**Part I – The basics of Shell Scripting**

***Unix is case sensitive.***

***Special Characters and how to escape them***

The following characters have a special meaning or function:

\ / ; , . ~ # $ ? & \* ( ) [ ] ` ‘ “ + - ! ^ = | < >

Mentioned in the book are 26 characters with special meaning or function.

To overcome its special meaning or function:

Immediate preceded with a backslash \ or enclosed within ‘ ‘ (forward tick marks / single quotes)

***Shells***

[Shell is an environment where you can run your commands, programs and shell scripts.

There are different flavours of shells and different flavours of Unix OS. ]

Shells are located in:

* Bourne Shell - /bin/sh or /usr/bin/sh
* Bash Shell - /bin/Bash or /usr/bin/Bash
* Korn Shell - /bin/ksh or /usr/bin/ksh

* /usr/bin or /bin directory based on the flavour (type $(uname) to know which unix flavour you are using. command usage same as hostname or echo $(hosrname).

Interpreted and not compiled

*Shell scripts and functions are interpreted and not compiled*. Both shell scripts and functions are ASCII text that are read by shell command interpreter. So when we execute a shell script, or function, a command interpreter goes through the ASCII text line-by-line, loop-by-loop, test-by-test, and executes each statement as each line is reached from the top to the bottom.

Compiler versus Interpreter

http://web.archive.org/web/20090414123001/http://ugweb.cs.ualberta.ca/resources/java/faq/compile.html

I highly recommend akapelko's link above. The comparison w/ translating between a French and English speaker is the simplest explanation of the difference I have found: Interpreters are present (during the conversation/execution) to translate line-by-line as you speak... it's more convenient and flexible and quicker to begin, but takes more time for a given piece of content. Compilation, meanwhile, is like translating a written document that the other person can read in their own free time. It's much less flexible, but they can read it at their native language speed

Functions in a shell script

*Syntax:*

function *function\_name*

{

Commands to execute

}

*function\_name*()

{

Command to execute

}

***Control Structures***

If [ test\_command]

then

commands

elif [ test\_commands]

then

commands

elif [test\_commands]

then

commands

fi

for loop\_variable in argument\_list

do

commands

done

while test\_condition\_is\_true

do

commands

done

until test\_condition\_is\_true

do

commands

done

case $variable in

match\_1)

commands

;;

Match\_2)

Commands

;;

\*)

Commands to execute for no\_match

;;

esac

***What can be executed in a shell script***

Mostly any command in a shell script that we can execute from the command line. (One

exception is trying to set an execution suid or sgid, sticky bit, within a shell script; it is

not supported for security reasons.)

***Executing command on a remote host***

$ rsh hostname command\_to\_execute

Two files required for this:

* .rhosts file stored in the user home directory with 600 permissions
* hosts.equiv file contained in the /etc/ directory

OpenSSH command (freeware encrypted replacement for ftp, telnet, and ftp):

$ ssh user@hostname command\_to\_execute

$ ssh –l username hostname command\_to\_execute

Will prompt for password if encryption pair has not been setup. Details can be setup in the man ssh pages.

$ su - <userid-of-the-user-for-which –the-keys-are-to-be-set>

$ ssh-keygen –t dsa 🡪 to setup DSA keys

$ ssh-keygen –t rsa 🡪 to setup RSA keys

More details on the ssh-keygen man pages

***Setting traps:***

To catch an exit signal when a program has terminated before it would normally end we use traps.

* $ trap ‘echo “ \nExiting on a trapped signal”;exit’ 1 2 3 15
* Kill -9 cannot be trapped
* $ kill –l 🡪 to see all the supporting signals for your OS.
* The exit signals are as follows:
  + 0 – normal termination; end of the script
  + 1- SIGHUP – hang up, line disconnected
  + 2- SIGITN – Terminal interrupt; usually Ctrl +C
  + 3- SIGQUIT – Quit key, child processes to die before terminating
  + 9-SIGKILL – kill -9 command, cannot trap this type of exit status
  + 15-SIGTERM – kill commands default action
  + 19 – SIGSTOP – Stop, usually Ctrl + Z

command and shortcuts :

------------------------

$uname or echo $(uname) --> unix flavour

in a shell script, you can use set -x for debugging.

set -n for syntax check but needs to be commented before execution

???

vi editor:

how to go to the first line of a vi editor

how to go directlyy to the first and last character of a file

how to directly go to a specific line in a file.

do we really need to have .sh extension for the shelll scripts

how to create a user in unix, login with multiple users and then execute the last command

for fibonicci series:

<http://www.bashguru.com/2010/12/shell-script-to-generate-fibonacci.html>

example on chmod sticky bits:

<http://computernetworkingnotes.com/managing-file-system-security/sticky-bit.html>

tee command:

<http://dbaspot.com/shell/410980-tee-command-print.html>

eg.,

echo "Hello there" | tee output\_file | sed 's/there/here/'

so the file now contains “Hello there” and on the console displayed is: “Hello here”

echo “Record error : $RECORD” | tee –a $LOGFILE

this actually appends the global logfile $LOGFILE with the statements – “Record error: <<<value of record>>>” and also displays it on the screen