

Forum:[Hack Forums](#)

Thread: [A parameter controlled calculator | Example \[BASH\]](#)

Mefie - November 5, 2015, 4:42 am

Hello!

I have written this source code in Bash. I tested it on Xubuntu 14.04 LTS within VMWare. Execution of the script asked for no additional software to be installed there. Also, no need for administrative permissions if you use default settings. As I have written in the title, I am showing to you a simple calculator, which is able to do basic mathematical operations. The educational purpose of this script is demonstrating some mathematical and parameter related operations in Bash. It is able to handle more than two input numbers.

At '-' and '/' operations, the first number following values will be '-' or '/' from the first one. So, \. argcalc.sh - 4 2 2 will have the result of 0.

You can call the script from Terminal like the following

General form

Code \. <scriptname> <operation> <value1> <value2> ... <valueN>

An example

Code \. paramcalc.sh + 5 6 7

The first .sh file contains the following code

<scriptname> (Example paramcalc.sh)

Code #!\bin\bash

```
op="$1"
params=""
remfst="0"
```

```
for i in $@
do
  if [ $remfst -eq 1 ]; then
    params="$params$i "
  fi
  remfst=1
done
```

```
case $op in
"+") pm="\. plusminus.sh $params";;
"-") pm="\. plusminus.sh $params";;
"\*") pm="\. multivide.sh $params";;
"/") pm="\. multivide.sh $params";;
*) return 1;;
esac
```

```
echo "Your result is `eval $pm`"
```

You can see within the code that a second script code is called from the first one. It is OPTIONAL so making one script file from them should be easy. For me, parsing them into more files resulted in better understanding. If you leave them like in this way, you will need two additional .sh files as well with the same names as they are in the code (or change the code to your files' names)

One that handles '+' and '-'

```
plusminus.sh
Code #!\bin\bash
if [ $op == "-" ];then
    result=`expr $1 + $1`
else
    result="0"
fi

for on in $@
do
    result=`expr $result $op $on`
done
```

echo \$result

In addition, one that handles '*' and '/'

```
multivide.sh
Code #!\bin\bash
if [ $op == "/" ];then
    code="expr $1 \* $1"
    result=`eval $code`
else
    result="1"
fi

for on in $@
do
    code="expr $result $op $on"
    result=`eval $code`
done
```

echo \$result

It is important, when you use '*', give it in parameter as "*" WITH "". The other operations should work simply as them like /, + and -, no need for ""'s either.

Least but not last, let me add, this source code was the result of my [attempt](#) in order to create a calculator in Bash from arguments that would be able to handle more than two input values. Finally, it works, but if I can suggest a more professional solution, feel free to check the linked thread and read [Evolution](#)'s feedback in order to see how you can do it. When I was writing this script, I had no internet connection so could not rely on Evolution's code, but his idea worths a check for sure.

Now, enjoy the example!