

Week 6 Topics:

Shell Scripting in FreeDOS, Windows Command Line (DOS), and Linux

As you saw from the Midterm Exam, these questions actually DID form the basis of the questions I used for the exam. Didn't believe me, did you? Now that you know this, why not study these now and review them weekly with me on Mondays! I may actually ask you to do something with these in class!

1. What method other than the "choice" command does Windows offer to prompt for input?
set /P <varname>=
2. What does the /A switch do to SET in a Windows batch file? Forces what is on the right side of the = to be interpreted as an arithmetic equation - interprets the product as an integer.
3. What does SET /P do? It prompts the user for input - See question #1. (Don't make me ask you again!)
4. What does the echo command do in the Linux bash shell do? It outputs the ensuing string (after dereferencing the variables, brackets, and operators) to "standard out" (the command line), followed by a line-feed / new-line character.
... and what happens if you add the -n switch to it? It suppresses the line-feed.
5. Why would I write an assigned variable as \${foo} rather than \$foo? If said variable were embedded in a statement where dereferencing it has become ambiguous - such as;
echo "\${foo}'s grades this quarter are \$bar"
6. Are spaces or the lack thereof important in the statement "foo=17" in Linux bash? Yes!
Why or why not? With spaces included, bash will interpret " = " as an equality test. (Some of you have seen this in your lab code. "Whaaaat?")
7. What is the significance of the ctrl-Z (^Z*) in bash? This is the key-combo that suspends (not backgrounds)
*(^Z is pressing the "Ctrl" key and the "Z" key at the same time)
a running process. The process cannot be killed or continue until you use the "fg" command to re-awaken it.
If you should do this, how would you go about undoing it? Hmmm.. Let me think...
You use the "fg" command plus the process ID number (PID) of the suspended process, OR by itself if you want the most recently suspended process, OR along with %0 (...or %1, %2, ..., %n - depending on how many processes you have backgrounded) to select the number of the process you'd like to bring back.
8. What is contained in the Bash variables "\$0", "\$*" and "\$#?"
\$0 - The very first positional parameter (The first item before a space) on "stdin" (Standard Input - the command line) usually the script or command name.
\$* - All of the positional parameters, when in quotes returns each as an individual entity
\$# - A count of all of the positional parameters.
9. What is the syntax for doing arithmetic in Bash? 1.) let \$var2=\$var1+10
2.) \$var2=\$((\$var1+10)) 3.) \$var2=`expr \$var1 + 10`
10. How is a variable cleared in Bash and FreeDos? DOS: val=
bash: val=

11. What are wildcards? Special specific characters which are interpreted to act as placeholders for none, one, or many characters in a sub-string of a string of characters, depending on the interpreter and the character. (EG: bash; "*" - zero or more, "?" - one or more / regular expressions; "." - one or more printing characters)

12. Show the syntax for two types of iteration statements in bash.

#!/bin/bash

for i in \$(<array of items>); do

echo Array item is : \$i

done

#!/bin/bash

ITEM=0

while [\$ITEM -lt 10]; do

echo This is item number \$ITEM

\$ITEM += 1

done

13. Show the syntax for two types of if statements in bash.

#!/bin/bash

if [-d \$dirname] ; then

(Test) echo "\$dirname is a directory"

else

echo "\$dirname is NOT a directory"

fi

#!/bin/bash

if ((\$value > 10)) ; then

echo "\$value is greater than 10" (Evaluate)

elif ((\$value < 10)) ; then

echo "\$value is less than 10"

else ; echo "\$value is 10"

fi

14. Why are the spaces in the if “[\$a -lt \$b]” mandatory? “[” is actually a command in bash and its complement ”]” is a parameter for the command. In latter iterations of bash (not POSIX compliant) the keywords “[” and ”]” have replaced this command - which still exists on your CentOS machines.