Shell Scripting for Computer Programmers

Andrew Thorp

Linux@App

What is shell scripting?

- BASH pipes

Why use shell scripting

- Learning the language
- Not having to learn the language
- Avoiding large overhead/libraries
- Very easy to write

Documentation / References

Documentation can be found my typing:

```
$man <program_name>
```

Review

Pipes

- `cmd1 | cmd2` takes output of `cmd1` and feeds it as input to `cmd2`
- `cmd1 >> file` takes the output of `cmd1` and appends `file` with the contents
- `cmd2 > file` takes the output of `cmd1` and overwrites `file` with the contents
- `cmd3 < file` takes the contents of `file` and extracts them as input to `cmd3`

Basic program

Hello World

`HelloMe.sh`

Basic program

Hello Me

```
#!/bin/bash
# This is a comment!
echo "Hello $USER!" # This is also a comment
```

Anatomy

```
#!/bin/bash
```

The “Shabang”: path to binary

Anatomy

```
echo "Hello $USER!"
```

The echo command

Anatomy (aside)

```
echo "Hello $USER!"

    vs

echo 'Hello $USER!'
```

Anatomy

```
# This is a comment
```

Golly I wonder what this is

Usage?

Running the script

```
$ ./helloMe.sh
```

Usage

Running the script:

run `$chmod +x ./<your_script>`

run `$./helloMe.sh`

Fundamentals: Variables

Variables

```
# Variable definition
MESSAGE="Hello there.\n"
NAME="Genreal Kenobi..."
#Variable usage
echo "$MESSAGE $NAME"
```

Fundamentals: Arguments

Arguments

```
# Called with argScript.sh <firstName> <lastName>
FNAME=$1
LNAME;$2
echo "Hello $1 $2. You called this script with $# arguments
```

Fundamentals: Loops

FOR loop

```
for i in $( ls ); do
    echo item: $i
done
```

Fundamentals: Loops

FOR loop

```
for i in *; do
    echo item: $i
done
```

Fundamentals: Loops

FOR loop

```
for i in `seq 1 10; do
    echo item: $i
done
```

Fundamentals: Loops

WHILE loop

```
echo "Press Ctrl-C to escape"
while read f; do
    echo $f
done
```

Fundamentals: Conditionals

Testing

```
if [ true ]; then
    echo true
elif [ true ]
    echo "This should not be reached"
else
    echo "If you're here something is very wrong"
fi
```

Notice the spaces around the brackets

Fundamentals: Tools

- `grep` - find
- `awk` - just about anything, mainly dealing with word manipulation / pattern matching
- `sed` - manipulate streams of text, find and replace, etc
- `echo` - print a value
- `cat` - print the contents of a file