

Useful Actions for Users at Solar System

Unit 03

Angel Berlanas Vicente

Introduction



Warning to Seafarers

PATHS, **URLS** and common stuff for scripts could be placed as the first lines of the scripts, in order to establish at the beginning the values and be ready to be changed if the **sysadmin** needs.

The examples for this task are presented in a Windows OS, but you can imagine the correct paths for GNU/Linux.

Task 01

Choose one language: BASH, PowerShell or Python, and then code a Script that:

- Accepts 3 argument.
- The first one must be a **valid user** of the System. In other case, the script must exit and display a message:

```
1 * Error - The User given not exists.
```

- The second argument must be one of these words:
 - **replenish**
 - **test**
 - **clean**

The behaviours of the actions are described below:



Useful Actions for Users at Solar System

replenish

In the [HOME](#) directory of the Given user, the script must create the next folder structure:

```
1  -- SolarSystem/  
2      Mercury/  
3      Venus/  
4      Mars/  
5      Jupiter/  
6      Saturn/  
7      Uranus/  
8      Neptune/
```

In each of the folders of the Solar System Planets the script must create a single text file with the name of the user as the content of the file with the date of the creation. The file must be named as: `control-planet.txt`.

```
1  -- SolarSystem/  
2      Mercury/  
3          control-planet.txt  
4      Venus/  
5          control-planet.txt  
6      Mars/  
7          control-planet.txt  
8      Jupiter/  
9          control-planet.txt  
10     Saturn/  
11     ??????  
12     Uranus/  
13         control-planet.txt  
14     Neptune/  
15         control-planet.txt
```

In the Saturn folder (because has rings), you must NOT include this file and instead of it, you must include a file named as :

```
1  Environment + "the user given " + "the date in format YYYYmmdd-hhMM " +  
    ".txt"
```

Wich content will be the list of the current Environment Values for this user.

```
1  user@mars:~$ ./solarSystem.sh admin replenish "All is done"  
2  # Doing stuff  
3  # Creating file at:  
4  /home/admin/SolarSystem/Saturn/Environment-admin-20221212-1030.txt  
5  .....  
6  .....  
7  All is done
```

Useful Actions for Users at Solar System

In the **replenish** action, the third argument must be Written as a message when everything was complet, as a “All is done” message.

test

If the second argument is test, then the script must accepts a third argument that must be a valid PATH to a SolarSystem folder. The script must test if the folder is in a *correct place* and test if the user given has Read Permissions of all the files and folders under the PATH given.

Examples of usage:

```
1 user@mars:~$ ./solarSystem.sh admin test /non-solarSystem-directory
2 * Test 1 : The user admin exists.
3 * Test 2 : The action test is a valid action.
4 * Error! - The folder given to test not exists: /non-solarSystem-
  directory
```

Now, if the previous behaviour (with replenish) was run for the user *Administrador*, the output must be:

```
1 user@mars:~$ ./solarSystem.sh admin test $HOME/SolarSystem
2 * Test 1 : The user admin exists.
3 * Test 2 : The action test is a valid action.
4 * Test 3 : The folder given to test exists: /home/admin/SolarSystem
5 -- The folder : /home/admin/SolarSystem/Venus exists and could be read
  by admin.
6 -- The folder : /home/admin/SolarSystem/Mars exists and could be read
  by admin.
7 ...
```

clean

If the second argument is clean, then the script must accepts a third argument that must be a valid PATH to a SolarSystem folder. The script must test if the folder is in a *correct place* and remove it and its files and sub-folders.

