Bash Scripting Examples

Prof. Brendan Kochunas

September 2021

1 Hello World

```
#!/bin/bash

echo "Hello World"
```

Listing 1: Bash Hello World Example

Bash scripts can be run using one of two methods. First is to invoke bash with the script name as the argument:

```
$ bash script.sh
```

Listing 2: Running bash scripts

second is to change permissions of the script to be executable and then run the script directly:

```
$ chmod u+x script.sh
$ ./script.sh
```

Listing 3: Running bash scripts

2 Bash Variables

```
#!/bin/bash
echo $USER
echo '$USER'   #The type of quote matters!
echo "$USER"
echo "\$USER"  #escape with "\" for special symbols

#variable names are case sensitive
class=NERS570
Class= "NERS-570"
export CLASS=NERS570_F21
foo=0
foobar=3

echo "$class, $Class and $CLASS"
echo "Today's lecture is $foobar"
echo "Today's lecture is ${foo}bar"
```

Listing 4: Bash example of variables

3 Bash Arguments and variable parsing

```
#!/bin/bash

echo "Arg 0 is $0"
echo "Arg 1 is $1"
echo "Arg 2 is $2"
echo "whole arg list $@"

echo "The number of arguments $#"
echo "The script name is $0"
echo "But we can trim file extensions ${0%.*}"
echo "We can also start at specific characters ${0:2}"
echo "Or select substrings: ${0:2:3}"
```

Listing 5: Bash Example Illustrating argument processing

You can execute this script as:

```
$ ./script foo bar
```

4 Bash Constructs

```bash
#!/bin/bash

n=$1

# n should be a positive integer
if [ -z "$n" ]; then
  echo "n is not defined"
  exit 1
else
  if [[ $n =~ ^-?[0-9]+$ ]]; then
    if [ $n -le 0 ]; then
      echo "n is less than 1! \"$n\""
      exit 3
    fi
  else
    echo "n is not an integer! \"$n\""
    exit 2
  fi
fi

echo ""
echo "Loop to $n"

for i in $(seq 0 $n); do
    echo "i=$i"
done
```

Listing 6: Bash Scripting to loop from 0 to $n > 0$