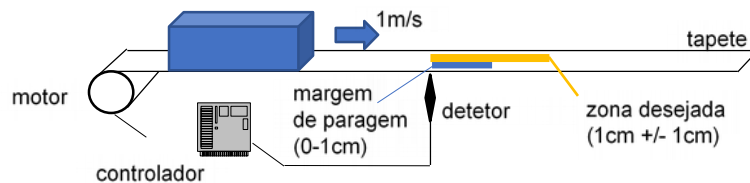


- 1 Considere um tapete de transporte industrial que transporta peças e que as deve parar numa dada posição, com uma precisão de  $\pm 1\text{cm}$ . O tapete tem um controlador que deteta as peças numa posição adequada e determina a sua paragem. A velocidade do tapete é constante e de  $1\text{m/s}$  e a sua paragem pode deixar cada peça até  $1\text{cm}$  à frente da posição em que estava quando o comando de paragem foi atuado.



- 1.a(1 valor) Qual o tempo máximo de reação do controlador, desde que a peça aparece no alcance do detetor até à atuação de paragem do tapete, para que a precisão desejada seja alcançada?
- 1.b(2 valores) Suponha que o controlador executa um programa escrito em C que na sua função *main* tem o seguinte código:

```
while (1) {
    if (read_detector()) stop_belt();
    usleep(T);
}
```

A função `read_detector()` retorna um valor booleano e demora  $2\text{ms}$  a executar e a função `stop_belt()` pára o tapete e demora  $0,5\text{ms}$  a executar. Qual o valor máximo de  $T$  (em  $\mu\text{s}$ ) que permite cumprir o tempo de reação identificado na alínea anterior? (se não respondeu à alínea anterior considere  $5\text{ms}$ ).

- 2 Um sistema de controlo de um robô móvel e autónomo tem 3 tarefas que executam comportamentos reativos básicos, localização e planeamento de rotas, respetivamente. Essas tarefas executam periodicamente, de acordo com *Deadline Monotonic*, mas sem sincronização, e têm as seguintes propriedades (em  $\text{ms}$ ):

$\tau_1 \rightarrow$	$C_1=1, D_1=5, T_1=10$	(comportamentos reativos básicos)
$\tau_2 \rightarrow$	$C_2=5, D_2=10, T_2=20$	(localização)
$\tau_3 \rightarrow$	$C_3=20, D_3=T_3=100$	(planeamento de rotas)

- 2.a (2 valor) Qual a utilização e a densidade totais deste sistema de tarefas?
- 2.b(2 valores) Determine os tempos de resposta de pior caso.
- 2.c(2 valores) Desenhe um gráfico de Gantt da execução das tarefas entre uma ativação síncrona ( $t=0$ ) e  $t=40\text{ms}$ .
- 2.d(2 valores) As tarefas  $\tau_1$  e  $\tau_3$  trocam dados através de um *buffer* partilhado, ao qual acedem durante  $0,5$  e  $1\text{ms}$ , respetivamente. Para garantir consistência dos dados, o acesso é protegido com um semáforo. Qual o bloqueio que as 3 tarefas podem sofrer se o semáforo tiver herança de prioridades e se não tiver. Comente a escalonabilidade do sistema em cada caso.
- 2.e (1 valor) Que diferença faria se em vez de *Deadline Monotonic* executasse estas tarefas (sem considerar recursos partilhados) com *Earliest Deadline First*?

- 3.a (2 points) Explain how the `SCHED_RR` and `SCHED_FIFO` scheduling policies of a POSIX operating system can co-exist during runtime. What impact do they have on each other?
- 3.b (2 points) Two tasks running in distinct processes wish to have access to the same memory area (so as to exchange data). Explain why this is/is not possible under a POSIX operating system.
- 4 The gcc (GNU C compiler) manual states that it may be called with the option “-fpic”:  
*“Generate position-independent code (PIC) suitable for use in a shared library, if supported for the target machine. Such code accesses all constant addresses through a global offset table (GOT). The dynamic loader resolves the GOT entries when the program starts (the dynamic loader is not part of GCC; it is part of the operating system).”*
- 4.a (2 point) Explain what is a shared library and why the code for a shared library should be position independent.
- 4.b (2 point) Explain the different objectives of a typical .h and a .c file. How does the compiler handle an “#include” directive?

### FORMULAS

Menor majorante	$U(n) = \sum_{i=1}^n (C_i/T_i) \leq n(2^{1/n}-1)$
Majorante hiperbólico	$\prod_{i=1}^n (C_i/T_i+1) \leq 2$
Response Time	$R_{wc_i}(0) = C_i + B_i + \sum_{k \in hp(i)} C_k$ $R_{wc_i}(m+1) = C_i + B_i + \sum_{k \in hp(i)} \lceil R_{wc_i}(m)/T_k \rceil * C_k$
Synchronous busy period	$L(0) = \sum_i C_i$ $L(m+1) = \sum_i \lceil L(m)/T_i \rceil * C_i$
Load function	$h(t) = \sum_{D_i \leq t} (1 + \lfloor (t - D_i)/T_i \rfloor) * C_i$