

Operating System – Linux Programming

Sunbeam Infotech



BASH shell script → interpreter

- Shell script is collection of shell commands along with programming constructs.

- Shebang line

- #!/bin/bash

- echo "Hello World"

- -n : skip trailing newline
 - -e : enable esc seq. \n, \t, \r, ..

- Shell variables

- var=value → init var
 - echo "\$var"

- expr command → int arithmetic

- Command substitution

- var=`command` → traditional
 - var=\$(command) → modern

- User input

- read var → no \$ sign

variables

→ if-else,
loops, case,
functions.

interpreter

*POS

- simplified syntax
- quick development

*CNS

- fixed syntax (not free-form).
- tough debugging
- slower execution

applications

① installers

② administration



BASH shell script

- if-else

- if [condition]
then
...
fi

test cond

- if [condition]
then
...
elif [condition]
then
...
else
...
fi

- test command

- -eq, -ne, -gt, -lt, -ge, -le
 - -a, -o, !
 - -f, -d, -w, -r, -x

- loops

- while [condition]
do
...
done

*loop repeated if
cond is true*

- until [condition]
do
...
done

*loop repeated if
cond is false*



BASH shell scripts

- case

- case expr in

c1)

...
..

..
..
..

← break

c2)

...
..

..
..
..

c3)

...
..

..
..
..

*)

← default case (last case)

esac

- for loop

- C like for loop

- for ((initialization; condition; modification))

do

...

done

- for-each loop

- for var in collection

do

...

done

break
continue

a
a a
a a a
a a a a
a a a a a

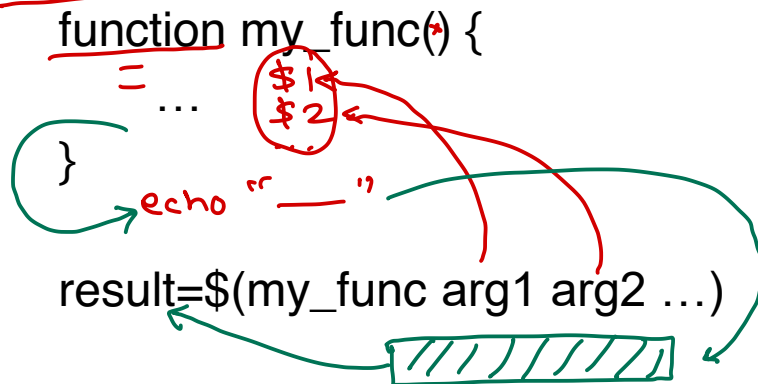


BASH shell script

- Positional parameters *→ cmd line args*
 - terminal> ./script.sh arg1 arg2 arg3
 - Special variables
 - \$0 - *script name*
 - \$1, \$2, ..., \$9 → *arg1, arg2, ..., arg9*
 - \$# - *cnt of args*
 - \$* → *list of all args - for each*
 - shift command → *shift/scrap first n args. further args are renumbered as \$1, \$2, ...*

- BASH functions

```
function my_func() {  
    = ...  
    }  
    echo "___"  
    result=$(my_func arg1 arg2 ...)
```



BASH shell script

- Arrays
 - `arr=(val1 val2 val3 ...)`
 - `${arr[0]}`, `${arr[1]}`, ...
 - `${arr[*]}` – collection of values
 - `${#arr[*]}` – count of values
- Strings
 - `str='string value'`
 - `${#str}` – string length
 - `${str:start_index}` – substring
 - `${str:start_index:count}` – substring
 - `if [[$str =~ regex]]; then echo "true"; fi`
 - `${str/find/replace}`

- Directory operations

- `pushd dirpath`
- `popd`
- `dirs -v`

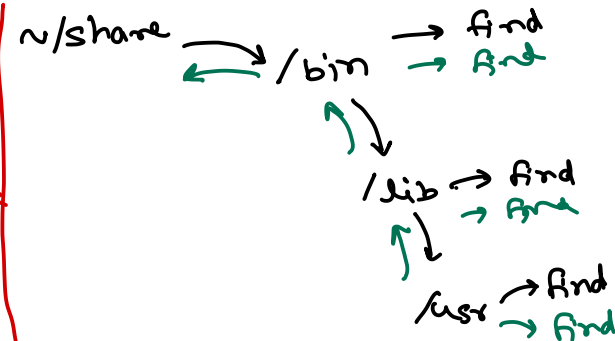
dirs
↳ print cur
dir stack

current dir
always on
top of stack.

cd → change dir

pushd dirpath
↳ go to dir
↳ put it on stack

popd
↳ pop topmost dir
from stack
↳ go to that dir.





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

