

CMSC 330: Organization of Programming Languages

Project 5 Multithreaded Space Simulation

Space Simulation

- ▶ You are given
 - List of starports & their sizes
 - List of starships & their sizes
 - List of travelers & their stops
 - Parser for reading simulation parameters / events
 - Code for printing simulation events
- ▶ You need to simulate
 - Starships moving between ports (in a fixed circle)
 - Travelers boarding / exiting starships

Simulation Parameters

► Format

```
=== Starports ===  
<name> <size>  
=== Starships ===  
<name> <size>  
=== Passengers ===  
<name> <port1> <port2>...  
=== Output ===  
<event>
```

Simulation Parameters

► Example

```
=== Starports ===  
Earth 2  
Vulcan 1  
=== Starships ===  
Enterprise 2  
=== Passengers ===  
Kirk Earth Vulcan Earth  
Spock Vulcan Earth  
=== Output ===
```

Simulation Events

► Format

- <starship> docking at <starport>
- <starship> departing from <starport>
- <traveler> boarding <starship> at <starport>
- <traveler> departing <starship> at <starport>

CMS3 330

5

Simulation Events

► Example

Enterprise docking at Earth
Kirk boarding Enterprise at Earth
Enterprise departing from Earth
Enterprise docking at Vulcan
Spock boarding Enterprise at Vulcan
Kirk departing Enterprise at Vulcan
Enterprise departing from Vulcan
Enterprise docking at Earth
Spock departing Enterprise at Earth
Enterprise departing from Earth

CMS3 330

6

Goals

1. Implement simulation display
 - Examine log file of simulation events
 - Display state of simulation
2. Implement simulation verifier
 - Examine log file of simulation events
 - Discover illegal / missing simulation events
3. Implement multithreaded simulation
 - Separate threads for starships, travelers
 - Use synchronization to avoid data races
 - Use wait / notify for efficiency

CMSC 330

7

Simulation Display

- ▶ Read simulation events & display state
- ▶ Example

Earth Kirk
Vulcan Spock

Enterprise docking at Earth

Earth Kirk
Enterprise
Vulcan Spock

CMSC 330

8

Simulation Verifier

- ▶ Read simulation events
- ▶ Check for data race & logic errors
 - Starships are traveling between starports in correct order
 - Starships always dock at a starport before leaving it
 - Starships do not exceed the capacity of a starport
 - Travelers follow their itinerary
 - Travelers only board /leave starship while it is at starport
 - Travelers do not exceed the capacity of a starship
 - All travelers have reached their final destination when simulation ends

CMSC 330

9

Space Simulation

- ▶ Multithreading
 - One thread per starship
 - One thread per traveler
- ▶ Synchronization
 - Starships dock at starport if size permits
 - Travelers only board / exit starship when in port
 - Use enough locks to permit concurrent execution
 - Use wait / notify to avoid busy waiting

CMSC 330

10

Notes

- ▶ Multithreaded simulation
 - Without using conditional variables, simulation may run very slowly and timeout due to busy waiting
 - Initially you may use a single monitor and conditional variable for the entire simulation
 - Later for more efficient implementation you should have a separate monitor for each starport, and multiple conditional variables for each monitor
- ▶ Simulation output
 - Need not be identical to example output provided
 - Lines beginning with % are ignored by submit server