

Lab 01 - Bash tutorial

Harish Krupo

August 12, 2016

The following tutorial is by no means a complete tutorial for bash. It can help you start working with bash. You are required to read more about each of the commands explained here in order to solve the questions given at the end.

1 Introduction

Bash is one of the most important and widely used programs in linux. It is an example of a shell. A shell provides an interface through which users can communicate with the kernel (you will study more about it in the coming classes). The work of a shell is simple, execute programs and display output of those programs.

2 Lets begin with the Sha Bang!

Every scripting language requires an appropriate interpreter, for example, python scripts are run by the python interpreter. A shell needs to know which interpreter to be used. This is where sha bang comes in. It is the first line of the document which specifies which interpreter to use.

```
#!/bin/bash #this forces the shell to execute the file using bash.
```

3 Hello world with bash

Echo command is used to print output to the screen just like printf in c.

```
echo "Hello World!"
```

4 Variables

```
#!/bin/bash
STRING="HELLO WORLD!!!" #there should be no space before and after '='
echo $STRING # the value of a variable is fetched using $
```

5 Reading User input

```
#!/bin/bash

echo -e "Hi, please type the word: \c "
read word
echo "The word you entered is: $word"
echo -e "Can you please enter two words? "
read word1 word2
echo "Here is your input: \"$word1\" \"$word2\""
echo -e "How do you feel about bash scripting? "
# read command now stores a reply into the default build-in variable $REPLY
read
echo "You said $REPLY, I'm glad to hear that! "
echo -e "What are your favorite colours ? "
# -a makes read command to read into an array
read -a colours
echo "My favorite colours are also ${colours[0]}, ${colours[1]} and ${colours[2]}:-)"
```

6 Arrays

```
#!/bin/bash
#Declare array with 4 elements
ARRAY=( 'Debian Linux' 'Redhat Linux' Ubuntu Linux )
# get number of elements in the array
ELEMENTS=${#ARRAY[@]}
```

7 IF/ELSE

```
#!/bin/bash
directory="./BashScripting"

# bash check if directory exists
```

```

if [ -d $directory ]; then
    echo "Directory exists"
else
    echo "Directory does not exists"
fi

```

8 Bash comparisons used in if conditions

Bash	C operator
-lt	<
-gt	>
-le	<=
-ge	>=
-eq	'=='
-ne	!='

9 Loops

```

#!/bin/bash

# bash for loop
for f in $( ls /var/ ); do
    echo $f
done

```

10 While loop

```

#!/bin/bash
COUNT=6
# bash while loop
while [ $COUNT -gt 0 ]; do
    echo Value of count is: $COUNT
    let COUNT=COUNT-1
done

```

11 File system management with bash

Everything in linux is a file. Even your hard drive, printer, mouse and keyboard, all are files. Bash can be used to navigate through the file system,

create files, write to files and delete them.

11.1 Create file in bash

```
touch filename
```

11.2 Delete file in bash

```
rm filename
```

11.3 Create directory in bash

```
mkdir filename
```

11.4 Delete directory in bash

```
rmdir filename # works only if the directory is empty
```

11.5 Copy files from one place to another

```
cp source destination
```

11.6 Move files from one place to another

```
mv source destination
```

11.7 Write to files using cat command

```
cat file << EOF  
This is a simple text file.  
EOF
```

11.8 Read to files using cat command

```
cat file
```

12 Finding help in bash

The bash man command is very useful if one wants to find more information about a command. To use simple type

```
man <command>
```

Learn more about the following commands using the man utility. Almost all manpages provide examples of how to use a command. try reading the manpages of the commands given in the next section.

13 Various other commands one can use in bash :

- cat
- grep
- which
- whoami

14 Commands used to control processes:

- ps : list all attributes about the current processes
- top : visual version of ps
- kill : kill a program
- exec : execute a program given as an argument.

15 Programming questions:

15.1 Create the following directory structure under your home directory:

Documents

- development
 - cs101
 - * lab1b
 - * test1
- essays
 - engl101

Which means, that you would create Documents subdirectory, and in it development and essays subdirectory, etc. Once completed, go to your home directory and type: `ls -R Documents`

15.2 Modify the directory structure to look like this:

Documents

- cs101
 - lab1b
 - test1
- engl101

Once completed, go to your home directory and type: `ls -R Documents`

- 15.3 Move the whole content of the test1 directory to engl101 directory. Show that test1 directory is empty and then show that the engl101 has its files.
- 15.4 Delete all the subdirectories of the Documents directory. Show that Documents directory has no files and subdirectories in it.
- 15.5 Delete the Documents directory from your home directory. Show that the Documents directory is deleted.
- 15.6 Write a bash script to find the factorial of a given number. Do not use the existing command
- 15.7 Open firefox in you computer, use ps to find out the process id of firefox and use kill command to kill it.
- 15.8 Using grep find the word cheerful in the file random.txt
- 15.9 Count the number of occurances of “the” in the file random.txt [Hint: read the manpage of grep]
- 15.10 Implement bubble sort in bash.