Bash Shell Scripting

Shell Arithmetic

Persistent University

Contents

Persistent

Ways to do maths in shell

- Arithmetic Operators
  - User can do maths in the following ways in shell
    - **Using expr command**
    - **Using $(()) construct**
    - **Using bc command**
    - **Using $[] construct**

Arithmetic expansion using expr

- expr – Evaluates arguments according to operation
  - Syntax :- expr integer1 operator integer2
  - expr command performs arithmetic operations on integers
- Arithmetic operators

- **Addition**	: expr 2 + 3	..returns 5
- **Subtraction**	: expr 4 - 1	..returns 3
- **Multiplication**	: expr 3 * 3	..returns 9
- **Division**	: expr 6 / 3	..returns 2
- **Modulus**	: expr 5 % 3	..returns 2

Be aware that many operators need to be escaped or quoted to be interpreted correctly by shell.

Arithmetic expansion using expr

- Expr -
  - expr is often used with command substitution to assign a variable. For example, you can set a variable x to the sum of two numbers:

    -
      ```
      $ x=`expr $a + $b`
      $ echo $x
      15
      ```
- **Note:** As you can see, for expr, you must put spaces around the arguments: "expr 123+456" doesn’t work.
- "expr 123 + 456" works.

Arithmetic expansion using expr

- Logical operators

Equal	: expr 24 = 25	..returns 0
Unequal	: expr 20 != 30	..returns 1
Greater than	: expr 20 \> 25	..returns 0(false)
Less than	: expr 20 \< 25	.. returns 1(true)
Less than equal	: expr 20 \<= 20	..returns 1
Greater than equal	: expr 20 \>= 25	..returns 0
OR	: expr 20 \\| 25	..returns 20
AND	: expr 20 \& 25	..returns 20
	expr 20 \& 0	..returns 0

Arithmetic expansion using expr

- String operators
  - **length**
    - a=abcddcba
    - echo `expr length $a`
    - **Output -> 8**
  - **index**
    - b=`expr index $a dc`
    - **Output -> 3**
  - **substr**
    - b=`expr substr $a 3 4`
    - **Output -> cddc**

Arithmetic expansion with backticks

- Often used in conjunction with expr.
- Example:
  - x=`expr 20 + 10`
  - echo $x

  - y=20
  - x=`expr $y + 10`
  - echo $x

Complete expression should be enclosed between

Arithmetic expansion with parentheses

- Syntax :
  - $((expression))
- Example :
  - echo $((2 + 2))
  - z=$(($z+3))
  - z=$((z+3))
- Supports only integers:
- Operations without assignment :
  - n=0
  - ((n += 1)) or ((n++)) or $[n++] or $((n++))
  - echo $n

Arithmetic expansion with parentheses

- ((..)) can be used as conditional expression
- Examples:
  - ((uid == 0))
  - ((uid > 1000))
  - ((i=i+1))
  - echo $((i+=2 , j++))
  - echo $(($a<10?100:1000))

Rules of using expansion with parentheses

- Rules in detail:
  - $(()) and (()) can be used for integer arithmetic expressions
  - You may have may not have spaces around the operators, but you must **not** have spaces around the equal sign, as with any bash variable assignment

  -
    ```
    $ c=$(($a+9))
    $ echo $c
    19
    $ c=$((a+9))                    #Also correct, no need of $ sign.
    $ c=$((a + 9))                  #Also correct, no restriction on spaces.
    $ c= $((a + b))          #Incorrect, space after assignment operator
    ```
  - Note that You may also use operations within double parentheses without assignment.
    E.g ((a++)) echo $a

Arithmetic expansion with bc

- For arithmetic on real numbers
- Can be run interactively, or as a shell script command
- Example :

```
r=3.5
s=`echo $r + 2.2 | bc`
echo $s

if [ $( echo "$t > 4.5" | bc ) -eq 1 ]; then
        echo “Your rating is Excellent !!”
fi
```

Quiz

- Many operators need to be ________ to be interpreted correctly by shell.
- For expression expr 2 \& 0, return value will be ____.
- Index returns position of ______ character in substring that matches a character in string.
- Arithmetic expansion with parentheses only supports _______.
- To perform arithmetic on real numbers, use _____ command.

Quiz Answers

- Many operators need to be **escaped** to be interpreted correctly by shell.
- For expression expr 2 \& 0, return value will be **0.**
- Index returns position of **first** character in substring that matches a character in string.
- Arithmetic expansion with parentheses only supports **integers.**
- To perform arithmetic on real numbers use **bc** command.

Assignments

- Write a script to accept 2 numbers from user and perform addition, subtraction, multiplication, division and modulus operations.
- Write a script to accept full name from user and print its length.
- Write a script to give Fahrenheit equivalent of centigrade values passed from user.

Assignment Solution

- Assignment 1:
  - read -p "Enter number 1: " num1
  - read -p "Enter number 2: " num2
  - val=`expr $num1 + $num2`
  - echo "$num1 + $num2 = “ $val
  - val=`expr $num1 - $num2`
  - echo "$num1 - $num2 = “ $val
  - val=`expr $num1 \* $num2`
  - echo "$num1 * $num2 = “$val
  - val=`expr $num2 / $num1`
  - echo "$num2 / $num1 = “$val
  - val=`expr $num2 % $num1`
  - echo "$num2 % $num1 = “$val

Assignment Solution

- Assignment 2:
  - read -p "Enter fullname : " fullname
  - echo Length of full name : `expr length "$fullname"`
- Assignment 3:
  - read –p "Enter temperature (C) : “ tc
  - # formula Tf=(9/5)*Tc+32
  - tf=$(echo "scale=2;((9/5) * $tc) + 32" |bc)
  - echo "$tc C = $tf F"

Summary : Shell Arithmetic

- In this module, we have learnt about shell arithmetic.
- Now, you should be able to answer following questions:
  - How to use expression evaluator expr?
  - What are different operators to be used with expr?
  - How to perform arithmetic expansion with backticks and parentheses?
  - How to use bc command for real number arithmetic?

Reference Material

- http://tldp.org/LDP/abs/html/arithexp.html
- http://www.softpanorama.org/Scripting/Shellorama/arithmetic_expressions.shtml
- http://www.tutorialspoint.com/unix/unix-what-is-shell.htm

Key contacts

- **Persistent Interactive**

  Bhimashankar Gavkare

  bhimashankar_gavkare@persistent.com

Persistent

Persistent

Thank You !!!

Persistent University