Shell Script Cheat Sheet

Editor

Use vi or mcedit or any editor to write your shell script.

Permissions

```
chmod +x your-script-name
chmod ugo your-script-name

Note: In the second syntax u: Owner/User; g: group; o: others
```

Execute

bash your-script-name

sh your-script-name

./your-script-name

Variables

(SD)System Variables: Upper Case, created by unix itself.

(UDV)User Defined Variables: Lower Case

List of some System Variables:

System Variable	Meaning
BASH	Our shell name
BASH_VERSION	Our shell version name
COLUMNS	No. of columns for our screen
HOME	Our home directory
LINES	No. of columns for our screen
LOGNAME	Our logging name
OSTYPE	Our Os type
PATH	Our path settings
PS1	Our prompt settings
PWD	Our current working directory
SHELL	Our shell name
USERNAME	User name who is currently login to this PC

- To print contains of the variables use $ echo $vasiable_name

Define UDV

Variable_name=value

Redirecting Standard Input Output

> Redirector Symbol

Syntax:

Linux-command > filename

Overwrites if file exists otherwise creates file.

>> Redirector Symbol

Syntax:

Linux-command >> filename

Appends if file exists otherwise creates file.

< Redirector Symbol

Syntax:

Linux-command < filename

Take input from file for linux commands.

Decision Making and Loops

If condition

```
if condition
        then
                command1 if condition is true or if exit status
                of condition is 0 (zero)
                ...
                ...
        fi
```

test command or [expr]

Syntax:

test expression OR [expression]

If… else… fi

Syntax:

```
           if condition
           then
                      condition is zero (true - 0)
                      execute all commands up to else statement

           else
                      if condition is not true then
                      execute all commands up to fi
           fi
```

Nested if-else-fi

Syntax:

```
        if condition
        then
                if condition
                then
                        .....
                        ..
                        do this
                else
                        ....
                        ..
                        do this
                fi
        else
                ...
                .....
                do this
        fi
```

Multilevel if-then-else

Syntax:

```
           if condition
           then
                      condition is zero (true - 0)
                      execute all commands up to elif statement
           elif condition1
           then
                      condition1 is zero (true - 0)
                      execute all commands up to elif statement
           elif condition2
           then
                      condition2 is zero (true - 0)
                      execute all commands up to elif statement
           else
                      None of the above condtion,condtion1,condtion2 are
true (i.e.
                      all of the above nonzero or false)
                      execute all commands up to fi
           fi
```

Loops in Shell Script

for loop

Syntax:

```
for { variable name } in { list }
do
        execute one for each item in the list until the list
is
        not finished (And repeat all statement between do and
done)
done
```

while loop

Syntax:

```
while [ condition ]
do
      command1
      command2
      command3
      ..
      ....
done
```

The case statement

Syntax:

```
case  $variable-name  in
     pattern1)   command
                     ...
                     ..
                     command;;
     pattern2)   command
                     ...
                     ..
                     command;;
     patternN)   command
                     ...
                     ..
                     command;;
     *)             command
                     ...
                     ..
                     command;;
esac
```

How to debug a Shell Script

Syntax:

sh option { shell-script-name }

OR

bash option { shell-script-name }

Option can be

-v : Print shell input lines as they are read.

-x : After expanding each simple-command, bash displays the expanded value of PS4 system variable, followed by the command and its expanded arguments.

Finding Matching Patterns

grep

Syntax:

grep {flag} "word-to-find" {file-name}

The most useful **grep** flags are shown here:

-i : Ignore uppercase and lowercase when comparing.
-v : Print only lines that do *not* match the pattern.
-c : Print only a count of the matching lines.
-n : Display the line number before each matching line.
-r : Search recursively
-w : Search words only
-l : List names of matching files
-c : count lines when words have been matched.

Search two different words

Syntax:

```
$ egrep -w 'word1|word2' /path/to/file
```