Contents

1 Bash best practices

An attempt to bring order in good advice on writing Bash scripts I collected from several sources.

1.1 General

- Always check for syntax errors and use ShellCheck. Integrate this in your text editor (e.g. Syntastic plugin in Vim)
- The principles of Clean Code apply to Bash as well

1.2 Variables

- Prefer local variables within functions over global variables
- If you need global variables, make them readonly
- Variable should always be referred to in the `${var}` form (as opposed to `$var`.
- Variables should always be quoted: ”`${var}`”
- Capitalization:
  - Readonly/environment variables: `${ALL_CAPS}`
  - Local/mutable variables: `${lower_case}`
- Declare variables with a meaningful name for positional parameters of functions

```
foo() {
  local first_arg=”${1}”
  local second_arg=”${2}”
  [...]
}
```

- Positional parameters of the script should be checked, those of functions should not

1.3 Substitution

- Always use `$(cmd)` for command substitution (as opposed to backquotes)

1.4 Functions

Bash can be hard to read and interpret. Using functions can greatly improve readability. Principles from Clean Code apply here.

- Apply the Single Responsibility Principle: a function does one thing.

- Create functions with a meaningful name for complex tests

```
# Don’t do this
if [ ”$#” -ge ”1” ] && [ ”$1” = ’-h’ -o ”$1” = ’--help’ -o ”$1” = ”-?” ]; then
  usage
  exit 0
fi

# Do this
help_wanted() {
  [ ”$#” -ge ”1” ] && [ ”$1” = ’-h’ -o ”$1” = ’--help’ -o ”$1” = ”-?” ]
}

if help_wanted ”$@”; then
  usage
  exit 0
fi
```

- Don’t mix levels of abstraction

- Describe the usage of each function: number of arguments, return value, output

1.5 Shell script template

An annotated template for Bash shell scripts:

1.6 Resources

- Bash Reference Manual, https://www.gnu.org/software/bash/manual/bashref.html
- bash(1) man page
- The Advanced Bash Scripting Guide (ABS) http://www.tldp.org/LDP/abs/html/
- Defensive Bash Programming, http://www.kfirlavi.com/blog/2012/11/14/defensive-bash-programming