**Aula Prática Nº 6**

## Objetivos

* Acesso a ficheiros de texto
* Acesso ao conteúdo de directorias

## Guião

1. Leia atentamente o código-fonte myCat.c.
   1. Compile e execute o programa passando-lhe o nome de um ficheiro de texto como argumento e interprete o resultado.
   2. Altere o programa de modo a garantir o seu funcionamento quando o utilizador passa vários ficheiros como argumentos.
   3. Acrescente ao programa a capacidade de numerar as linhas impressas (isto significa que cada linha deve ser precedida do número de linhas desde o início do ficheiro).
   4. Execute o programa obtido na alínea anterior usando como argumento o ficheiro myCat.c. Deve notar algo estranho na impressão da linha mais longa e na contagem de linhas. Investigue a razão. Altere o código de forma a resolver o problema.
2. Leia atentamente o código-fonte sortNumbers.c.
   1. Compile e execute o programa passando-lhe vários números como argumentos e interprete o resultado.
   2. Altere o programa de modo a ordenar o conteúdo de um ficheiro de texto contendo números inteiros, um por linha, cujo nome seja passado como argumento. Assuma que o ficheiro a ordenar não tem mais que 100 números. Crie duas versões do programa:

i) usando a função fgets() para ler o ficheiro.

ii) usando a função fscanf() para ler o ficheiro.

* 1. Considere agora que não quer limitar o número de valores armazenados no ficheiro. Explore diferentes soluções para resolver este problema. [Sugestões: pode efetuar duas consultas ao ficheiro (uma para contar os valores seguida de outra para efectuar a leitura) ou considerar que o primeiro número contido no ficheiro é a indicação do número de valores.]

1. Leia atentamente o código-fonte myActions.c.
   1. Compile e execute o programa; interaja com ele escrevendo comandos (que podem incluir *pipes* e redireccionamentos) até introduzir a palavra “end”. Interprete o resultado.
   2. Altere-o de modo a gravar o texto que o utilizador insere, bem como a hora a que o faz, num ficheiro de nome “command.log”. Sucessivas chamadas ao programa devem acrescentar conteúdo a esse ficheiro.
2. Leia atentamente o código-fonte dirList.c.
   1. Compile e execute o programa passando-lhe como argumento o nome de uma directoria e interprete o resultado.
   2. Altere o programa de modo que, na listagem, as directorias sejam precedidas por “d ” (d seguido de espaço) e os ficheiros por “ ” (2 espaços).
   3. Complete o programa acrescentando a capacidade de fazer a listagem recursiva de todas as directorias encontradas. [Sugestão: invoque recursivamente a função listDir().]