Experimento #3

Sistemas Operacionais A

Adriano de Oliveira Munin RA:17066960

Fábio Seiji Irokawa RA:17057720

Lucas Rodrigues Coutinho RA:17776501

Marcos Lelis de F. Oliveira RA:16248387

Paulo M. Birocchi RA:16148363

Introdução

Neste projeto, realizamos dois experimentos, nos quais consistiam em executar e corrigir os erros de um programa exemplo. Tal programa demonstra como funciona um mecanismo de semáforos e memória compartilhada. Seu funcionamento ocorre da seguinte forma, o programa possui uma string global, na qual possui uma cadeia de caracteres alfanuméricos, além disso ele possui uma região de memória compartilhada que servira como um índice contador. Quando o programa começa a ser executado, ele cria alguns filhos os quais irão imprimir na tela uma quantidade pseudoaleatória de vezes de caracteres da string acima citada, tendo o papel da memória compartilhada guardar em qual posição do vetor de char o processo anterior parou. Porém, pelo fato de vários processos de impressão estarem sendo executados concorrentemente, a impressão poderia vir desordenada, além disso, poderia ocorrer *race condition* no índice, portanto, foi utilizado um mecanismo de exclusão mutua, que no caso foi o mecanismo de semáforos, no qual permite que apenas um processo por vez acesse a região de memória compartilhada e faca a impressão.

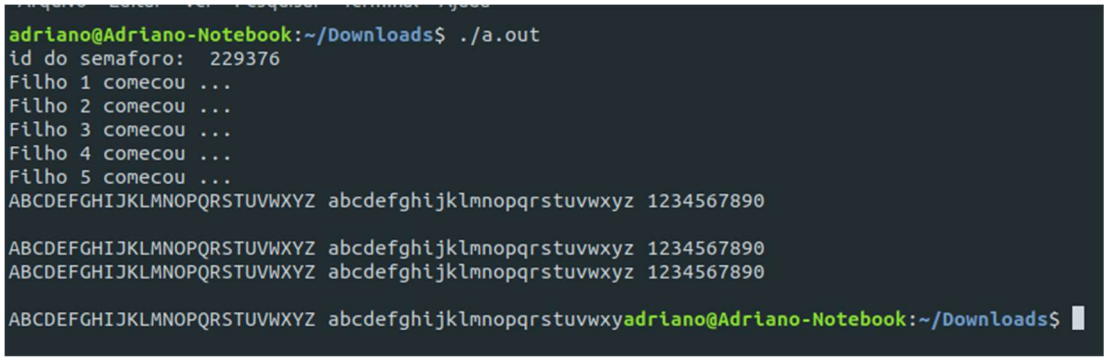
Além do programa acima citado, foi criado outro, no qual é uma modificação deste que tem a tarefa de criar oito filhos, dos quais quatro são iguais aos já existentes no exemplo (produtores de caracteres) e quatro serão consumidores. A tarefa do produtor é de pegar um número pseudorrandômico de caracteres de uma string e inserir em um buffer que é uma memória compartilhada. Já a tarefa do consumidor é a de pegar uma quantidade também pseudorrandômica de caracteres do buffer acima dito e substitui-los por #, a após o buffer cheio, imprimir todos os caracteres.

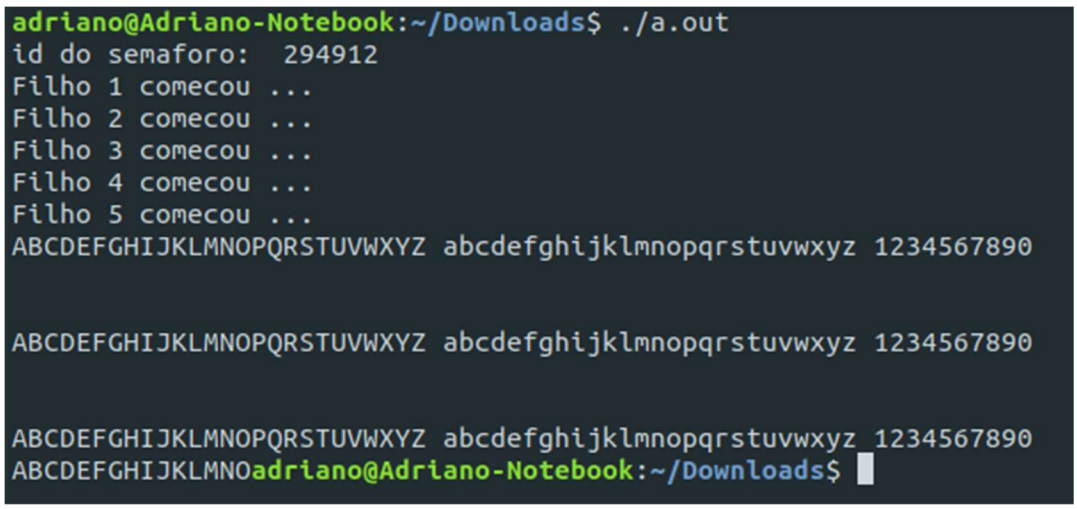
Com tais tarefas acima descritas, este experimento tem como objetivo melhorar a compreensão do funcionamento de mecanismos de memória compartilhada, além disso, visa-se compreender a ocorrência de *race condition* e suas possíveis soluções, com uso de mecanismos de exclusões mútuas, no caso deste experimento foi usado os mecanismos de semáforos e regiões críticas.

Resultados da Execução

Execuções com Semáforo Ativado:

Nessas execuções com o semáforo ativado foi possível observar que houve ordem na impressão da cadeia de caracteres, ou seja, o mecanismo de exclusão mútua foi eficaz para não deixar que um processo imprimisse enquanto o outro não tivesse terminado.



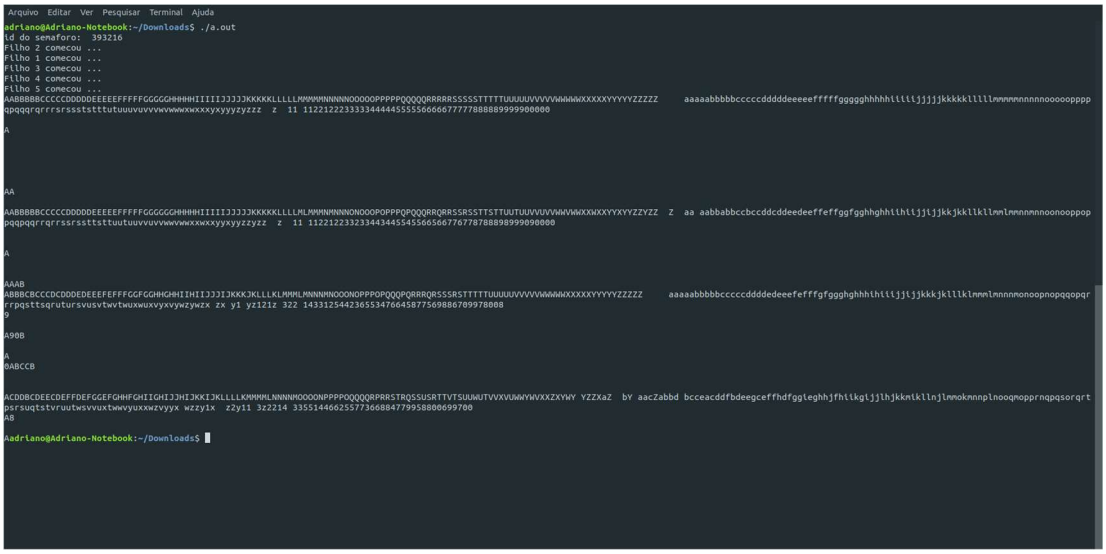


```
adriano@Adriano-Notebook:~/Downloads$ ./a.out
id do semaforo:  327680
Filho 1 comecou ...
Filho 2 comecou ...
Filho 3 comecou ...
Filho 4 comecou ...
ABFilho 5 comecou ...
CDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890

ABCDEFGHIJKLMNOadriano@Adriano-Notebook:~/Downloads$
```

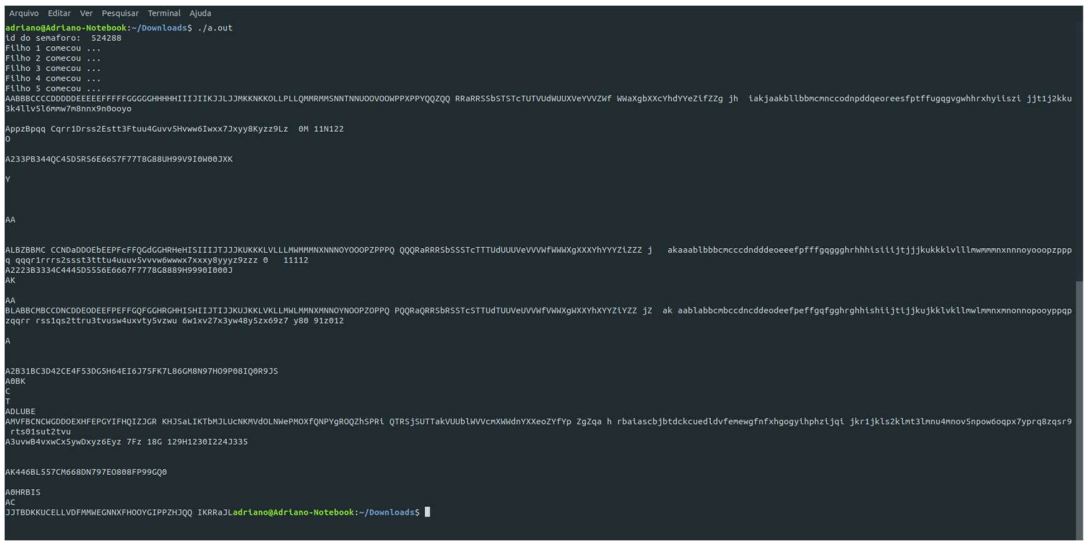
```
adriano@Adriano-Notebook:~/Downloads$ ./a.out
id do semaforo:  360448
Filho 1 comecou ...
Filho 2 comecou ...
Filho 3 comecou ...
AFilho 4 comecou ...
Filho 5 comecou ...
BCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890


ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890

ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890

ABCadriano@Adriano-Notebook:~/Downloads$
```

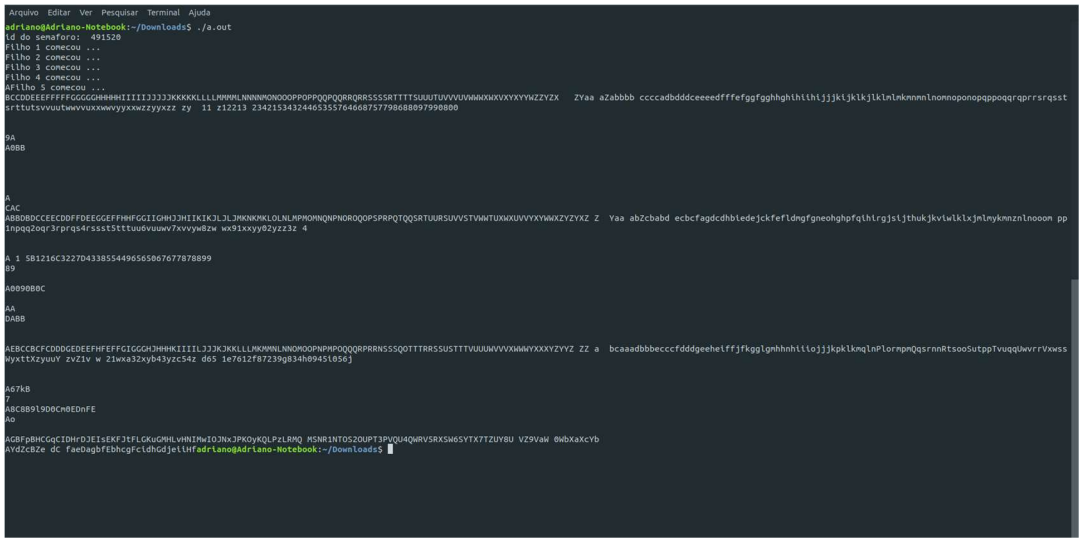
Execuções com Semáforo Desativado:

Nessa execução foi possível observar um desordenamento na impressão da cadeia de caracteres devido ao fato de processos concorrentes imprimirem suas partes antes de outro terminar.




Arquivo Editar Ver Pesquisar Terminal Ajuda
adrlano@Adrlano-Notebook:~/Downloads$ ./a.out
ld do semaforo: 458752
Filho 1 comecou ...
Filho 2 comecou ...
Filho 3 comecou ...
Filho 4 comecou ...
Filho 5 comecou ...
AABBCCCDDDDEEEEEFFFFFGGGGGHHHHIIIIIJJJJJKKKKLLLLKMMMLMMNMNNOONOOPPOPPQQPQQRRQRSRSRSTSTSTUUTTUVVUUVWVVWXXWWXYYXXYZZYYZ ZZ aa a bbabaccbcbddcdceededffefeggfghfhghigihijjhkkijkljklljklmnlkmnnnlnloonnoppoppqqpqqrr
qrrssrsstsstuutuuvvuvvwwvwxwxyxyxyzyzyz zz 11 12211233223443345545466565776768878799898080909 0
A
A
AB
ABB
CABCCDBCDDECDEEFDEFFGEFGGHFGHHIHGIIJIHJJKJIKKLKJLLMNLKMMHQLNNNPQNQQQPNPPRQQQQSRPRRTSQSSUTTRTVUUSUWVVTVXWWUWYXXXVZYYYW ZZZXa YbaaaZcbbb dcccaedddbfeeecgfffdhgggelhhhfjiiigkjjjlhkkkmilllnmjnmonknnpooloqppnnpqqro
qrrsprsstqsttturtuuvusvvwvtwwxwuxxyxvyyzywzz z x 1 iy1212z232334 34451455625676367847795888869997 0008
A9
A
BAA0BBBCCCD
CADDEDBEEFECFFGFDGGHGEHHIIHFIJJKIJGLKJKHMLKNLIOMLMPJNMNQKONOLRPOPMQSPQNRTQROSURPSTVSQTUWTRUVUXSVWVYTWXWZUXYX VYZYWa ZZXba YcbaZadcbb edccafddbegeecfhffdgiggehjhhfikiigjljjhkkkklmlllmnnmmnonnnopooopqpppqrqqqrsrr
rstsssstutttuvuuuuvwwwxvwxwwyxyxzxyzyy zz 1 z 12 11232224333354444565655776686987707988
A8B099C9D08E0F
AGHB
IAA
AJCBBBKDCCCLEDDDMFEEEGNHFFFIGOGGJHPHHIKQIILJRJJKKKSKLLLTLMMUMNNVNNOOWOOPPPXPQQQYQRRRZRSSS STTTaTUUUbUVVVcVWWWdWXXXeXYYYfYZZZgZ h aaalabbbjbccckcdddldadrlano@Adrlano-Notebook:~/Downloads$


Arquivo Editar Ver Pesquisar Terminal Ajuda
adrlano@Adrlano-Notebook:~/Downloads$ ./a.out
ld do semaforo: 524288
Filho 1 comecou ...
Filho 2 comecou ...
Filho 3 comecou ...
Filho 4 comecou ...
Filho 5 comecou ...
AABBBCCCCDDDDDEEEEEFFFFFGGGGGHHHHHIIIJIIKJJLJJMKKNKKOLLPLLQMMRMMSNNTNNUOOVOOWPPXPPYQQZQQ RRaRRSSbSTSTcTUTVUdWUUXVeYVVZWf WWaXgbXXcYhdYYeZifZZg jh iakjaakbllbbmcmnccodnpddqeoreesfptffugqgvgwhhrxhyiiszi jjt1j2kku
3k4llv5l6mmw7m8nnx9n0ooyo
AppzBpqq Cqrr1Drss2Estt3Ftuu4Guvv5Hvww6Iwxx7Jxyy8Kyzz9Lz 0M 11N122
O
A233PB344QC4SD5RS6E66S7F77T8G88UH99V9I0W00JXK
Y
AA
ALBZBBMC CCNDaDDOEbEEPFcFFQGdGGHRHeHISIIIJTJJJKUKKKLVLLLMWMMMNXNNNOYOOOPZPPPQ QQQRaRRRSbSSSTcTTTUdUUUVeVVVWfWWWXgXXXYhYYYZiZZZ j akaaablbbbcmcccdnddddeoeeefpfffggggghrhhhisiiijtjjjkukkklvlllmwmmmnxnnnoyooopzppp
q qqqr1rrrs2ssst3ttt4uuuv5vvvw6wwwx7xxxy8yyyz9zzz 0 11112
A2223B3334C4445D5556E6667F7778G8889H9990I00J
AK
AA
BLABBCMBCCDNCDDEODEEFPEFFGQFGGHRGHHISHIIJTJIJKUJKKLVKLLMWLMMNXMNNOYNOOPZOPPQ PQQRaQRRSbRSSTcSTTUdTUUVeUVVWfVWWXgWXXYhXYYZiYZZ jZ ak aablabbcmbccdncddeodeefpeffgqfggghrghhishiijtijjkujkklvkllmwlmmnxmnonnopooyppqp
zqqrr rss1qs2ttru3tvusw4uxvty5vzwu 6w1xv27x3yw48y5zx69z7 y80 91z012
A
A2B31BC3D42CE4F53DG5H64EI6J75FK7L86GM8N97HO9P08IQ0R9JS
A0BK
C
T
ADLUBE
AHVFBCNCWGDDOEXHFEPGYIFHQIZJGR KHJSaLIKTbMJLUcNKMVdOLNWePMOXfQNPYgROQZhSPRi QTRSjSUTTakVUUblWVVcmXWdnYXXeoZYfYp ZgZqa h rbaiascbjbtdckcuedldvfenewgfnfxhgogyihphziqi jkr1jkls2klmt3lnnu4mnov5npow6oqpx7yprq8zqsr9
rts01sut2tvu
A3uvwB4vxwCx5ywDxyz6Eyz 7Fz 18G 129H1230I224J335
AK446BLS57CM68DN797EO808FP99GQ0
A0HRBIS
AC
JJTBDKKUCELLVDFMWEGNNXFHOOYGIPPZHJQQ IKRRaJLadrlano@Adrlano-Notebook:~/Downloads$


Arquivo Editar Ver Pesquisar Terminal Ajuda
adrlano@Adrlano-Notebook:~/Downloads$ ./a.out
ld do semaforo: 491520
Filho 1 comecou ...
Filho 2 comecou ...
Filho 3 comecou ...
Filho 4 comecou ...
AFilho 5 comecou ...
BCCDDEEEFFFFFGGGGGHHHHHIIIIIJJJJJKKKKKLLLLMMMMLNNNNMONOOOPPOPPQQPQQRRQRRSSSSRTTTTSUUUTUVVVUVWWWXWXVXYXYYWZZYZX ZYaa aZabbbb ccccadbdddceeeedfffefggfggghhghihiihijjjkijkljklklmlmknnnnlnonnoponopqppoqqrqprrsrqsst
srttutsvvuutwwvvuxxwwvyyxxwzzyyxzz zy 11 z12213 2342153432446535576466875779868809799 0800
9A
A0BB
A
CAC
ABBDBDCCEECDDFFDEEGGEFFHHFGGIIGHHJJHIIKIKJLJLJMKNKMKLOLNLMPMOMNQNPNOROQOPSPRPQTQQSRTUURSUVVSTVWWTUXWXUVVYXYWWXZYZYXZ Z Yaa abZcbabd ecbcfagdcdhbiedejckfefldmgfgneohghpfqihirgjsljthukjkviwlklxjmlmykmnznlnooom pp
1npqq2oqr3rprqs4rssst5tttuu6vuuwv7xvvyw8zw wx91xxyy02yzz3z 4
A 1 5B1216C3227D43385544965650676778789 9
89
A0090B0C
AA
DABB
AEBCCBCFCDDDGEDEEFHFEFFGIGGGHJHHHKIIIILJJJKJKKLLLMKMNLNNOMOOPNPMPOQQQRPRRNSSSQOTTTRRSSUSTTTVUUUWVVVXWWYXXXYZYYZ ZZ a bcaaadbbbecccfdddgeeheiffjfkgglgmhhnhiiiojjjkpklkmqlnPlormpmQqsrnnRtsooSutppTvuqqUwvrrVxwss
WyxttXzyuuY zvZiv w 21wxa3Zxyb43yzc54z d65 1e7612f87239g834h0945l056j
A67kB
7
A8C8B9l9D0Cm0EDnFE
Ao
AGBFpBHCGqCIDHrDJEIsEKFJtFLGKuGMHLvHNIMwIOJNxJPKOyKQLPzLRMQ MSNR1NTOS2OUPT3PVQU4QWRV5RXSW6SYTX7TZUY8U VZ9VaW 0WbXaXcYb
AYdZcBZe dC faeDagbfEbhcgFcldhGdjelLHfadrlano@Adrlano-Notebook:~/Downloads$

Análise dos Resultados

A execução do programa com e sem PROTECT funcionou como esperado. Com o PROTECT a string foi imprimida corretamente, enquanto a ausência do mesmo acabou imprimindo letras e números aleatórios, resultado da race condition o qual fez com que os filhos disputassem pela mesma região compartilhada de forma paralela.

Infelizmente não foi possível realizar a segunda parte do experimento pois, houve dificuldade durante a modificação do programa para adicionar o Consumidor e Produtor.

Erros no Código do Programa

- Falta da biblioteca stdlib.h
- Duas estruturas g_sem_op1 existentes, alterado o g_sem_op1 com g_sem_op1[0].sem_op = 1 para g_sem_op2 que será a liberação do semáforo
- 'break' em vez de 'exit' no 'for' durante a criação dos filhos
- Tipo de variável incorreta no fprintf da string de %7.f para %c
- Semáforo fechando em vez de abrir no segundo PROTECT, trocado g_sem_op1 (tranca) por g_sem_op2 (libera)

Perguntas

Perguntas Fora do Código

Pergunta 1: Uma região por ser crítica tem garantida a exclusão mútua? Justifique.

R: Não. A região crítica é a área de um código que possui seus recursos compartilhados, portanto precisa de mecanismos de exclusão mútua para evitar as condições de corrida que poderiam ocorrer, porém não é garantida a exclusão mútua em qualquer região crítica, tendo a necessidade de ser implementada.

Pergunta 2: É obrigatório que todos os processos que acessam o recurso crítico tenham uma região crítica igual?

R: Não, pois processos distintos podem acessar por diferentes modos um mesmo recurso crítico, por exemplo memorias compartilhadas.

Pergunta 3: Por que as operações sobre semáforos precisam ser atômicas?

R: Para que outros processos não acessem o semáforo simultaneamente.

Pergunta 4: O que é uma diretiva ao compilador?

R: As diretivas de compilação são comandos que não são compilados, sendo dirigidos ao pré-processador, executado pelo compilador antes da execução do processo de compilação propriamente dito. Exemplo: o #include, que diz ao pré-processador para incluir naquele ponto um arquivo um arquivo especificado.

Pergunta 5: Porque o número é pseudoaleatório e não totalmente aleatório?

R: Porque não existe um método matemático para se obter um número totalmente aleatório, logo não existe aleatoriedade na computação.

Perguntas Dentro do Código

Pergunta 1: Se usada a estrutura g_sem_op1 terá qual efeito em um conjunto de semáforos?

R: Se a estrutura g_sem_op1 for usada, com o valor de sem_op sendo 1, o conjunto de semáforos será destrancado.

Pergunta 2: Para que serve esta operação semop(), se não está na saída de uma região crítica?

R: A operação semop() serve para operar os semáforos, por exemplo, na entrada de uma região crítica, ele tranca o mesmo.

Pergunta 3: Para que serve essa inicialização da memória compartilhada com zero?

R: Essa inicialização em zero se dá devido ao fato da memória compartilhada estar armazenando um contador, sendo assim necessário começá-lo em 0.

Pergunta 4: se os filhos ainda não terminaram, semctl e shmctl, com o parâmetro IPC-RMID, não permitem mais o acesso ao semáforo / memória compartilhada?

R: Não, pois tais comandos com esse parâmetro sinalizam ao sistema operacional que a área de memória compartilhada e semáforo podem ser removidos.

Pergunta 5: quais os valores possíveis de serem atribuídos a number?

R: Os valores possíveis são 1, 2 e 3 somente.

Conclusão

Neste experimento podemos observar a eficiente de mecanismos de Mutex, no caso específico foi utilizado semáforo, o qual não permitiu o acesso mútuo a uma área crítica do programa, evitando assim uma desordem na impressão do conteúdo desejado. Em sistemas multprogramados sistemas de mútua exclusão são amplamente utilizados para evitarem as chamadas condições de corridas, que muitas vezes pode acontecer em regiões de memória compartilhada, assim gerando inconsistência nos dados.

Este experimento nos permitiu compreender melhor o funcionamento de semáforos, regiões de memória compartilhada e a utilização de regiões críticas. Com isso conseguimos melhorar nossa percepção das condições de *race conditions* e seus perigos.