**Overview**

* A shell script is a script written for the shell, or command line interpreter, of an operating system.
* The shell is often considered a simple domain-specific programming language.
* Typical operations performed by shell scripts include file manipulation, program execution, and printing text.
* Bash Shell Script is an interpreted language. This means that the shell analyzes each statement in the language one line at a time, then executes it.
* This differs from languages such as C, in which programs are compiled into executable files.
* Interpreted languages are generally easier to debug and modify; however, they usually take much longer to execute than compiled programs.

**Special Shell Variables**

* $0 The name of the program is assigned here.
* $1 - $9 The arguments typed on the command line are assigned here.
* ${10} Any argument after $9 must be accessed using curly braces.
* $# Number of arguments passed to the program or number of parameters set by executing the set statement.
* $\* Collectively references all positional parameters as $1, $2, …
* $@ Same as $\* except when double quoted; collectively references all positional parameterrs as “$1”, “$2”, …
* $? Exit status of the last command not executed in the background.
* $! The process ID number of the last program sent to the background for execution.
* $$ The process ID number of the program being executed.

**Variables**

* Just about every programming language in existence has the concept of *variables* - a symbolic name for a chunk of memory to which we can assign values, read and manipulate its contents.

VAR=value

Echo $VAR

* Export Variable

**Wildcards:**

cp /tmp/a/\* /tmp/b/  
$ cp /tmp/a/\*.txt /tmp/b/  
$ cp /tmp/a/\*.html /tmp/b/

**Examples**

1. Test.sh

#!/bin/bash

echo "Number of arguments passed are $#"

Execute the below command after you created Test.sh to have executable permissions to the file

chmod 755 Test.sh

We need to execute above command for each Shell Script that you creates

1. InstallHttpd.sh

#!/bin/bash

sudo yum install httpd -y

sudo service httpd start

**awk (Aho, Weinberger, and Kernighan)**

Example 1:

1) Amit Physics 80

2) Rahul Maths 90

3) Shyam Biology 87

4) Kedar English 85

5) Hari History 89

[jerry]$ awk '{print $3 "\t" $4}' marks.txt

Example 2:

employee.txt

ajay manager account 45000

sunil clerk account 25000

varun manager sales 50000

amit manager account 47000

tarun peon sales 15000

deepak clerk sales 23000

sunil peon sales 13000

satvik director purchase 80000

**Default behavior of Awk :**By default Awk prints every line of data from the specified file.

$ awk '{print}' employee.txt

**Print the lines which matches with the given pattern.**

$ awk '/manager/ {print}' employee.txt

**Splitting a Line Into Fields :**For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively. Also, $0 represents the whole line.

$ awk '{print $1,$4}' employee.txt

**SED (Stream Editor)**

**file.txt**

unix is great os. unix is opensource. unix is free os.

learn operating system.

unix linux which one you choose.

unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

**Replacing or substituting string :** Sed command is mostly used to replace the text in a file. The below simple sed command replaces the word “unix” with “linux” in the file.

**$sed 's/unix/linux/' file.txt**

Here the “s” specifies the substitution operation. The “/” are delimiters. The “unix” is the search pattern and the “linux” is the replacement string.

By default, the sed command replaces the first occurrence of the pattern in each line and it won’t replace the second, third…occurrence in the line.

**Replacing the nth occurrence of a pattern in a line :**Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word “unix” with “linux” in a line.

**$sed 's/unix/linux/2' file.txt**

**Replacing all the occurrence of the pattern in a line :**The substitute flag /g (global replacement) specifies the sed command to replace all the occurrences of the string in the line.

**$sed 's/unix/linux/g'file.txt**

**Replacing from nth occurrence to all occurrences in a line :**Use the combination of /1, /2 etc and /g to replace all the patterns from the nth occurrence of a pattern in a line. The following sed command replaces the third, fourth, fifth… “unix” word with “linux” word in a line.

**$sed 's/unix/linux/3g' geekfile.txt**

1. **Parenthesize first character of each word :**This sed example prints the first character of every word in parenthesis.
2. **$ echo "Welcome To The Geek Stuff" | sed 's/\(\b[A-Z]\)/\(\1\)/g'**

Output:

(W)elcome (T)o (T)he (G)eek (S)tuff

1. **Replacing string on a specific line number :**You can restrict the sed command to replace the string on a specific line number. An example is

**$sed '3 s/unix/linux/' file.txt**

**Examples of General Shell Scripting**

* Create a script add.sh and pass arguments like ‘./add.sh 1 2 3 4’

#!/bin/bash

result=0

for index in $@

do

result=$((result+index))

done

echo "Result is $result"

Functions

#!/bin/bash

show\_contents()

{

filename\_local=$1

echo "function argument is $filename\_local"

echo "function arguments are $@"

}

#shell script arguments

otherargs=$@

echo "Shell arguments are $otherargs"

for filename in ["/home/ubuntu/bin/test.sh"]

do

show\_contents $filename

done

* Write a shell script which adds first two arguments and multiplies third and fourth argument

Sample Testcases

./math.sh 1 2 3 4

---------Result----------

addition result of (1+2) = 3

multiplication result of (3\*4) = 12

./math.sh 1 3

--------Result----------

number of arguments should be 4

Solution:

#!/bin/bash

add()

{

number1=$1

number2=$2

result=$((number1+number2))

echo "addition result of $number1+$number2 is $result"

}

mul()

{

number1=$1

number2=$2

result=$((number1\*number2))

echo "multiplication result of $number1\*$number2 is $result"

}

if [ $# -eq 4 ]

then

add $1 $2

mul $3 $4

exit 0

else

echo "invalid arguments passed"

exit 1

fi