IT UNIVERSITY OF CPH

DISYS Mandatory Exercise 2 - Distributed Mutual Exclusion

Gustav Christoffersen - `guch@itu.dk`
Jacob Walter Bentsen - `jawb@itu.dk`
Markus Grand Petersen - `mgrp@itu.dk`

DISYS
IT-Universitetet i København

November 17, 2021

GitHub Repository

https://github.com/Markusgp/DISYS_MandaExercise2

1 Implementation

Our implementation is a UDP-based approach to the Ricart-Agrawala algorithm for mutual exclusion on a distributed system.

The algorithm uses two types of messages: **REQUEST** and **REPLY**

- A process sends a **REQUEST** message to all other processes to request their permission to enter the *critical section*.
- A process sends a **REPLY** message to a process to give its permission.

Processes use **Lamport-style logical clocks** to assign timestamp to critical section requests and timestamps are used to decide the priority of requests should they appear simultaneously.

We're aware that a requirement of the code, is usage of gRPC for communication between nodes. But we decided to follow a more bare-bones UDP implementation, because we felt we lacked understanding of the gRPC library and generated proto-files.

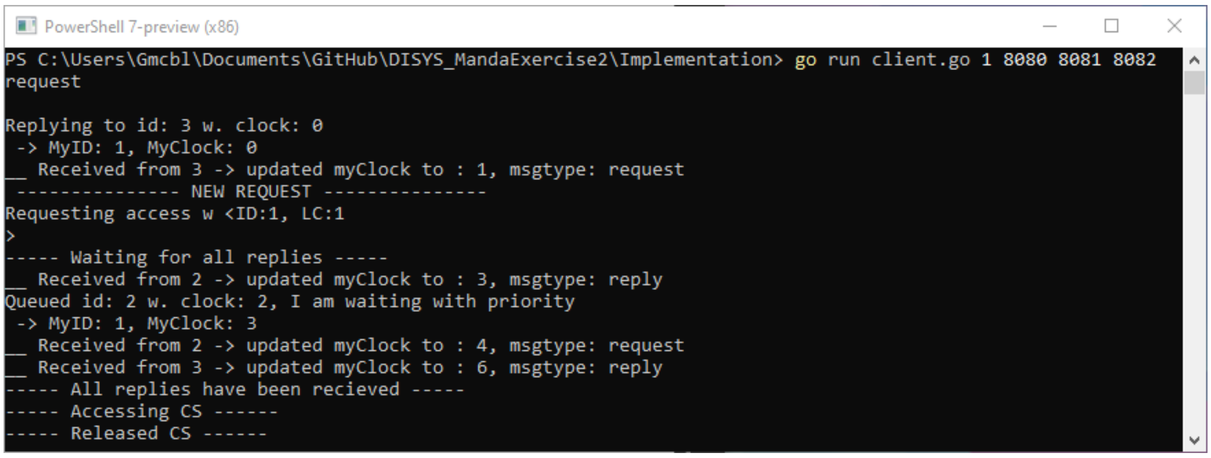
2 Execution

Below are attached screenshots of four terminals running respectively three instances of client.go and one instance of shared_resource.go.

A **REQUEST** has been issued, almost simultaneously between the clients, and the queuing of requests can therefore be seen in the logs of the different clients.

The order of execution is therefore:

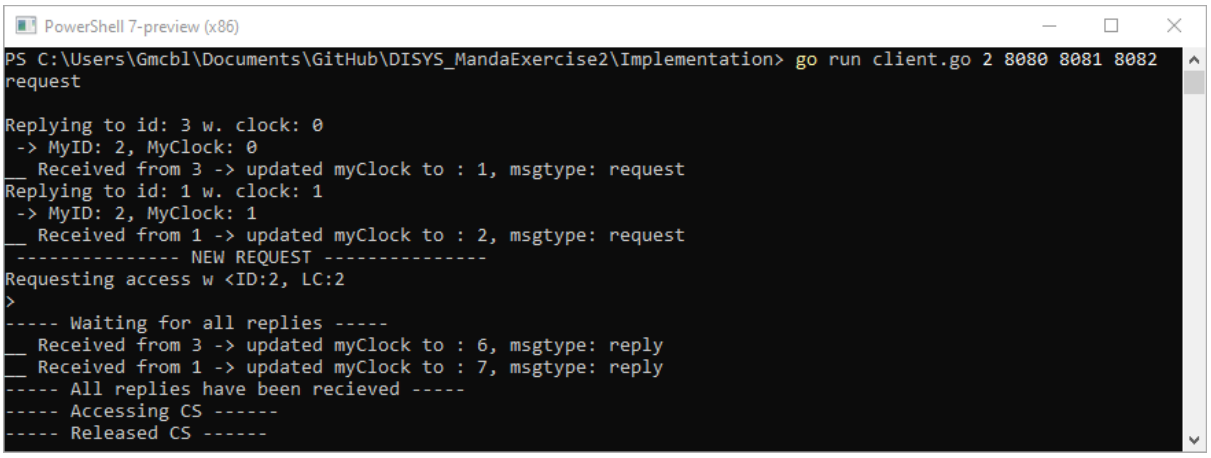
1. Client 3.
2. Client 1.
3. Client 2.



```
PS C:\Users\Gmcbl\Documents\GitHub\DISYS_MandaExercise2\Implementation> go run client.go 1 8080 8081 8082
request

Replying to id: 3 w. clock: 0
 -> MyID: 1, MyClock: 0
__ Received from 3 -> updated myClock to : 1, msgtype: request
 --------------- NEW REQUEST ---------------
Requesting access w <ID:1, LC:1
>
----- Waiting for all replies -----
__ Received from 2 -> updated myClock to : 3, msgtype: reply
Queued id: 2 w. clock: 2, I am waiting with priority
 -> MyID: 1, MyClock: 3
__ Received from 2 -> updated myClock to : 4, msgtype: request
__ Received from 3 -> updated myClock to : 6, msgtype: reply
----- All replies have been recieved -----
----- Accessing CS ------
----- Released CS ------
```

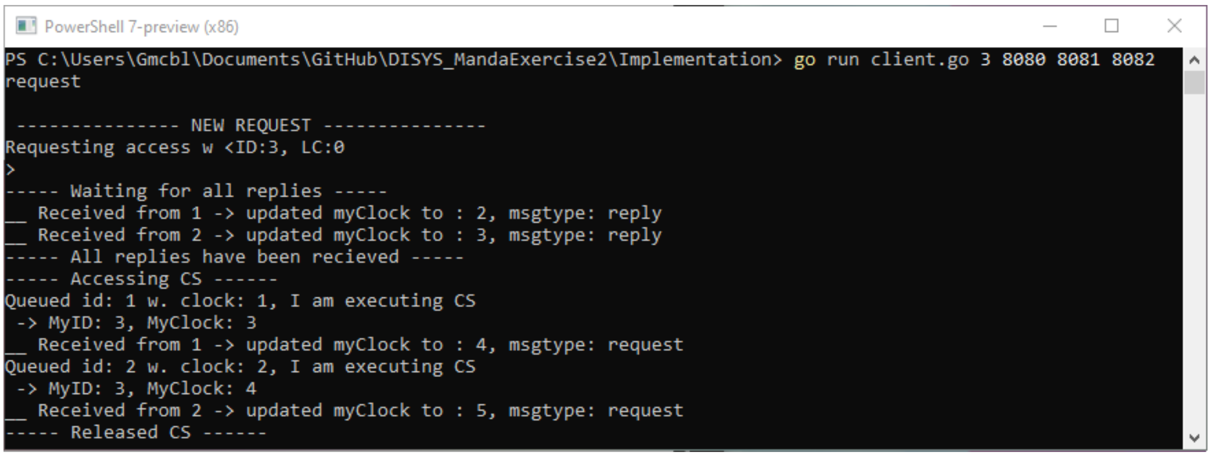
Figure 1: Client 1.



```
PS C:\Users\Gmcbl\Documents\GitHub\DISYS_MandaExercise2\Implementation> go run client.go 2 8080 8081 8082
request

Replying to id: 3 w. clock: 0
 -> MyID: 2, MyClock: 0
__ Received from 3 -> updated myClock to : 1, msgtype: request
Replying to id: 1 w. clock: 1
 -> MyID: 2, MyClock: 1
__ Received from 1 -> updated myClock to : 2, msgtype: request
 --------------- NEW REQUEST ---------------
Requesting access w <ID:2, LC:2
>
----- Waiting for all replies -----
__ Received from 3 -> updated myClock to : 6, msgtype: reply
__ Received from 1 -> updated myClock to : 7, msgtype: reply
----- All replies have been recieved -----
----- Accessing CS ------
----- Released CS ------
```

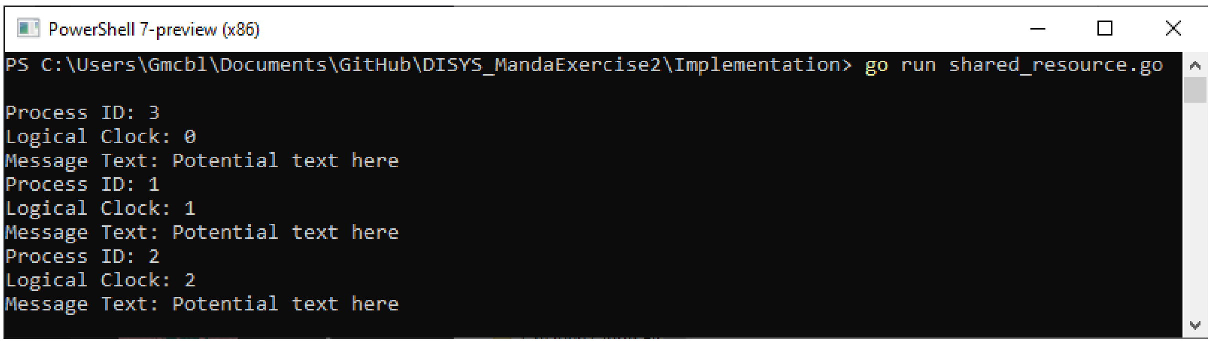
Figure 2: Client 2.



```
PS C:\Users\Gmcbl\Documents\GitHub\DISYS_MandaExercise2\Implementation> go run client.go 3 8080 8081 8082
request

 --------------- NEW REQUEST ---------------
Requesting access w <ID:3, LC:0
>
----- Waiting for all replies -----
__ Received from 1 -> updated myClock to : 2, msgtype: reply
__ Received from 2 -> updated myClock to : 3, msgtype: reply
----- All replies have been recieved -----
----- Accessing CS ------
Queued id: 1 w. clock: 1, I am executing CS
 -> MyID: 3, MyClock: 3
__ Received from 1 -> updated myClock to : 4, msgtype: request
Queued id: 2 w. clock: 2, I am executing CS
 -> MyID: 3, MyClock: 4
__ Received from 2 -> updated myClock to : 5, msgtype: request
----- Released CS ------
```

Figure 3: Client 3.



```
PS C:\Users\Gmcbl\Documents\GitHub\DISYS_MandaExercise2\Implementation> go run shared_resource.go

Process ID: 3
Logical Clock: 0
Message Text: Potential text here
Process ID: 1
Logical Clock: 1
Message Text: Potential text here
Process ID: 2
Logical Clock: 2
Message Text: Potential text here
```

Figure 4: Shared_resource