

Report 12

C-Strings and File Operations

In this laboratory wor I have created 2 different programs for two tasks - function creating and file operations

First file mainFunc.c has a strcmp function, whose main task is strings comparing.

Function name	Purpose	Parameters	Return value	Example Usage
strcmp	Compare two strings	const char *firstString, const char *secondString	Integer value - 0, if strings equal, some ither value if not value	strcmp(firstInputOne, firstInputTwo);

Here is documentation of my function, and also output for three compinations of string

```
(malex-kali@MA)-[~/Programming/Lab12]
└─$ gcc -g -O0 mainFunc.c -o expr

(malex-kali@MA)-[~/Programming/Lab12]
└─$ gdb ./expr
GNU gdb (Debian 13.2-1+b2) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later http://gnu.org/licenses/gpl.html
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
https://www.gnu.org/software/gdb/bugs/.
Find the GDB manual and other documentation resources online at:
http://www.gnu.org/software/gdb/documentation/.

For help, type "help".
--Type for more, q to quit, c to continue without paging--c
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./expr...
(gdb) r
Starting program: /home/malex-kali/Programming/Lab12/expr
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
```

```
First option - "gfg" and "gfg"
Strings are equal
Value returned by strcmp() is: 0

Second option - "gFg" and "gfg"
Strings are unequal
Value returned by strcmp() is: -32

Third option - "gfg" and "gFg"
Strings are unequal
Value returned by strcmp() is: 32

[Inferior 1 (process 5089) exited normally]
(gdb) q
```

Second one file main.c is a file operating programm. It can take input file, read it, modify - replace all TODO words with DONE and and calculate amount of lines in the file. After this programm will return new file, which will include all changes.

Here I will include input and output files and their full versions will also be included as separated files

Input

```
// TODO: Review the header files
#include <stdio.h>
#include <stdlib.h>

// TODO: Add error handling
int main() {
    printf("Starting the program...\n"); // TODO: Add a debug log
    int result = 0;

    // TODO: Implement the calculation logic
    for (int i = 0; i < 10; i++) {
        result += i; // TODO: Optimize this loop
    }

    printf("Result: %d\n", result);

    // TODO: Write unit tests for this function }

    // TODO: Add additional helper functions here void helperFunction() { // TODO: Implement
    this function }
```

And the output file

```
// DONE: Review the header files
#include <stdio.h>
#include <stdlib.h>
```

```
// DONE: Add error handling
int main() {
printf("Starting the program...\n"); // DONE: Add a debug log
int result = 0;

// DONE: Implement the calculation logic
for (int i = 0; i < 10; i++) {
result += i; // DONE: Optimize this loop
}

printf("Result: %d\n", result);

// DONE: Write unit tests for this function }

// DONE: Add additional helper functions here
void helperFunction() {
// DONE: Implement this function
}
```

Total Line Count: 23