

Instituto Superior de Engenharia de Lisboa  
Licenciatura em Engenharia Informática e de Computadores  
**Programação de Sistemas Computacionais**  
Verão de 2021/2022  
Série de Exercícios 1

---

Realize os exercícios seguintes usando a linguagem C. Não se esqueça de testar devidamente o código desenvolvido, bem como de o apresentar de forma cuidada, apropriadamente indentado e comentado. Assegure-se de que o compilador não emite qualquer aviso sobre o seu código, mesmo com a opção `-Wall` activa. Contacte o docente se tiver dúvidas. Não é necessário relatório. Encoraja-se a discussão de problemas e soluções com outros colegas, mas a partilha directa de soluções leva, no mínimo, à anulação das entregas de todos os envolvidos.

1. Programe a função `count_trailing_zeros` que conta o número de zeros existentes, numa palavra de 64 *bits*, entre a posição de menor peso e a posição a um com menor peso. Por exemplo: se a palavra for `0xf0` o resultado é 4 e se for `0x6000000` o resultado é 25.

```
int count_trailing_zeros(unsigned long word);
```

Use um algoritmo mais eficiente que a simples iteração sobre todos os bits da palavra. Pode basear-se na informação desta página: [https://en.wikipedia.org/wiki/Find\\_first\\_set](https://en.wikipedia.org/wiki/Find_first_set).

2. Considere a necessidade de criação de um vetor de valores inteiros codificados a 24 *bits*, sobre o qual se podem executar as funções `vec24_write` e `vec24_read` que realizam, respectivamente, a escrita e a leitura de valores no vetor. `vec24_write` escreve o valor passado em `value` na posição especificada por `index`, `vec24_read` devolve o valor contido na posição `index`.

```
void vec24_write(unsigned int vector[], int index, unsigned int value);
```

```
unsigned vec24_read(unsigned int vector[], int index);
```

Use como estrutura de dados de suporte um array de inteiros do tipo `unsigned int` e armazene a informação de forma compactada de modo a utilizar todos os bits de memória disponíveis. Na versão de compilador de C que estamos a utilizar na unidade curricular, o tipo `unsigned int` é representado com 32 bits. Assim, três posições do array podem suportar o armazenamento de quatro posições do vetor, pois correspondem ambas a 96 *bits*.

3. Implemente, em linguagem C, a função `string_find_substitute` que procura em `string` as ocorrências da *substring* `string1` e substitui-as pela *substring* `string2`. Admita que a memória de suporte a `string` é suficiente para suportar o aumento da dimensão de `string`, que ocorre quando a dimensão de `string2` é superior à dimensão de `string1`.

Na programação desta função utilize funções da biblioteca, tais como `strncmp`, `strlen`, `memmove`, `memcpy` ou outras.

```
void string_find_substitute(char *string, char *string1, char *string2);
```

4. Considere a utilização dos tipos `Block` e `Stream` na criação de uma estrutura de dados para armazenar sequência de *bytes* de dimensão diversa. A sequência é suportada numa cadeia de blocos, em que cada bloco é representado por uma *struct* `Block` - o campo `size` indica a dimensão do bloco e o campo `data` aponta para o início dos dados do bloco. A sequência total é representada por uma *struct* `Stream`, em que o campo `n_blocks` indica o número de blocos utilizados e o campo `blocks` aponta para um *array* de ponteiros para *struct* do tipo `Block`.

Programe a função `stream_read` que preenche o *buffer* apontado por `buffer` com o conteúdo da sequência de *bytes*, indicada por `stream`, a partir da posição `offset` e de dimensão `size`.

```
typedef struct block { size_t size; char *data; } Block;
```

```
typedef struct stream { size_t n_blocks; Block **blocks; } Stream;
```

```
size_t stream_read(Stream *stream, size_t offset, size_t size, char *buffer);
```

5. Realize o programa utilitário **file\_dump**, que apresenta o conteúdo de um ficheiro no *standard output*. O ficheiro pode ser indicado por argumento do programa ou recebido via *standard input*. O programa deve comportar-se com funcionalidade idêntica à do utilitário **hexdump** quando invocado com a opção **-C** (experimentar: **\$ hexdump -C filename**).

Data recomendada de entrega: 9 de abril de 2022

ISEL, 8 de março de 2022