

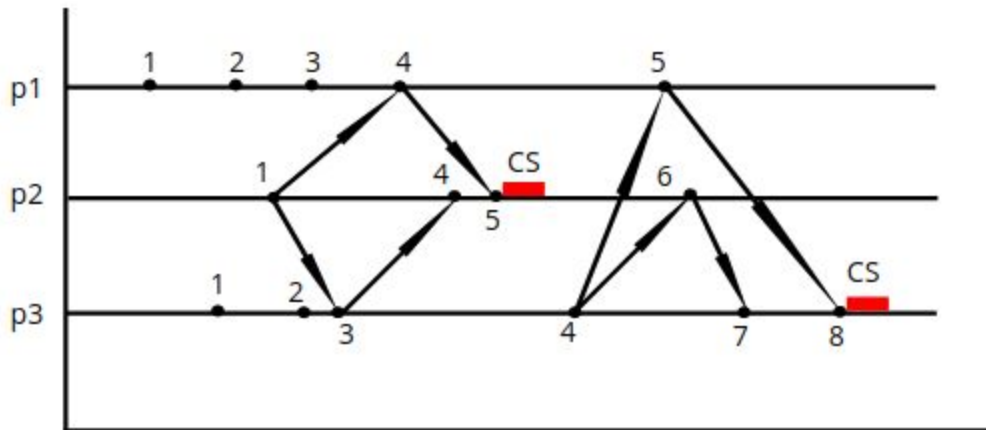
Laboratório 02 de CES-27: Mutual Exclusion

Aluno: Léo Lobo Gomes

Turma: Comp-21

Teste 01:

Resultado esperado:



Resultado obtido:

Primeiramente todos os processos tiveram apenas tarefas internas para aumentar os seus relógios lógicos.

Processo 1:

```
1
Relógio Atualizado: 1
1
Relógio Atualizado: 2
1
Relógio Atualizado: 3
```

Processo 2:

```
2
Relógio Atualizado: 1
```

Processo 3:

```
3
Relógio Atualizado: 1
3
Relógio Atualizado: 2
```

Em seguida, a tecla 'x' foi apertada no terminal do processo 2, mandando, assim, as request para a entrada em CS aos processos restantes.

```
2
Relógio Atualizado: 1
x
Processo pediu para entrar no CS
state: 1 timestamp: 1
Mandando requests.
Request mandada para process de id: 1
Request mandada para process de id: 3
```

Os demais processos recebem a mensagem e atualizando seu relógio lógico, além disso, como não estão em CS e nem desejam entrar, eles mandam a reply imediatamente.

Reply do processo 1:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 1 de 127.0.0.1:49833
Meu clock atualizado: 4
Reply para: 2 enviada.
```

Reply do processo 3:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 1 de 127.0.0.1:60183
Meu clock atualizado: 3
Reply para: 2 enviada.
```

O processo 2 recebe, primeiramente a reply do processo 3 e atualiza seu relógio lógico.

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:44914
Meu clock atualizado: 4
```

Em seguida, o processo 2 recebe a reply do processo 1 e atualiza novamente seu relógio lógico.

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 4 de 127.0.0.1:33453
Meu clock atualizado: 5
```

Como o processo 2 recebeu todas as replies necessárias para entrar em CS, ele começa a se comunicar com SharedResource.

```
Processo entrou em CS
Mensagem: Oi!
Proveniente do processo de ID: 2
Esse processo tem clock: 5
```

Após sair da CS o processo 2 tenta mandar as replies dos processos que estão na fila de espera, porém ela está vazia.

```
Processo saiu de CS
Mandando Reply para processos da fila
Tamanho do fila: 0
```

Em seguida o processo 3 aumenta o seu relógio lógico em uma unidade e pede para entrar em CS.

```
3
Relogio Atualizado: 4
x
Processo pediu para entrar no CS
state: 1   timestamp: 4
Mandando requests.
Request mandada para process de id: 1
Request mandada para process de id: 2
```

Os demais processos recebem a mensagem e atualizando seu relógio lógico, além disso, como não estão em CS e nem desejam entrar, eles mandam a reply imediatamente.

Reply do processo 1:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 4 de 127.0.0.1:40092
Meu clock atualizado: 5
Reply para: 3 enviada.
```

Reply do processo 2:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 4 de 127.0.0.1:44914
Meu clock atualizado: 6
Reply para: 3 enviada.
```

O processo 3 recebe, primeiramente a reply do processo 2 e atualiza seu relógio lógico.

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 6 de 127.0.0.1:60183
Meu clock atualizado: 7
```

Em seguida, o processo 2 recebe a reply do processo 1 e atualiza novamente seu relógio lógico.

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 5 de 127.0.0.1:37399
Meu clock atualizado: 8
```

Como o processo 3 recebeu todas as replys necessárias para entrar em CS, ele começa a se comunicar com SharedResource.

```
Processo entrou em CS
Mensagem: Oi!
Proveniente do processo de ID: 3
Esse processo tem clock: 8
```

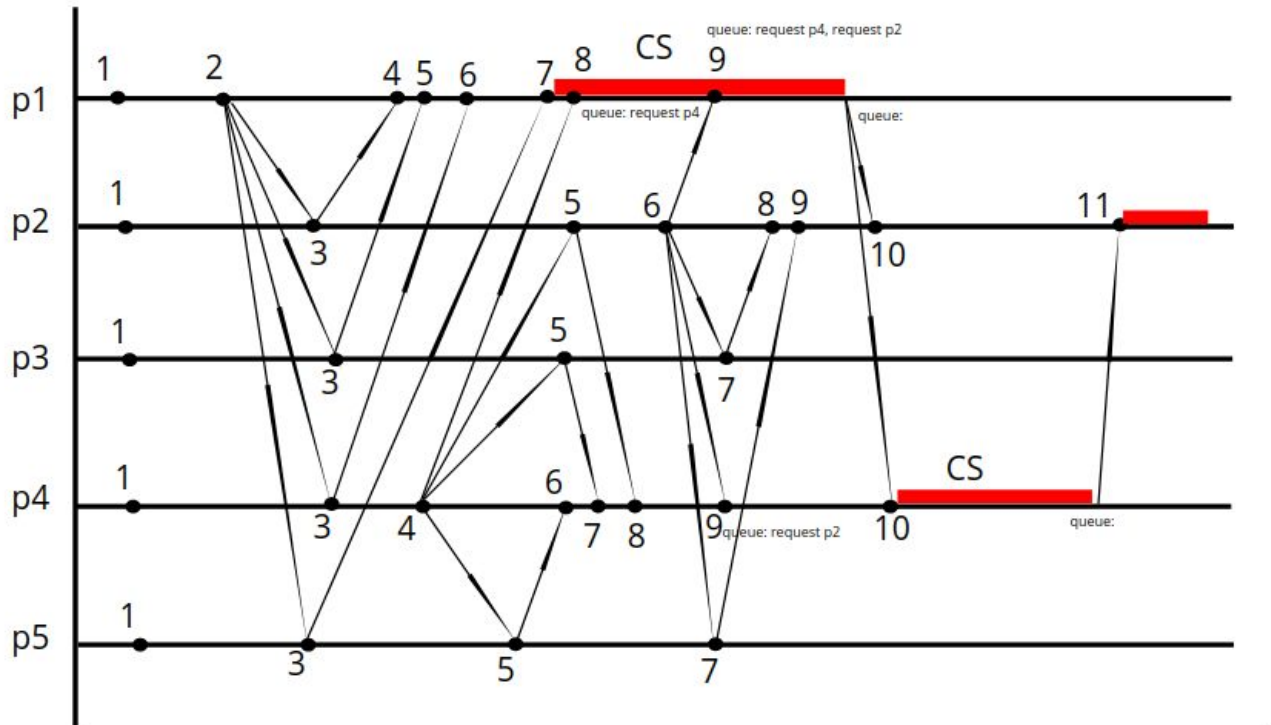
Após sair da CS o processo 3 tenta mandar as replys dos processos que estão na fila de espera, porém ela está vazia.

```
Processo saiu de CS
Mandando Reply para processos da fila
Tamanho do fila: 0
```

Logo, como a propagação do relógio lógico aconteceu de acordo com o diagrama apresentado acima, este teste aconteceu como o esperado.

Teste 02:

Resultado esperado:



Resultado obtido:

Primeiramente todos os processos tiveram apenas tarefas internas para aumentar os seus relógios lógicos.

Processo 1:

```
1
Relógio Atualizado: 1
1
Relógio Atualizado: 2
```

Processo 2:

```
2
Relógio Atualizado: 1
```

Processo 3:

```
3
Relogio Atualizado: 1
```

Processo 4:

```
4
Relogio Atualizado: 1
```

Processo 5:

```
5
Relogio Atualizado: 1
```

Em seguida, a tecla 'x' foi apertada no terminal do processo 1, mandando, assim, as request para a entrada em CS aos processos restantes.

```
x
Processo pediu para entrar no CS
state: 1 timestamp: 2
Mandando requests.
Request mandada para process de id: 2
Request mandada para process de id: 3
Request mandada para process de id: 4
Request mandada para process de id: 5
```

Os demais processos recebem a mensagem e atualizando seu relógio lógico, além disso, como não estão em CS e nem desejam entrar, eles mandam a reply imediatamente.

Reply do processo 2:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 2 de 127.0.0.1:52268
Meu clock atualizado: 3
Reply para: 1 enviada.
```

Reply do processo 3:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 2 de 127.0.0.1:39447
Meu clock atualizado: 3
Reply para: 1 enviada.
```

Reply do processo 4:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 2 de 127.0.0.1:54796
Meu clock atualizado: 3
Reply para: 1 enviada.
```

Reply do processo 5:

```
Mensagem Recebida.
Clock do processo que mandou a mensagem: 2 de 127.0.0.1:49215
Meu clock atualizado: 3
Reply para: 1 enviada.
```


O processo 1 recebe as replys dos demais processos e entra em CS para se comunicar com SharedResource.

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:55470  
Meu clock atualizado: 4  
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:45954  
Meu clock atualizado: 5  
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:41028  
Meu clock atualizado: 6  
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:53227  
Meu clock atualizado: 7  
Processo entrou em CS
```

```
Mensagem: Oi!  
Proveniente do processo de ID: 1  
Esse processo tem clock: 7
```

Em seguida o processo 4 pede para entrar em CS e manda as request para os demais processos.

```
X  
Processo pediu para entrar no CS  
state: 1 timestamp: 3  
Mandando requests.  
Request mandada para process de id: 1  
Request mandada para process de id: 2  
Request mandada para process de id: 3  
Request mandada para process de id: 5
```

Os demais processos recebem a mensagem e atualizando seu relógio lógico, além disso, os processos que não estão em CS e nem desejam entrar mandam a reply imediatamente.

Processo 2:

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:59402  
Meu clock atualizado: 4  
Reply para: 4 enviada.
```

Processo 3:

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:33070  
Meu clock atualizado: 4  
Reply para: 4 enviada.
```

Processo 5:

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:56196  
Meu clock atualizado: 4  
Reply para: 4 enviada.
```

Como o processo 1 já está em CS ele enfileira o request do processo 4.

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 3 de 127.0.0.1:53227  
Meu clock atualizado: 8  
Request de: 4 colocada na queue.
```

O processo 4 recebe os replays dos processos 2, 3 e 5.

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 5 de 127.0.0.1:56799  
Meu clock atualizado: 6  
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 5 de 127.0.0.1:40661  
Meu clock atualizado: 7  
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 5 de 127.0.0.1:51143  
Meu clock atualizado: 8
```

Em seguida o processo 2 pede para entrar em CS e manda as request para os demais processos.

```
X  
Processo pediu para entrar no CS  
state: 1 timestamp: 6  
Mandando requests.  
Request mandada para process de id: 1  
Request mandada para process de id: 3  
Request mandada para process de id: 4  
Request mandada para process de id: 5
```

Os demais processos recebem a mensagem e atualizando seu relógio lógico, além disso, os processos que não estão em CS e nem desejam entrar mandam a reply imediatamente.

Processo 3:

```
Clock do processo que mandou a mensagem: 6 de 127.0.0.1:52098  
Meu clock atualizado: 7  
Reply para: 2 enviada.
```

Processo 5:

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 6 de 127.0.0.1:44673  
Meu clock atualizado: 7  
Reply para: 2 enviada.
```

Como o processo 1 já está em CS ele enfileira o request do processo 2.

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 6 de 127.0.0.1:46394  
Meu clock atualizado: 9  
Request de: 2 colocada na queue.
```

Como o processo 4 já deseja entrar em CS e possui um timestamp menor que a do processo 2, ele enfileira o request do mesmo.

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 6 de 127.0.0.1:56799  
Meu clock atualizado: 9  
Request de: 2 colocada na queue.
```

O processo 2 recebe os replys dos processos 3 e 5.

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 7 de 127.0.0.1:54183  
Meu clock atualizado: 8  
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 7 de 127.0.0.1:58884  
Meu clock atualizado: 9
```

Então, o processo 1 sai de CS e manda as replys para os requests que estava enfileirados, nesse caso são os requests do processo 4 e do processo 2.

```
Mandando Reply para processos da fila  
Tamanho do fila: 2  
Reply para: 4 enviada.  
Reply para: 2 enviada.
```

Já que o processo 4 recebeu a reply que estava esperando ele entrar em CS.

```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 9 de 127.0.0.1:34213  
Meu clock atualizado: 10  
Processo entrou em CS
```

```
Mensagem: Oi!  
Proveniente do processo de ID: 4  
Esse processo tem clock: 10
```

Ao sair de CS o processo 4 envia a reply para o processo 2.

```
Processo saiu de CS  
Mandando Reply para processos da fila  
Tamanho do fila: 1  
Reply para: 2 enviada.
```

Depois de receber as replys dos processo 1 e 4, o processo 2 entra em CS.


```
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 9 de 127.0.0.1:53264  
Meu clock atualizado: 10  
Mensagem Recebida.  
Clock do processo que mandou a mensagem: 10 de 127.0.0.1:58767  
Meu clock atualizado: 11  
Processo entrou em CS
```

```
Mensagem: Oi!  
Proveniente do processo de ID: 2  
Esse processo tem clock: 11
```

Após sair da CS o processo 2 tenta mandar as replies dos processos que estão na fila de espera, porém ela está vazia.

```
Processo saiu de CS  
Mandando Reply para processos da fila  
Tamanho do fila: 0
```

Logo, como a propagação do relógio lógico aconteceu de acordo com o diagrama apresentado acima, este teste aconteceu como o esperado.