

## Lista de Exercícios - Sistemas Operacionais

### Aula 5: *Comunicação entre Processos*

**Professores:** Valmir C. Barbosa e Felipe M. G. França

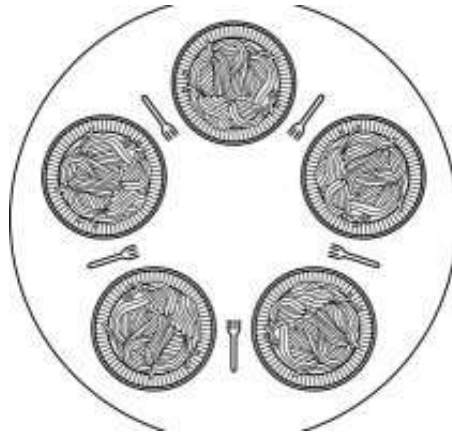
**Assistente:** Alexandre H. L. Porto

1. Considere um computador que não tenha uma instrução TEST AND SET LOCK (TSL) mas tenha uma instrução para comutar o conteúdo de um registrador e de uma palavra da memória em uma única ação indivisível. É possível utilizar essa instrução para escrever uma rotina **ENTER\_REGION** como a reproduzida a seguir (vista na Aula 5)?

ENTER\_REGION:

```
TSL REGISTER, LOCK
CMP REGISTER, #0
JNE ENTER_REGION
RET
```

2. Um problema clássico da comunicação de processos é o problema dos filósofos. Neste problema, os filósofos estão dispostos em torno de uma mesa circular e cada filósofo passa por períodos alternados de comer e de pensar. Sobre a mesa existe um prato de espaguete à frente de cada filósofo e um garfo entre dois pratos consecutivos, como podemos ver na figura a seguir, em que temos 5 filósofos (e 5 garfos e pratos de espaguete):



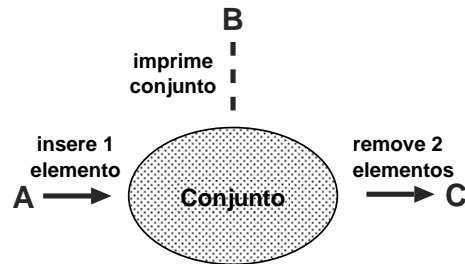
Quando um filósofo está com fome, ele tenta pegar os garfos à sua esquerda e à sua direita, pois o espaguete está muito escorregadio e por isso são necessários dois garfos para comê-lo. Ao conseguir pegar os garfos, o filósofo come por um tempo um pouco do espaguete e depois recoloca os garfos na mesa e volta a pensar.

A seguir é dada uma solução para o problema com  $n$  filósofos baseada em  $n$  semáforos binários, sendo que o semáforo  $fork[i]$  está associado ao garfo  $i$ ,  $1 \leq i \leq n$ . Os semáforos são todos inicializados com o valor 1. Esta solução funcionará se os processos que implementam os filósofos executarem a função *philosopher* dada a seguir, sendo que cada um destes processos está associado a um identificador  $i$ ,  $1 \leq i \leq n$ ? Justifique a sua resposta.

```
void philosopher(int i)
{
    while (1) {
        pensar();
        P(fork[i]);
        P(fork[(i+1) % n]);
        comer();
        V(fork[i]);
        V(fork[(i+1) % n]);
    }
}
```

3. O conjunto dado na figura a seguir, que pode armazenar até  $n$  elementos, é compartilhado por três processos, A, B e C. O processo A sempre coloca um elemento no conjunto, o processo B sempre imprime

o conteúdo atual do conjunto e o processo C sempre remove dois elementos do conjunto. Supondo que a estrutura de dados que representa o conjunto somente armazene os elementos do conjunto, como os semáforos podem ser usados para garantir o correto funcionamento dos processos A, B e C? Justifique a sua resposta.



4. Suponha que dois processos, A e B, compartilhem uma pilha, usada para armazenar números. Suponha ainda que esta pilha, inicialmente vazia, possa armazenar até  $n$  números, e que não exista uma variável para contabilizar a quantidade de números armazenados na pilha. O processo A continuamente insere números na pilha. Já o processo B continuamente remove dois números da pilha, calcula a soma destes números, e depois insere o resultado da soma na pilha. Como os semáforos devem ser usados para garantir a correta execução dos processos A e B? Justifique a sua resposta.