**CS/CE/TE 6378.003 Advanced Operating Systems**

**Summer 2015**

**Testbed Implementation (10%)**

**Submission due by Friday, 7/24 at 11:59pm CT**

1. **Purpose**

The purpose of this assignment is for your team to implement its designed testbed.

1. **Assignment**

Your team is expected to **implement Maekawa’s algorithm** and **your new algorithm** in **your designed testbed**, which should allow for the algorithms to be empirically compared. Your team will use this testbed in the next assignment.

Your implementation of **your new algorithm** should adhere to the following requirements:

* Is **quorum-based** and includes **quorum assignments**.
* Is **token-based** with only **one token**.
* **Does not violate mutual exclusion** and **all requests are eventually granted**.

Your implementation of **Maekawa’s algorithm** should adhere to the following requirements:

* Is **faithful to the algorithm described** in Maekawa’s paper.
* Uses **the same quorum assignment method** that your algorithm uses.
* **Does not violate mutual exclusion** and **all requests are eventually granted**.

Your implementation of **your testbed** should adhere to the following requirements:

* **Supports both** your new algorithm and Maekawa’s algorithm.
* Is **easily configurable** to set which nodes make requests and when.
* Supports **at least 16 system nodes**.
* Can **demonstrate** that mutual exclusion **is not violated**.
* Can **demonstrate** that all requests are **eventually granted**.
* Records and reports identified **performance metrics**.

Finally, keep in mind that your software implementations **must be written in C/C++** and **capable of compiling and running on the CS Department’s networking machines** (i.e., netXX.utdallas.edu, where XX ∈ {01, 02, …, 45}).

1. **Submission**

Your team’s Manager and/or Lead Developer are expected to submit this assignment **through email to TA** by the deadline indicated above. The submission **must be a ZIP (.zip) file** containing **all of the files needed to compile and run your testbed**. **A README file is also required** and should instruct how to compile, setup, configure, and run your testbed. **Use your team name separated by spaces to name your file** (e.g., “AOS Comets.zip”).

1. **Grading**

Your team’s grade for this assignment will start at 10 points. For each criterion your team fails to meet, the team grade will be reduced by the indicated deduction. The minimum possible grade is 0.

* The implementation of your algorithm is quorum-based **(10 points)**.
* The implementation of your algorithm is token-based with only one token **(10 points)**.
* The implementation of your algorithm does not violate mutual exclusion **(5 points)**.
* The implementation of your algorithm eventually grants all requests **(5 points)**.
* The implementation of Maekawa’s algorithm is faithful to its description **(10 points)**.
* The implementation of Maekawa’s algorithm uses the same quorum assignment method as your algorithm uses **(10 points)**.
* The implementation of Maekawa’s algorithm does not violate mutual exclusion   
  **(5 points)**.
* The implementation of Maekawa’s algorithm eventually grants all requests **(5 points)**.
* The testbed supports both your new algorithm and Maekawa’s algorithm **(10 points)**.
* The testbed is easily configurable, with the ability to indicate which nodes makes requests and when **(5 points)**.
* The testbed supports at least 16 system nodes **(5 points)**.
* The testbed can demonstrate that mutual exclusion is not violated **(10 points)**.
* The testbed can demonstrate that all requests are eventually granted **(10 points)**.
* The testbed records and reports your predetermined performance metrics **(5 points)**.
* The assignment is submitted via eLearning by the indicated deadline **(2 points/day late)**.
* The submission is a ZIP (.zip) file **(2.5 points)**.
* The submission contains all the files required to compile, setup, configure, and run the testbed on the CS Department’s networking machines **(5 points)**.
* The submission contains a README file for setting up the prototype **(5 points)**.

1. **Academic Integrity**

While this is a group assignment, groups are expected to submit their own work. ***Code sharing between groups is strictly prohibited and will result in disciplinary proceedings***. Analysis software will be used to determine if code sharing has occurred.

**These descriptions and timelines are subject to change at the discretion of the instructor.**

**Demonstration Cases**

**CASE 1**

CONFIGURATION

|  |  |
| --- | --- |
| **Time** | **Requesting Nodes** |
| T1 | 1 |
| T2 | 2 |
| T3 | 3 |
| T4 | 4 |
| T5 | 5 |
| T6 | 6 |
| T7 | 7 |
| T8 | 8 |
| T9 | 9 |
| T10 | 10 |
| T11 | 11 |
| T12 | 12 |
| T13 | 13 |
| T14 | 14 |
| T15 | 15 |
| T16 | 16 |

POSSIBLE OUTPUT (order of nodes should not change)

Node 1 Enters; Node 1 Exits;

Node 2 Enters; Node 2 Exits;

Node 3 Enters; Node 3 Exits;

Node 4 Enters; Node 4 Exits;

Node 5 Enters; Node 5 Exits;

Node 6 Enters; Node 6 Exits;

Node 7 Enters; Node 7 Exits;

Node 8 Enters; Node 8 Exits;

Node 9 Enters; Node 9 Exits;

Node 10 Enters; Node 10 Exits;

Node 11 Enters; Node 11 Exits;

Node 12 Enters; Node 12 Exits;

Node 13 Enters; Node 13 Exits;

Node 14 Enters; Node 14 Exits;

Node 15 Enters; Node 15 Exits;

Node 16 Enters; Node 16 Exits;

**CASE 2**

CONFIGURATION

|  |  |
| --- | --- |
| **Time** | **Requesting Nodes** |
| T1 | 1, 2, 3, 4 |
| T2 | 5, 6, 7, 8 |
| T3 | 9, 10, 11, 12 |
| T4 | 13, 14, 15, 16 |

POSSIBLE OUTPUT (order of nodes may vary within every block of four, but a node from a later time should not precede a node from an earlier time)

Node 4 Enters; Node 4 Exits;

Node 1 Enters; Node 1 Exits;

Node 2 Enters; Node 2 Exits;

Node 3 Enters; Node 3 Exits; // End of Block 1

Node 5 Enters; Node 5 Exits;

Node 6 Enters; Node 6 Exits;

Node 7 Enters; Node 7 Exits;

Node 8 Enters; Node 8 Exits; // End of Block 2

Node 10 Enters; Node 10 Exits;

Node 9 Enters; Node 9 Exits;

Node 11 Enters; Node 11 Exits;

Node 12 Enters; Node 12 Exits; // End of Block 3

Node 13 Enters; Node 13 Exits;

Node 14 Enters; Node 14 Exits;

Node 15 Enters; Node 15 Exits;

Node 16 Enters; Node 16 Exits; // End of Block 4

**CASE 3**

CONFIGURATION

|  |  |
| --- | --- |
| **Time** | **Requesting Nodes** |
| T1 | 13, 14, 15, 16 |
| T2 | 9, 10, 11, 12 |
| T3 | 5, 6, 7, 8 |
| T4 | 1, 2, 3, 4 |

POSSIBLE OUTPUT (order of nodes may vary within every block of four, but a node from a later time should not precede a node from an earlier time)

Node 13 Enters; Node 13 Exits;

Node 14 Enters; Node 14 Exits;

Node 15 Enters; Node 15 Exits;

Node 16 Enters; Node 16 Exits; // End of Block 1

Node 10 Enters; Node 10 Exits;

Node 9 Enters; Node 9 Exits;

Node 11 Enters; Node 11 Exits;

Node 12 Enters; Node 12 Exits; // End of Block 2

Node 5 Enters; Node 5 Exits;

Node 6 Enters; Node 6 Exits;

Node 7 Enters; Node 7 Exits;

Node 8 Enters; Node 8 Exits; // End of Block 3

Node 4 Enters; Node 4 Exits;

Node 1 Enters; Node 1 Exits;

Node 2 Enters; Node 2 Exits;

Node 3 Enters; Node 3 Exits; // End of Block 4

**CASE 4**

CONFIGURATION

|  |  |
| --- | --- |
| **Time** | **Requesting Nodes** |
| T1 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 |

POSSIBLE OUTPUT (order of nodes may vary slightly)

Node 4 Enters; Node 4 Exits;

Node 1 Enters; Node 1 Exits;

Node 2 Enters; Node 2 Exits;

Node 3 Enters; Node 3 Exits;

Node 5 Enters; Node 5 Exits;

Node 6 Enters; Node 6 Exits;

Node 7 Enters; Node 7 Exits;

Node 8 Enters; Node 8 Exits;

Node 9 Enters; Node 9 Exits;

Node 10 Enters; Node 10 Exits;

Node 11 Enters; Node 11 Exits;

Node 12 Enters; Node 12 Exits;

Node 13 Enters; Node 13 Exits;

Node 14 Enters; Node 14 Exits;

Node 15 Enters; Node 15 Exits;

Node 16 Enters; Node 16 Exits;

**CASE 5**

CONFIGURATION

|  |  |
| --- | --- |
| **Time** | **Requesting Nodes** |
| T1 | 16 |
| T2 | 1 |
| T3 | 12 |
| T4 | 5 |
| T5 | 4 |
| T6 | 13 |
| T7 | 8 |
| T8 | 2 |
| T9 | 11 |
| T10 | 6 |
| T11 | 15 |
| T12 | 9 |
| T13 | 3 |
| T14 | 14 |
| T15 | 7 |
| T16 | 10 |

POSSIBLE OUTPUT (order of nodes should not change)

Node 16 Enters; Node 16 Exits;

Node 1 Enters; Node 1 Exits;

Node 12 Enters; Node 12 Exits;

Node 5 Enters; Node 5 Exits;

Node 4 Enters; Node 4 Exits;

Node 13 Enters; Node 13 Exits;

Node 8 Enters; Node 8 Exits;

Node 2 Enters; Node 2 Exits;

Node 11 Enters; Node 11 Exits;

Node 6 Enters; Node 6 Exits;

Node 15 Enters; Node 15 Exits;

Node 9 Enters; Node 9 Exits;

Node 3 Enters; Node 3 Exits;

Node 14 Enters; Node 14 Exits;

Node 7 Enters; Node 7 Exits;

Node 10 Enters; Node 10 Exits;