NOVA SCHOOL OF SCIENCE & TECHNOLOGY
DEPARTAMENTO DE INFORMÁTICA

Tempo

HOMEWORK 1

To be submitted for evaluation to mooshak.di.fct.unl.pt until end of 6th October 2023

This work is individual, and all solutions will be compared. Don't look at other students' code and don't show your own to others.

Objective

Implement in C/Unix your version of the Unix/Posix *time* command (see manual). This will measure the time it takes to execute a given command.

Program

Your program will be used like the standard *time* command, receiving from the command line the command to execute and its arguments. Your program should execute the requested command and wait for its completion, measuring the time it takes to complete. Also, if the last argument starts with '=' this argument defines the file for the standard output of the target command and is not part of this command.

As an example, to execute the command "ls -l" and getting its execution time:

```
tempo ls -l
total 424
-rw-r--r--@ 1 vad  staff  95323 Sep 25 16:31 HW-1.docx
-rwxr-xr-x  1 vad  staff  49792 Sep 22 16:36 tempo
-rw-r--r--  1 vad  staff    706 Sep 22 16:36 tempo.c
-rw-r--r--  1 vad  staff   1059 Sep 25 12:02 tempo2.c
```

`Elapsed time = 0.007433 s`

The first line is the one given by the user to execute the command, next is the output of "ls -l" and, in bold, the output from the "tempo" command. The tempo command will write a **newline** followed by the message with the seconds measured, accordingly to the *printf* format "**`Elapsed time = %6f s\n`**".

As an example of output redirection, you can execute the command "tempo ls -l =out.txt" and the 'ls' should get its standard output redirected to the 'out.txt' file:

```
tempo ls -l =out.txt
```

`Elapsed time = 0.007433 s`

The output of '`ls -l`' goes to file '`out.txt`'.

Any problems when launching the target command should produce an error message accordingly to the system error in **`errno`**. For example, if the command to execute, *cmd*, doesn't exist in the PATH directories, the traditional message "*cmd*: No such file or directory" is printed to *stderr* (see `perror()`).

Suggestions

Look at the code from Lab 01, at your code from Lab 02 and examples from theoretical classes or recommended books.