Learn Shell

1. Introductions

Start the scripts with “#!/bin/sh”

1. Philosophy
   1. Increasing speed sequence.
      1. Shell script->Perl->C/C++
   2. There are several factors which can go into good, clean, quick, shell scripts.
      1. The most important criteria must be a clear, readable layout.
      2. Second is avoiding unnecessary commands.
   3. cat /tmp/myfile | grep "mystring"

which would run much faster as:

grep "mystring" /tmp/myfile

* 1. Above may not be significant in single operation, but would effectively slow down, if goes in loops.

1. A First Script- All about echo and variable assignment.
2. Variables Part 1:
   1. The shell does not care about types of variables; they may store strings, integers, real numbers - anything you like.
   2. Note though that special characters must be properly escaped to avoid interpretation by the shell.
   3. We must **export** the variable for it to be inherited by another program - including a shell script.
   4. Sourcing will run the commands in the current shell process. executing will run the commands in a new shell process.
3. Wildcards
   1. Echo Vs ls Vs mv – think about it
4. Escape Characters:
   1. $,”,\ and ` are interpreted, they need to be preceded by \ , to be taken literally and not get interpreted.
5. Loops:
   1. For I in 1 2 3 4 do…done…..iterates for the item list
   2. For I in abc 1 \* 2 end…do…done….star replace all the files in current directory
   3. While [ condition(Space is important on both side) ] do….done
   4. While : do …done :- always true.
6. Test
   1. [ is Test .
   2. make sure to keep SPACE before and after [
   3. In fact , it is a good practice to keep spaces before and after every shell command , always.
   4. For equality check use = for string and -eq for integers.
   5. if..fi, case..esac
   6. if [condition] **newline then** or if[condition]**;then**
7. Case $option
   1. o1) something ;;
   2. o2) something break/exit;;
   3. \*) everything else
   4. esac
8. Variable 2
   1. $0 = calling script name/basename(use cmd basename on it to get name)
   2. $1..$9 = All the rest 9 parameter passed
   3. $# = number of parameter passed
   4. $@ = all the parameters
   5. $\* =As above, without preserving white spaces
   6. $$ = PID
   7. $! = PID of last background process
   8. $? = Exit status, good=0, otherwise bad
   9. IFS = Inter Filed Separator
10. functions:../common.lib
11. TBD