www.mirantis.com

Bash-scripting

Guzikova Anastasia

Introduction

Bourne-Again Shell

shebang, sha-bang, hashbang, pound-bang, hash-exclam, hash-pling

```
#!/bin/bash
gmake superclean
#!/bin/bash – just a comment
gmake −j 4 vace & >../ build_vace.log && \
gmake −j 4 vace_anm & >../ build_vace_anm.log
```

```
#! /bin/sh
#! /bin/bash
#! /usr/bin/perl
#! /usr/bin/tcl
#! /bin/sed −f
#! /usr/awk −f
```

```
sh scriptname
bash scriptname
```

```
#!/bin/more
```

```
chmod +rx scriptname
./scriptname
```

```
#! /usr/bin/env perl
```

MIRANTIS

Variables

```
variable1=abacaba
echo $variable1
echo ${variable1}
export variable1
unset variable1
```

```
#!/bin/bash
a=2334              #integer
let "a += 1"
echo "a = $a "   # a = 2335

b=${a/23/BB}
echo "b = $b"    # b = BB35
declare -i b
echo "b = $b"    # b = BB35
let "b += 1"       # BB35 + 1 =
echo "b = $b"    # b = 1
```

```
c=BB34
echo "c = $c"         # c = BB34
d=${c/BB/23}
echo "d = $d"         # d = 2334
let "d += 1"          # 2334 + 1 =
echo "d = $d"         # d = 2335

e=""
echo "e = $e"         # e =
let "e += 1"
echo "e = $e"         # e = 1

echo "f = $f"          # f =
let "f += 1"
echo "f = $f"          # f = 1
```

MIRANTIS

Special characters

#

- # comment
- ${#var}
- ${#*}
- ${#@}
- $((2#101011))
- ${var#0}

" "
...

- bash$ ls [Ff]*
  File file.txt
  bash$ ls "[Ff]*"
  ls: [Ff]*: No such file or directory
- bash$ var=abc; echo "$var"
  abc

:

- if condition
  then :
  else
    take-some-action
  fi
- while : # while true
  do
    commands
    if condition
    then
      break
    fi
  done

;

- cd $dir; make
- ;;

' '
...

- bash$ echo '$var'
  $var
- bash$ echo '"'"'$var'"'"'
  '$var'

.

- . filename (source file)
- ls -l .hidden
- cp ../file .

` `
...

- i=`expr $i + 1`

MIRANTIS

Special characters

!

- [! -e $file]
- [”$str1” != ”$str2”]
- ${!var} # $${var}

$

- echo $USERNAME
- ${filename##*.}
- $*
- $@
- $?
- $$

(...)

- a=123
  (a=321;)
  echo ”a = $a” # a = 123

*****

- echo *
- expr 1 * 2
- let "a = 2 ** 3"

?

- ${ parameter?err_msg }
- ((t = a<45?7:11))

{...}

- find *.{ko,klm}
  { names=`get_names`
  echo $names
  } > names.$$

  echo $names
- find -exec grep -Hns “$pattern” **{} \;**

MIRANTIS

Special characters

[...]

- [-d "$dir"] || echo "directory $dir not found."

[[...]]

- [["$a" -ne "$b" && "$a" -eq "$c"]] && echo "a != b and a == c"

((...))

- a=$((5 + 3))

|

- cat *.lst | sort | uniq

&

- *bash$* **make &>make.log &**

```
bash$ jobs -l
[1]+  19129 Running                 make &>make.log &
bash$ fg 1
make &>make.log
^Z
[1]+  Stopped                 make &>make.log
bash$ bg
[1]+ make &>make.log &
bash$ jobs -p
19129
```

MIRANTIS

Special characters

>, <
0 – stdin
1 – stdout
2 – stderr

[1]>filename
[1]>>filename
2>filename
2>>filename
&>filename
i>&j
< filename
[j]<>filename
n<&- , n>&-
0<&-, <&-
1>&-, >&-

```
: > filename
> filename

exec 3<>filename
```

```
exec 3>&1
ls -l 2>&1 >&3 3>&- | grep bad 3>&-
exec 3>&-
```

```
#!/bin/bash
interactive-program <<LimitString
command #1
command #2
...
LimitString
```

```
exec 6<&0
exec < data-file
exec <&6 6<&-
```

```
exec 6>&1
exec > $LOGFILE
exec 1>&6 6>&-
```

```
: << COMMENTBLOCK
====Documentation====
&*@!!++=
echo ”text”
======================
COMMENTBLOCK
```

```
cat <<End-of-message
#cat <<-End-of-message
#cat <<'End-of-message'
      ===============
      Text: $text
      ===============
End-of-message
```

Assigning values to variables

- a=78901
- let a=19+123
- for a in 7 8 9 11
  do echo -n "$a "
  done
- read a
- a=`ls -l`
  echo $a
  echo "$a"
- arch=$(uname -m)

MIRANTIS

Special Variable Types

```
#!/bin/bash
change_name() {
        NAME="$1"
}
print_name() {
        echo "`basename $0`: $NAME"
}
print_name $NAME # ""
change_name name2
print_name $NAME # name2
change_name name3
print_name $NAME # name3
./script2.sh     # "", name4
print_name $NAME # name3
export NAME=name5
./script2.sh # name5, name4
print_name $NAME # name5
```

```
#!/bin/bash
# script2.sh
echo "`basename $0`: $NAME"
NAME=name4
echo "`basename $0`: $NAME"
```

```
bash$ NAME=default_name
bash$ ./script.sh >/dev/null
bash$ echo $NAME
default_name
```

```
$0, $1, $2, $3, … , ${10}, ${11}, …
args=$#
lastarg=${!args}
echo "$@"    # 1 2 3 4 5
shift
echo "$@"    # 2 3 4 5
```

MIRANTIS

Debugging Bash scripts

```
echo "Hello, $USERNAME!"
echo "List of connected users:"
w
echo 'I'\''m setting variable now.'
variable=variable_value
echo "variable: $variable"
```

```
set -x # activate debugging from here
w
set +x # stop debugging from here
```

```
bash$ set -v
bash$ ls
ls
script1.sh
bash$ set +v
set +v
bash$ ls *
script1.sh
bash$ set -f
bash$ touch *
bash$ ls
*  script1.sh
bash$ set +f
```

```
bash$ /bin/bash -x script1.sh
+ echo 'Hello, aguzikova!'
Hello, aguzikova!
+ echo 'List of connected users:'
List of connected users:
+ w
 14:52:51 up 28 days,  2:00,  4 users,  load average: 0.88, 0.77, 0.61
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
aguzikov pts/0    :1               Mon20   18:36m  1.10s  0.57s ssh -o ConnectT
aguzikov pts/1    :1               Mon20   42:40   0.76s  0.76s bash
aguzikov pts/2    :1.0             14:14   26:28  24.96s 24.44s texmaker
aguzikov pts/3    :1.0             14:41    1.00s  3.45s  0.04s w
+ echo 'I'\''m setting variable now.'
I'm setting variable now.
+ variable=variable_value
+ echo 'variable: variable_value'
variable: variable_value
```

MIRANTIS

Tests

```
if [ condition1 ] # ; then
then
   command1
   command2
elif [ condition2 ] # else if
then
   command3
else
   default-command
fi
```

```
if 0              # true
if 1              # false
if -1             # false
if [ ]            # false (NULL)
if [ abc ]        # true  (string)
if [ $x ]         # true if x is defined
if [ -n "$x" ] # true if x is not empty
if [ "false" ] # true  (just string)
```

```
if cd "$dir" 2>/dev/null; then
  echo "changed directory to $dir "
else
  echo "can't change directory to $dir."
fi
```

```
if [ "$exp1" -a "$exp2" ]
if [ "$exp1" -o "$exp2" ]
[[ condition1 && condition2 ]]
[[ condition1 || condition2 ]]
```

```
-e, -f, -s, -d, -h, -r, -w, -x, -O, -G, -N, …
f1 -nt f2, f1 -ot f2, f1 -ef f2
[ -h "$file" -a ! -e "$file" ] && echo "broken"
-eq, -ne, -gt, -ge, -lt, -le
=, !=, >, >=, <, <=
=, ==, !=, >, <
if [[ "$a" < "$b" ]]
if [ "$a" \< "$b" ]
[[ $a == z* ]]
[[ $a == "z*" ]]
```

```
[ $a == z* ]
[ "$a" == "z*" ]
if [ -z "$a" ]
if [ -n "$a" ]
```

MIRANTIS

Strings

```
${#string}
expr length $string
expr "$string" : “.*”
expr match "$string" “$substring”
expr "$string" : “$substring”
```

```
${string#substring}
${string##substring}
${string%substring}
${string%%substring}
```

```
${string:position}
${string:position:length}
expr substr $string $position $length
${string: -length}
```

```
${string/substring/replacement}
${string//substring/replacement}
${var/#Pattern/Replacement}
${var/%Pattern/Replacement}
```

```
expr match "$string" '\($substring\)'
expr "$string" : '\($substring\)'
expr match "$string" '.*\($substring\)'
expr "$string" : '.*\($substring\)'
```

```
a="12345"
echo ”a=$a” #  a=12345
echo ${a:3}  #  123
echo ${a#12} # 345
echo "${a/12/21}" # 21345
${filename##*.}
```

MIRANTIS

Parameter substitution

```
${parameter}
${parameter-default}, ${parameter:-default}
${parameter=default}, ${parameter:=default}
${parameter+alt_value}, ${parameter:+alt_value}
${parameter?err_msg}, ${parameter:?err_msg}
```

```
bash$ echo ${username:-`whoami`}

bash$ : ${username:=`whoami`}

bash$ ${username:+new_name}

bash$ echo ${username:?no username}
bash: username: no username
```

MIRANTIS

Loops

```
for arg in [list]
do
 command(s)...
done
```

```
for ((a=1, b=LIMIT; a <= LIMIT ; a++, b--)); do
  echo "a=$a; b=$b "
done
```

```
for i in 1 2 3 4 5; do
      echo $i
done
```

```
for i in `seq 5`; do
      echo $i
done
```

```
for i in {1..10}
do
  echo -n "$i "
done
```

```
for param
do
      echo $param
done
```

```
files="text.txt script.sh output.log"
for file in $files
do
  if [ ! -e $file ]; then
      echo "$file doesn't exist"
      continue
  fi
  stat $file
done
```

```
OUTFILE=symlinks.list
directory=${1-`pwd`}

for file in "$( find $directory -type l )"
do
  echo "$file"
done | sort >> "$OUTFILE"
```

MIRANTIS

Loops

```
while [ condition ]
do
 command(s)...
done
```

```
LIMIT=10
a=1
while [ "$a" -le $LIMIT ]; do
  echo -n "$a "
  let "a+=1"
done
```

```
var0=0
LIMIT=10
while [ "$var0" -lt "$LIMIT" ]
do
      echo -n ”$var0 ”
      var0 = `expr $var0 + 1`
      # var0=$(($var0+1))
      # var0=$((var0 + 1))
      # let "var0 += 1"
done
```

```
((a = 1, LIMIT=10))
while (( a <= LIMIT ))
do
      echo -n "$a "
      ((a += 1))
done
```

```
while [ $# -ne 0 ]; do
        echo "$1"
        shift
done
```

```
while read line
do
      echo $line
done < files.txt
```

MIRANTIS

Loops

```
until [condition-is-true]
do
 command...
done
```

```
until [ "$var1" = end ]
do
  read var1
done
```

```
for i in `seq 0 9`; do
   for j in `seq 10`; do
      num=`expr $i \* 10 + $j`
      if [  "$num" -ge "$LIM" ]
      then
            break 2
      else
            echo $num
      fi
   done
done
```

```
for outer in I II III IV V ; do
    echo -n "Group $outer: "
    for inner in 1 2 3 4 5 6 7 8 9 10 ; do
        if [[ "$inner" -eq 7 && "$outer" = "III" ]]
        then
            continue 2
        fi
        echo -n "$inner"
    done
done
```

MIRANTIS

case

```
case "$variable" in

 "$condition1" )
 command...
 ;;

 "$condition2" )
 command...
 ;;
esac
```

```
echo "Do you wish to proceed? [y..n] \c"
while read ANS; do
        case $ANS in
        [yY]*)
                break;
        ;;
        [nN]*|exit)
                exit 0
        ;;
        *)
                echo "try again: yes, no, exit"
        esac
done
echo "Hello, $USER!"
```

MIRANTIS

select

```
select variable [in list]
do
      command...
      break
done
```

```
select color in yellow white black red
do
      echo "Your color - $color"
      break
done
```

```
1) yellow
2) white
3) black
4) red
#? 3
Your color - black
```

MIRANTIS

Functions

```
[function] function_name() {
    command(s)
}
```

```
/etc/profile
/etc/bashrc
~/.bash_profile
~/.bashrc
```

```
bash$ . functions.file
bash$ hello
Hello, aguzikova!
bash$ square
bash: $1 must be integer
bash$ square 12
144
bash$ set | grep "square ()"
square ()
bash$ unset square
bash$ set | grep "square ()"
bash$
```

```
bash$ cat functions.file
hello() {
      echo "Hello, $USERNAME!"
}
square() {
  _number=`echo $1 | sed -n '/^[0-9]*$/p'`
  if [ -z "$_number" ]; then
    echo "`basename $0`: \$1 must be integer"
    return 1
  fi
  expr $_number \* $_number
}
```

```
alias timestamp='date +"%F_%H-%M-%S\"'
alias ll="ls -l"
alias :q='exit'
unalias ll
```

MIRANTIS

Builtin commands

eval

```
y=`eval ls -l`
echo $y
```

echo, read, cd, pwd, let, source

type [cmd]

```
bash$ type perl
perl is /usr/bin/perl
bash$ which perl
/usr/bin/perl

bash$ type '['
[ is a shell builtin
bash$ type -a '['
[ is a shell builtin
 [ is /usr/bin/[
```

MIRANTIS

Builtin commands

getopts

```
scriptname -abc -e /usr/local
while getopts c:h opt; do
      case $opt in
      h)   usage
           exit
      ;;
      c)   count=$OPTARG
           echo "count is $count"
      ;;
      *)
           echo "`basename $0`: unknown option."
           exit 1
      ;;
      esac
done
```

MIRANTIS

Builtin commands

$RANDOM

```
RANDOM=$$
rnumber=$((RANDOM%25+1))
```

wait

```
#!/bin/bash
. /users/aguzikov/.bashrc

vace_gmake all.$1.vace.final.j16 && vace_gmake all.$1.vace.anm.final.j16 &
vace_gmake all.$1.vace.j16 &
vace_gmake all.$1.npe.j16 &
vace_gmake all.$1.npe.final.j16 &
vace_gmake all.$1.unittest.j16 &

wait
```

MIRANTIS

External commands

cat

cat listfile*

bash$ cat file
first line (tab)

second line

third line
bash$ cat -nvbs file
    1 first line (tab)^L
    2 second line

    3 third line

tac

bash$ tac file
third line

second line
first line (tab)

rev

bash$ rev file
)bat(enil tsrif
enil dnoces

enil driht

tee

cat listfile* | sort | tee check.file | uniq > result.file

MIRANTIS

External commands

find

```
find $dirname -name "[a-z][0-9]*.txt" -o -name ".*"
find -perm 755
find . -name "bin" -prune  -o -print
find /var/adm -mtime +1
touch -t 04162140 dstamp; find -newer dstamp
find -type l
```

-exec COMMAND \;

```
find logs/ -size +100k -exec ls -lh {} \;
find logs/ ! -type d -size +100M -exec rm {} \;
find logs/ -name "*.log" -mtime +5 -ok rm {} \;
find . -name "* *" -exec rm -f {} \;
find build/defs/ -name "*.mk" -exec grep -Hns "x86_kernel_host" {} \;
```

xargs

```
ls | xargs -n 8 echo
find -type f | xargs grep "device"
find / -type f -print0 | xargs -0 grep -liwZ GUI | xargs -0 rm -f
grep -rliwZ GUI / | xargs -0 rm -f
```

MIRANTIS

Time/Date commands

date

```
date +"%m/%d/%Y %H:%M:%S"
date +"%D %T"
03/27/12 16:15:25
date +"%s"

BEGIN=`date +%s`
./script
END=`date +%s`
DIFF=$((END-BEGIN))
printf "%dh %dm %ds" $((DIFF/3600)) $((DIFF/60)) $((DIFF%60))
```

at

```
at 2pm January 15
at 2:30 am Friday < at-jobs.list
at 2:30 am Friday -f at-jobs.list
```

time

```
time gmake
```

sleep

```
sleep 3 h
```

MIRANTIS

Text processing commands

cut

cat /etc/mtab | cut −d ’ ’ −f1,2
uname −a | cut −d" " −f1,3,11,12

paste

paste names.txt results.txt

watch

watch −n 5 tail file.log

join

join file1 file2

head

head −20 file
head −c4 file

tail

tail −1 file
tail −f
tailf

MIRANTIS

Text processing commands

grep

grep pattern [file...]
grep -Hns $pattern $file
grep −A2 −B2 "$pattern" file
grep -r "$pattern" *.txt
grep -l "$pattern" *
ps aux | grep "$named" | grep −v "grep"
egrep "(reboot|shutdown)s?" *
fgrep $word $file

wc

bash $ wc file
20 127 838 file
-w, -l, -c, -L

tr

tr "A−Z" "a−z" < text.txt
t r −d 0−9 <filename
tr ’ [:lower:] ’ ’ [:upper:]’ < file
bash$ echo "a b c " | t r −−squeeze−repeats ’ ’
a b c

MIRANTIS

sed

```
[range of lines]/p
[range of lines]/d
s/pattern1/pattern2
[range of lines]/s/pattern1/pattern2
s/pattern1/pattern2/g

sed "/$pattern/d" "$filename"
sed -n "/$pattern/p" $filename
/^$/d
1,/^$/d
s/ *$//
s/$pattern//g
for eth in `ifconfig | sed -n 's/.*\(eth[1-9][0-9]*\).*/\1/p'`; do
      ifconfig down $eth
done
```

MIRANTIS

awk

```
awk [-F<delimiter>] 'script' $input_filename
awk -f $scriptname $input_filename

BEGIN, END
awk '{print $1 $5 $6}' $filename
{ total += ${column_number} } END { print total }
awk 'BEGIN{print "COUNT\tNAME"} {print $2 "\t" $1} \
{sum+=$2} END{print "TOTAL:\t"sum}' text.txt
awk -F": " '/model name/ {print $2}' < /proc/cpuinfo
awk '{ for(i = 1; i <= NF; i=i+1) if ($i < 0) $i = -$i; print  }' matrix.txt
awk '{if ($0 ~ /pattern/) print $0}'
awk '$0 ~ /pattern/ {print $0}'
awk '/pattern/ {print}'
awk '/pattern/'
awk '$2 ~ /pattern/ {print $1}'
awk '{if(sub(/pattern1/,"pattern2")){print}}'
awk '{sub(/pattern1/,"pattern2")}1'
awk 'NR % 6'
```

MIRANTIS

awk

```
awk 'NR > 5'
tail -n +6
sed '1,5d'

awk '$2 == "foo" '
awk 'NF >= 6'
awk '/foo/ && /bar/'
awk '/foo/ && !/bar/'
awk '/foo/ || /bar/'
awk 'NF'
awk 'NF--'
awk '$0 = NR" "$0'

cat test.txt | head -n +1 | grep foo | \
sed 's/foo/bar/' | tr '[a-z]' '[A-Z]' | cut -d ' ' -f 2

cat test.txt | awk 'NR>1 && /foo/{sub(/foo/,"bar"); print toupper($2)}'
```

MIRANTIS

Thank you!

MIRANTIS