

UNIVERSIDADE DE SÃO PAULO

JOÃO AUGUSTO FERNANDES BARBOSA - 11953348

JANTAR DOS FILÓSOFOS

22 DE MAIO DE 2022

1 - Lógica do código:

A estruturação do código seguiu como base o modelo de texto do livro texto. Neste, alguns filósofos se reúnem para jantar. Suas ações são limitadas à pensar, comer, agarrar os garfos e deixá-los. O problema se dá no momento em que cada filósofo, ao tentar comer, busca agarrar os recursos mais próximos a si, ou seja, à sua esquerda ou direita. Entretanto, nesse ponto, haverá um conflito, uma vez que um dos recursos pode ser motivo de competição.

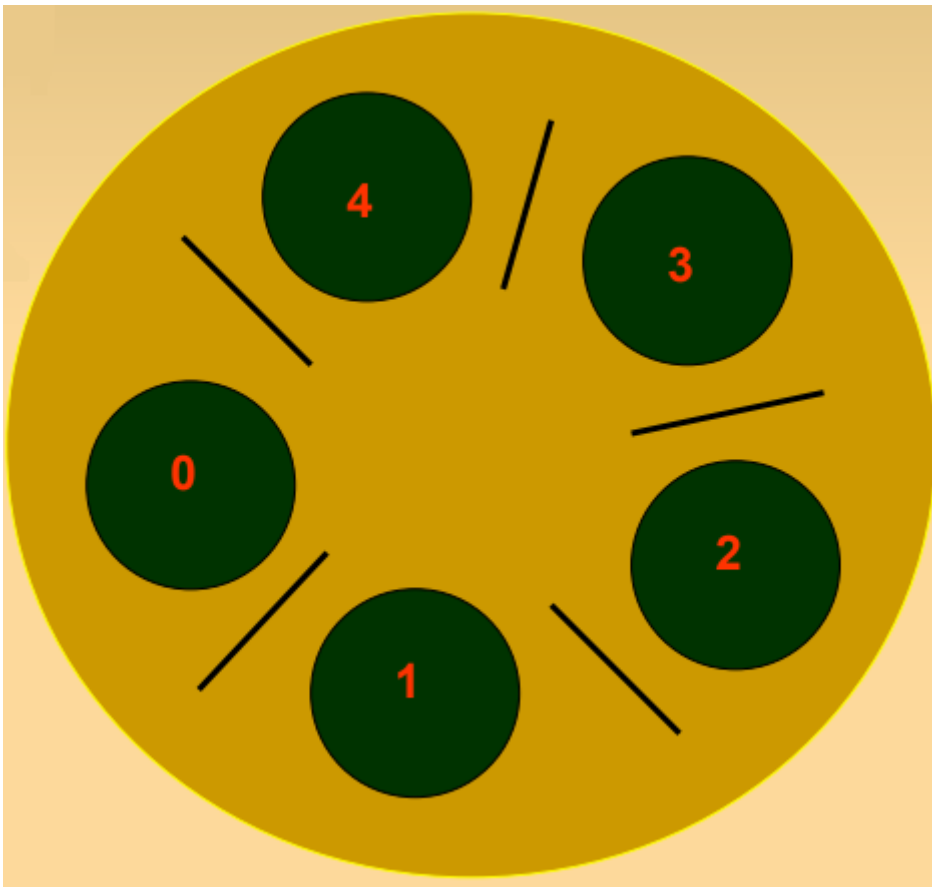


imagem 1- representação esquemática do problema. Os hashis, por estarem dividido entre os filósofos, são recursos escassos

Desta maneira, inicialmente, definiu-se algumas variáveis para o sistema, como o número de filósofos, seu estado e monitores. No geral, os monitores foram usados tanto para impedir duas threads de acessarem ao mesmo tempo o mesmo recurso quanto para que, enquanto um dos filósofos estivesse usando o recurso, àqueles ao seu redor não pudessem interferir.

Assim, criou-se uma thread para que, indefinidamente, os filósofos passassem pelo seu ciclo de comer e pensar. Ao agarrar um garfo, o filósofo

inicialmente verifica se este está disponível. Caso esteja, ele travará o recurso e atualizará o valor do semáforo. Ao deixar o garfo, verifica-se se este está sendo requisitado - ou seja, se o filósofo ao lado está com fome.

2 - Resultados

Os resultados obtidos podem ser observados na seguinte imagem:

```
0 filosofo 0 esta pensando.  
0 filosofo 1 esta pensando.  
0 filosofo 2 esta pensando.  
0 filosofo 3 esta pensando.  
0 filosofo 4 esta pensando.  
0 filosofo 0 está com fome.  
0 filosofo 0 agarrou os garfos  
0 filosofo 1 está com fome.  
0 filosofo 2 está com fome.  
0 filosofo 2 agarrou os garfos  
0 filosofo 0 deixou os garfos.  
0 filosofo 0 esta pensando.  
0 filosofo 3 está com fome.  
0 filosofo 4 está com fome.  
0 filosofo 4 agarrou os garfos  
0 filosofo 0 está com fome.  
0 filosofo 2 deixou os garfos.  
0 filosofo 2 esta pensando.  
0 filosofo 1 agarrou os garfos  
0 filosofo 4 deixou os garfos.  
0 filosofo 4 esta pensando.  
0 filosofo 3 agarrou os garfos  
0 filosofo 1 deixou os garfos.  
0 filosofo 1 esta pensando.
```

imagem 1 - print dos resultados para o código implementado do trabalho

Como pode ser observado, o retorno foi uma situação que à priori segue os conceitos estabelecidos em aula. Inicialmente, os filósofos estão todos pensando. Pode ser percebido que o primeiro filósofo a agarrar os garfos é o 0. Ou seja, nessa situação, é impossível para o filósofo 1 comer, já que um de seus garfos está sendo usado. Assim, mesmo que logo em seguida apareça “O filósofo 1 está com fome”,

ele não entra no loop de comer. Já o filósofo 2 que aparece com fome posteriormente ao filósofo 1, ainda come antes, por não ser travado.

4 - Problemas encontrados

No meu ver, o maior problema foi a falta de contato que tive com threads em C. A maior parte do ensino nos cursos anteriores, principalmente POO, foi em JAVA, então ao implementar em C tive certa dificuldade. Além disso, conceitualmente falando, tive uma dificuldade semântica na interpretação de livros textos sobre o problema. Isso porque alguns tratavam os recursos como garfos, e na minha cabeça não fazia sentido um filósofo precisar de dois garfos para comer. Só quando vi a abordagem tratando de Hashis que consegui compreender melhor o que a situação representava.

5 - Instruções para rodar o programa

Compilar:

```
gcc -pthread -o a.out DiningPhil.c
```

Rodar:

```
./a.out
```