Lab 1

Name: Oshoumya Verma
Register Number: 220905540
Roll Number:58
Title: Basic File Handling Operations

Solved Example

C program to copy the contents of source file to destination file

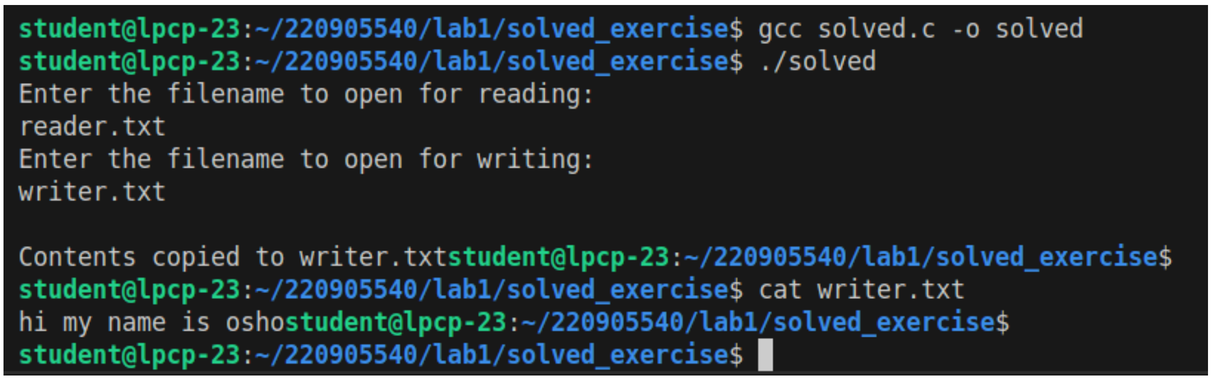
```
#include <stdio.h>
#include <stdlib.h> // For exit()

int main()
{
        FILE *fptr1, *fptr2;
        char filename[100], c;

        printf("Enter the filename to open for reading: \n");
        scanf("%s", filename);

        // Open the file for reading
        fptr1 = fopen(filename, "r");
        if (fptr1 == NULL)
        {
        printf("Cannot open file %s \n", filename);
        exit(0);
        }

        printf("Enter the filename to open for writing: \n");
        scanf("%s", filename);

        // Open another file for writing
        fptr2 = fopen(filename, "w+");
        if (fptr2 == NULL)
        {
        printf("Cannot open file %s \n", filename);
        fclose(fptr1);
        exit(0);
        }

        // Read contents from the first file and write to the second file
        c = fgetc(fptr1);
        while (c != EOF)
        {
        fputc(c, fptr2);
        c = fgetc(fptr1);
        }

        printf("\nContents copied to %s", filename);

        fclose(fptr1);
        fclose(fptr2);

        return 0;
}
```

Create 2 files named reader.txt and writer.txt in the same directory. Write your input in reader.txt file. Leave writer.txt empty. Then compile and execute the code.

Contents of reader.txt:

Hi my name is osho

Output



```
student@lpcp-23:~/220905540/lab1/solved_exercise$ gcc solved.c -o solved
student@lpcp-23:~/220905540/lab1/solved_exercise$ ./solved
Enter the filename to open for reading:
reader.txt
Enter the filename to open for writing:
writer.txt

Contents copied to writer.txtstudent@lpcp-23:~/220905540/lab1/solved_exercise$
student@lpcp-23:~/220905540/lab1/solved_exercise$ cat writer.txt
hi my name is oshostudent@lpcp-23:~/220905540/lab1/solved_exercise$
student@lpcp-23:~/220905540/lab1/solved_exercise$
```

Lab exercises

1. C program to count the number of lines and characters in a file

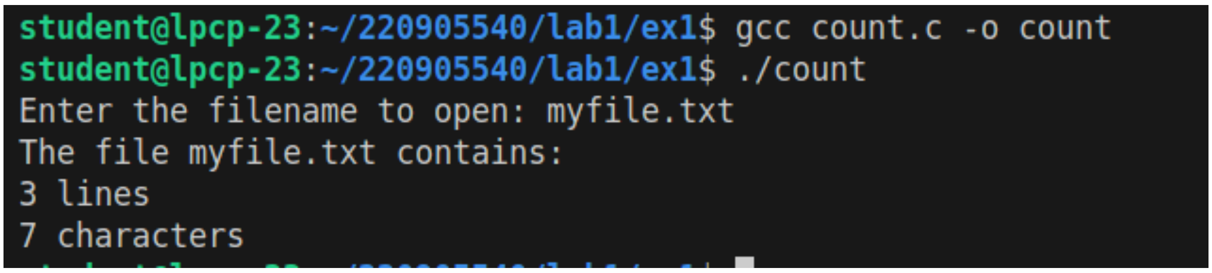
```
#include <stdio.h>
#include <stdlib.h> // For exit()

int main()
{
    FILE *fptr;
    char filename[100], c;
    int lineCount = 0, charCount = 0;

    // Prompt user for the filename
    printf("Enter the filename to open: ");
    scanf("%s", filename);

    // Open the file for reading
    fptr = fopen(filename, "r");
    if (fptr == NULL)
    {
    printf("Cannot open file %s \n", filename);
    exit(0);
    }

    // Read file character by character
    while ((c = fgetc(fptr)) != EOF)
    {
    charCount++; // Increment character count
    if (c == '\n')
    {
    lineCount++; // Increment line count on newline
    }
    }

    // Close the file
    fclose(fptr);

    // Display the results
    printf("The file %s contains:\n", filename);
    printf("%d lines\n", lineCount);
    printf("%d characters\n", charCount);

    return 0;
}
```

Create a file called myfile.txt in the same directory.

Contents of myfile.txt:

```
1
1
1
1
```

Output



```
student@lpcp-23:~/220905540/lab1/ex1$ gcc count.c -o count
student@lpcp-23:~/220905540/lab1/ex1$ ./count
Enter the filename to open: myfile.txt
The file myfile.txt contains:
3 lines
7 characters
```

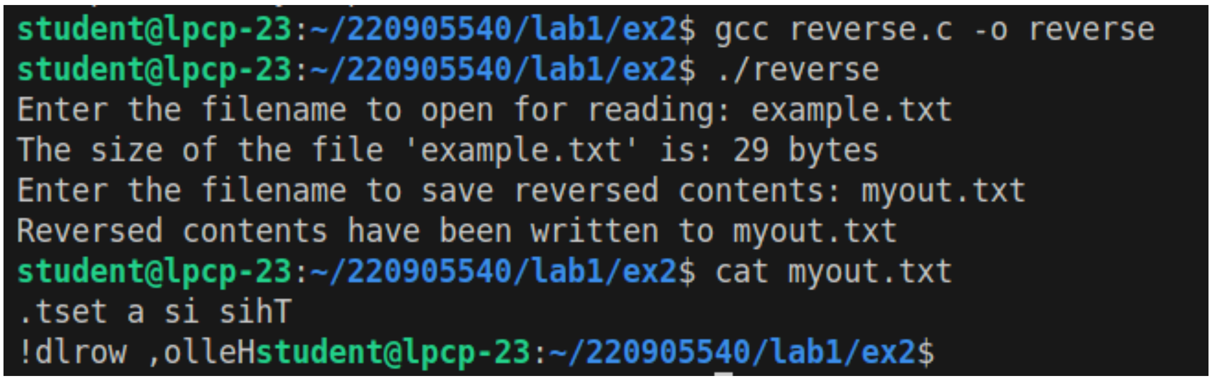
2. C program to reverse the file contents and store in another file. Also display the size of file using file handling function.

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

void reverseFileContents(const char *inputFile, const char *outputFile);

int main()
{
        char inputFile[100], outputFile[100];
        FILE *fptr;
        long fileSize;

        // Prompt user for input file name
        printf("Enter the filename to open for reading: ");
        scanf("%s", inputFile);

        // Open the input file in read mode
        fptr = fopen(inputFile, "r");
        if (fptr == NULL)
        {
        printf("Cannot open file %s\n", inputFile);
        exit(0);
        }

        // Determine the file size
        fseek(fptr, 0, SEEK_END); // Move the file pointer to the end
        fileSize = ftell(fptr);   // Get the current position (file size)
        fclose(fptr);

        printf("The size of the file '%s' is: %ld bytes\n", inputFile, fileSize);

        // Prompt user for output file name
        printf("Enter the filename to save reversed contents: ");
        scanf("%s", outputFile);
        // Reverse the contents of the file
        reverseFileContents(inputFile, outputFile);

        printf("Reversed contents have been written to %s\n", outputFile);

        return 0;
}

void reverseFileContents(const char *inputFile, const char *outputFile)
{
        FILE *fptr1, *fptr2;
        long fileSize, i;
        char c;

        // Open the input file in read mode
        fptr1 = fopen(inputFile, "r");
        if (fptr1 == NULL)
        {
        printf("Cannot open file %s\n", inputFile);
        exit(0);
        }

        // Open the output file in write mode
        fptr2 = fopen(outputFile, "w");
        if (fptr2 == NULL)
        {
        printf("Cannot open file %s\n", outputFile);
        fclose(fptr1);
        exit(0);
        }

        // Determine the file size
        fseek(fptr1, 0, SEEK_END);
        fileSize = ftell(fptr1);

        // Read and write the file contents in reverse order
        for (i = fileSize - 1; i >= 0; i--)
        {
        fseek(fptr1, i, SEEK_SET); // Move the file pointer to the position
        c = fgetc(fptr1);          // Read the character
        fputc(c, fptr2);           // Write the character to the output file
        }

        // Close the files
        fclose(fptr1);
        fclose(fptr2);
}
```

Create a file names example.txt with the text:

Hello, world!
This is a test.

Output



```
student@lpcp-23:~/220905540/lab1/ex2$ gcc reverse.c -o reverse
student@lpcp-23:~/220905540/lab1/ex2$ ./reverse
Enter the filename to open for reading: example.txt
The size of the file 'example.txt' is: 29 bytes
Enter the filename to save reversed contents: myout.txt
Reversed contents have been written to myout.txt
student@lpcp-23:~/220905540/lab1/ex2$ cat myout.txt
.tset a si sihT
!dlrow ,olleHstudent@lpcp-23:~/220905540/lab1/ex2$
```

3. C program that merges lines alternatively from 2 files and stores it in a resultant file.

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

void mergeFiles(const char *file1, const char *file2, const char *resultFile);

int main()
{
        char file1[100], file2[100], resultFile[100];

        // Prompt user for input filenames
        printf("Enter the first filename: ");
        scanf("%s", file1);

        printf("Enter the second filename: ");
        scanf("%s", file2);

        // Prompt user for output filename
        printf("Enter the resultant filename: ");
        scanf("%s", resultFile);

        // Merge the files
        mergeFiles(file1, file2, resultFile);

        printf("Lines have been merged alternately into %s\n", resultFile);
        return 0;
}

void mergeFiles(const char *file1, const char *file2, const char *resultFile)
{
        FILE *fptr1, *fptr2, *fptrResult;
        char line[256];

        // Open the first file for reading
        fptr1 = fopen(file1, "r");
        if (fptr1 == NULL)
        {
        printf("Cannot open file %s\n", file1);
        exit(0);
        }

        // Open the second file for reading
        fptr2 = fopen(file2, "r");
        if (fptr2 == NULL)
        {
        printf("Cannot open file %s\n", file2);
        fclose(fptr1);
        exit(0);
        }

        // Open the resultant file for writing
        fptrResult = fopen(resultFile, "w");
        if (fptrResult == NULL)
        {
        printf("Cannot open file %s\n", resultFile);
        fclose(fptr1);
        fclose(fptr2);
        exit(0);
        }

        // Merge lines alternately
        while (!feof(fptr1) || !feof(fptr2))
        {
        // Read and write a line from the first file
        if (fgets(line, sizeof(line), fptr1) != NULL)
        {
        fputs(line, fptrResult);
        }

        // Read and write a line from the second file
        if (fgets(line, sizeof(line), fptr2) != NULL)
        {
        fputs(line, fptrResult);
        }
        }

        // Close all files
        fclose(fptr1);
        fclose(fptr2);
        fclose(fptrResult);
}
```

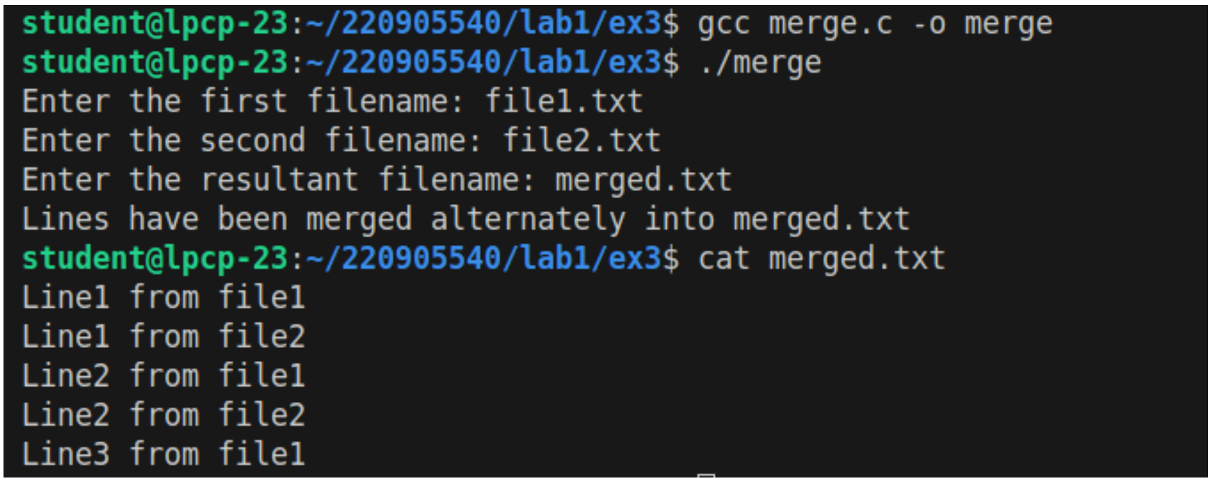
Create 2 files, file1.txt and file2.txt in the same directory with the following inputs

File1.txt:
Line1 from file1
Line2 from file1
Line3 from file1

File2.txt:

Line1 from file2
Line2 from file2

Output



```
student@lpcp-23:~/220905540/lab1/ex3$ gcc merge.c -o merge
student@lpcp-23:~/220905540/lab1/ex3$ ./merge
Enter the first filename: file1.txt
Enter the second filename: file2.txt
Enter the resultant filename: merged.txt
Lines have been merged alternately into merged.txt
student@lpcp-23:~/220905540/lab1/ex3$ cat merged.txt
Line1 from file1
Line1 from file2
Line2 from file1
Line2 from file2
Line3 from file1
```