

Aula Prática 5

Resumo:

- Robustez e Excepções.

Exercício 5.1

Altere o programa 4.2 da aula anterior por forma a torná-lo totalmente robusto. Em particular, o programa deve garantir a sanidade dos argumentos do programa assim como as entradas resultante da interacção com o utilizador.

Exercício 5.2

Crie um programa que copie um ficheiro de texto¹. Os nomes dos dois ficheiros envolvidos devem ser dados como argumentos na linha de comandos². Assim a execução do programa com os argumentos `Texto1.txt` `Texto2.txt` deve criar um ficheiro `Texto2.txt` com um conteúdo igual ao do ficheiro `Texto1.txt`.

Nota: Faça com que o programa seja robusto, não só detectando a existência do ficheiro original (apresentando uma mensagem de erro quando este não existe), como também a possível existência do ficheiro destino (neste caso fará sentido perguntar ao utilizador se de facto deseja destruir esse ficheiro). Deve também verificar se sobre os ficheiros se podem realizar as operações de leitura e escrita, e se algum deles é um directório (caso em que deve ser apresentada uma mensagem de erro).

Exercício 5.3

Crie um programa que apresente alguma informação sobre os ficheiros existentes num determinado directório. O nome do directório é passado como argumento do programa, excepto quando o programa não tem argumentos caso em que se considera que o directório é o actual (directório: `.`). Por cada ficheiro existente no directório deve ser indicado se o mesmo corresponde a um directório e quais as suas permissões de leitura e escrita.

A saída do programa deve ter o seguinte aspecto (nome escrito em, pelo menos, 20 colunas):

¹Pretende-se que seja uma versão simplificada do comando UNIX `cp`.

²No programa Java esses valores farão parte do *array* de *strings* que é passado como argumento da função `main`.

dir1	(DIR,READ,WRITE)
dir2	(DIR,READ)
dir3	(DIR)
fich1	(READ,WRITE)
ficheiro2	()

O texto **DIR** indica que o ficheiro é um directório, e os textos **READ** e **WRITE** indicam, respectivamente, que existe permissão de leitura e escrita.

Exercício 5.4

Altere o programa 4.4 da aula anterior por forma a torná-lo totalmente robusto. Em particular, o programa deve garantir a sanidade dos argumentos do programa assim como as entradas resultante da interacção com o utilizador.

Exercício 5.5

Construa um programa que dado um ficheiro de texto de entrada escreva somente as palavras existentes numa determinada coluna em cada linha do texto (se não existir nenhuma palavras deve escrever uma linha em branco). Considere que as palavras são separadas por espaços em branco ou por caracteres de tabulação.

Quer o nome do ficheiro quer o número da coluna devem ser dados como argumentos do programa.

Por exemplo, no caso de termos o seguinte texto de entrada:

1	2	3	4	5	6
		boa		tarde	
um	dois	tres	quatro	cinco	seis

A saída do programa, se for escolhida a coluna número 1, deverá ser:

1
boa
um

No caso da coluna número 4, a saída passará a ser:

4
quatro