Sporadic Server

Shihab Ud Doula

Sporadic task VS Sporadic Server

Purposes

- Handling Sporadic Task Arrivals
- Timing Guarantees
- Resource Management:
- Flexibility and Adaptability
- System Responsiveness:
- Predictability and Determinism

TASK DESIGN OF SPORADIC SERVER

- In Context of RTS mathematical formula of sporadic server;
- $\Phi s=(Ps,Es, \theta, \rho)$

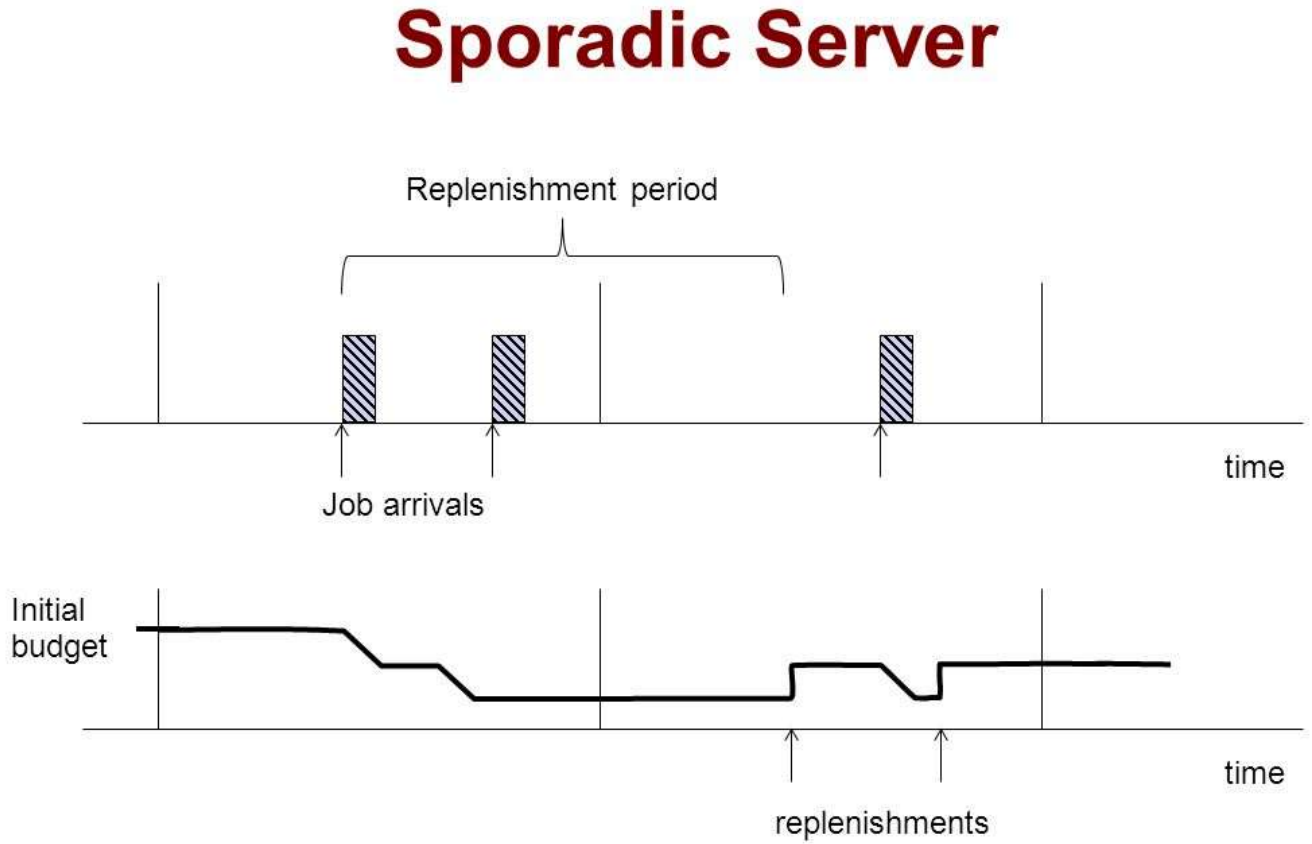
- PS represents the inter-arrival time.
- ES represents the execution time.
- θ represents the threshold time.
- ρ represents the replenishment.

Replenishment

- Refers to the process of restoring the server's resources or budget after it has served a task.
- Process:

- Internal Data Structures
- Resetting Variables
- Acquiring resources

Sporadic Server Scheduling



Deferrable Server

- Used As a conjunction of sporadic server
- Flexibility with uncertain arrival times
- Defer or postpone task with compromising performance

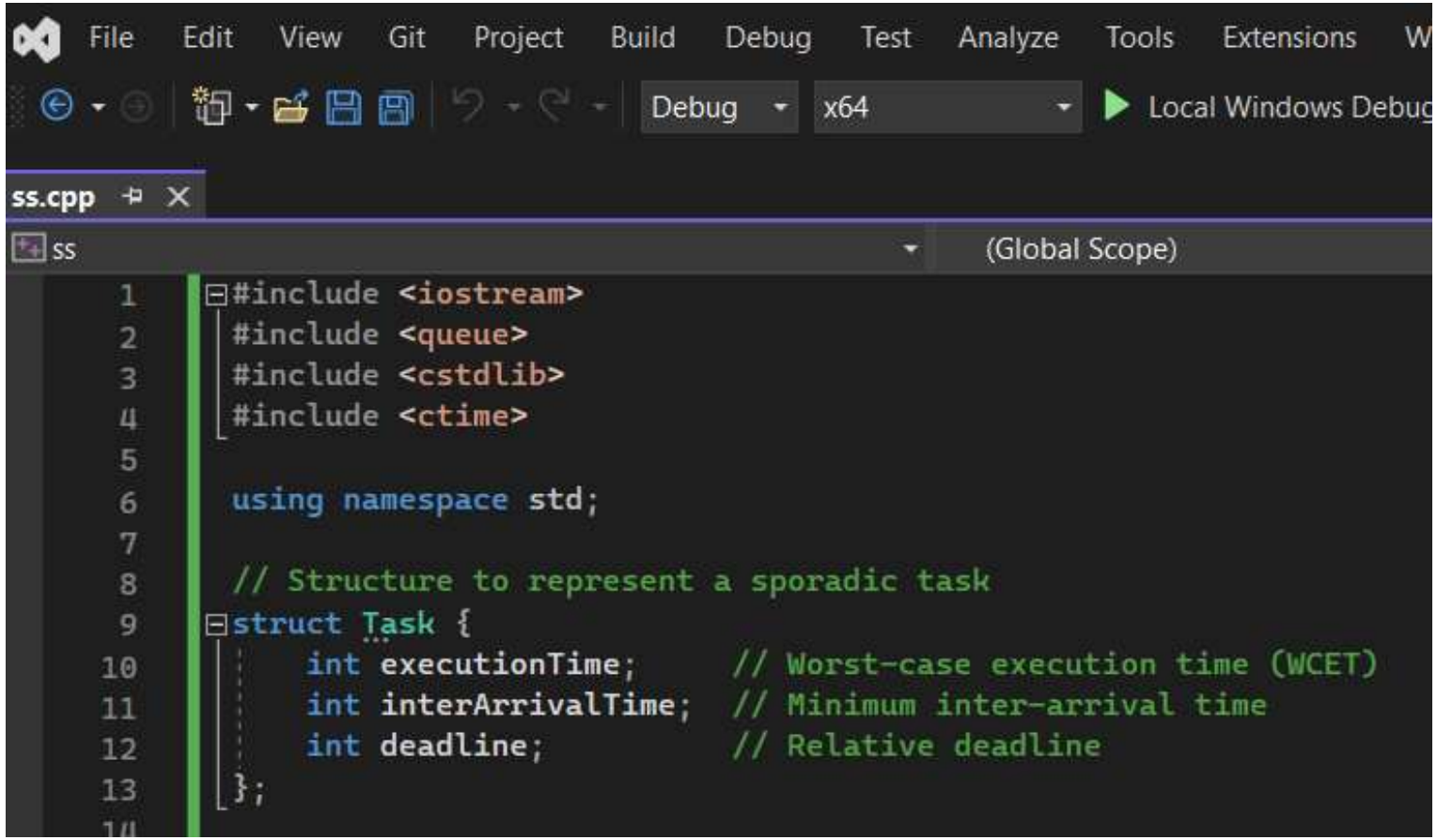
Future Implementation of DS

- Advanced Deferral Policies:
- Dynamic Deferral Time Calculations:
- Adaptive Task Prioritization:
- • Energy-Efficient Deferral Strategies:

About Code

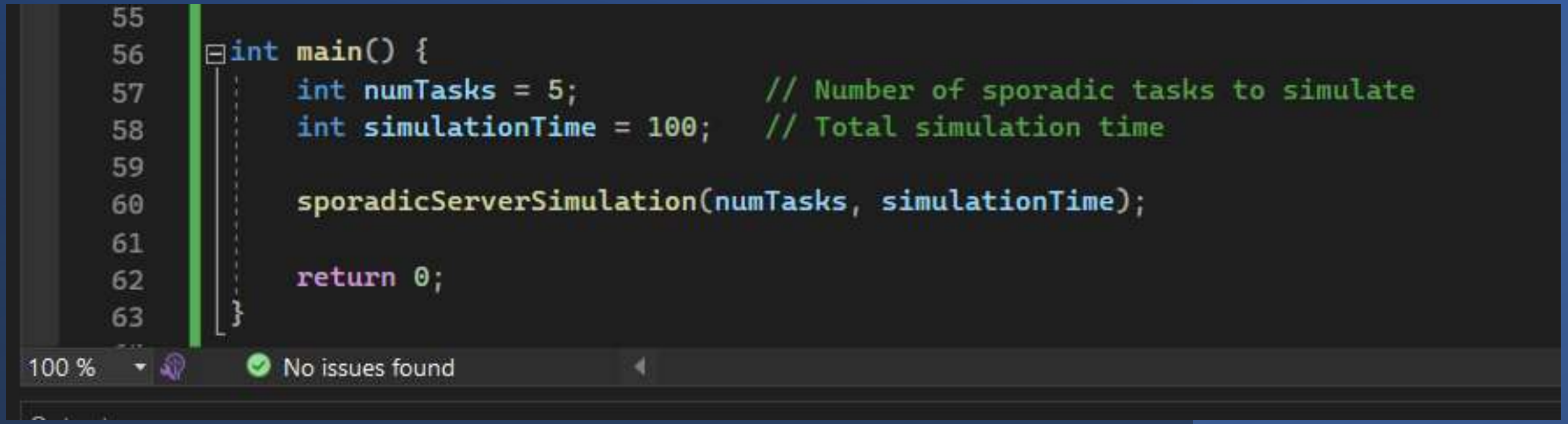
- In this code, the sporadic server simulation generates a specified number of sporadic tasks with random execution times, inter-arrival times, and deadlines. It then simulates the behavior of the sporadic server by executing the tasks based on their parameters and checking if they meet their deadlines. The simulation runs for a defined duration (simulationTime) and outputs the execution results for each task.

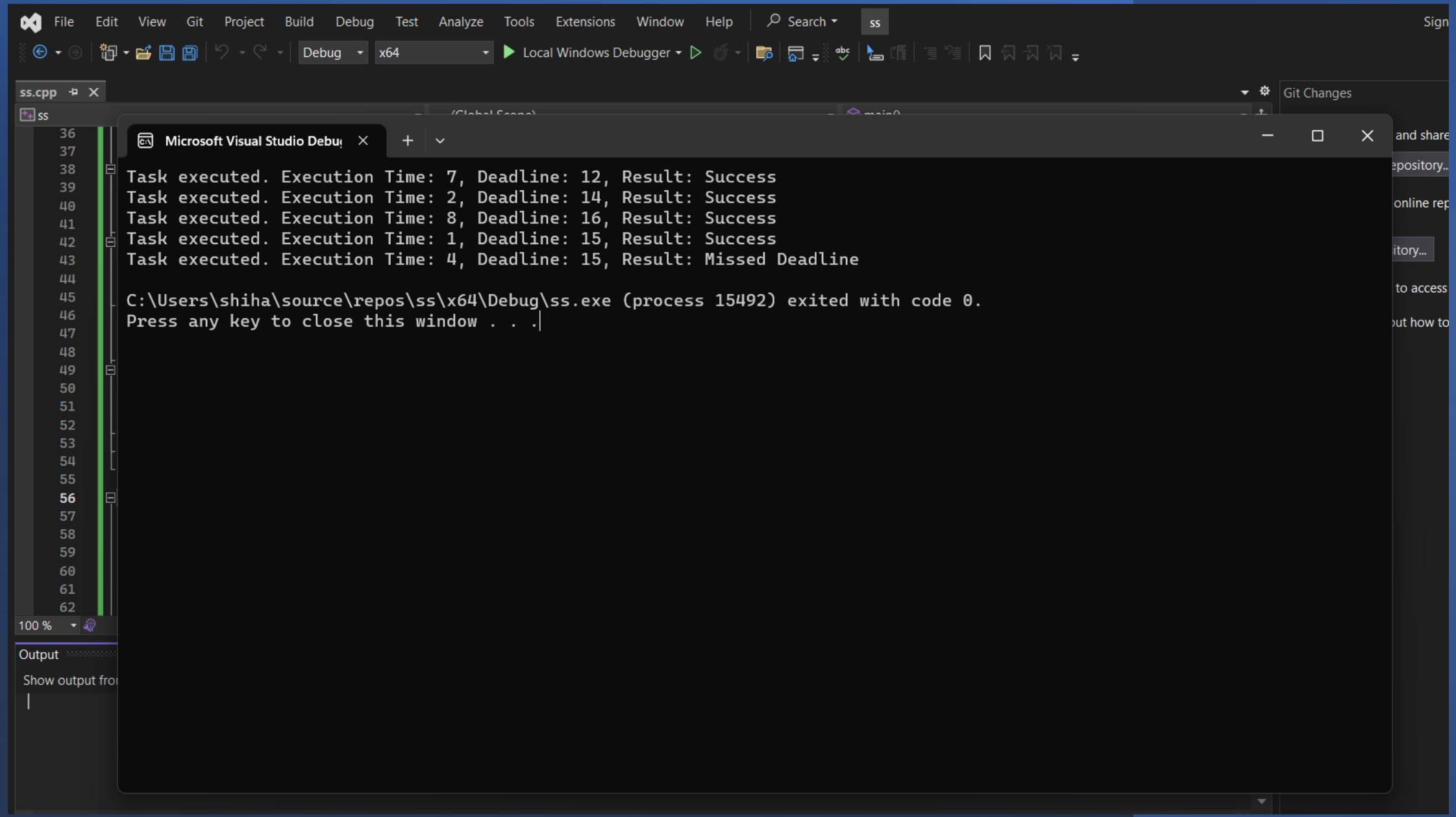
Implementation in C++



```
14
15  // Function to simulate the sporadic server
16  void sporadicServerSimulation(int numTasks, int simulationTime) {
17      srand(time(NULL));
18
19      queue<Task> taskQueue;
20      int currentTime = 0;
21
22      // Generate sporadic tasks with random parameters
23      for (int i = 0; i < numTasks; ++i) {
24          Task task;
25          task.executionTime = rand() % 10 + 1;      // Random execution time between 1 and 10
26          task.interArrivalTime = rand() % 10 + 1;   // Random inter-arrival time between 1 and 10
27          task.deadline = rand() % 10 + 10;          // Random relative deadline between 10 and 19
28          taskQueue.push(task);
29      }
30
```

```
30
31      // Simulate sporadic server behavior
32      while (currentTime <= simulationTime) {
33          if (!taskQueue.empty()) {
34              Task currentTask = taskQueue.front();
35              taskQueue.pop();
36
37              // Check if task can be executed within its allocated budget
38              if (currentTask.executionTime <= currentTask.deadline - currentTime) {
39                  cout << "Task executed. Execution Time: " << currentTask.executionTime
40                      << ", Deadline: " << currentTask.deadline << ", Result: Success" << endl;
41              }
42              else {
43                  cout << "Task executed. Execution Time: " << currentTask.executionTime
44                      << ", Deadline: " << currentTask.deadline << ", Result: Missed Deadline" <<
45              }
46
47              currentTime += currentTask.interArrivalTime;
48          }
49          else {
50              // No tasks in the queue, advance the time
51              currentTime++;
52          }
53      }
54  }
```


55
56 int main() {
57 int numTasks = 5; // Number of sporadic tasks to simulate
58 int simulationTime = 100; // Total simulation time
59
60 sporadicServerSimulation(numTasks, simulationTime);
61
62 return 0;
63 }
100 %
No issues found


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
ss
Debug
x64
Local Windows Debugger
ss.cpp
Git Changes
Microsoft Visual Studio Debu
Task executed. Execution Time: 7, Deadline: 12, Result: Success
Task executed. Execution Time: 2, Deadline: 14, Result: Success
Task executed. Execution Time: 8, Deadline: 16, Result: Success
Task executed. Execution Time: 1, Deadline: 15, Result: Success
Task executed. Execution Time: 4, Deadline: 15, Result: Missed Deadline
C:\Users\shiha\source\repos\ss\x64\Debug\ss.exe (process 15492) exited with code 0.
Press any key to close this window . . .
100 %
Output
Show output fro

Thank You.

- Questions?