#!/bin/bash (first line)

vi .bash\_profile

export PATH=$PATH:$HOME/bin

(assign variable and show variable)

var1="ahmet"

echo $var1

var2=2

echo $var2

\*\*

\*\*use always lowercase variables\*\*

(some of environment variables)

$USER : current user

$HOME: home directory

$(date): date

**.bash\_profile:** when logins (different for each user on system)

alias cmd='ls -la'

**.bashrc:** when terminal opens (different for each user on system)

**.bash\_history**: user shell activity (duplicate command’ler ignored)

[root@localhost ~]# env | grep HIST

HISTSIZE=1000

HISTCONTROL=ignoredups

\*\*hide commands from bash history using SPACE char infront \*\*

vi .bash\_profile

export HISTCONTROL=$HISTCONTROL:ignorespace

\*\*increase historysize

export HISTSIZE='2000'

**.bash\_logout**: execute when you log out

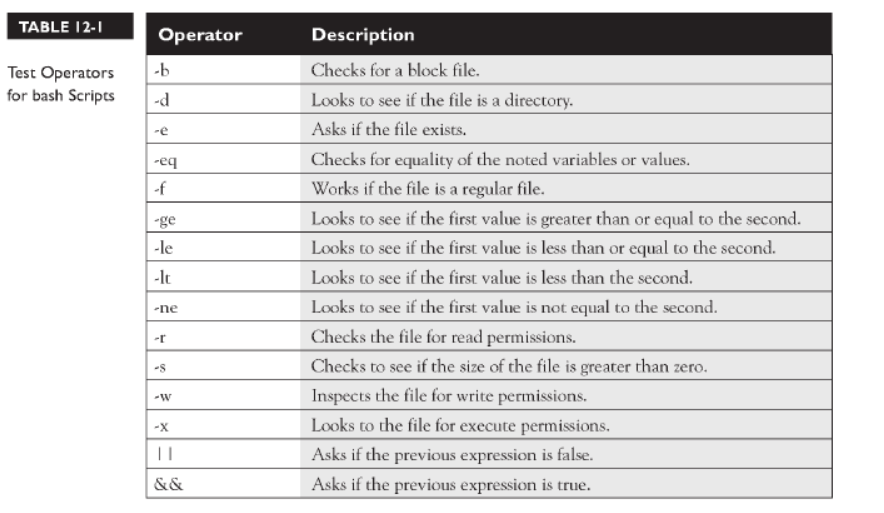
echo 'bye'

echo `date` : printer cmd result

#cmd execution check

echo $? >> 0: if cmd completed successfully

>> 1 if cmd completed unsuccessfully



#check file exists

if [ -e /etc/fstab ];

then

echo 'command completed successfully'

else

echo 'command completed unsuccessfully'

fi

#command syntax check#

if [ $? -eq 0 ];

then

echo 'command completed successfully'

else

echo 'command completed unsuccessfully'

fi

**set -e** : exit shell if there is error

echo `expr 2 + 2`

Escape characters

[root@localhost bin]# var=5

[root@localhost bin]# echo "$var"

5

[root@localhost bin]# echo "\$var"

$var

[root@localhost bin]# echo '$var'

$var

Redirecting to output to /dev/null

cmd >> /dev/null

Reading from user input and do arithmetic

read var1

echo $var1

echo `expr $var1 + 10 `

Home directory sign: ~

Declare variable

declare -r var1=1 OR readonly var1=1 (read only variable)

declare -i number (integer)

declare -a indices (array)

Arrays

[root@localhost bin]# myarray=("1" "2" "3")

[root@localhost bin]# echo ${myarray[1]}

2

[root@localhost bin]# echo ${myarray[0]}

1

[root@localhost bin]# echo ${myarray[2]}

3

[root@localhost bin]# echo ${myarray[\*]}

1 2 3

root@localhost bin]# myarray[3]=4

For Loop and Arrays Example

serverlist=("websrv01" "websrv02" "websrv03")

count=0

for index in ${serverlist[@]}; do

echo "processing: ${serverlist[count]}"

count="`expr $count + 1`"

done

Passing variables to script from terminal

echo "The following item was passed in to the script at run time $1 $2"

[root@localhost bin]# test.sh var1 var2

The following item was passed in to the script at run time var1 var2

check existence of file

file="/root/input.txt"

cd /root

if [ -a $file ];

then

echo 'file exists'

else

echo 'file doesnt exist'

fi

<https://linuxacademy.com/blog/linux/conditions-in-bash-scripting-if-statements/>

Sending error output to /dev/null

rm file.txt 2>/dev/null

(don’t print error)

Adding admin users with 10 char random password from text file

code=`date +%s | sha256sum | base64 | head -c 10`

for user in `more /root/input.txt` ; do

useradd $user -s "/bin/bash" -d "/home/$user" -m >> /dev/null 2>/dev/null

gpasswd -a $user wheel >> /dev/null 2>/dev/null

echo "$code" | passwd --stdin "$user" >> /dev/null 2>/dev/null

done

WHILE LOOP

#!/bin/bash

clear

while [ "$CHOICE" != "Q" ] && [ "$CHOICE" != "1" ] && [ "$CHOICE" != "2" ] ; do

echo "MAIN MENU"

echo "1 Exit"

echo "2 Exit"

echo "Q Exit"

echo ""

read CHOICE

done

ERROR HANDLING WITH EXIT STATUS

#!/bin/bash

clear

directory=$1 #takes user input

cd $directory 2>/dev/null #display nothing if no such path

if [ $? -eq 0 ];

then

echo "`ls -la`"

else

echo 'no such path'

exit 1

fi

sleeping

#!/bin/bash  
  
echo “Wait for 5 seconds”  
sleep 5  
echo “Completed”

Date Parsing

#!/bin/bash  
Year=`date +%Y`  
Month=`date +%m`  
Day=`date +%d`  
Hour=`date +%H`  
Minute=`date +%M`  
Second=`date +%S`  
echo `date`  
echo "Current Date is: $Day-$Month-$Year"  
echo "Current Time is: $Hour:$Minute:$Second"