We have selected expected 70 shell scripting question and answers for your interview preparation. Its really vital for all system admin to know scripting or atleast the basics which in turn helps to automate many tasks in your work environment. In the past few years we have seen that all linux job specification requires scripting skills.

**1) How to pass argument to a script ?**

./script argument

**Example :** Script will show filename

./show.sh file1.txt

cat show.sh  
#!/bin/bash  
cat $1

**2) How to use argument in a script ?**

First argument: $1,  
Second argument : $2

Example : Script will copy file (arg1) to destination (arg2)

./copy.sh file1.txt /tmp/

cat copy.sh  
#!/bin/bash  
cp $1 $2

**3) How to calculate number of passed arguments ?**

$#

**4) How to get script name inside a script ?**

$0

**5) How to check if previous command run successful ?**

$?

**6) How to get last line from a file ?**

tail -1

**7) How to get first line from a file ?**

head -1

**8) How to get 3rd element from each line from a file ?**

awk '{print $3}'

**9) How to get 2nd element from each line from a file, if first equal FIND**

awk '{ if ($1 == "FIND") print $2}'

**10)  How to debug bash script**

Add -xv to #!/bin/bash

Example

#!/bin/bash –xv

**11) Give an example how to write function ?**

function example {  
echo "Hello world!"  
}

**12) How to add string to string ?**

V1="Hello"  
V2="World"  
let V3=$V1+$V2  
echo $V3

Output

Hello+World

**13) How to add two integers ?**

V1=1  
V2=2  
V3=$V1+$V2  
echo $V3

Output  
3

Remember you need to add "let" to line V3=$V1+$V2  
then echo $V3 will give 3

if without let , then it will be

echo $V3 will give 1+2

**14) How to check if file exist on filesystem ?**

if [ -f /var/log/messages ]  
then  
echo "File exists"  
fi

**15) Write down syntax for all loops in shell scripting ?**

**for loop :**

for i in $( ls ); do  
echo item: $i  
done

**while loop :**

#!/bin/bash  
COUNTER=0  
while [ $COUNTER -lt 10 ]; do  
echo The counter is $COUNTER  
let COUNTER=COUNTER+1  
done

**until loop :**

#!/bin/bash  
COUNTER=20  
until [ $COUNTER -lt 10 ]; do  
echo COUNTER $COUNTER  
let COUNTER-=1  
done

**16) What it means by #!/bin/sh or #!/bin/bash at beginning of every script ?**

That line tells which shell to use. #!/bin/bash script to execute using /bin/bash. In case of python script there there will be #!/usr/bin/python

**17) How to get 10th line from the text file ?**

head -10 file|tail -1

**18) What is the first symbol in the bash script file**

#

**19) What would be the output of command: [ -z "" ] && echo 0 || echo 1**

0

**20) What command "export" do ?**

Makes variable public in subshells

**21) How to run script in background ?**

add "&" to the end of script

**22) What "chmod 500 script" do ?**

Makes script executable for script owner

**23) What ">" do ?**

Redirects output stream to file or another stream.

**24) What difference between & and &&**

& - we using it when want to put script to background  
&& - when we wand to execute command/script if first script was finished successfully

**25) When we need "if" before [ condition ] ?**

When we need to run several commands if condition meets.

**26) What would be the output of the command: name=John && echo 'My name is $name'**

My name is $name

**27) Which is the symbol used for comments in bash shell scripting ?**

#

**28) What would be the output of command: echo ${new:-variable}**

variable

**29) What difference between ' and " quotes ?**

' - we use it when do not want to evaluate variables to the values  
" - all variables will be evaluated and its values will be assigned instead.

**30) How to redirect stdout and stderr streams to log.txt file from script inside?**

Add "exec >log.txt 2>&1" as the first command in the script

**31) How to get part of string variable with echo command only?**

echo ${variable:x:y}  
x - start position  
y - length  
example:  
variable="My name is Petras, and I am developer."  
echo ${variable:11:6} # will display Petras

**32) How to get home\_dir with echo command only if string variable="User:123:321:/home/dir" is given ?**

echo ${variable#\*:\*:\*:}  
or  
echo ${variable##\*:}

**33) How to get “User” from the string above ?**

echo ${variable%:\*:\*:\*}  
or  
echo ${variable%%:\*}

**34) How to list users which UID less that 100 (awk) ?**

awk -F: '$3<100' /etc/passwd

**35) Write the program which counts unique primary groups for users and displays count and group name only**

cat /etc/passwd|cut -d: -f4|sort|uniq -c|while read c g  
do  
{ echo $c; grep :$g: /etc/group|cut -d: -f1;}|xargs -n 2  
done

**36) How to change standard field separator to ":" in bash shell ?**

IFS=":"

**37) How to get variable length ?**

${#variable}

**38) How to print last 5 characters of variable ?**

echo ${variable: -5}

**39) What difference between ${variable:-10} and ${variable: -10} ?**

${variable:-10} - gives 10 if variable was not assigned before  
${variable: -10} - gives last 10 symbols of variable

**40) How to substitute part of string with echo command only ?**

echo ${variable//pattern/replacement}

**41) Which command replaces string to uppercase ?**

tr '[:lower:]' '[:upper:]'

**42) How to count local accounts ?**

wc -l /etc/passwd|cut -d" " -f1  
or  
cat /etc/passwd|wc -l

**43) How to count words in a string without wc command ?**

set ${string}  
echo $#

**44) Which one is correct "export $variable" or "export variable" ?**

export variable

**45) How to list files where second letter is a or b ?**

ls -d ?[ab]\*

**46) How to add integers a to b and assign to c ?**

c=$((a+b))  
or  
c=`expr $a + $b`  
or  
c=`echo "$a+$b"|bc`

**47) How to remove all spaces from the string ?**

echo $string|tr -d " "

**48) Rewrite the command to print the sentence and converting variable to plural: item="car"; echo "I like $item" ?**

item="car"; echo "I like ${item}s"

**49) Write the command which will print numbers from 0 to 100 and display every third (0 3 6 9 …) ?**

for i in {0..100..3}; do echo $i; done  
or  
for (( i=0; i<=100; i=i+3 )); do echo "Welcome $i times"; done

**50) How to print all arguments provided to the script ?**

echo $\*  
or  
echo $@

**51) What difference between [ $a == $b ] and [ $a -eq $b ]**

[ $a == $b ] - should be used for string comparison  
[ $a -eq $b ] - should be used for number tests

**52) What difference between = and ==**

= - we using to assign value to variable  
== - we using for string comparison

**53) Write the command to test if $a greater than 12 ?**

[ $a -gt 12 ]

**54) Write the command to test if $b les or equal 12 ?**

[ $b -le 12 ]

**55) How to check if string begins with "abc" letters ?**

[[ $string == abc\* ]]

**56) What difference between [[ $string == abc\* ]] and [[ $string == "abc\*" ]]**

[[ $string == abc\* ]] - will check if string begins with abc letters  
[[ $string == "abc\*" ]] - will check if string is equal exactly to abc\*

**57) How to list usernames which starts with ab or xy ?**

egrep "^ab|^xy" /etc/passwd|cut -d: -f1

**58) What $! means in bash ?**

Most recent background command PID

**59) What $? means ?**

Most recent foreground exit status.

**60) How to print PID of the current shell ?**

echo $$

**61) How to get number of passed arguments to the script ?**

echo $#

**62) What difference between $\* and $@**

$\* - gives all passed arguments to the script as a single string  
$@ - gives all passed arguments to the script as delimited list. Delimiter $IFS

**63) How to define array in bash ?**

array=("Hi" "my" "name" "is")

**64) How to print the first array element ?**

echo ${array[0]}

**65) How to print all array elements ?**

echo ${array[@]}

**66) How to print all array indexes ?**

echo ${!array[@]}

**67) How to remove array element with id 2 ?**

unset array[2]

**68) How to add new array element with id 333 ?**

array[333]="New\_element"

**69) How shell script get input values ?**

a) via parameters

./script param1 param2

b) via read command

read -p "Destination backup Server : " desthost

**70) How can we use "expect" command in a script ?**

/usr/bin/expect << EOD  
spawn rsync -ar ${line} ${desthost}:${destpath}  
expect "\*?assword:\*"  
send "${password}\r"  
expect eof  
EOD

############### 2 #############

**What are environmental variables?**

Environmental variables are global settings that control the shell’s function as well as that of other Linux programs. Another common term for environmental variables is global shell variables.

**What is redirection?**

Redirection is the process of directing data from one output to another. It can also be used to direct an output as an input to another process.

**What is grep command?**

grep a search command that makes use of pattern-based searching. It makes use of options and parameters that is specified along the command line and applies this pattern into searching the required file output.

**What are the contents in /usr/local?**

It contains locally installed files. This directory actually matters in environments where files are stored on the network. Specifically

**Write a command that will look for files with an extension “c”, and has the occurrence of the string “apple” in it.**

Answer: find ./ -name “\*.c” | xargs grep –i “apple”

**Write a command that will display all .txt files, including its individual permission.**

**Ans:** ls -a -l \*.txt

**What is the command to calculate the size of a folder?**

To calculate the size of a folder use the command **du –sh folder1.**

**How can you find status of a process?**

Use the command

ps ux

**How can you check the memory status ?**

You can use the command

free -m  to display output in MB

free -g  to display output in GB

**How can you append one file to another in  Linux?**

To append one file to another in Linux you can use command **cat file2 >> file 1.** The operator >> appends the output of the named file or creates the file if it is not created.  While another command **cat file 1 file 2 > file 3** appends two or more files to one

**Explain how you can find a file  using Terminal?**

To find a file  you have to use command, **find . –name “process.txt”** .  It will look for the current directory for a file called process.txt.