

# Sistemas Operativos

## Licenciatura em Engenharia Informática

### Exercícios de problemas de sincronização de processos – soluções

Para os seguintes exercícios assuma que um **semáforo** *S* tem disponível as instruções *wait(S)* e *post(S)*.

- 1) Considere um problema de uma passagem de nível em que existem carros e comboios que querem passar. Insira semáforos no código seguinte de modo a evitar que haja colisões, passando uma viatura de cada vez.

```
mutex = semaphore(1)
```

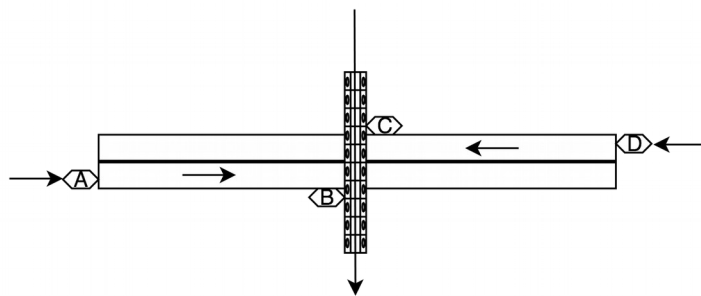
**Car:**

```
wait(mutex)
go()
post(mutex)
```

**Train:**

```
wait(mutex)
go()
post(mutex)
```

- 2) Considere uma via com trânsito nos dois sentidos em que existe uma passagem de nível controlada pelos semáforos *B* e *C*. Por se situar numa zona sensível, existem também os semáforos *A* e *D* que devem limitar o número de carros que podem estar simultaneamente entre as zonas *A-B* e *C-D* (máximo de 8).



```
A = semaphore(8), B = semaphore(1)
D = semaphore(8), C = semaphore(1)
```

**Train:**

```
wait(C)
wait(B)
go_down()
```

**Car-from-left:**

```
wait(A)
go_right()
wait(B)
```

**Car-from-right:**

```
wait(D)
go_left()
wait(C)
```

<b>post(B)</b>	<b>go_right()</b>	<b>go_left()</b>
<b>post(C)</b>	<b>post(B)</b>	<b>post(C)</b>
	<b>post(A)</b>	<b>post(D)</b>

- a) Sabendo que os comboios apenas se deslocam de cima para baixo, e que demoram algum tempo a se deslocarem, modifique o código de modo a incluir os semáforos necessários para garantir que o comboio passa pela estrada sem colidir com nenhum carro. Assuma que os comboios têm sempre prioridade em relação aos carros.
  - b) Modifique o código dos carros que vêm da esquerda de modo a garantir que não colidem com nenhum comboio e que respeitam os limites de carros na zona A-B.
  - c) Repita o anterior para os carros que vêm da direita.
- 3) Um professor do departamento de informática tem horário de atendimento aos alunos às segundas-feiras. Como o seu gabinete é pequeno, apenas pode atender um aluno de cada vez. Existem três cadeiras à entrada em que os alunos devem sentar-se e aguardar enquanto o professor está a atender outros alunos. Enquanto não houver alunos, o professor pode dormir uma sesta (a vida de professor é dura!). Se um aluno entretanto aparecer, este deverá acordar o professor. Se o professor já estiver a atender um aluno, este deverá aguardar numa das cadeiras à entrada, senão deverá aguardar que haja uma cadeira vazia. Usando semáforos, implemente uma solução que coordene as actividades do professor e dos alunos.

```

Prof = semaphore(1)
Chair = semaphore(3)
Nap = semaphore(1)

```

**Professor:**

```

repeat
    wait(Nap)
    meet_student()
forever

```

**Student:**

```

wait(Chair)
    sit_in_chairs()
wait(Prof)
post(Chair)
post(Nap)
    meet_professor()
post(Prof)

```