

## Exercício Prático de COM120 – EP03 – 09/09/19

### Problema da Montanha Russa

Existem  $n$  passageiros, que repetidamente aguardam para entrar em um carrinho da montanha russa, fazem o passeio, e voltam a aguardar. Vários passageiros podem entrar no carrinho ao mesmo tempo, pois este tem várias portas. A montanha russa tem somente um carrinho, onde cabem  $C$  passageiros (  $C < n$  ). O carrinho só começa seu percurso se estiver lotado. Sincronize as ações dos processos Passageiro e Carrinho usando semáforos.

Uma possível solução é mostrada abaixo:

```
semaforo passageiro = C;
semaforo carrinho = 0;
semaforo andando = 0;
semaforo mutex = 1;
int Npass = 0;

Passageiro() {
    while (true) {
        DOWN(passageiro)
        entra_no_carrinho() /* vários passageiros podem entrar "ao mesmo tempo" */
        DOWN(mutex)
        Npass++
        if (Npass == C) {          /* carrinho lotou */
            UP(carrinho)           /* autoriza carrinho a andar */
            DOWN(andando)         /* espera carrinho parar */
            UP(mutex)
        }
        else {
            UP(mutex)
            DOWN(andando) /* espera carrinho lotar, passear e voltar */
        }
    }
}

Carrinho() {
    while (true){
        DOWN(carrinho) /* espera autorização para andar */
        passeia()      /* faz o passeio e volta */
        Npass := 0     /* esvazia carrinho */
        for (int i=0; i<C; i++){
            UP(andando); /* libera passageiro que andou de volta à fila */
            UP(passageiro); /* libera entrada no carrinho */
        }
    }
}
```

Tente implementar este algoritmo utilizando Threads, Semáforos e Mutex em linguagem C/C++.