OPS235

Bash Shell Scripting - Part 3

Bash Shell Scripting Essentials / Continued
Creating a Software Information Report

OPS235

Bash Shell Scripting - Part 3

Bash Shell Scripting Essentials / Continued
Creating a Software Information Report

Bash Shell Scripting - Modifying Text

In labs1 and 2 you have learned some useful tools to create useful Bash Shell scripts which included: variables, command substitution, mathematical operators, logic, and loops.

In Lab3, you will be learning to modify text by using the following tools:

- The Here Document
  A "trick" for displaying multiple lines of text without using the **echo** command

- Using the sed command to modify text
  A Linux command that can modify text that is read from files or send to the sed command by using pipes.

The "Here" Document

In ULI101, you learned redirection relating to standard input (stdin), standard output (stdout), and standard error (stderr).

The following symbols for redirection are:

< Redirect stdin of file (e.g. mail -s "mssg" e-mail-address < file.txt)

> Redirect stdout and overwrite contents (eg. ls > listing.txt)
>> Redirect stdout and append (eg. ls >> log-file.txt)

You may have asked yourself, "Is there a redirection symbol **<<** ?"

The "Here" Document

The answer to that question is: **yes**
The redirection symbol **<<** is used to redirect stdin from within the command itself.

Command Example:

```
cat << +
Line 1
Line 2
Line 3
+
```

```
Output:
Line 1
Line 2
Line 3
```

Comments: The symbols **+** are used to mark the stdin section. The last + must be on a line contain NO OTHER characters. You can use anything other than + as long as they match. Stdin is redirected into the cat command to display text.

The "Here" Document

Although you can also use the echo command (quoting the beginning of the line) and continuing over multiple lines (ending with a quote), the Here Document works with any Linux / Unix command that accepts stdin.

Examples (Issue from shell to see what happens):

```
mail -s "test message" youremailaddr <<+
This is a test message
I hope you like it.
+
```

```
tr [a-z] [A-Z] <<+
i like ops235
i love scripting.
+
```

It is called the Here Document since the stdin is already contained "here" in the same command!

Modifying Text with sed Command

There are many Linux / Unix commands that can be used to manipulate text, possibly via pipeline commands.

In ULI101, those commands included: grep, head, tail ,cut ,sort ,tr, wc, sed and awk.

The Linux / Unix commands called sed and awk are very powerful tools for text manipulation. We will quickly discuss the sed command in this lesson (providing examples), and then discuss the awk command in another future lesson.

The sed Command

You learned in your ULI101 course the sed command stands for: **S**treaming **Ed**itor

This command can accept stdin from a file, or from pipeline commands.

Therefore, the sed command can be used as a filter to modify or manipulate text. This is particularly useful for shell scripting.

The sed Command

The sed command can manipulate matching text on a variety of criteria (such as line number(s), regular expression matches, etc).

Commands can then be used for manipulation such as omitting, printing, substituting, adding, and inserting text. The sed option -n suppresses display of text so the print command can be used; otherwise, the text will be displayed (with edits via the sed command instructions).

In Lab3, we use the sed command to print yum info output that matches keywords that are selected by the user that runs the shell script. Notice how the Here document and a loop is used for a menu to correctly select the keyword information.

The sed Command

Here are some additional examples of how the sed command can manipulate text by the substitute command (run for yourself to see what happens):

```
sed 's/|/ /g' <<+
I|like|weekends!
+
```

```
sed 's/$/\n/g' <<+
This text
should be
double-spaced!
+
```

OPS235

Bash Shell Scripting - Part 3

Bash Shell Scripting Essentials / Continued
Creating a Software Information Report

The "Here" Document

Although you can also use the echo command (quoting the beginning of the line) and continuing over multiple lines (ending with a quote), the Here Document works with any Linux / Unix command that accepts stdin.

Examples (Issue from shell to see what happens):

```
mail -s "test message" youremailaddr <<+
This is a test message
I hope you like it.
+
```

```
tr [a-z] [A-Z] <<+
i like ops235
i love scripting.
+
```

It is called the Here Document since the stdin is already contained "here" in the same command!

The "Here" Document

Although you can also use the echo command (quoting the beginning of the line) and continuing over multiple lines (ending with a quote), the Here Document works with any Linux / Unix command that accepts stdin.

Examples (Issue from shell to see what happens):

```
mail -s "test message" youremailaddr <<+
This is a test message
I hope you like it.
+
```

```
tr [a-z] [A-Z] <<+
i like ops235
i love scripting.
+
```

It is called the Here Document since the stdin is already contained "here" in the same command!