TITLE

Programming a script with the BASH interpreter

GOALS

To learn the possibilities of the command interpreter as a basic tool for the Linux system administrator.

DESCRIPTION

We will develop a *script* in the BASH language. This *script* is a tool for management of student practical assignments in Linux.

This practical assignment has to be defended before December 16th, asking for an appointment for me to see it working in the lab. Then, the program will be uploaded so that other aspects that are considered can be checked, as described in the last section of this handout.

It is recommended to develop two different scripts. The main *script* is collect-prac.sh, it does not receive any arguments and all the working options are introduced through a simple menu described below.

The second script (store-prac.sh) will collect the practical assignments of students when called. So, the main script will program a task for cron that will call the store-prac.sh script. Since the second script does not have any interaction with the user, it should receive all the information needed through arguments sent by the main script (at least, the course and origin and destination directories for the practical assignments).

The errors that occur during the execution of both scripts and the actions performed will be registered in a log file called prac.log.

Following, we see the menu presented by the main script collect-prac.sh. This script will execute until the user chooses the option "End program".

```
ASO 24/25 – Script Assignment
Student name:

Practical assignment management
-------------------------------

Menu
   1) Program collection of assignment solutions
   2) Pack course assignments
   3) See size and date of a course backup file
   4) End program

   Option:
```

Option 1 programs a task for cron to collect all the assignments of a given course. To program the task, the main script needs to obtain from the user certain data about the

course: original location of the assignments and destination directory. If the original directory does not exist, an error message will be sent to the standard output and to the log file, and the user will be asked for the information again. If the destination directory does not exist, the script will just create it.

This assignment is thought to be executed by the administrator (root user). If anybody needs to execute it as a regular user, the directories used should be created under the home directory of the user so that there are no problems with permissions.

Following, we see the menu presented when the user chooses the first option of the main script and an example of answers that we could provide.

```
Menu 1 – Program collection of assignment solutions

Course: ASO <RET>
Path containing student accounts: /home/aso <RET>
Path to store assignments: /prac/aso <RET>

The ASO assignment collection process is programmed for tomorrow
at 8:00. Origin: /home/aso. Destination: /prac/aso

Do you want to continue? (y/n) y <RET>
```

<RET> marks the end of input data.

We assume that under the directory containing the student accounts there is a directory for each course student, and under this there is a file called prac.sh with the solution to the practical assignment. For example, /home/aso/aperez/prac.sh contains the solution for the practical assignment for the student with login aperez.

On the other hand, the script store-prac.sh will store the assignments of all the students in the destination directory, giving them the name login.sh. For example, the assignment just mentioned will be stored in the file /prac/aso/aperez.sh.

As a result of the execution of this option, a task for cron will be programmed. This task will consist of a call to store-prac.sh to store the assignments of the specified course.

Option 2 packs the assignments of a given course in a file. The menu presented when choosing this option is:

```
Menu 2 – Pack course assignments

Course: ASO <RET>
Absolute path of directory with the assignments: /prac/aso <RET>

The assignments of the course ASO present in the directory
/prac/aso will be packed.

Do you want to continue? (y/n) y <RET>
```

If there is any problem (for example, the directory to save does not exist), the script presents an error message in the standard output and in the log file, and returns to the main menu.

Following, we describe some details about the packing process:

- We will use the command `tar` for packing, and you need to specify a compressing option for the file to occupy less space.
- The packed file should be stored in the directory where we previously stored the assignments (for example, `/prac/aso`).
- The name of the file is `course-YYMMDD.tgz`. If the date of today is November 13, 2024, the file generated in the process of packing will be called "`aso-241113.tgz`".

Option 3 obtains information about the size and date of the file generated when packing the course assignments.

```
Menu 3 – Obtain size and date of backup

Course: ASO <RET>

The file generated is aso-241113.tgz. Its size is <n> bytes
and it was generated on <date>.
```

Option 4 ends the execution of the main script.

Besides registering traces about operations, in the file **prac.log** we have to register all the errors that occur during the execution of the scripts. The format of the trace lines is:

```
Date  Hour  Description of operation/error
```

REVIEW
- Some commands that may be useful for the implementation of this assignment are: **echo, tar, date, cut, find, basename, dirname, wc, crontab**
- Control sentences of the BASH language
- Use of functions to structure the program
- **read** sentence for data input

EVALUATION
The grading will consider the following aspects:

- The program should work according to the description.
- Good presentation of the program: spaces, indentation, use of functions and control sentences, and any other aspect contributing to the program's legibility.
- Error control so that the program does not perform illegal operations.
- Registration of errors and cron information in the log file.