

Scripting

Advanced Embedded Linux Development

with **Dan Walkes**



University of Colorado **Boulder**

Learning objectives:

Scripts Overview

Script Variables

Script Arguments

Script Conditionals

Scripts

- Scripting allows us to:
 - Automate frequently used commands
 - Customize commands for different scenarios using arguments and variables
- Think of scripts as documentation!
 - If you find yourself listing commands used to perform a task, consider scripting instead.
- Often are in files ending with `.sh` (short for shell)

Simple Script Example

- `#!/bin/sh` is a comment which tells the shell which shell interpreter to use
 - Also known as “shebang”
- Also common to see `#!/bin/bash`

```
dan@DESKTOP-BQMVP69:~/CU/aesd-lectures/lecture2$ cat helloworld.sh
#!/bin/bash
greeting=hello
echo "${greeting} world!"
dan@DESKTOP-BQMVP69:~/CU/aesd-lectures/lecture2$ ./helloworld.sh
hello world!
```

Script Variables

- Script Variables include:
 - Variables we define ourselves

```
dan@DESKTOP-BQMVP69:~/CU/aesd-lectures/lecture2$ cat helloworld.sh
#!/bin/bash
greeting=hello
echo "${greeting} world!"
dan@DESKTOP-BQMVP69:~/CU/aesd-lectures/lecture2$ ./helloworld.sh
hello world!
```

- Variables passed into the script as arguments

```
dan@DESKTOP-BQMVP69:~/CU/aesd-lectures/lecture2$ cat helloworldargs.sh
#!/bin/bash
greeting=$1
echo "${greeting} world!"
dan@DESKTOP-BQMVP69:~/CU/aesd-lectures/lecture2$ ./helloworldargs.sh hola
hola world!
```

Script Argument

- Script special variables
 - \$0 - the invoked name of the bash script
 - \$1 - \$9 - first 9 arguments to a script
 - \$# - Number of arguments passed to the script
 - \$? - The exit status of the most recent process

Bash Conditional if/else

- Conditional syntax for bash
- [] is shorthand for the “test” command

```
if [ conditional ]  
then  
    action  
else  
    action  
fi
```

```
if [ -d "$WRITEDIR" ]  
then  
    echo "$WRITEDIR created"  
else  
    exit 1  
fi
```

-d FILE

FILE exists and is a directory

Manual page test(1) line 48/165 45% (press h for help or q to quit)

Script Failure Exit Codes

- Scripts can communicate success/failure through exit status and the exit command

```
if [ $? -eq 0 ]; then
    echo "success"
    exit 0
else
    echo "failed: expected ${MATCHSTR} in ${OUTPUTSTRING} but instead found"
    exit 1
fi
```

```
dan@DESKTOP-BQMVP69:~/CU/assignments-complete-private$ ./finder.sh
ERROR: Invalid Number of Arguments.
Total number of arguments should be 2.
The order of the arguments should be:
  1)File Directory Path.
  2)String to be searched in the specified directory path.
```