

LAB5-PartB-Atomic_Flag

Visual Studio interface showing the execution of a C++ program (LAB5.cpp) using a debugger. The program is running on a 32-bit (x86) system, and the process is CPEN333-LAB5-B.exe.

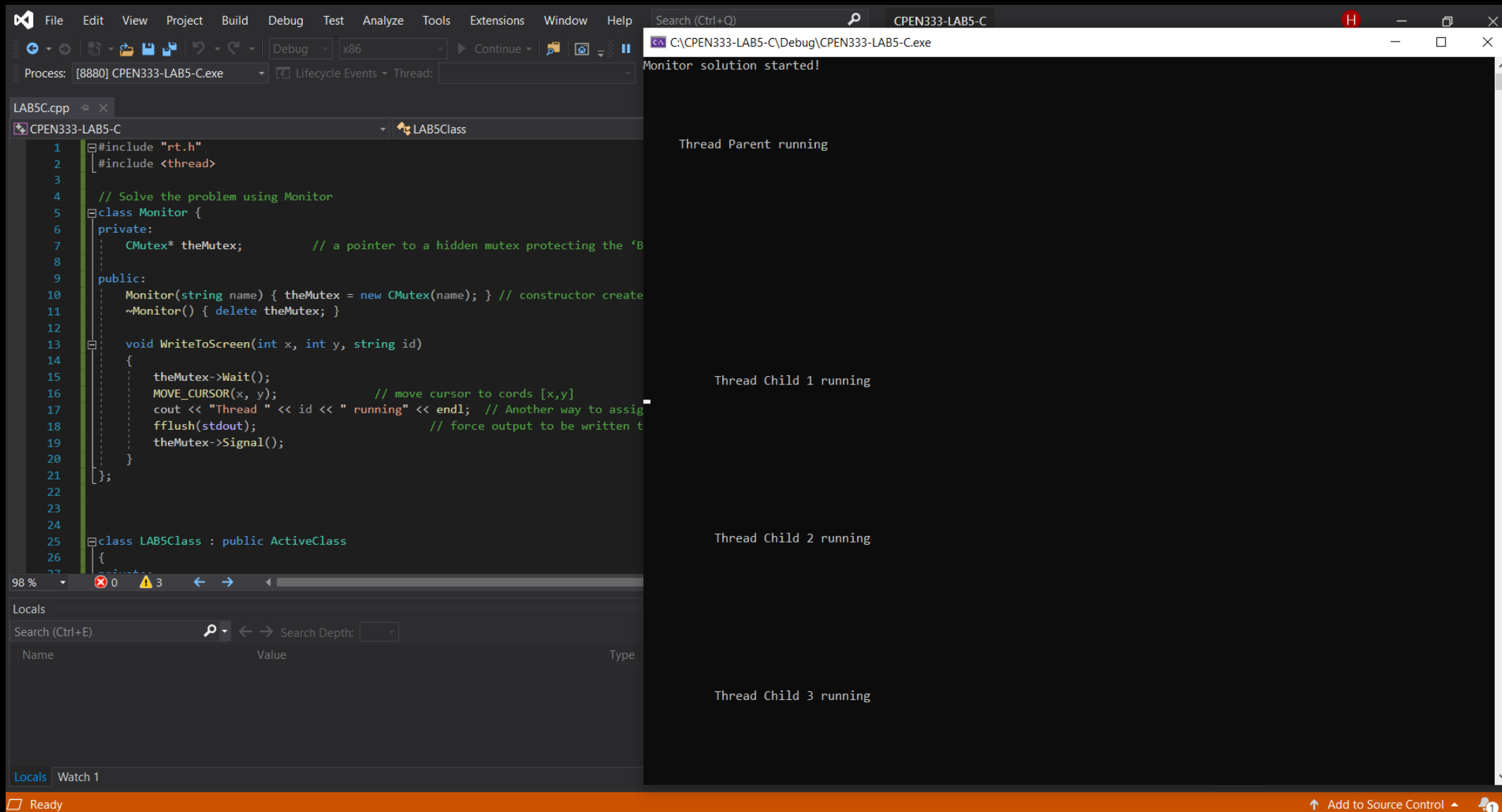
The code defines a `ChildThreads` function and a `main` function. The `ChildThreads` function takes a `void*` argument, creates a local mutex, and enters a loop where it waits for the mutex, moves the cursor, prints a message, and releases the mutex. The `main` function creates a local mutex, enters a loop, and prints "Parent Thread 1 running".

The debugger shows the execution flow, with the following output messages:

- CMutex solution started!
- Parent Thread 1 running
- Child Thread 1 running
- Child Thread 2 running
- Child Thread 3 running

The debugger interface includes a menu bar (File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window), a toolbar, a status bar (98 %), and a Locals/Watch panel.

LAB5-PartB-Mutex



LAB5-PartC-Monitor

Visual Studio IDE showing the execution of a C++ program (Q9.exe) and its child processes (Q9Child.exe and Q9Child2.exe) in a debug session.

The main window displays the source code of Q9.cpp, which implements a Producer-Consumer problem using semaphores and a data pool. The code includes comments and variable declarations for semaphores (ps1, ps2, cs1, cs2), a data pool (dp1), and two processes (p1, p2).

The output window shows the execution flow:

- Producer Process Running.....
- Hit RETURN to Produce Data.....
- Produced 0.....
- Hit RETURN to Produce Data.....
- Produced 1.....
- Hit RETURN to Produce Data.....
- Produced 2.....
- Hit RETURN to Produce Data.....
- Produced 3.....
- Hit RETURN to Produce Data.....
- Produced 4.....
- Hit RETURN to Produce Data.....
- Produced 5.....
- Hit RETURN to Produce Data.....
- Produced 6.....
- Hit RETURN to Produce Data.....
- Produced 7.....
- Hit RETURN to Produce Data.....
- Produced 8.....
- Hit RETURN to Produce Data.....

The child process windows show the consumer's execution:

- Consumer 1 Process Running.....
- Hit RETURN to Consume Data.....
- Consumed: 0
- Hit RETURN to Consume Data.....
- Consumed: 1
- Hit RETURN to Consume Data.....
- Consumed: 2
- Hit RETURN to Consume Data.....
- Consumed: 3
- Hit RETURN to Consume Data.....
- Consumed: 4
- Hit RETURN to Consume Data.....
- Consumed: 5
- Hit RETURN to Consume Data.....
- Consumed: 6
- Hit RETURN to Consume Data.....
- Consumed: 7
- Hit RETURN to Consume Data.....
- Consumed: 8
- Hit RETURN to Consume Data.....

The child process window Q9Child2.exe shows the consumer's execution:

- Consumer 2 Process Running.....
- Hit RETURN to Consume Data.....
- Consumed: 0
- Hit RETURN to Consume Data.....
- Consumed: 1
- Hit RETURN to Consume Data.....
- Consumed: 2
- Hit RETURN to Consume Data.....
- Consumed: 3
- Hit RETURN to Consume Data.....
- Consumed: 4
- Hit RETURN to Consume Data.....
- Consumed: 5
- Hit RETURN to Consume Data.....
- Consumed: 6
- Hit RETURN to Consume Data.....
- Consumed: 7
- Hit RETURN to Consume Data.....
- Consumed: 8
- Hit RETURN to Consume Data.....

The Output window shows the final state of the program:

- Show output from: Debug
- 'Q9.exe' (Win32): Loaded 'C:\Windows\SysWOW64\hit RETURN to Consume Data.....'
- 'Q9.exe' (Win32): Loaded 'C:\Windows\SysWOW64\hit RETURN to Consume Data.....'
- 'Q9.exe' (Win32): Loaded 'C:\Windows\SysWOW64\apphelp.dll'.
- The thread 0x2308 has exited with code 0 (0x0).
- The thread 0x4080 has exited with code 0 (0x0).
- The thread 0x990 has exited with code 0 (0x0).

LAB5-PartD-SingleProducer_DoubleConsumer

The screenshot displays the Visual Studio IDE with a C++ project named "Q9" open. The main window shows the source code for "Q9Child.cpp", which implements a Producer-Consumer problem using semaphores and data pools. The code includes comments and uses `std::mutex`, `std::semaphore`, and `std::condition_variable` for synchronization. The `main` function creates two data pools and two producers, and then runs a consumer process.

Below the code editor, two console windows are open, showing the output of the program. The left window shows the output of "Q9.exe" (the producer process), and the right window shows the output of "Q9Child.exe" (the consumer process).

Q9.exe Output:

```
Producer 1 Process Running.....
Hit RETURN to Produce Data.....

Produced 0.....
Hit RETURN to Produce Data.....

Produced 1.....
Hit RETURN to Produce Data.....

Produced 2.....
Hit RETURN to Produce Data.....

Produced 3.....
Hit RETURN to Produce Data.....

Produced 4.....
Hit RETURN to Produce Data.....

Produced A.....
Hit RETURN to Produce Data.....

Produced B.....
Hit RETURN to Produce Data.....

Produced C.....
Hit RETURN to Produce Data.....

Produced D.....
```

Q9Child.exe Output:

```
Consumer Process Running.....
Hit RETURN to Consume Data.....

Consumed: 0
Hit RETURN to Consume Data.....

Consumed: 1
Hit RETURN to Consume Data.....

Consumed: 2
Hit RETURN to Consume Data.....

Consumed: 3
Hit RETURN to Consume Data.....

Consumed: 4
Hit RETURN to Consume Data.....

Consumed: A
Hit RETURN to Consume Data.....

Consumed: B
Hit RETURN to Consume Data.....

Consumed: C
Hit RETURN to Consume Data.....

Consumed: D
```

LAB5-PartD-DoubleProducer_SingleConsumer