21MIS1001
Sabbarish S

30-07-2024
Tuesday

SWE-4001
System Programming

LAB-1

Working on Files and creation of Symbol table

Open and Close a File:

• Write a program to open a file named "test.txt" in read mode. Check if the file is successfully opened. If it is, close the file.

```c
//To Open and close a file

#include <stdio.h>

int main() {

    FILE *file;


    file = fopen("C:/Users/student/Downloads/ex1.txt", "r");

    if (file == NULL) {
        perror("Error opening file");
        return 1; // Return an error code
    }

    fclose(file);

    printf("File opened and closed successfully.\n");

    return 0;
}
```

```
File opened and closed successfully.

--------------------------------
Process exited after 0.01713 seconds with return value 0
Press any key to continue . . .
```

Write to a File:

• Write a program that opens a file named "output.txt" in write mode. Write the string "Hello, World!" to the file. Close the file after writing.

```c
#include <stdio.h>

int main() {

    FILE *file = fopen("C:/Users/student/Downloads/output.txt", "w");

    if (file == NULL) {
        perror("Error opening file");
        return 1;
    }

    fprintf(file, "Hello, World!");

    fclose(file);

    printf("Hello World text is written to the file output.txt successfully");

    return 0;
}
```

```
Hello World text is written to the file output.txt successfully
--------------------------------
Process exited after 0.0156 seconds with return value 0
Press any key to continue . . .
```

```
Hello, World!
```

Read from a File:

• Write a program to read the contents of a file named "input.txt" and print the contents to the console.

```c
#include <stdio.h>

int main() {

    FILE *file = fopen("C:/Users/student/Downloads/input.txt", "r");

    if (file == NULL) {
        perror("Error opening file");
        return 1;
    }

    char ch;
    while ((ch = fgetc(file)) != EOF) {
        putchar(ch);
    }

    fclose(file);

    return 0;
}
```

```
Hello, World! I am 21MIS1001
--------------------------------
Process exited after 0.02303 seconds with return value 0
Press any key to continue . . .
```

```
Hello, World! I am 21MIS1001
```

Append to a File:

• Write a program that opens a file named "log.txt" in append mode. Append the current date and time to the file. Close the file after appending.

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

int main() {

    FILE *file = fopen("C:/Users/student/Downloads/log.txt", "a");

    if (file == NULL) {
        perror("Error opening file");
        return 1;
    }

    time_t now;
    time(&now);
    char *date_time = ctime(&now);

    fprintf(file, "Current date and time: %s", date_time);

    fclose(file);

    return 0;
}
```

log.txt

```
Hello, World! I am 21MIS1001Current date and time: Tue Jul 30 12:39:23 2024
```

Copy a File:

• Write a program to copy the contents of a file named "source.txt" to a new file named "destination.txt".

```c
#include <stdio.h>

int main() {

    FILE *sourceFile = fopen("C:/Users/student/Downloads/source.txt", "r");

    if (sourceFile == NULL) {
        perror("Error opening source file");
        return 1;
    }

    FILE *destinationFile = fopen("C:/Users/student/Downloads/destination.txt", "w");

    if (destinationFile == NULL) {
        perror("Error opening destination file");
        fclose(sourceFile);
        return 1;
    }

    int ch;
    while ((ch = fgetc(sourceFile)) != EOF) {
        fputc(ch, destinationFile);
    }

    fclose(sourceFile);
    fclose(destinationFile);

    return 0;
}
```

source.txt

```
Hello, I am Sabbarish S
```

destination.txt

```
Hello, I am Sabbarish S
```

Count the Number of Lines in a File:

• Write a program that opens a file named "data.txt" and counts the number of lines in the file. Print the total number of lines.

```c
#include <stdio.h>

int main() {

    FILE *file = fopen("C:/Users/student/Downloads/data.txt", "r");

    if (file == NULL) {
        perror("Error opening file");
        return 1;
    }

    int lineCount = 0;
    int ch;

    while ((ch = fgetc(file)) != EOF) {

        if (ch == '\n') {
            lineCount++;
        }
    }

    if (ftell(file) > 0 && ch != '\n') {
        lineCount++;
    }

    fclose(file);

    printf("Total number of lines: %d\n", lineCount);

    return 0;
}
```

```
Total number of lines: 5

--------------------------------
Process exited after 0.03027 seconds with return value 0
Press any key to continue . . .
```

```
include
main
data
integer
lineeee
```

Count the Number of Words in a File:

• Write a program to count the number of words in a file named "text.txt". Assume that words are separated by spaces, tabs, or newlines

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

int main() {

    FILE *file = fopen("C:/Users/student/Downloads/text.txt", "r");

    if (file == NULL) {
        perror("Error opening file");
        return 1;
    }

    int wordCount = 0;
    int inWord = 0;
    int ch;

    while ((ch = fgetc(file)) != EOF) {

        if (isspace(ch)) {

            if (inWord) {
                wordCount++;
                inWord = 0;
            }
        } else {

            inWord = 1;
        }
    }

    if (inWord) {
        wordCount++;
    }

    fclose(file);

    printf("Total number of words: %d\n", wordCount);

    return 0;
}
```

```
Total number of words: 6

--------------------------------
Process exited after 0.01144 seconds with return value
0
Press any key to continue . . .
```

```
include main
main
data
integer
line
```

Creation of Symbol table

Write a C program that implements a symbol table with functions to create, insert, modify, search, and display symbols. The program reads symbols from a file named "input.txt", with each symbol assigned a starting address of 1000 and each symbol occupying 3 bytes.

Requirements:

- Read symbols from "input.txt".
- Assign addresses starting from 1000, with each symbol occupying 3 bytes.
- Store each symbol and its corresponding address in a symbol table.
- Print the symbol table in a formatted manner

[*] ex1.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_SYMBOLS 100
#define SYMBOL_SIZE 256

typedef struct {
    char symbol[SYMBOL_SIZE];
    int address;
} Symbol;

Symbol symbolTable[MAX_SYMBOLS];
int symbolCount = 0;

void readSymbolsFromFile(const char *filename);
void insertSymbol(const char *symbol, int address);
void displaySymbolTable();

int main() {

    readSymbolsFromFile("C:/Users/student/Downloads/input.txt");


    displaySymbolTable();

    return 0;
}

void readSymbolsFromFile(const char *filename) {
    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        perror("Error opening file");
        exit(EXIT_FAILURE);
    }

    char symbol[SYMBOL_SIZE];
    int address = 1000;
    int i;

    while (fgets(symbol, sizeof(symbol), file) != NULL) {

        symbol[strcspn(symbol, "\n")] = '\0';

        if (strlen(symbol) > 0) {
```

C:\Users\student\Docume

```
Symbol Table:
Symbol                Address
-------------------- ----------
START                 1000
LOAD                  1003
ADD                   1006
STORE                 1009
END                   1012

--------------------------------
Process exited after 0.03208 seconds with return
value 0
Press any key to continue . . .
```

LAB-1 - [LAB-1.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

LAB-1

[*] ex1.c

```c
        exit(EXIT_FAILURE);
    }

    char symbol[SYMBOL_SIZE];
    int address = 1000;
    int i;

    while (fgets(symbol, sizeof(symbol), file) != NULL) {

        symbol[strcspn(symbol, "\n")] = '\0';

        if (strlen(symbol) > 0) {

            insertSymbol(symbol, address);

            address += 3;
        }
    }

    fclose(file);
}

void insertSymbol(const char *symbol, int address) {
    if (symbolCount < MAX_SYMBOLS) {
        strncpy(symbolTable[symbolCount].symbol, symbol, SYMBOL_SIZE - 1);
        symbolTable[symbolCount].symbol[SYMBOL_SIZE - 1] = '\0';
        symbolTable[symbolCount].address = address;
        symbolCount++;
    } else {
        fprintf(stderr, "Symbol table is full. Cannot insert new symbol.\n");
    }
}

void displaySymbolTable() {
    printf("Symbol Table:\n");
    printf("%-20s %-10s\n", "Symbol", "Address");
    printf("-------------------- ----------\n");

    int i;
    for (i = 0; i < symbolCount; i++) {
        printf("%-20s %d\n", symbolTable[i].symbol, symbolTable[i].address);
    }
}
```

C:\Users\student\Docume

```
Symbol Table:
Symbol               Address
-------------------- ----------
START                1000
LOAD                 1003
ADD                  1006
STORE                1009
END                  1012

--------------------------------
Process exited after 0.03208 seconds with return
value 0
Press any key to continue . . .
```