

bash shell

Gloria W
aagg@comcast.net

Bash is awesome

- Take control of your machine
- Be able to work on any other *NIX system anywhere in the world
- Know how to circumvent GUI issues re: system configuration
- Do quick iterative things without using a formal programming language.

SysV vs. BSD: the war left some scars

- “service” (/etc/init) vs. “init.d” (/etc/init.d)
 - Example:
 - `sudo service postgresql start`
 - `sudo /etc/init.d/postgresql start`
- Poor Mac users:
 - “launchd”

Let's start with permissions

```
api : bash - Konsole
File Edit View Bookmarks Settings Help
[gloriajw@g-monster api]$ vi generic_query_chg.py
[gloriajw@g-monster api]$ pwd
/home/loriajw/LightCloud_web/lightcloud/api
[gloriajw@g-monster api]$ ls -l
total 108
-rwxrwxr-x. 1 gloriajw gloriajw 31586 Jun 23 12:25 api_test.py
-rwxrwxr-x. 1 gloriajw gloriajw 3966 Jun 23 08:05 forms.py
-rw-rw-r--. 1 gloriajw gloriajw 39840 Jun 25 10:19 generic_query_chg.py
-rw-rw-r--. 1 gloriajw gloriajw 0 May 19 15:39 hardware_api_calls-000.
-rwxrwxr-x. 1 gloriajw gloriajw 0 May 19 15:27 __init__.py
-rw-rw-r--. 1 gloriajw gloriajw 143 May 28 14:28 __init__.pyc
-rw-rw-r--. 1 gloriajw gloriajw 2904 Jun 23 19:15 lightcloud_log-000.log
-rw-rw-r--. 1 gloriajw gloriajw 57 May 19 15:27 models.py
drwxrwxr-x. 2 gloriajw gloriajw 4096 Jun 25 10:19 __pycache__
-rwxrwxr-x. 1 gloriajw gloriajw 3251 Jun 12 15:15 urls.py
-rwxrwxr-x. 1 gloriajw gloriajw 11368 Jun 23 08:02 views.py
```

Permissions

- `-rwxrwxr-x 1 gloriajw dev 101`
`Mar 6 11:58 test_uwsgi.sh`
 - Why is this known as “755”?
- `drw-r--r-- 23 root root 4096 Apr`
`3 13:41 uwsgi-1.9.4`
 - Why is this known as “644”
- `chmod 644 filename`
- `chmod g+w filename`

Permissions Bitmasks

- - rwx r-x r-- ('-' = off)
 - - or “d” (directory)
 - User (owner) permissions bitmask ('-' or 'r', '-' or 'w', '-' or 'x') (r-x = binary 101 = 5)
 - Same with group
 - Same with other
- What is the binary and decimal equivalent of the above example?

Permissions Bitmasks

- How about these?
 - `-r-Xr--r--`
 - `-r--r----X`
 - `-rwxr--r-x`
- Not all of these combos are practical:
 - User usually needs 'rwx' on directories to be able to cd into them.
 - Why give others execute permissions but not group?
- See both linear and loop variants of this script.

Quick note about umask

- umask is the opposite bitmask from what you want file permissions to be:
 - `umask 0022 :755`
 - `umask 0002: 775`
- Notice the difference between files and directories! Files never get execute perms by default. Dirs do, because you cannot “cd” into a dir if you do not have execute perms.
- Also note: 'ls -l' shows directory contents, 'ls -ld' shows directory permissions.

User Input and Functions

- 'read' command waits for stdin (usually typed input, but it can be redirected from elsewhere as well)
- Our read script shows two examples of functions, one messes with letters, the other numbers:
 - \$1, \$2, etc. Are script parameters. “shift” allows us to keep referencing \$1 in our script.
 - “echo” inside of a function returns a value to the function caller, instead of echoing to stdout.
 - 'seq' and 'rev' commands are not common to all bash shells. Let me know if your shell does not support these commands.

I/O, STD(IN/OUT/ERR)

- Common to all programming languages
- File descriptors or 'handles', open by default, which allow input, output and error output.
- `STDIN = 0`, `STDOUT = 1`, `STDERR = 2`
-