

Assessment - 5

The due date for submitting this assignment has passed.

Due on 2017-03-09, 15:00 IST.

Submitted assignment

1 point

The PATH shell variable contains

Fully qualified path of the home directory

Directories that must be searched for executable files

Relative path of the home directory

the login shell environment.

1 point

PS2 refers to

the prompt for Interactive shell.

the prompt for shell environment

the prompt for multi-line commands

See the second page of the ps command

1 point

From the following find the correct declaration and initialization of an array in shell?

`$arrayname=(var1, var2, ... var n)`

arrayname=(var1 var2 ... var n)

array name=(var1 var2 ... var n)

arrayname=\${arrayname[index]}

1 point

Some of the tools available for doing mathematical operations in shell scripts include

bc and expr

expr and eval

eval and bc

+, -, *, % and /

1 point

Assume that you want to compare integer values “a” and “b”. Give the correct representation for this in shell script:

[\$a=\$b]

[\$a == \$b]

[\$a == \$b]

[\$a -eq \$b]

1 point

Assume a=100, b=200 and c= 300. How you will represent the condition “**\$b is greater than \$a but is less than \$c**” in shell language?

[\$a -gt \$b -lt \$c]

[\$b -gt \$a -a \$b -lt \$c]

[\$b -gt \$a -but \$b -lt \$c]

[\$b -gt \$a -a \$b -lt \$c]

1 point

Having **#!/bin/bash** in the first line of a shell script informs us to

Invoke the shell and execute the commands following #!/bin/bash in the file.

Tell a comment that this program is going to use BASH shell.

Invoke the bash shell and execute the commands following #!/bin/bash in the file.

execute the shell script from a file present in /bin/bash

1 point

[\$# -ge 20]

Tests whether the number of lines in the program is greater than or equal to 20.

Tests whether the ASCII value of # is greater than or equal to 20

Tests whether \$# is a shell variable

Tests whether the number of arguments passed to the program/function is greater than 19

1 point

Which of the following statement about procedure is correct in Shell scripting?

It returns a single value

It does not return any value

It returns one or more values

It takes exactly one parameter

1 point

Use of which of the following statement at the end of a function, will handover control back to the calling shell script ?

exit

end

return

exit 0

1 point

How will you represent the condition $B < A < C$ in an if statement in shell script?

[B < A < C]

[B -lt A -lt C]

[B -lt A && A -lt C]

[B -lt A || A -lt C]

1 point

In File Test Operation of Shell Scripting, the condition [-h <file name>] returns true

if file exists and is a Hidden directory

if file exists and is a Symbolic link file

if file exists and is Directory

if file exists and a Regular file

2 points

The syntax for printing each element in a list using FOR statement is ?

```
for VARIABLE in <list of elements>
```

```
do
```

```
echo $VARIABLE
```

```
done
```

```
for VARIABLE in <list of elements>
```

```
do
```

```
echo VARIABLE
```

```
done
```

```
for <list of elements>
```

```
do
```

```
echo VARIABLE
```

```
done
```

```
for VARIABLE in <list of elements>
```

```
do
```

```
COMMAND LIST
```

1 point

Identify the one that is not a repetition construct in the shell.

```
for i in 10 9 8 7 6 5 4 3 2 1; do echo $i; done
```

```
while [ $i -gt 10 ]; do echo $i; i=`expr $i - 1`; done
```

```
until [ $i -eq 0 ]; do echo $i; i=`expr $i - 1`; done
```

```
do; echo $i; i=`expr i - 1`; done; until [ i -eq 0 ];
```