

Bash Scripting - Basics

...

07-131 Great Practical Ideas in CS

Instructors: Eduardo Feo-Flushing & Giselle Reis

Course website: <https://web2.qatar.cmu.edu/cs/07131/>

In the Beginning... Was the Command Line

- What's a shell?
 - Execution commands
 - Scripting commands
 - Interactive
- A bit of history
 - 1971: Thompson shell
 - 1979: Bourne shell
 - 1989: Bourne-again shell (bash)



Commands (review)

Task	Command
Show directory contents	<code>ls</code>
Change directory	<code>cd directory_name</code>
Move a file	<code>mv file.txt location</code>
Rename a file	<code>mv file.txt renamed.txt</code>
Copy a file	<code>cp file.txt copy.txt</code>
Execute a binary	<code>./binary_file</code> <code>binary_file</code> <code>path/binary_file</code>
Print something	<code>echo "Hello World"</code>

Shell scripting

- Sometimes, you want to run the same set of commands multiple times.
- Shell scripts allow you to run a sequence of shell commands from a file.
- Shells can be executed as binary files



Example

Shebang

```
1 #!/usr/bin/env bash
```

```
2
```

```
3 touch source_file.c
```

```
4 mkdir src
```

```
5 mv source_file.c ./src
```

```
6
```

```
7
```

```
8
```

Regular
commands

```
bash_script.sh
```

chmod

- Files are not executable by default

```
twildenh@unix5:~/private/script$ ./script.sh  
-bash: ./script.sh: Permission denied
```

- Have to add executable permission

- `chmod +x script.sh`

- Then we can run the script

```
twildenh@unix5:~/private/script$ ./script.sh  
Hello World!
```

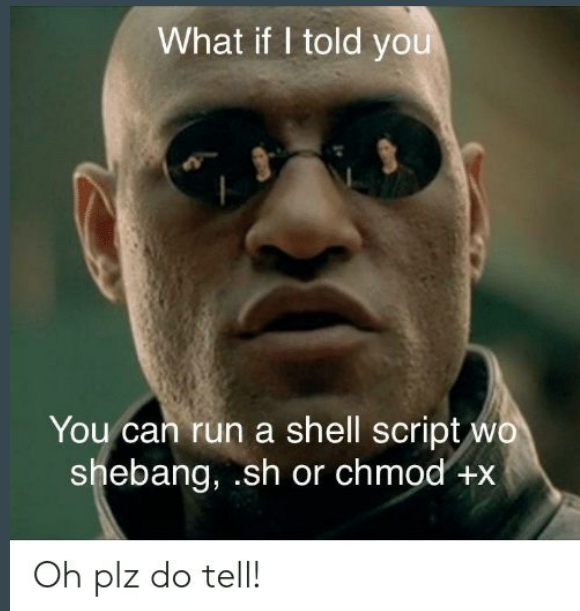
- chmod changes the mode to add(+) executable

Bash programming

- Apart from interfacing with the operating system, they support other programming constructs as well
 - Variables
 - Conditionals
 - Loops
- Variable assignment: VAR=value
 - Note: NO SPACES IN VARIABLE ASSIGNMENT!!!
- Variable Access: echo \$VAR
- Command Line Arguments: \$1, \$2, \$3, and so on

Scripting summary

- Bash scripts end in a .sh extension
- Always start with a shebang
 - `#!/usr/bin/env bash`
- Add permissions with
`chmod +x script.sh`



Example

- You want to make a new directory for each 15112 HW initialized with git to stay organized
- You give the script a HW number, and it would create the directory for you
- You could create a script like the one on the right

```
1 #!/usr/bin/env bash
2
3 HW_NAME=15112_hw$1
4 mkdir $HW_NAME
5 cd $HW_NAME
6 touch README.md
7 touch main.py
8 git init
9 git add .
10 git commit -m "Create homework directory"
```

Globs

Bash does carry out filename expansion

known as globbing

- **Question mark - (?)**: matches any single character.
 - You can use '?' for multiple times for matching multiple characters.
- **Asterisk - (*)**: matches zero or more characters
- **Square Brackets ([])**: matches a character in the range. E.g.,
 - [:upper:] or [A-Z]: all upper characters
 - [:lower:] or [a-z]: all lowercase characters
 - [:digit:] or [0-9]
 - [:alnum:] or [a-zA-z0-9]
- Many more ...

Ranges - { .. }

- Surrounded with brackets
- Expands into all possible permutations
- Comma-separated for list of values
- Use “..” to indicate a range of letters or numbers

Matches	Range
• {1,2,3} OR {1..3}	• 1 2 3
• {cmu,gpi,rocks}	• cmu gpi rocks
• bug{1..3}	• bug1 bug2 bug3
• {a..c}.{1..3}	• a.1 a.2 a.3 b.1 b.2 b.3 c.1 c.2 c.3

Strings

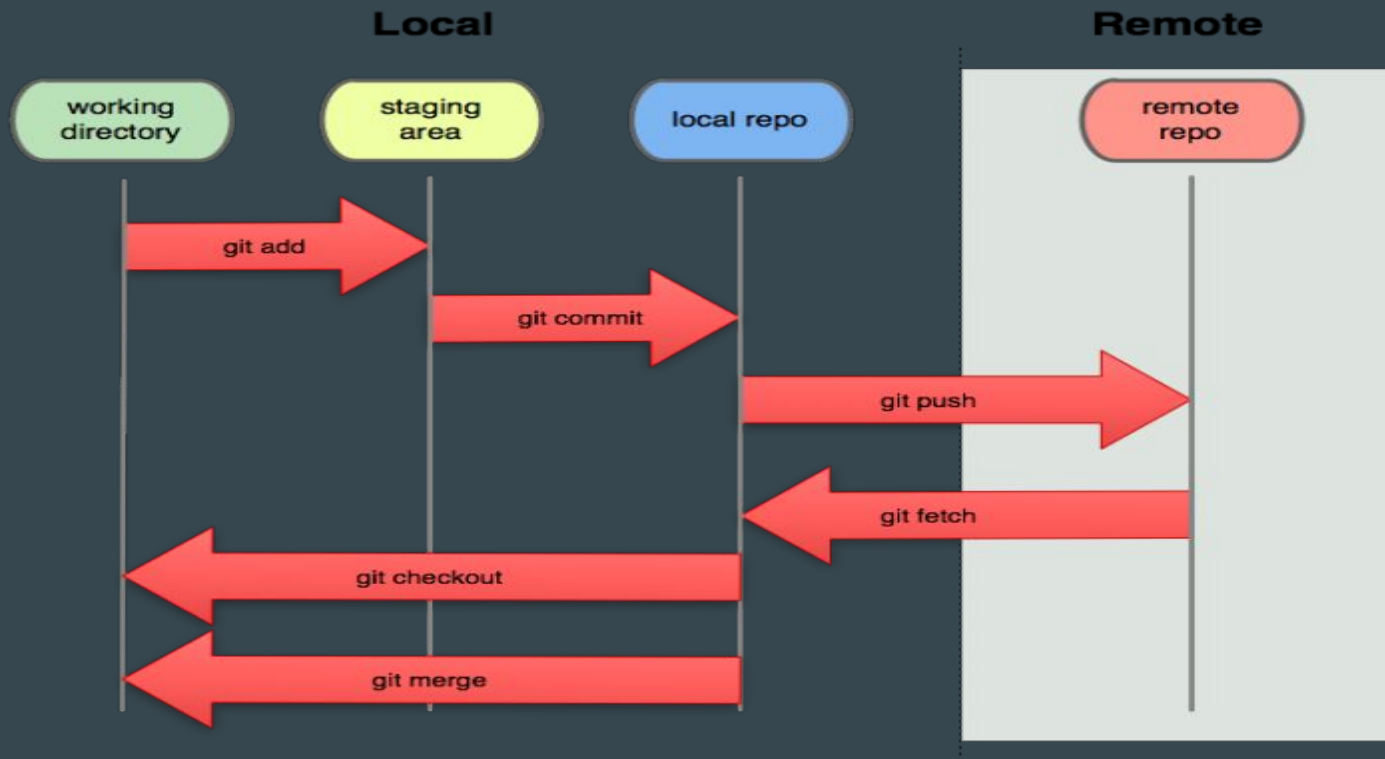
- `echo * Bash scripting is fun *`
- `echo "Bash scripting is "*fun*"`
- **Arguments containing spaces/special characters can be written in quotes**
 - `echo "* Bash scripting is fun *" -> Bash scripting is fun`
- **They can also be written in single quotes**
 - `echo 'Bash scripting is "*fun*"' -> Bash scripting is "fun"`

Strings

- Special characters can also be escaped with backslash
 - `echo "Bash scripting is \"fun\""` -> Bash scripting is "fun"
- In single quotes, escape characters are ignored.
 - `echo 'Bash scripting is \"fun\"'` -> Bash scripting is \"fun\"

Time for ForceLab

Remotes



Github only concepts

- Fork
 - A copy of a repository in another user's space or organization.
 - The connection between those is done by github.
- Issues
 - List of bugs, feature requests, improvements, etc, in a project.
 - Good place to start when contributing to open source projects!
 - Issues can be closed automatically if your commit message includes the text `Fixes #n` (or any of: `close`, `closes`, `closed`, `fix`, `fixes`, `fixed`, `resolve`, `resolves`, `resolved`)
- Pull requests
 - Asking owners of the forked repository to use commits from your repository

It's Collab time! (Attention, this lab has two parts!)