

**Instituto de Matemática e Computação**  
**SRSC02 – Sistemas Operacionais**  
**Exercício Prático 03 – EP03 02/10/2024**

Vida de Hoare

Em um determinado stand de uma feira, um demonstrador apresenta um filme sobre a vida de Hoare. Quando 10 pessoas chegam, o demonstrador fecha o pequeno auditório que não comporta mais do que essa platéia. Novos candidatos a assistirem o filme devem esperar a próxima exibição. Esse filme faz muito sucesso com um grupo grande de fãs (de bem mais de 10 pessoas), que permanecem na feira só assistindo o filme seguidas vezes. Cada vez que um desses fãs consegue assistir uma vez o filme, ele vai telefonar para casa para contar alguns detalhes novos para sua mãe. Depois de telefonar ele volta mais uma vez ao stand para assistir o filme outra vez.

Usando semáforos, foi modelado o processo fã e o processo demonstrador, lembrando que existem muitos fãs e apenas um demonstrador. Como cada fã é muito ardoroso, uma vez que ele chega ao stand ele não sai dali até assistir o filme. Suponha que haja muitos telefones disponíveis na feira e, portanto, que a tarefa de telefonar para casa não impõe nenhuma necessidade de sincronização.

OBS:

Observe que o demonstrador só pode começar a exhibir o filme quando há 10 pessoas no stand, e que as pessoas que chegam durante uma exibição têm que esperar a próxima.

Importante: observe que um fã só pode ir telefonar para a mãe depois que acaba a exibição do filme! Isso tem que estar modelado na sincronização entre os processos demonstrador e fãs.

Pede-se implementar o algoritmo a seguir em C/C++ utilizando semáforos e mutex permitidos pela biblioteca Pthreads:

constante N 10

inteiro nFans=0;

semaforo mutex = 1;

semaforo dem = 0;

semaforo fila = 0;

fan (){

    enquanto(verdadeiro){

        down(mutex);

        nFans = nFans + 1;

        up(mutex);

        up(dem);

        down(fila);

        assisteFilme();

        telefona();

    }

}

demonstrator (){

    enquanto(verdadeiro){

        enquanto( nFans < N )

            down(dem);

        down(mutex);

```

        nFans = nFans - N;
        up(mutex);
        para ( i = 0; i < N; i++)
            up(fila);
        exhibeFilme();
    }
}

```

Escreva o programa principal para criar vários fans e apenas um demonstrador e verifique se o algoritmo funciona, poderá utilizar a função `sleep()` nas funções de `assisteFilme()`, `telefona()` e `exibeFilme()`, `assisteFilme()` e `exibeFilme()` devem demorar o mesmo tempo. Criar mais fans que o N que foi definido.

Será necessário entregar somente o arquivo .C.