

## Descrição do problema e abordagem utilizada no código com Impasse

**Problema Inicial:** Criar um programa que permita a ocorrência de um impasse (deadlock).

Para esta implementação foi determinado que as colônias pares [0,2,4, ...] irão obter primeiro o recurso A e depois o B. Já as ímpares [1,3,5, ...] irão obter primeiro o recurso B e depois o A.

## Análise do comportamento sem a prevenção de impasse

À partir da compilação e execução do programa, as threads deveriam tentar pegar ambos os recursos para crescerem, no entanto, conforme à abordagem utilizada e testes realizados, as primeiras colônias [0 e 1], pegavam respectivamente os recursos A e B, o que resultava em um impasse, já que ambas bloqueiam os recursos que estão em sua posse e só liberam estes após crescerem e como não ocorreu o crescimento em ambas ocasionou o impasse.

Para saber se ocorreu o impasse, foi impresso prints dos recursos que as colônias tinham, os recursos que queriam e quais estavam envolvidas neste impasse. O mecanismo utilizado foi um mutex trylock se o resultado fosse zero à colônia poderia pegar aquele recurso se o resultado fosse diferente significa que outra colônia já possui aquele recurso e portanto ambas estavam diante de um impasse.

As colônias que não conseguiram nenhum recurso não foram representadas no console.

## Testes realizados para a verificação do Impasse

Foram realizados 4 testes para a verificação do impasse.

*1º Teste - Realizado com 2 colônias de bactérias [0,1]*

```
9 #define numero_colonias 2/*numero de colonias(threads)*/
10
SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURACÃO  PROBLEMAS

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> gcc -o comImpasse comImpasse.c -p
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> .\comImpasse
Colonia 0 obteve o recurso A
Colonia 0 quer o recurso B
Colonia 1 obteve o recurso B
Colonia 1 quer recurso A
Detectado um impasse!!!
Colonia 0 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
Detectado um impasse!!!
Colonia 1 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
```

2º Teste - Realizado com 5 colônias de bactérias [0,1,2,3,4]

```
10 #define numero_colonias 5/*numero de colonias(threads)*/
11

SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURAÇÃO  PROBLEMAS  powershell

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> gcc -o comImpasse comImpasse.c -pthread -lm
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> .\comImpasse
Colonia 1 obteve o recurso B
Colonia 2 obteve o recurso A
Colonia 2 quer o recurso B
Detectado um impasse!!!
Colonia 1 quer recurso A
Detectado um impasse!!!
Colonia 1 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
Colonia 2 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
```

3º Teste - Realizado com 10 colônias de bactérias [0,1,2,3,4, 5, 6, 7, 8, 9]

```
10 #define numero_colonias 10/*numero de colonias(threads)*/
11

SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURAÇÃO  PROBLEMAS  powershell

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> gcc -o comImpasse comImpasse.c -pthread -lm
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> .\comImpasse
Colonia 0 obteve o recurso A
Colonia 0 quer o recurso B
Colonia 1 obteve o recurso B
Colonia 1 quer recurso A
Detectado um impasse!!!
Colonia 0 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
Detectado um impasse!!!
Colonia 1 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
```

4º Teste - Realizado com 100 colônias de bactérias [0,1,2,3,4, 5, 6, 7, 8, 9, ..., 99]

```
10 #define numero_colonias 100/*numero de colonias(threads)*/
11

SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURAÇÃO  PROBLEMAS  powershell

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> gcc -o comImpasse comImpasse.c -pthread -lm
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> .\comImpasse
Colonia 1 obteve o recurso B
Colonia 2 obteve o recurso A
Colonia 2 quer o recurso B
Detectado um impasse!!!
Colonia 1 quer recurso A
Detectado um impasse!!!
Colonia 1 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
Colonia 2 esta em impasse. Conseguiu apenas 1 recurso e portanto nao podera crescer
```

## Descrição do problema e abordagem utilizada no código com prevenção de impasses

Conforme demonstrado no código acima estava ocorrendo impasse entre as colônias e estas não podiam crescer, portanto foi feito uma modificação do código e acrescido a técnica de

prevenção de ordenação de recursos, que significa que todas as colônias deverão obter os recursos em uma mesma ordem, assim garantindo que não ocorram impasses entre as colônias.

## Análise do comportamento com a prevenção de impasse

Após à compilação e execução do programa as threads são inicializadas e a primeira thread pega o recurso A, após isso o recurso B, realiza o seu crescimento, imprime a sua população final no console e após isso libera ambos os recursos para que as demais colônias também possam crescer.

No terminal, é impressa uma colônia por vez, ou seja, a que está realizando o seu crescimento naquele momento, as demais esperam a sua vez.

## Testes realizados para com o código de prevenção do Impasse

Do mesmo modo que o código com impasse, foram realizados 4 testes distintos.

*1º Teste - Realizado com 2 colônias de bactérias [0,1]*

```
10 #define numero_colonias 2 /*numero de colonias(threads)*/
11
SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURAÇÃO  PROBLEMAS  powershe
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> gcc -o semImpasse semImpasse.c -pthread -lm
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> ./semImpasse

Colonia 1 obteve o recurso A
Colonia 1 obteve o recurso B
Colonia 1 comecou a crescer!!!
A populacao final da colonia 1 : 30.802168

Colonia 0 obteve o recurso A
Colonia 0 obteve o recurso B
Colonia 0 comecou a crescer!!!
A populacao final da colonia 0 : 30.802168
```

2º Teste - Realizado com 5 colônias de bactérias [0,1,2,3,4]

```
10 #define numero_colonias 5 /*numero de colonias(threads)*/
11

SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURAÇÃO  PROBLEMAS

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> ./semImpasse

Colonia 1 obteve o recurso A
Colonia 1 obteve o recurso B
Colonia 1 comecou a crescer!!!
A populacao final da colonia 1 : 30.802168

Colonia 0 obteve o recurso A
Colonia 0 obteve o recurso B
Colonia 0 comecou a crescer!!!
A populacao final da colonia 0 : 30.802168

Colonia 2 obteve o recurso A
Colonia 2 obteve o recurso B
Colonia 2 comecou a crescer!!!
A populacao final da colonia 2 : 30.802168

Colonia 3 obteve o recurso A
Colonia 3 obteve o recurso B
Colonia 3 comecou a crescer!!!
A populacao final da colonia 3 : 30.802168

Colonia 4 obteve o recurso A
Colonia 4 obteve o recurso B
Colonia 4 comecou a crescer!!!
A populacao final da colonia 4 : 30.802168
```

3º Teste - Realizado com 10 colônias de bactérias [0,1,2,3,4, 5, 6, 7, 8, 9], foram representados somente alguns print's do terminal e o mesmo não foi representado por completo

```
10 #define numero_colonias 10 /*numero de colonias(threads)*/
11

SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURAÇÃO  PROBLEMAS  powershell

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> gcc -o semImpasse semImpasse.c -pthread -lm
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> ./semImpasse

Colonia 0 obteve o recurso A
Colonia 0 obteve o recurso B
Colonia 0 comecou a crescer!!!
A populacao final da colonia 0 : 30.802168
```

```
10 #define numero_colonias 10 /*numero de colonias(threads)*/
11
```

SAÍDA   TERMINAL   PORTAS   CONSOLE DE DEPURAÇÃO   PROBLEMAS


Colonia 7 obteve o recurso A  
Colonia 7 obteve o recurso B  
Colonia 7 comecou a crescer!!!  
A populacao final da colonia 7 : 30.802168

Colonia 8 obteve o recurso A  
Colonia 8 obteve o recurso B  
Colonia 8 comecou a crescer!!!  
A populacao final da colonia 8 : 30.802168

Colonia 9 obteve o recurso A  
Colonia 9 obteve o recurso B  
Colonia 9 comecou a crescer!!!  
A populacao final da colonia 9 : 30.802168

4º Teste - Realizado com 100 colônias de bactérias [0,1,2 ,3,4, 5, 6, 7, 8, 9, ....., 99], este teste segue a mesma lógica citada no teste acima.

```
10 #define numero_colonias 100 /*numero de colonias(threads)*/
11
```

SAÍDA   TERMINAL   PORTAS   CONSOLE DE DEPURAÇÃO   PROBLEMAS    powershell

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> gcc -o semImpasse semImpasse.c -pthread -lm  
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> .\semImpasse

Colonia 1 obteve o recurso A  
Colonia 1 obteve o recurso B  
Colonia 1 comecou a crescer!!!  
A populacao final da colonia 1 : 30.802168

Colonia 2 obteve o recurso A  
Colonia 2 obteve o recurso B  
Colonia 2 comecou a crescer!!!  
A populacao final da colonia 2 : 30.802168

```
10 #define numero_colonias 100 /*numero de colonias(threads)*/
11
```

SAÍDA   TERMINAL   PORTAS   CONSOLE DE DEPURAÇÃO   PROBLEMAS

A populacao final da colonia 97 : 30.802168

Colonia 98 obteve o recurso A  
Colonia 98 obteve o recurso B  
Colonia 98 comecou a crescer!!!  
A populacao final da colonia 98 : 30.802168

Colonia 99 obteve o recurso A  
Colonia 99 obteve o recurso B  
Colonia 99 comecou a crescer!!!  
A populacao final da colonia 99 : 30.802168

## Discussão sobre o impacto da prevenção do impasse no desempenho e complexidade do programa

Para discutir sobre este impacto, irei utilizar como exemplo os testes realizados com 2 colônias de bactérias, sendo com e sem impasse respectivamente.

Considerando ambos os códigos, é possível perceber que o código com prevenção de impasse é mais simples, visto que, todas as colônias devem seguir uma mesma ordem e não seguir ordens diferentes conforme o código com impasse.

Já o desempenho atrelado ao tempo de execução retornou uma pequena diferença, sendo que o código sem impasse demorou 0,94 segundos a mais para executar.

*Tempo de Execução com Impasse: aproximadamente 1,16 segundos*

```
PS C:\Users\Cliente\Desktop\Colônia de Bactérias> Measure-Command {.\comImpasse.exe}

Days           : 0
Hours          : 0
Minutes        : 0
Seconds        : 1
Milliseconds    : 154
Ticks          : 11544326
TotalDays      : 1,33614884259259E-05
TotalHours     : 0,000320675722222222
TotalMinutes   : 0,0192405433333333
TotalSeconds   : 1,1544326
TotalMilliseconds : 1154,4326
```

*Tempo de Execução sem Impasse: aproximadamente 2,10 segundos*

```
10 #define numero_colonias 2 /*numero de colonias(threads)*/
11
SAÍDA  TERMINAL  PORTAS  CONSOLE DE DEPURAÇÃO  PROBLEMAS

PS C:\Users\Cliente\Desktop\Colônia de Bactérias> Measure-Command {.\semImpasse.exe}

Days           : 0
Hours          : 0
Minutes        : 0
Seconds        : 2
Milliseconds    : 92
Ticks          : 20922897
TotalDays      : 2,42163159722222E-05
TotalHours     : 0,000581191583333333
TotalMinutes   : 0,034871495
TotalSeconds   : 2,0922897
TotalMilliseconds : 2092,2897
```