UNIVERSIDADE DA CORUÑA

FACULTY OF COMPUTER SCIENCE
Programming II

Running the Script

In each practical, the main program will implement a loop to process the requests to be executed one by one. To simplify both the development and testing of the system, instead of prompting the user to type the data corresponding to each request, the program must **work in *batch* mode** (i.e. without interactivity with the user). Instead, it will take as input a text file containing the sequence of requests to be executed. For each loop iteration, the program will read a new request from the file and, then, the request will be processed.

To be able to test different files, the name (and path, if needed) of the input file will NOT be *hardcoded* into the program. Instead, it will be passed as a parameter when running it. Suppose that, for example, the input file is `/tmp/input.txt`. So, when running the program, we will need to indicate (assuming the executable file is `main`):

```
./main /tmp/input.txt
```

If no input file is specified, the program will use by default the file indicated in the file `main.c` provided (`new.txt`). In this case, it will be called just with:

```
./main
```

The arguments `nargs` and `args` of function `main` in file `main.c` indicate, respectively, the number of arguments that are passed to function `main` and a list with all of those arguments. They will be useful for the implementation of this new functionality.

By reading the input in this way, we can use a script to test different files all at once. Thus, together with the documentation of the practical, a ZIP file is provided. This file, named `script.zip`, contains a script (`script.sh`) that allows us to test both the `test.c` and the `main.c`. The zipped file includes 2 folders (`script_test_unit` and `script_minimos`) that contain all the input files needed joint with their corresponding expected outputs (files ending in `ref`). In order to run this script correctly, all the `.c` files with the code of our practical (`test.c`, `main.c`, `static_list.c` and `dynamic_list.c`) must be placed in a directory of the teaching lab computers or the reference server and, then, unzip the `script.zip` there too. Next, from the terminal and after going to that folder, we must give the file `script.sh` permissions for execution using:

```
chmod u+x script.sh
```

and, finally, run said script. Two options are available:

```
1) ./script.sh  (equivalent to ./script.sh -p main)
2) ./script.sh -p test
```

In the first case, the main program is run with all the input files contained in the folder script_minimos. As a result, two output files are created in this same folder (e.g. new_Dynamic.txt and new_Static.txt). These files contain the output of running the main program (main.c) with each of the input test files (new.txt, vote.txt, illegalize1.txt and illegalize2.txt) using the static and dynamic implementations of the ADT, respectively. In the case of the second option, the program test.c is executed, again, for both ADT implementations, and the output obtained is compared with the expected one (ref.txt).

Note that the script itself modifies the main.c and test.c files to include the name of the corresponding library, compile, and then execute. If a compilation error is found for any of the libraries, the corresponding result file will only contain the names of the different files that are being tested.

The script includes an option -v that, apart from indicating whether the output is correct or not, also shows on screen those lines where the program's output differs from the reference output:

```
./script.sh -v
```