

## -> ## Timing a bash function <-

Time is little used function.

### !/bin/bash

```
time ( sleep 1; sleep 1; sleep 1 )
```

```
real 0m3.036s user 0m0.006s sys 0m0.000s
```

---

## -> ## Get Bash to Give you a Clue

% bash -x will echo each line of the bash script as it is run

e.g. have a file a.sh with the following contents

### !/bin/bash

```
echo "a"
```

#### time (sleep 3)

## -> ## Bash -x

% bash a.sh

output as follows

```
hi
```

```
real 0m3.003s user 0m0.003s sys 0m0.000s
```

---

## -> ## Bash -x now with -x flag

% bash -x a.sh

output as follows

- echo hi hi
- sleep 3

```
real 0m3.009s user 0m0.004s sys 0m0.000s
```

---

## -> ## Bash -X, what it means

You will be notified as each line runs. This can be very useful when you need it.

---

## -> ## Shellcheck, Lint for Bash

Shellcheck is a program that will go through your shell script and let you know about any "iffy" code.

E.g.

SC1040 When using <<, you can only indent with tabs SC1081 Scripts are case sensitive. Use 'if' not 'If' SC1086 Don't use \$ on the iterator name in for loops

These can really help you out when you've got a subtle bug.

Can also be a major pain in the butt if it is part of your build process

---

## **-> ## Use this not that <-**

1. tldr instead of man
  2. ip instead of ifconfig
  3. fd instead of find
- 

## **-> ## tldr instead of man <-**

tldr is the modern replacement for man

Each piece of documentation is pretty short

Make sure to do do "tldr -u" to download the latest version of the pages

E.g. tldr vim; tldr wget

---

## **-> ## ip instead of ifconfig <-**

ip is the replacement for ifconfig. net-tools is not installed in a lot of distros including docker instances.

**ip -4 a - show ipv4 address**

**ip -6 a - show ipv6 address**

**ip r - show routes on system**

**ip a show enp0s3 - show stats for enp0s3 interface**

**ip link ls up - show active interfaces**

---

## **-> ## ip instead of ifconfig <-**

Modifying an Interface

**ip a add 192.168.0.243/255.255.255.0 dev enp0s3**

you can also add a CIDR at the end as well

**ip a add 192.168.0.243/24 dev enp0s3**

Deleting an Interface

**ip a del 172.16.0.1/24 dev eth0**

Flush address from interfaces

**ip -s -s a f to 172.16.0.1/24**

---

## **-> ## ip instead of ifconfig <-**

Bring an interface up and down

**ip link set eth0 down**

**ip link set eth0 up**

Add a default route

**ip route add default via 192.168.0.254**

Add a route

**ip route add 192.168.0.0/255.255.255.0 dev eth0**

---

**-> ## ss instead of netstat <-**

ss is the replacement for netstat.

from the man page.

**ss -t -a # display all tcp sockets**

**ss -u -a # display all udp sockets**

**ss -t -a -Z # show sockets with SELinux security contexts**

**ss -u -a -Z # display all udp sockets with SELinux contents**

show all established SS connections

**ss -o state established '( dport = :ssh or sport = :ssh )'**

---

**-> ## fd instead of find <-**

fd is a replacement for find, with an easier syntax

find messages file in /var/log is

```
% fd messages /var/log
```

as opposed to

```
% find /var/log -name '*messages*'
```

---

**-> ## fd instead of find <-**

How to Vi all the files that match a pattern in the current directory

```
% fd md -X vi
```

Find equivalent

```
% find -name '*md*' -exec vi {} \;
```

How to do an ls -l for ALL the files in a folder

```
% fd -H -l
```

find equivalent

```
% find -exec ls -l {} \;
```

Note: you have to explicitly tell it to show hidden files with fd

---

## -> ## fd instead of find <-

Also has options to either do globbing or regular expressions

Caveat:

Isn't included with a lot of distros so violates the reason for using ip instead of ifconfig

---

## -> ## grep -r instead of find/xargs/grep <-

In the average day I will type a variation of this command upwards of 10 times

```
% find . -type f -print0 | xargs -0 grep -i value
```

looks for a value recursively with grep

```
% grep -r value
```

---

## -> ## Fun things <-

1. <https://cheat.sh/>
  2. bash insulter
- 

## -> ## Future things <-

GNU Coreutils being rewritten in Rust

<https://github.com/uutils/coreutils>

Why is it the Future?

Moving away from classic languages like C. Interesting to see how long it will take to reach feature parity with the classic Coreutils.

---

## -> ## Github BASH Awesome <-

Google: bash awesome github - and you'll get links to a lot of good bash resources

E.g. Bash Hacker's wiki is pretty good - <https://wiki.bash-hackers.org/doku.php>

---

## -> ## 6 Tips <-

1. -x is your friend
  2. Be fearless, unless you've got an rm in your script
  3. You will screw up, but you'll learn a lot in the process
  4. The command line is a very efficient way to do things
  5. Powershell runs on Linux
  6. Have fun
- 

## -> ## Last words <-

[GitHub Repo for this talk.](#)