PRCMP PL10
Shell scripts 2

May 2023

1 Shell scripts

1. Create a script that generates a random integer between 1 and 10 and counts the number of attempts the user needs to guess the number.

   Remember that the `RANDOM` variable produces a random number between 0 and 32767 whenever it is invoked.

2. Create a second version of the previous game, implementing the following changes.

   - the script generates a number between 1 and 100, and
   - for each wrong guess, the script indicates whether the guess was greater or less than the hidden number.

3. A given directory that has a large number of files of the same type (file extension `.dat`), with names that were randomly generated. It is easier for a human to distinguish between files if each one is identified by a number. For this reason, we intend to add a sequential counter as a prefix to the file names, as shown in the following example:

```
xljjrtje.dat ---> 1-xljjrtje.dat
jlkriuss.dat ---> 2-jlkriuss.dat
iubaspeo.dat ---> 3-iubaspeo.dat
...
```

   Develop a script that meets the requirements listed below. The script must terminate whenever one of the requirements is not met.

   R1: Receives exactly one argument passed as a parameter on the command line: the path to a directory.

   R2: The directory indicated as a parameter must be valid.

   R3: Files with the .dat extension that are found in the indicated directory will be renamed with a sequential number as a prefix, as in the example given. Each new name must be presented to the user.

   R4: The script should inform the number of files renamed.

4. You were asked to create a shell script that automatically downloads a series of files from the web. This script should receive the name of a text file specified by the user as a command line parameter. The file must contain a list of uniform resource locators (URL), one URL per line, as shown in the example below:

```
http://api.ipma.pt/open-data/forecast/meteorology/cities/daily/1131200.json
https://isep.sport.pt/atleta/1220000/act-20230120-1830.json
https://isep.sport.pt/atleta/1220000/act-20230121-1900.json
...
```

Develop a script that meets the requirements listed below. If an error occurs and the script is not able to continue, an error message must be displayed to the user before exiting.

R1: The script takes exactly one argument passed as a parameter on the command line: the name of the text file.

R2: The file given as a parameter must exist.

R3: The script must inform the user of the URLs that were not downloaded (by error).

R4: The script should report the number of URLs successfully downloaded.

TIP 1: The `curl` command allows you to download a URL by creating a local copy, using the `-O` option, as in the example below. `curl` returns the exit code 0 (zero) on success.

Example: `curl -O https://some_URL_here.go`

TIP 2: A file can be read line-by-line into a variable as follows:

```
while read -r line
do
    echo $line
done < some_file_name_here
```