Linux/Unix System Programming

CSCI 2153

David L. Sylvester, Sr., Professor

BASH Programming

Bash scripts can be used for various purposes, such as executing a shell command, running multiple commands together, customizing administrative tasks, performing task automation etc. So knowledge of bash programming basics is important for every Linux user.

Variables have no scope, they are all global, there is no standard library, you don’t have a module system, for instance. But the advantages are pretty great: you can very easily invoke any CLI (command line) tool just like you were in the shell, and the Unix approach of having many little utility commands really makes shell scripting shine.

BASH Programming

Scripts are stored in files. You can give any name and extension to a shell script, it does not matter. The important thing is that it must start with a “shebang” on the first line:

#!/bin/bash

The file must be an executable file.
The file is set as executable using the chmod command.

This can be done by using the following command:

$ chmod u+x filename

BASH Programming

COMMENTS:
Comments are one of the most important things when writing programs. A line starting with the # symbol is a comment (with the exception of the shebang line.

```
#!/bin/bash
# This is a comment
```

A comment can also start at the end of a line.

```
#!/bin/bash
echo “This is a test” # This is a comment
```

BASH Programming

Variables:

You can set (assign) variables using the = operator.

name=value

Examples:

year=2020
name=Rouge
nickname="Tuff"

You can print a variable by using the **echo** built-in command and prepending a **$** to the variable name.

echo $year
echo $nickname

BASH Programming

Operators:

Arithmetic Operators		Comparison Operators		
+	add	<	less than	-lt
-	subtract	<=	lower or equal than	-le
*	multiply	==	equal to	-eq
/	divide	>=	greater or equal than	-ge
%	modulo	>	greater than	-gt
**	exponentiation			

BASH Programming

Operators:

```
#!/bin/bash

# Add two numeric value
((sum=25+35))

#Print the result
echo $sum
```

This code adds two number and stores the value into a variable named sum. Then prints sum.

```
#!/bin/bash
: '
The following script calculates
the square value of the number, 5.
'
((area=5*5))
echo $area
```

This code calculates the area of a square and stores the value into a variable named area. Then prints area.

BASH Programming

Operators:

Logical Operators

&&	logical AND
||	logical OR

Shortcut Arithmetic Operators

+=

-=

***=**

/=

%=

BASH Programming

Print to the Screen:

You can print anything to the screen using the **echo** command.

```
#!/bin/bash
echo “test”
echo test
echo testing something
```

BASH Programming

Control Structures (if/else statements

if

```
if condition
then
   command
fi
```

If then else

```
if condition
then
   command
else
   anothercommand
fi
```

Nested if then else

```
if condition
then
   command
elif
   anothercommand
else
   yetanothercommand
fi
```

BASH Programming

Control Structures (if/else statements

```
#!/bin/bash
DOGNAME=Roger
if [ "$DOGNAME" == "" ]; then
  echo "Not a valid dog name!"
else
  echo "Your dog name is $DOGNAME"
fi
```

```
#!/bin/bash
n=10
if [ $n -lt 10 ];
then
echo "It is a one digit number"
else
echo "It is a two digit number"
fi
```