

# Arquitetura de Computadores 2017/18

## TPC 4

**Prazo de entrega: 23:59 de 25 de maio de 2018**

**Tópicos:** IO e processamento de ficheiros em C.

Este trabalho de casa consiste em dois exercícios de programação **individual**. Pode esclarecer dúvidas gerais com colegas, mas a solução e a escrita do código deve ser estritamente individual. Todas as resoluções serão comparadas de forma automática e os casos de plágio serão punidos de acordo com os regulamentos em vigor.

A entrega será via Mooshak do DI (<http://mooshak.di.fct.unl.pt/~mooshak/>). Os programas serão compilados com base no comando `make` no primeiro exercício e, no segundo, com:

```
cc -m32 -Wall -std=c11 -o prog prog.c
```

### Exercício 1

Recorde o exercício da ficha 9. Pretende-se agora implementar a leitura de bytes que chegam pela porta série. Obtenha o código em `iosim2.zip` e implemente a função `recvByte` em `driver.c`, tendo em conta que deve aguardar a chegada de cada novo byte corretamente recebido. Em caso de erro, tal é assinalado no registo LSR, devendo ignorar esse byte, como se não tivesse sido recebido. No registo LSR, cada um dos seus bits fica a 1 caso a condição enunciada seja verdadeira e a 0 se falsa:

<b>bit 0</b>	Chegou um byte que está disponível no registo RBR
<b>bit 1</b>	Ocorreu um erro durante a recepção. O byte em RBR está corrompido.
<b>bit 5</b>	O registo de transmissão (THR) está livre e pode-se ordenar o envio de um novo byte

Os bits 0 e 1 passam a 0 assim que RBR for lido.

Para testar compile o programa `teste2` (use `make teste2`). Este programa mostra no ecrã todos os bytes lidos da “porta série”. Durante a execução, a simulação da chegada de bytes pela porta série é efetuada se escrever no seu teclado, assim, durante a execução do `teste2`, de cada vez que escreve uma linha de texto deve ver aparecer uma cópia **exata** do que escreveu.

Submeta ao mooshak apenas o seu ficheiro “`driver.c`” e nada mais.

### Exercício 2

Pretende-se implementar um programa que permita trocar todas as ocorrências de determinada sequência de bytes num ficheiro por outra sequência fornecida. O programa deve ler um ficheiro indicado e criar um novo ficheiro para onde vai copiar o conteúdo do primeiro, alterado de acordo com o pretendido. Para tal, recebe da linha de comando (via `argv` do `main`) os seguintes argumentos, por esta ordem:

1. a sequência a trocar, indicada como uma string.
2. a sequência de caracteres que vai substituir a anterior.
3. o nome do ficheiro a ler.
4. o nome do ficheiro a criar com a cópia do anterior depois da substituição pretendida.

Note também que as dimensões das duas strings não têm de ser iguais, podendo mesmo a segunda ser vazia (“”). Por exemplo, pode-se pedir para apagar a sequência “abc”. Neste caso o seu programa será executado da seguinte forma:

```
prog abc "" fich1 fich2
```

Como resultado, se `fich1` contiver:

```
Zcababcde  
Xabc
```

o `fich2` deve ficar com:

```
Zcabde  
X
```

**Nota:** no geral podem existir casos de sequências com subsequências repetidas, como por exemplo, procurar “aab” num ficheiro que algures contém “aaab”. Considere que a sequência a procurar não começa por nenhum padrão que se repete, podendo ignorar situações como a anterior para facilitar a sua solução.