

CES-27 Distributed Processing

2nd Activity

Prof Hirata and Prof Juliana
Carlos Matheus Barros da Silva

September 2019

2th Activity

It was implemented the Ricart-Agrawala's algorithm. The implementation was made in Go and it can be seen on the Repository¹.

First Case

As It was suggested, it was conducted a test with 3 terminal windows, on two of them were open the process and on the third one was open the shared process.

The test was made according to the model represented on Figure 1. The results can be seen on the terminal windows shown from Figure 2 to Figure 4.

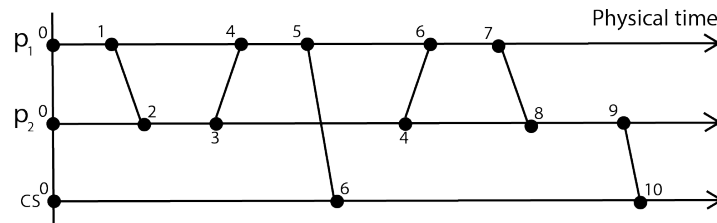


Figure 1: Model representing the execution of Case 1.

```
carlosmatheus@carlos-matheus-Pre-1: ~/Documents/CES-27-Labs/lab02$ clear
carlosmatheus@carlos-matheus-Pre-1: ~/Documents/CES-27-Labs/lab02$ go run Process.go 1 :10002 :10003
a
[01] Message: a
[04] Received reply message " a " from process: 2 at clock 4
[04] Entrei na CS at clock 4
[05] Sai da CS at clock 5
[06] Received request message " b " from process: 2 at clock 6
```

Figure 2: Process 1 after execution of Case 1 example test case.

As expected, the mutual exclusion worked; thus, when a process request access the other reply allowing the access when it is on *released* state, then the first process accesses the shared

¹<https://github.com/CarlosMatheus/CES-27/tree/master/lab02>

```
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ clear
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ go run Process.go 1 :10002 :10003
a
[01] Message: a
[04] Received reply message " a " from process: 2 at clock 4
[04] Entrei na CS at clock 4
[05] Sai da CS at clock 5
[06] Received request message " b " from process: 2 at clock 6
```

Figure 3: Process 2 after execution of Case 1 example test case.

```
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ clear
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ go run SharedResource.go
[06] Received logical Clock: 5
[06] Received a
[10] Received logical Clock: 9
[10] Received b
```

Figure 4: Shared Resource after execution of Case 1 example test case.

resource. When it finishes, it returns to released state. Then the the second process do the same steps. All the logic worked as well as the logical clocks on each process, therefore the simulation on the terminals matched the model representd on Figure 1.

Second Case

It was built a test case with 6 terminal windows. The test was made according to the model represented on Figure 5. The results can be seen on the terminal windows shown from Figure 6 to Figure 11.

As expected, the mutual exclusion woked; Even when multiple process were wanting to access the shared resource at the same time, the mutual exclusion garanteed the safity and the lifeness of the operation. All the logic worked as well as the logical clocks on each process, therefore the simulation on the terminals matched the model representd on Figure 1.

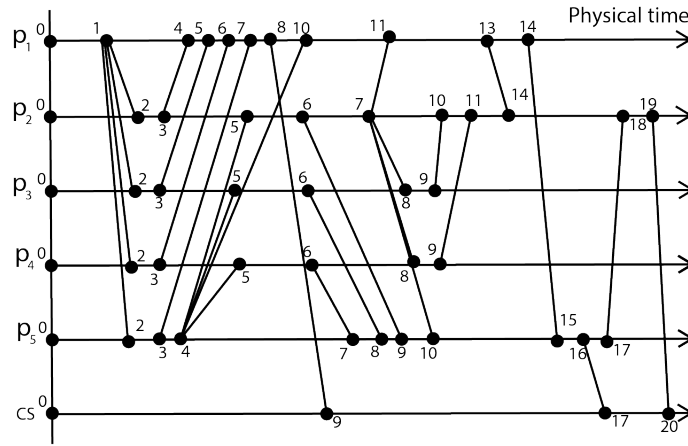


Figure 5: Model representing the execution of Case 1.

```

Default
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ clear
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ go run Process.go 1 :10002 :10003 :10004 :10005 :10006
a
[01] Message: a
[04] Received reply message " a " from process: 3 at clock 4
[05] Received reply message " a " from process: 2 at clock 5
[06] Received reply message " a " from process: 4 at clock 6
[07] Received reply message " a " from process: 5 at clock 7
[07] Entrei na CS at clock 7
[10] Received request message " b " from process: 5 at clock 10
[12] Received request message " c " from process: 2 at clock 12
[12] Sai da CS at clock 12

```

Figure 6: Process 1 after execution of Case 2 example test case.

```

Default
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ clear
carlosmatheus@carlos-MacBook-Pro-1: ~/Documents/CES-27-Labs/Lab02$ go run Process.go 2 :10002 :10003 :10004 :10005 :10006
[02] Received request message " a " from process: 1 at clock 2
[05] Received request message " b " from process: 5 at clock 5
c
[07] Message: c
[10] Received reply message " c " from process: 3 at clock 10
[11] Received reply message " c " from process: 4 at clock 11
[15] Received reply message " c " from process: 1 at clock 15
[18] Received reply message " c " from process: 5 at clock 18
[18] Entrei na CS at clock 18
[19] Sai da CS at clock 19

```

Figure 7: Process 2 after execution of Case 2 example test case.

```
carlosmatheus@carlos-MacBook-Pro-3: ~/Documents/CES-27-Labs/Lab02$ clear
carlosmatheus@carlos-MacBook-Pro-3: ~/Documents/CES-27-Labs/Lab02$ go run Process.go 3 :10002 :10003 :10004 :10005 :10006
[02] Received request message " a " from process: 1 at clock 2
[05] Received request message " b " from process: 5 at clock 5
[08] Received request message " c " from process: 2 at clock 8
```

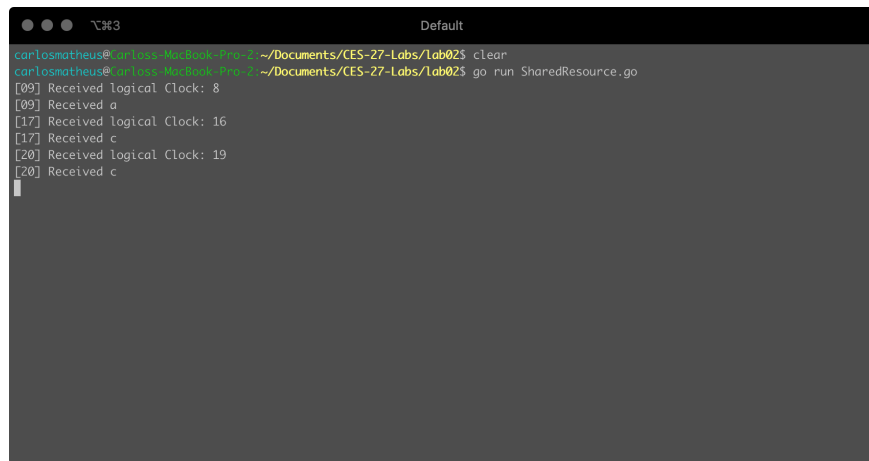
Figure 8: Process 3 after execution of Case 2 example test case.

```
carlosmatheus@carlos-MacBook-Pro-3: ~/Documents/CES-27-Labs/Lab02$ clear
carlosmatheus@carlos-MacBook-Pro-3: ~/Documents/CES-27-Labs/Lab02$ go run Process.go 4 :10002 :10003 :10004 :10005 :10006
[02] Received request message " a " from process: 1 at clock 2
[05] Received request message " b " from process: 5 at clock 5
[08] Received request message " c " from process: 2 at clock 8
```

Figure 9: Process 4 after execution of Case 2 example test case.

```
carlosmatheus@carlos-MacBook-Pro-3: ~/Documents/CES-27-Labs/Lab02$ go run Process.go 5 :10002 :10003 :10004 :10005 :10006
[02] Received request message " a " from process: 1 at clock 2
b
[04] Message: b
[07] Received reply message " b " from process: 2 at clock 7
[08] Received reply message " b " from process: 4 at clock 8
[09] Received reply message " b " from process: 3 at clock 9
[10] Received request message " c " from process: 2 at clock 10
[11] Received request message " c " from process: 2 at clock 11
[15] Received reply message " c " from process: 1 at clock 15
[15] Entrei na CS at clock 15
[16] Sai da CS at clock 16
```

Figure 10: Process 5 after execution of Case 2 example test case.

A terminal window with a dark background and light text. The window title bar shows three window control buttons and the text "Default". The terminal content shows a user named "carlosmatheus" at a prompt. The user enters "clear" and then "go run SharedResource.go". The program outputs several lines of log messages: "[09] Received logical Clock: 8", "[09] Received a", "[17] Received logical Clock: 16", "[17] Received c", "[20] Received logical Clock: 19", and "[20] Received c". A cursor is visible at the end of the last line.

```
carlosmatheus@carlosmatheus-MacBook-Pro: ~/Documents/CES-27-Labs/Lab02$ clear
carlosmatheus@carlosmatheus-MacBook-Pro: ~/Documents/CES-27-Labs/Lab02$ go run SharedResource.go
[09] Received logical Clock: 8
[09] Received a
[17] Received logical Clock: 16
[17] Received c
[20] Received logical Clock: 19
[20] Received c
```

Figure 11: Shared Resource after execution of Case 2 example test case.