



# Trabalho 2

Comunicação, Sincronismos,  
Deadlock, Threads

André Bermudes Viana - 10684580

Marcos Antonio Nobre Coutinho - 10716397

# Race Condition

Quando duas ou mais threads acessam simultaneamente um mesmo recurso compartilhado.

```
1 void codigoA(int *counter) {  
2     (*counter)++;  
3 }  
4
```

```
6 void codigoB(int *counter) {  
7     (*counter)++;  
8 }  
9
```

```
11 int main() {  
12     int counter = 0;  
13  
14     codigoA(&counter);  
15     codigoB(&counter);  
16     print("%d\n", *counter);  
17     return 0;  
18 }
```

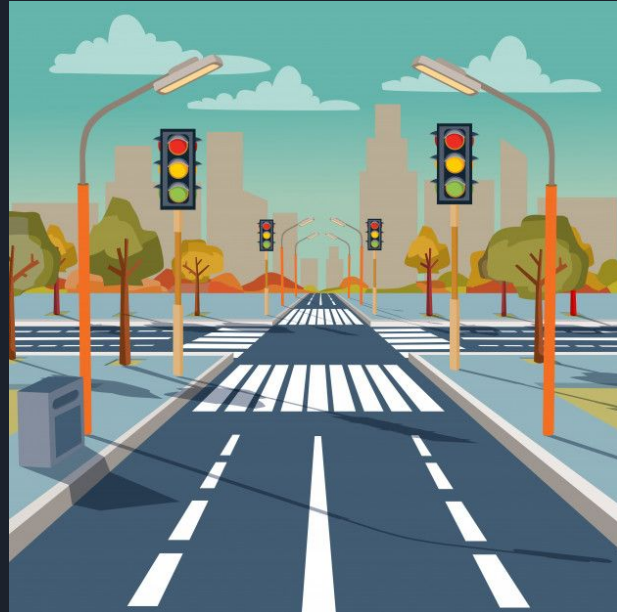


## Race Condition

Thread 1	Thread 2		Inteiro
			0
lê o valor		←	0
	lê o valor	←	0
incrementa-o			0
	incrementa-o		0
escreve nele		→	1
	escreve nele	→	1

# Semáforo

O semáforo é uma variável proposta por Dijkstra, ele pode conter o valor 0 (caso nenhum sinal de acordar está salvo), ou um valor positivo (valor que representa o número de sinais de acordar que estão pendentes).





# Semáforo

A proposta de Dijkstra foi da existência de duas operações “down” e “up”.

- Down: Verifica se o valor do semáforo é maior do que zero, se for, o decrementa e continua, caso contrário, o processo será suspenso “sleep”.
- Up: Esta operação incrementa o valor de um semáforo, porém caso existam processos suspensos nele, um deles será executado, e o valor do semáforo continuará zero.



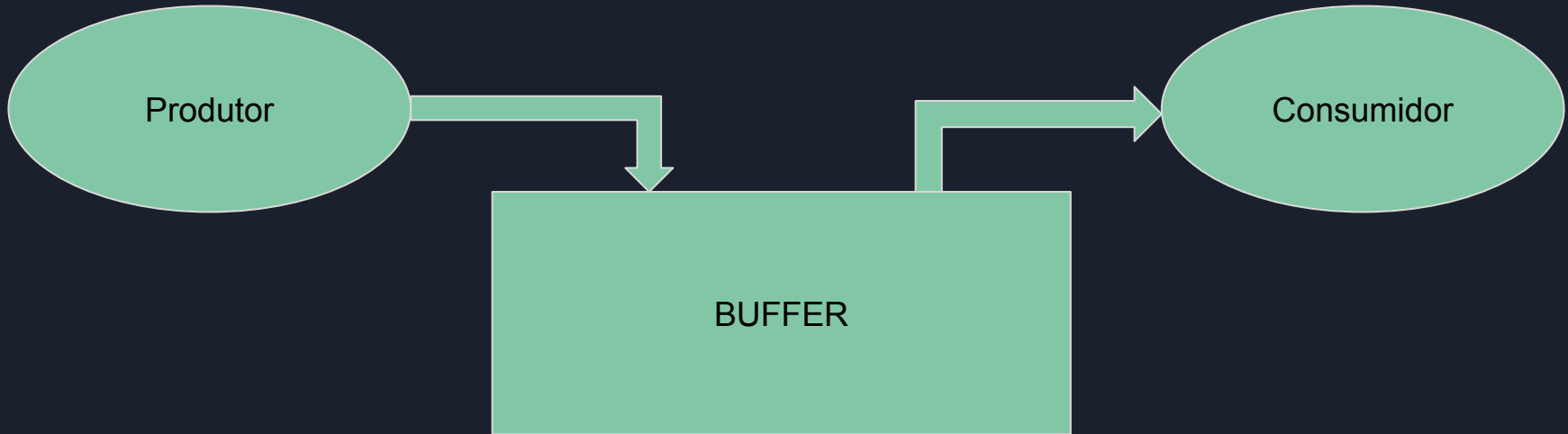
# Mutex

- Mutex = MUTual EXclusion
- Garante acesso exclusivo a um recurso compartilhado.
- Como funciona?



# Produtor x Consumidor

Dois processos compartilham acesso ao mesmo buffer, um dos processos insere informação no buffer (produtor), e o outro retira informações do buffer (consumidor)



# Produtor x Consumidor (Possíveis Problemas)

Os dois processos não podem ter acesso ao recurso ao mesmo tempo, para evitar race conditions, além disso, o consumidor não pode “consumir” de uma posição vazia do buffer, e o produtor não pode inserir em uma posição ocupada (não-consumida) do buffer.







Solução com Semáforo e Mutex

VAMOS AO CÓDIGO :)

# Solução com Semáforo e Mutex

```
gso15@tau05-vm3:~/Trabalho2$ ./ComSemaforoMutex
Produzi no indice 0, Total Produzido 1
Consumi do indice 0, Total Consumido 1
Produzi no indice 1, Total Produzido 2
Consumi do indice 1, Total Consumido 2
Produzi no indice 2, Total Produzido 3
Consumi do indice 2, Total Consumido 3
Produzi no indice 3, Total Produzido 4
Consumi do indice 3, Total Consumido 4
Produzi no indice 4, Total Produzido 5
Consumi do indice 4, Total Consumido 5
Produzi no indice 0, Total Produzido 6
Consumi do indice 0, Total Consumido 6
Produzi no indice 1, Total Produzido 7
Consumi do indice 1, Total Consumido 7
Produzi no indice 2, Total Produzido 8
Consumi do indice 2, Total Consumido 8
Produzi no indice 3, Total Produzido 9
Consumi do indice 3, Total Consumido 9
Produzi no indice 4, Total Produzido 10
Finalizando Produtor!
Consumi do indice 4, Total Consumido 10
Finalizando Consumidor!
```

```
gso15@tau05-vm3:~/Trabalho2$ ./ComSemaforoMutex
Produzi no indice 0, Total Produzido 1
Produzi no indice 1, Total Produzido 2
Produzi no indice 2, Total Produzido 3
Produzi no indice 3, Total Produzido 4
Produzi no indice 4, Total Produzido 5
Consumi do indice 0, Total Consumido 1
Consumi do indice 1, Total Consumido 2
Consumi do indice 2, Total Consumido 3
Consumi do indice 3, Total Consumido 4
Consumi do indice 4, Total Consumido 5
Produzi no indice 0, Total Produzido 6
Produzi no indice 1, Total Produzido 7
Produzi no indice 2, Total Produzido 8
Produzi no indice 3, Total Produzido 9
Produzi no indice 4, Total Produzido 10
Finalizando Produtor
Consumi do indice 0, Total Consumido 6
Consumi do indice 1, Total Consumido 7
Consumi do indice 2, Total Consumido 8
Consumi do indice 3, Total Consumido 9
Consumi do indice 4, Total Consumido 10
Finalizando Consumidor!
```

# Solução com Condicional e Mutex

```
gso15@tau05-vm3:~/Trabalho2$ gcc -pthread ComMutexCondicional.c -o CondicionalMutex
gso15@tau05-vm3:~/Trabalho2$ ./CondicionalMutex
Produziu no 1
Consumiu no 1
Produziu no 2
Consumiu no 2
Produziu no 3
Consumiu no 3
Produziu no 4
Consumiu no 4
Produziu no 5
Consumiu no 5
Produziu no 6
Consumiu no 6
Produziu no 7
Consumiu no 7
Produziu no 8
Consumiu no 8
Produziu no 9
Consumiu no 9
Produziu no 10
Consumiu no 10
gso15@tau05-vm3:~/Trabalho2$
```

**Agradecemos a atenção!**





# Links

Vídeo:

<https://youtu.be/zvvEZj2OkEo>

Repositório:

[https://github.com/andrecoco/SSC0640\\_SistemasOperacionaisI/tree/Trabalho02](https://github.com/andrecoco/SSC0640_SistemasOperacionaisI/tree/Trabalho02)