Bash scripting 1

processes practice for CSC420 Operating Systems Spring 2022 Lyon College

README

- This file accompanies lectures on the shell and `bash(1)`. To gain practice, you should type along in your own Org-mode file. You have to have Emacs and my `.emacs` file installed on your PC or the Pi you're working with.
- This section is based on chapter 24 of Shotts, The Linux Command Line (2e), NoStarch Press (2019), and on the DataCamp course "Introduction to Bash Scripting".
- To make this easier, use the auto expansion (`<s`). This will only work if you have my `.emacs` file (from GDrive) installed.

- Add the following two lines at the top of your file, and activate each line with `C-c C-c` (this is confirmed in the echo area as `Local setup has been refreshed`)):

```
#+PROPERTY: header-args:bash :results output
```

- Remember that `C-M-\` inside a code block indents syntactically (on Windows, this may only work if you have a marked region - set the mark with `C-SPC`).

Overview

- A shell script is a file containing a series of commands.
- The shell is both a **command line interface** to the OS and a **scripting language interpreter**.
- The shell reads the file, interprets and carries them out as if they had been entered on the command line.

How to write a shell script

- Write the script in a text editor (Emacs or vi or nano)
- Make the script executable by setting the file permissions
- Put the script somewhere the shell can find it

Script file format

- `[X]` Fire up an editor and create a "Hello World" program `hello.sh`. You can use `vi` or `nano` if you like!
  - In Emacs, `C-x C-f hello.sh` to create the file, and `C-x C-s` to save it
  - In vi, write `vi hello.sh` in the terminal, insert with `i`, save with `:w` and exit with `:q`
- `[X]` Put a comment in after the command, using `#`
- `[X]`

  You got to get the location of your `bash` program right.

```
which bash  # likely in /usr/bin/bash
```

  First line of your script should look like this:

```
#!/usr/bin/bash
```

- `[X]` If successful, run the same command in the terminal (including the comment.
- The first line of the script is the *shebang* to tell the kernel the name of the interpreter that should be used to execute the script.
- `[X]` Make a copy of the file, find a different interpreter (e.g. `csh`, the C shell) and modify the file accordingly.

Executable permissions

- `[X]` Check file permissions with the command `chmod`
- `[X]` Make your file executable. Check the permissions.

Script file location

- `[X]` Save and run the file on the shell (you can do that inside Emacs with `M-x shell`).
- For the file to run, an *explicit* path has to be provided, otherwise you get the `Command not found` error
- The 'source' operator `.` executes bash on the current location. It is a shell builtin that reads a specified file of shell commands and treats it like input from the keyboard.
- `[X]`

  Print the path that the OS searches when looking for a command:

  ```
  echo $PATH
  ```

- `[X]` Make a directory `/bin` in your home directory and add it to the `PATH` using the syntax `PATH=$HOME/bin:$PATH`
- `[X]` Check that `PATH` was altered as you wanted. The new directory should be the first in the list.
- `[X]` Copy `hello.sh` to that new directory and run the file again from your current location.
- `[ ]`

  To apply this change of `PATH` whenever `bash` is called, you need to include this line in your initialization file `$HOME/.bashrc`:

  To find the file:

  ```
  ls -la .bashrc
  ```

  ```
  export PATH=$HOME/bin:$PATH
  ```

  To append this line to `.bashrc` do:

  ```
  echo "export PATH=$HOME/bin:$PATH" >> .bashrc
  ```

  To check if the appending was successful (`cat` works, too):

  ```
  tail -1 .bashrc
  ```

  - `[ ]`

    To make the change, you have to source the `$HOME/.bashrc` file using the source operator `.`:

    ```
    . .bashrc
    ```

Summary

- `[X]` How to write a shell script in 3 steps
- `[X]` Script file format with *shebang*
- `[X]` Permission to execute with `chmod`
- `[X]` Location with `$PATH`

References

- Shotts, The Linux Command Line (2e), NoStarch Press (2019).
- DataCamp, Introduction to Bash Scripting (course).

Author: Marcus Birkenkrahe
Created: 2022-04-14 Thu 08:36