PRCMP PL09
Shell scripts

May 2023

1 Shell scripts

1. Write the "Hello, world" script that simply prints "Hello, world!" on the terminal.
   (a) Use the `nano` editor to edit the `hello_world.sh` file. Write the source code, save and exit.
   (b) Add execute permission to the `hello_world.sh` file.
   (c) Run your script by issuing the command line:

   ```
   ./hello_world.sh
   ```

2. Write a script that asks the user for his name. The script should greet the user by the name that was entered.
3. Write a script that generates an output composed of three parts.
   - First, you must print the text line "Content of the user's *username* home directory:" where *username* is the current system user name.
   - Second, you should list all content (files and directories) in the user's home directory, regardless of the current working directory.
   - Finally, you must print the current date (hint: see the `date` command).
4. Write a script that asks the user for the path to a directory and lists the contents of that directory.

   The script should check if the directory exists. If the path is not valid, the script must print an error message to the stderr and terminate with exit status 1.

   Tip: To print text to the stderr you can add the `>&2` redirector:

   ```
   echo This message goes to stderr >&2
   ```

5. Write a second version of the script from the previous question, where the path to the directory is passed as the first command line argument.

   The script should verify that the command line has exactly one argument. If it does not have exactly one argument, the script must print the error message "usage: exercise5.sh directory" to the stderr (to inform the user about how to use the script) and terminate with exit status 2.

6. Write a script that asks the user for a command to execute.

   After executing the command, you should inform the user whether or not the command was successful.

   Tip: use the commands `true` (which always ends successfully) and `false` (which always ends with failure) to test your script.

7. Create a script that, when receiving a file name as a command line argument, shows its content on the screen.

   If the file does not exist, the script should print "Error: file not found" to the stderr and terminate with exit status 1.

Do not forget to check if the command line has exactly one argument. If not, the script must print the error message "usage: exercise6.sh file" to the stderr (to inform the user about how to use the script) and terminate with exit status 2.

8. Create a script that asks the user to enter an exam score and displays a message based on this score:
   - "Very good" if the score is between 18 and 20;
   - "Good" when it is between 14 and 17;
   - "Average" if the score is between 10 and 13;
   - "Insufficient" if the score is less than 10.

   The script should alert the user if the score is not in the range between 0 and 20.

9. Create a script that receives from the command line a sequence of file names and determines the type of each of them.

   If the command line does not have any arguments, the script must print the error message "usage: exercise9.sh file ..." to the stderr (to inform the user about how to use the script) and terminate with exit status 1.

   Tip: Remember the command `file`.

10. Create a script that counts how many files and subdirectories the current working directory contains.

2 Solutions

1. Possible solution:

```
#!/bin/bash

echo 'Hello, world!'
exit 0

```

2. Possible solution:

```
#!/bin/bash

read -p "Name: " name
echo "Hi, $name\!"
exit 0

```

3.

4.

5. Possible solution:

```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "usage: $(basename $0) directory" >&2
    exit 2
fi

directory=$1
if [ ! -d "$directory" ]; then
    echo "$directory is not a directory\!" >&2
    exit 1
fi

ls $directory
exit 0

```

6.

7.

8.

9.

10.