# Exercício Prático 1 - encriptar - 3

Printed by Pedro Quaresma

Set 24, 10 10:56 **encriptarDS.c** Page 3/3

```
  //Aplicar a Cifra de Deslocamento Simples
  while((ch=fgetc(fl))!=EOF){
    resultado=encriptarDS(ch,chave);
    fprintf(fe,"%c",resultado);
  }


  //Fechar os ficheiros
  fclose(fl);
  fclose(fe);

  exit(0);

}
```

Sexta Setembro 24, 2010 encriptarDS.c 3/3





# Byteisses

Para a implementação da cifra FEAL é necessário recorrer a um conjunto de manipulações em "bytes".

- rotações de *N*bits - `ROTN`.
- "Mascarar" um dado conjunto de "bits".
- Separar uma sequência de "bits" em partes.
- Juntar várias partes numa só sequência de "bits".
- Ou exclusivo (XOR) entre "bytes".





# Tipos e Operações

Uma variável do tipo (*unsigned*) *int* pode ser manipulada na sua forma binária.

| Operação (em bits) | Significado |
|---|---|
| & | Conjunção |
| \| | Disjunção |
| ^ | Disjunção exclusiva (XOR) |
| >> *n* | Deslocamento de *n* bits para a direita |
| << *n* | Deslocamento de *n* bits para a esquerda |

Para lidar com inteiros com um número bem definido de "bytes" temos a biblioteca *stdint* (`#include <stdint.h>`) a qual define os seguintes tipos (com as respectivas operações).

```
int8_t  int16_t  int32_t  int64_t
uint8_t  uint16_t  uint32_t  uint64_t
```





# FEAL8 - Caixas S

Na implementação das caixas S temos:

$$S_d(x, y) = \mathrm{ROT2}(x + y + d \bmod 256)$$

isto é, temos:

- Rotação de 2 bits para a esquerda;
- adição módulo 256.

Soluções:

- `ROTN`(x) = (x << N) — (x >> 6)

```
ROT2(11101001) = (11101001 << 2) | (11101001 >> 6)
                 (10100100)      |   (00000011)
                          (10100111)
```

- Basta fazer a operação de adição em variáveis de 8bits (ignorando o transporte).

```
uint8_t d, uint8_t x, uint8_t y, uint8_t S;
```

## Separar "Bytes"

- Na função $f_k$ temos duas variáveis de entrada de 32bits que têm de ser divididas em secções de 8bits.
- Na função $f$ temos duas variáveis de entrada, uma de 32bits e outra de 16bits, ambas têm de ser divididas em secções de 8bits.

Soluções:

- "Máscaras" — designa-se por máscara ("mask") um padrão de bits que serve como forma de realçar um dado conjunto de "bits" através da conjunção binária.
  Por exemplo: 32bits quer-se realçar o segundo "byte".
  - Máscara:
    $65280_{10} = FF00_{16} = 00000000000000001111111100000000_2$
  - Conjunção com a máscara: $A \leftarrow A$ & $Mascara$

```
  11100101 11010101 11110010 10101010
& 00000000 00000000 11111111 00000000
-------------------------------------
  00000000 00000000 11110010 00000000
```





## Separar "Bytes" (continuação)

Temos então que, para dividir uma variável de 32bits em secções de 8bits, temos de fazer:

- Aplicar uma máscara apropriada e retirar o primeiro "byte";
- Deslocar a variável um "byte" para a direita;
- repetir o processo com os restantes "bytes".

```
A3 = A & 255; A >>= 8;
...
```

```
A & 255
  11100101 11010101 11110010 10101010
& 00000000 00000000 00000000 11111111
-------------------------------------
  00000000 00000000 00000000 10101010
A >>= 8
A = 00000000 11100101 11010101 11110010
```





## Juntar "Bytes"

Tanto na função $f_k$ como na função $f$ é necessário, no fim, "juntar" secções de 8bits numa só sequência de 32bits.
Solução:

- Definir uma variável de tamanho apropriado;
- Deslocar cada uma das secções para a sua posição correcta;
- Combinar todas as secções numa só sequência.

```
(X0 << 24) | (X1 << 16) | (X2 << 8) | X3;
```

```
X0 = 11100101; X1 = 11010101; X2 = 11110010; X3 = 10101010
X0 << 24 11100101 00000000 00000000 00000000
X1 << 16 00000000 11010101 00000000 00000000
X2 <<  8 00000000 00000000 11110010 00000000
X3       00000000 00000000 00000000 10101010
|        11100101 11010101 11110010 10101010
```





## Inteiros de Gama (quase) Infinita

**GMP — GNU Multiple Precision Arithmetic Library**

O que é a biblioteca GMP?

O GMP é uma biblioteca livre para a aritmética de precisão e gama de variação arbitrárias, implementa inteiros com sinal, racionais, e reais.
Não tem limites fixos para a precisão ou gama de variação, que não sejam os impostos pelas limitações em memória do sistema computacional que se está a usar.
A biblioteca GMP possuí um conjunto muito rico de funções, sendo que cada uma delas possuí um interface normalizado.

As principais aplicações da biblioteca GMP são, sistemas criptográficos, segurança da Rede, sistemas algébricos, etc.

`http://gmplib.org/`