

Interview Questions & Answer Sets : Shell Programming

Interview Questions Sets : Shell Programming

How do you find out what's your shell?

Answer: `echo $SHELL`

What's the command to find out today's date?

Answer: `date`

What's the command to find out users on the system?

Answer: `who`

How do you find out the current directory you're in?

Answer: `pwd`

How do you remove a file?

Answer: `rm`

How do you remove a file?

Answer: `rm -rf`

How do you find out your own username?

Answer: `whoami`

How do you send a mail message to somebody?

Answer: `mail somebody@techinterviews.com -s 'Your subject' -c 'cc@techinterviews.com'`

How do you count words, lines and characters in a file?

Answer: wc

How do you search for a string inside a given file?

Answer: grep string filename

How do you search for a string inside a directory?

Answer: grep string *

How do you search for a string in a directory with the subdirectories recursed?

Answer: grep -r string *

What are PIDs?

Answer: They are process IDs given to processes. A PID can vary from 0 to 65535.

How do you list currently running process?

Answer: ps

How do you stop a process?

Answer: kill pid

How do you find out about all running processes?

Answer: ps -ag

How do you stop all the processes, except the shell window?

Answer: kill 0

How do you fire a process in the background?

Answer: ./process-name &

How do you refer to the arguments passed to a shell script?

Answer: \$1, \$2 and so on. \$0 is your script name.

What's the conditional statement in shell scripting?

Answer: if {condition} then ... fi

How do you do number comparison in shell scripts?

Answer: -eq, -ne, -lt, -le, -gt, -ge

How do you test for file properties in shell scripts?

Answer: -s filename tells you if the file is not empty, -f filename tells you whether the argument is a file, and not a directory, -d filename tests if the argument is a directory, and not a file, -w filename tests for writeability, -r filename tests for readability, -x filename tests for executability

How do you do Boolean logic operators in shell scripting?

Answer: ! tests for logical not, -a tests for logical and, and -o tests for logical or.

How do you find out the number of arguments passed to the shell script?

Answer: \$#

What's a way to do multilevel if-else's in shell scripting?

Answer: if {condition} then {statement} elif {condition} {statement} fi

How do you write a for loop in shell?

Answer: for {variable name} in {list} do {statement} done

How do you write a while loop in shell?

Answer: while {condition} do {statement} done

How does a case statement look in shell scripts?

Answer: case {variable} in {possible-value-1}) {statement};; {possible-value-2}) {statement};; esac

How do you read keyboard input in shell scripts?

Answer: read {variable-name}

How do you define a function in a shell script?

Answer: `function-name() { #some code here return }`

How does getopt command work?

Answer: The parameters to your script can be passed as `-n 15 -x 20`. Inside the script, you can iterate through the `getopts` array as `while getopts n:x option`, and the variable `$option` contains the value of the entered option.

What's a way to do multilevel if-else's in shell scripting?

Answer: `if then elif fi`

How do you write a for loop in shell?

Answer: `for in do done`

How do you write a while loop in shell?

Answer: `while do done`

How does a case statement look in shell scripts?

Answer: `case in) ;;) ;; esac`

How do you define a function in a shell script?

Answer: `function-name()`

How do you find out about all running processes?

Answer: `ps -ag`

How do you stop a process?

Answer: `kill pid`

How do you remove a file?

Answer: `rm`

How do you remove recursively?

Answer: `rm -rf`

What are PIDs?

Answer: They are process IDs given to processes. A PID can vary...

How do you list currently running process?

Answer: `ps`

What is \$\$?

What is a named pipe?

What does `||` mean?

What does `&&` mean?

What is a loop?

What does `while` do?

What is a function?

What are the different kinds of loops available in shell script?

for if while case

What does \$# stand for?

`$#` returns the number of parameters that are passed to a shell script

`$?` returns the exit code of the last executed command (0 : Successful, 1 or other: Failed)

What does \$? return?

Will return the status of the command which is executed lastly.

0 > Success

2 > Error

How do u open a read only file in Unix?

"vi -R filename"

What is the difference between a shell variable that is exported and the one that is not exported?

If you have a string "one two three", Which shell command would you use to extract the strings?

How do you schedule a command to run at 4:00 every morning?

How will you list only the empty lines in a file (using grep)?

grep ^\$ filename.txt

When you login to a c shell, which script would be run first? (before the terminal is ready for the user)

first /etc/.login script is run & after that

~/login is run & then ~/.cshrc is run.

How would you get the character positions 10-20 from a text file?

cat filename.txt | cut -c 10-20

How would you print just the 25th line in a file (smallest possible script please)?

tail -n +25 | head -1 OR

head -n 25 | tail -1

How would you replace the n character in a file with some xyz?

sed 's/n/xyz/g' filename > new_filename

Grep | SED |

How you will list the ordinary files in your current directory that are not user-writable?

Use Command substitution with grep to list the names of the persons from emp.lst who were born today.

How will you remove blank lines from a file using (i) grep and (ii) sed?

Locate lines longer than 100 and smaller than 150 characters using (i) grep and (ii) sed?

How do you delete all leading and trailing spaces in all lines of a file?

Making Script Interactive

```
# read name
```

```
# echo "$name"
```

Special Parameters Used by Shell

```
# $* - It stores the complete set of positional parameters as a single string
```

```
# $? – Exit Status of last command
```

```
# $# - Number of arguments specified in command line
```

```
# $0 – Holds the command name itself
```

```
# $$ - PID of current Shell
```

```
# $! – PID of last the last Background job
```

Exit status of command

```
# exit 0 – Used when everything went fine
```

```
# exit 1 – Used when something went wrong
```

0 indicates the success and other values point to failure

The Logical Operator && and ||

cmd1 && cmd2 - cmd2 is executed only when cmd1 succeeds.

cmd1 || cmd2 - cmd2 is executed only when cmd1 Fails

Usage of -a and -o

The if Conditional

If command is successful

then

execute command

else

execute command

fi

If command is successful

then

execute command

fi

If command is successful

then

execute command

elif command is successful

then

execute command

else

execute command

fi

Using test AND [] to evaluate expressions

```
# test $x -eq $y
```

```
# [$x -eq $y]
```

Numeric Comparison Operators used by test

```
# -eq – Equal to
```

```
# -ne – Not Equal to
```

```
# -gt – Greater than
```

```
# -ge – Greater than or equal to
```

```
# -le – Less than
```


-le – Less than or equal to

String Tests used by test

Test	True if
s1=s2	String s1 = s2
s1 != s2	String s1 is not equal to s2
-n stg	String stg is not a null string
-z stg	String stg is a null string
stg	String stg is assigned and not null
s1 == s2	String s1 =s2

File related Tests with test

Test	True if
-f file	file exists and is a regular file
-r file	file exists and is a readable
-w file	file exists and is a writeable
-x file	file exists and is a executable
-d file	file exists and is a directory
-s file	file exists and has a size greater than zero
-e file	file exists

-u file	file exists and has SUID bit set
-L file	file exists and is a Symbolic link

The case CONDITIONAL

read choice

case "\$choice" in

ls -l ;;

ps -f ;;

date ;;

who ;;

exit ;;

*) echo "Invalid options"

esac

for

for variable in list

do

commands

done

for file in *.htm *.html

do

sed 's/strong/STONG/g' \$file >> \$\$

mv \$\$ \$file

gzip \$file

done

while

```
while condition is true  
do  
  commands  
done
```

basename – Changing Filename Extension

```
# basename /home/henry/project3/dec2bin.pl  
#dec2bin.pl  
# basename ux2nd.txt txt  
# ux2nd.          - txt stripped off
```

set AND shift – Manipulating the Positional Parameters

```
# set 9876 2345 6213  
# echo "$1 is $1, $2 is $2, $3 is $3"  
# $1 is 9876, $2 is 2346, $3 is 6213  
  
# set `date`  
# echo $*
```

Note- set parses its arguments on the delimiters specified in the environment variable IFS which by default is whitespace.

Shift – Shifting arguments left

Maximum number of Bash arguments

On a 32-bit Linux, this is ARGMAX/4-1 (32767). This becomes relevant if the average length of arguments is smaller than 4.

Linux 2.6.23, one argument must not be longer than MAX_ARG_STRLEN (131072).

How to find success/failure status of a command in different shells?

echo \$?

List all the users in Linux

```
cat /etc/passwd | cut -d":" -f1
```

How to change the permission of multiple files which has 655 to 755

```
for myfile in `find . -perm 655`
```

```
do
```

```
    chmod 755 $myfile
```

```
done
```

Shell script to Print only Size and file name using unix command

```
ls -l | tr -s " " " " | cut -d " " -f5,9
```

Shell script to read file line by line

<http://www.scmgalaxy.com/forum/shell-script/shell-script-to-read-file-line-by-line.html>

How do you remove blank line using grep and sed in shell scripting?

```
# grep -v "^$" filename > newfilename
```

```
# sed '/^$/d' /tmp/data.txt > /tmp/output.txt
```

```
# sed '/^[PRESS TAB]*$/d' filename
```

List all the Groups and users in linux

```
# less /etc/passwd
```

```
# less /etc/group
```

How Do I Find Out CPU is 32bit or 64bit?

```
# grep flags /proc/cpuinfo
```

```
# uname
```

How to gunzip and untar all at once!

```
gunzip myfile.tar.gz
```

```
tar -xvf myfile.tar
```

```
into one
```

```
tar -zxvf myfile.tar.gz
```

Another way to kill multiple processes easily is by adding the following two functions to the .bash_profile.

```
function psgrep ()
{
ps aux | grep "$1" | grep -v 'grep'
}

function psterm ()
{
[ ${#} -eq 0 ] && echo "usage: $FUNCNAME STRING" && return 0
local pid
pid=$(ps ax | grep "$1" | grep -v grep | awk '{ print $1 }')
echo -e "terminating '$1' / process(es):\n$pid"
kill -SIGTERM $pid
}

# psgrep http
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
apache	31186		0.0		1.6	23736	17556	?		S
Jul26	0:40	/usr/local/apache2/bin/httpd								
apache	31187		0.0		1.3	20640	14444	?		S
Jul26	0:37	/usr/local/apache2/bin/httpd								

```
# psterm httpd
```

```
terminating 'httpd' / process(es):
```

```
31186
```

```
31187
```