

Operating System Assignment # 04

Implementation Of Threads on String

Made by:

MUHAMMAD QALB E ALI

200901105 CS-01 B

Instructor: Ma'am Asia

4/1/23

Table of Contents

Code	2
Code Explanation	4
GitHub Link	5

Code

```
import threading

# Thread 1: Input thread
def input_thread():
    while True:
        try:
            # Get input from user
            user_input = input("Enter a string: ")
        except Exception as e:
            print("Exception occurred in input thread:", e)
        else:
            # Notify other threads to start working
            reverse_event.set()
            capital_event.set()
            shift_event.set()

# Thread 2: Reverse thread
def reverse_thread():
    while True:
        # Wait for input thread to finish
        reverse_event.wait()
        try:
            # Reverse the string
            reversed_string = user_input[::-1]
            print("Reversed string:", reversed_string)
        except Exception as e:
            print("Exception occurred in reverse thread:", e)
        finally:
            # Reset event for next iteration
```

```

reverse_event.clear()

# Thread 3: Capital thread
def capital_thread():
    while True:
        # Wait for input thread to finish
        capital_event.wait()

        try:
            # Capitalize the string
            capitalized_string = user_input.upper()
            print("Capitalized string:", capitalized_string)
        except Exception as e:
            print("Exception occurred in capital thread:", e)
        finally:
            # Reset event for next iteration
            capital_event.clear()

# Thread 4: Shift thread
def shift_thread():
    while True:
        # Wait for input thread to finish
        shift_event.wait()

        try:
            # Shift the string
            shifted_string = ""
            for c in user_input:
                shifted_string += chr(ord(c) + 2)
            print("Shifted string:", shifted_string)
        except Exception as e:
            print("Exception occurred in shift thread:", e)

```

```

finally:

    # Reset event for next iteration
    shift_event.clear()

# Create events for synchronization
reverse_event = threading.Event()
capital_event = threading.Event()
shift_event = threading.Event()

# Create and start threads
input_t = threading.Thread(target=input_thread)
reverse_t = threading.Thread(target=reverse_thread)
capital_t = threading.Thread(target=capital_thread)
shift_t = threading.Thread(target=shift_thread)

input_t.start()
reverse_t.start()
capital_t.start()
shift_t.start()

```

Code Explanation

- The **input_thread** function is responsible for getting input from the user in an infinite loop. If an exception occurs while getting the input, it is caught and a message is printed. If no exception occurs, the **reverse_event**, **capital_event**, and **shift_event** events are set to notify the other threads that they can start working.
- The **reverse_thread**, **capital_thread**, and **shift_thread** functions all run in an infinite loop. Each of these functions waits for its respective event to be set, and then performs its task on the **user_input** string. If an exception occurs while performing the task, it is caught and a message is printed. Finally, the event is reset for the next iteration.
- The **reverse_event**, **capital_event**, and **shift_event** events are created to synchronize the threads.
- The **input_thread**, **reverse_thread**, **capital_thread**, and **shift_thread** functions are each run in their own separate threads. These threads are started simultaneously.

GitHub Link

<https://github.com/Qalb-E-Ali/IST-Sem-5-Implementation-Of-Threads-On-String>