Operating Systems

Assignment 4

Roll no:	200901018
Name:	Muhammad Tayyab
Batch & Section:	BS-CS-01 (B)
Date of Submission:	04/01/2023

String Manipulation Using Threads

Task:

You must create four threads other than main thread. 1. Input thread 2. Reverse thread 3. Capital thread 4. Shift thread Input thread will take string input from user, reverse thread will reverse the string and output it, capital thread will capitalize the characters of string and output it and shift thread will shift each character of the string two time (e.g., a will become c) and output it. All the threads wait for input thread when input thread finishes his task all the waiting thread start their work simultaneously. You also must handle the exceptions of input thread. Also take care the state of each thread. Do not waste your memory resources.

Solution:

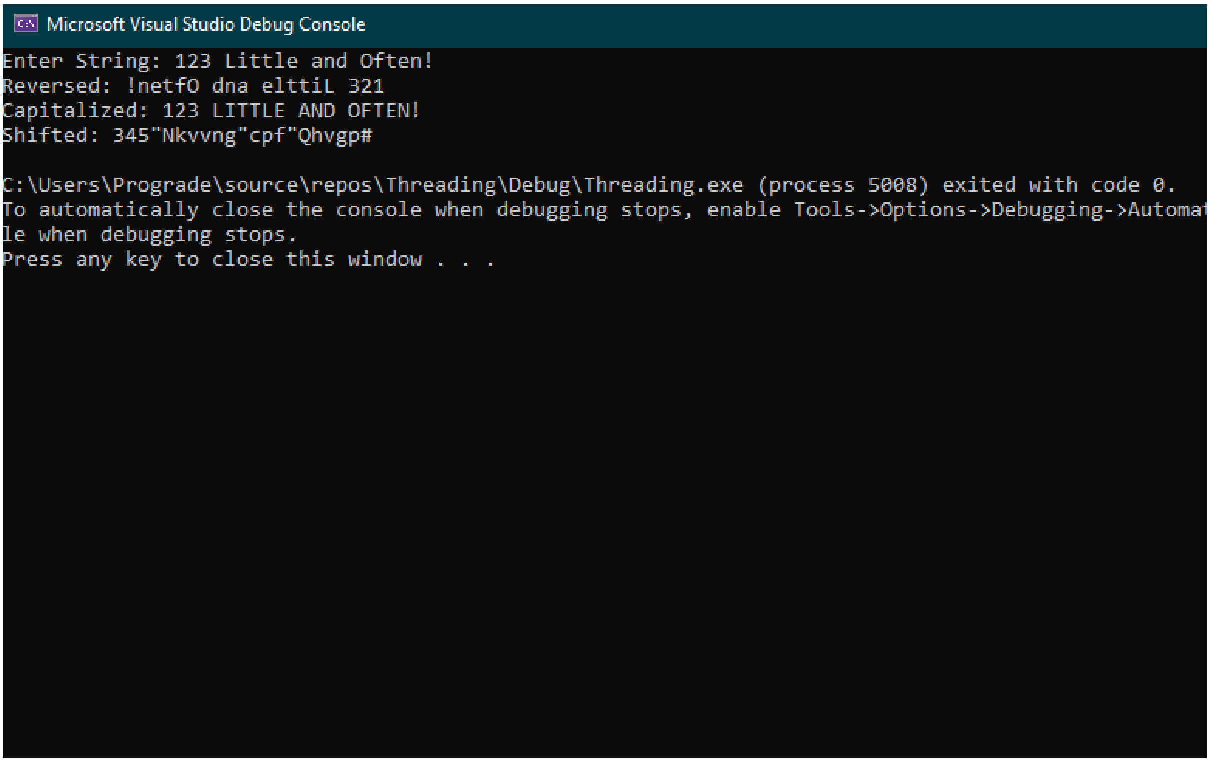
Here is an outline of the procedure for performing string manipulation using thread functions in C++:

- **Include the necessary header files:**
  To use thread functions in C++, you will need to include the <thread> header file. You may also need to include other header files, such as <string> for string manipulation and <iostream> for input/output operations.
- **Create a thread:**
  To create a thread in C++, you can use the std :: thread class, which is defined in the <thread> header file. You can create a new thread by calling the std :: thread constructor and passing it a function to execute as the thread's entry point.
- **Pass the string to the thread function**:
  To pass a string to a thread function, you can simply include it as an argument to the function.
- **Perform string manipulation in the thread function:**
  Once the string has been passed to the thread function, you can perform any desired string manipulation on it within the function. The <string> header file provides a variety of functions and methods for manipulating strings in C++, such as find, substr, and replace.
- **Join or detach the thread:**
  Once you have finished manipulating the string in the thread function, you will need to either join or detach the thread. Joining a thread blocks the calling thread until the target thread finishes execution, while detaching a thread allows the target thread to run freely in the background. To join a thread, you can call the join member function on the std::thread object. To detach a thread, you can call the `detach` member function.

Code:

```cpp
#include <iostream>
#include <string>
#include <thread>
#include <future>

using namespace std;

string TakeUserInput(string &InputString) {
    cout << "Enter String: "; getline(cin, InputString);
    return InputString;
};

void ReverseString(string UserString) {
    int StringSize = UserString.size();
    cout << "Reversed: ";
    for (int i = StringSize; i >= 0; i--){
        cout << UserString[i];
    }
    cout << endl;
};

void CapitalizeString(string UserString) {
    cout << "Capitalized: ";
    for (int i = 0; i < UserString.size(); i++)
    {
        putchar(toupper(UserString[i]));
    }
    cout << endl;
};

void ShiftString(string UserString) {
    cout << "Shifted: ";
    for (int i = 0; i < UserString.size(); i++)
    {
        putchar(UserString[i] + 2);
    }
    cout << endl;
};


int main() {
    string UserString;

    thread InputThread(TakeUserInput, ref(UserString));
    InputThread.join();

    thread ReverseThread(ReverseString, UserString);
    thread CapitalizeThread(CapitalizeString, UserString);
    thread ShiftThread(ShiftString, UserString);

    ReverseThread.join();
    CapitalizeThread.join();
    ShiftThread.join();
}
```

Output:



```
Enter String: 123 Little and Often!
Reversed: !netfO dna elttiL 321
Capitalized: 123 LITTLE AND OFTEN!
Shifted: 345"Nkvvng"cpf"Qhvgp#

C:\Users\Prograde\source\repos\Threading\Debug\Threading.exe (process 5008) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automat
le when debugging stops.
Press any key to close this window . . .
```

Github Profile Link:

https://github.com/ProgradeX

Task Link:

https://github.com/ProgradeX/Semester-Project-Archive/tree/master/OS%20Assignments%20and%20Projects