

Códigos e Criptografia

Cristina Caldeira & Pedro Quaresma

Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

2010/2011

1 / 22

C - Argumentos na Linha de Comandos

Em *C* é possível fornecer a um dado programa argumentos no momento em que se invoca o programa.

Exemplos disso são os múltiplos comandos acessíveis através da linha de comandos, os quais não são mais do que o resultado da compilação de um programa escrito e *C*.

```
ls -l
```

ls é o nome do programa *-l* é um argumento que é passado para o programa.

No caso das cifras esta característica é muito importante dado que nos permite incorporar as cifras como ferramentas auxiliares num dado sistema de criptografia/criptoanálise.

2 / 22

Argumentos da Função `main`

A passagem de argumentos na linha de comando é feito através do especificar da função de entrada num programa (a função `main`) com parâmetros.

A função `main` é capaz de aceitar dois argumentos especiais:

- o primeiro refere-se ao número de argumentos da linha de comando;
- o segundo é a lista completa dos argumentos dado na linha de comandos.

3 / 22

Argumentos da Função `main` (cont)

A declaração completa da função `main` é então do tipo:

```
int main ( int argc, char *argv[] )
```

- O inteiro `argc` é o contador do número de argumentos. É o número de argumentos que são escritos na linha de comandos aquando da invocação do programa, sendo que esse número também inclui o nome do programa.
- O vector de ponteiros para caracteres (vector de "strings") contém a lista de todos os argumentos dados na linha de comandos. O primeiro elemento, `argv[0]`, é o nome do programa, os restantes elementos contêm os diferentes argumentos dado na linha de comandos.

4 / 22

Como usar

Utilizando a funcionalidade dos argumentos da linha de comando é usual começar um programa por validar (pelo menos o número) os argumentos dados e, em caso de erro, enviar uma mensagem de ajuda.

Por exemplo:

```
int main(int argc, char *argv[]) {
    int i,chave;
    char *strChave,*msgClara,*msgCifrada;
    FILE *pfM, *pfC, *fopen();

    if (argc<=3) { // valores de entrada insuficientes (3+)
        printf("\nUtilização: encriptarDS chave msgClara msgCifrada\n\n");
        return;
    }
}
```

C - Ficheiros

Ler/Escriver/Acrescentar

- inclusão da biblioteca: `#include <stdio.h>`
- declaração de variáveis do tipo ficheiro: `FILE *f1,*fe;`
- abrir o ficheiro para leitura: `f1 = fopen("f_leitura","r");`
- abrir o ficheiro para escrita: `fe = fopen("f_escrita","w")`
- leituras e/ou escritas como normalmente, mas com a inclusão do canal de entrada/saída: `fscanf(f1,..., fprintf(fe,...`

Exercício Prático 1 - encriptar - 1

(falta o ficheiro `cifraDS.h` e os "abaixo" dele)

```
Set 24, 10 10:56 encriptarDS.c Page 1/3
// Copyright (c) 2008
// versão 1.0 desenvolvido por Sidnei Cruz
// (sob orientação de Doutor Pedro Quaresma)

#include <stdio.h>
#include <string.h> // permite a utilização de funcoes que manipulam strings
#include <stdlib.h> // biblioteca para funcao exit
#include "cifraDS.h"
#include "alfabeto.h"

int main(int argc, char *argv[]){
    //f1 -> o ficheiro de entrada(leitura)
    //fe -> o ficheiro de saida(escrita)

    FILE *f1,*fe;

    //verificar o número de argumento
    if(argc!=4){
        printf("\nUtilização: encriptarDS nome_ficheiro_entrada chave nome_ficheiro_saida\n\n");
        exit(1);
    }

    //Declaração das variáveis
    char *mensnormal, *menscifrada;
    int ch;
    int caracter;

    Sexta Setembro 24, 2010 encriptarDS.c 1/3
```

Exercício Prático 1 - encriptar - 2

```
Set 24, 10 10:56 encriptarDS.c Page 2/3
int resultado;
int chave;

//cópia do ficheiro de entrada
mensnormal = argv[1];

//funcao atoi recebe uma string e devolve o inteiro que nela esta representado
chave=atoi(argv[2]);

//cópia do ficheiro de saida
menscifrada = argv[3];

//abrir o ficheiro para a leitura
f1 = fopen(mensnormal,"r");

//verificar se o ficheiro foi aberto ou nao
//caso contrario o programa termina
if(f1==NULL){
    printf("Impossivel abrir o ficheiro %s\n", argv[1]);
    exit(2);
}

//Abrir o ficheiro para escrever a mensagem cifrada
if((fe = fopen(menscifrada,"w"))==NULL){
    printf("Impossivel criar o ficheiro %s\n", menscifrada);
    exit(3);
}

Sexta Setembro 24, 2010 encriptarDS.c 2/3
```