Scripting

Dr Peadar Grant

March 6, 2024

Contents

1 What is Scripting?

In CLI environments commands are used to carry out tasks (get-process, ls, mkdir, etc):

- These commands may be executed individually
- May be combined using pipelining and redirection.

Scripting adds familiar programming constructs to the command-line environment.

- May need to execute a number of commands together as a single unit.
- Commands can be saved to a file - called a script.
- Used to automate repetitive tasks.

2 Scripting concepts

- A script contains a number of commands that will be executed in the order they are presented in.
- Scripts are interpreted, line by line, as the script is run by the interpreter.
- Error in scripts may result in scripts failing or making unexpected changes.

3 Shells

- Particular installation may have a number of different shells:

  Windows: CMD, PowerShell

  Linux: Bash, zsh, C-shell, ksh

  Mac: zsh, Bash

- Use cat /etc/shells to view the available shells
- Scripts are written for a particular shell (BASH, c shell, etc)
  - First line of a script should identify the shell the script runs on.

4 Bash

- The default shell for Ubuntu is BASH.
- Scripts should be saved with a .sh extension
- Some text editors in Linux recognise scripts and format the text accordingly.

5 Facilities

Scripts contain all the usual programming components:

- Variables
- Input (from keyboard and via arguments)
- Output (using echo)
- Control Flow (looping – for, while, case, etc)
- Conditional Statements (if, ifelse, etc)
- Functions

6 Structure of a Script

A script should follow the basic structure:

```
#!/bin/bash
# Good idea to state brief purpose here
# Also put your name!

echo commands and scripting constructs go here
```

7 File Permissions

Knowledge of File Permission is essential for running scripts:

- By default, files cannot be executed.
- Not even root can execute a file unless the file owner adds execute permission manually.
  - This is a security feature.
- Use chmod and symbolic or numeric notation to change permissions:
  - `chmod a+x filename`
  - `chmod 777 filename`

8 Creating a Shell Script

1. Create a file using an editor
2. Start with header to specify interpreter:
    - Normally `#!/bin/bash`
3. Good idea to have comment lines with purpose and your name
4. Then commands (and scripting constructs) follow.
5. Save the file with the .sh extension
6. Change permissions on file to execute it e.g. `chmod +x ./myscript.sh`.
7. Run the file e.g. `./myscript.sh`