

## Aula Prática Nº 7

### Objetivos

- Acesso a ficheiros binários
- Uso do `switch` para tratamento de argumentos
- Utilização das funções `getopt` e `getopt_long`

### Guião

1. Leia atentamente o código fonte `writePeople.c` e `readPeople.c`.

- a) Compile e execute os programas passando-lhes o nome de um ficheiro binário como argumento e interprete o resultado.
- b) Altere o programa `writePeople.c` de modo a pedir ao utilizador a informação sobre cada pessoa que se pretende armazenar no ficheiro. O programa deve começar por perguntar ao utilizador quantas pessoas são para guardar.
- c) Adicione ao programa `readPeople.c` a capacidade de acrescentar informação sobre mais pessoas, garantindo no final a salvaguarda de todas as pessoas no mesmo ficheiro. Sugere-se que comece por alterar o programa de modo a ler o conteúdo do ficheiro para um *array*, que os dados das pessoas adicionais sejam lidas para o *array* e que no final o *array* seja escrito no ficheiro binário. Pode considerar que nunca existem mais do que 100 pessoas.

2. Leia atentamente o código fonte `testOpt.c`.

- a) Compile, execute o programa e interprete o seu resultado. Este programa mostra como se pode processar argumentos de um programa utilizando a função `getopt`. Este exemplo permite gerir duas opções do programa: `-n` que não tem valor associado e `-t val` que espera um valor depois da opção. É suposto também passarmos um argumento que não é opção.
- b) Utilizando este mecanismo de tratamento de argumentos, crie o programa `myls` (dica: ver exercício 4 da Aula 6) que aceita 3 opções: `-f` que apenas lista ficheiros, `-d` que apenas lista directorias, `-e ext` que apenas lista ficheiros com extensão `ext` e que pode ter (ou não) como argumentos várias directorias.

3. Leia atentamente o código fonte `testLongOpt.c`.

- a) Compile, execute o programa (experimentando combinações das opções `--add`, `--append`, `--create`, `--reset`, `-a`, `-b`, `-c`, `-0`, `-1`, `-2` e `-3`) e interprete o seu resultado. Este programa mostra como se pode processar argumentos de um programa

utilizando a função `getopt_long`. Este exemplo permite gerir várias opções, nos seus modos curtos ou longos.

- b) Utilizando este mecanismo de tratamento de argumentos, altere o programa `mys` para que este aceite as opções: `--file` equivalente a `-f`, `--dir` equivalente a `-d` e `--ext` equivalente a `-e`.