc123

A string

data:5
```

Append:

```
#include <stdio.h>

int main(void)

{

    FILE  *fh_out;


    fh_out=fopen("o.txt","a");


    int data=5;

    fprintf(fh_out,"data:%d\n",data);

     fputs("abc",fh_out);

    fputs("123\n",fh_out);

    fputs("A string\n",fh_out);

    fclose(fh_out);

    return 0;

}
```

Output:

```
gcc filehandlingappend.c -o filehandlingappend

./filehandlingappend

cat o.txt

abc123

A string

abc123

A string

data:5

abc123

A string
```

Readwritecreat

```
#include <stdio.h>

#include <stdbool.h>

int main(void){

   FILE *fh_output;

   fh_output=fopen("io.txt","w");

   int input=0;

   while(true)

   {

      printf("enter no (-1 quits):");

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

```
        if(input==-1)break;
        else fprintf(fh_output,"%d\n",input);
    }
    fclose(fh_output);
    //#READONLY
     FILE *fh_input;
     fh_input=fopen("io.txt","r");
     int finput=0;
     int lines=0;
     int numbers[100];
    while( fscanf(fh_input,"%d",&finput)!=EOF){
        numbers[lines]=finput;
        printf("number: %d\n",finput);
        lines++;
    }
     fclose(fh_input);
    return 0;
}
```

OUTPUT:

```
gcc TEST1.C -o TEST1
./TEST1
enter no (-1 quits):2
enter no (-1 quits):3
enter no (-1 quits):2
enter no (-1 quits):-1
number: 2
number: 3
number: 2
```

Q2.write a c program to demonstrate indexing I and associated operations

```
#include <stdio.h>
#define SIZE 5  // Define the size of the array

// Function to display the contents of the array
void displayArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("arr[%d] = %d\n", i, arr[i]);
    }
}
int main() {
    int array[SIZE] = {10, 20, 30, 40, 50};  // Initialize the array
    printf("Initial array contents:\n");
    displayArray(array, SIZE);

    // Accessing array elements using indexing
    printf("\nAccessing elements using indexing:\n");
    printf("Element at index 0: %d\n", array[0]);
    printf("Element at index 2: %d\n", array[2]);
    printf("Element at index 4: %d\n", array[4]);
    // Modifying elements at specific indices
    array[1] = 100;  // Change the element at index 1
    array[3] = 200;  // Change the element at index 3

    printf("\nArray contents after modification:\n");
    displayArray(array, SIZE);
    return 0;
}
```

Output:

```
Initial array contents:
arr[0] = 10
arr[1] = 20
arr[2] = 30
arr[3] = 40
arr[4] = 50
Accessing elements using indexing:
Element at index 0: 10
Element at index 2: 30
Element at index 4: 50
Array contents after modification:
arr[0] = 10
arr[1] = 100
arr[2] = 30
arr[3] = 200
```

```
arr[4] = 50
```

Q3.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.File;
import java.io.FileInputStream;
import java.io.IOException;

public class ExcelReader {

    public static void main(String[] args) {
        // Path to your Excel file
        String excelFilePath = " E:\new1.1\Book1.xlsx ";

        try {
            // Load the Excel file
           FileInputStream fileInputStream = new FileInputStream(new
           File(E:\new1.1\Book1.xlsx));

            // Create a workbook instance that refers to the .xlsx file
            Workbook workbook = new XSSFWorkbook(fileInputStream);

            // Get the first sheet from the workbook
            Sheet sheet = workbook.getSheetAt(0);
```

```
// Iterate through each row in the sheet
for (Row row : sheet) {
    // Iterate through each cell in the row
    for (Cell cell : row) {
        // Process based on the cell type
        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 Value\t");
        }
    }
    System.out.println();
}
// Close the workbook and file stream
```

```
            workbook.close();

            fileInputStream.close();


        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}
```

Output:

```
NAME            PASSWORD
user_name1      password345
user_name2      password678
user_name3      passeord654
```