While Loops Part 1

How and when to create infinite while loops

* Learn different ways to control the number of times a while loop is performed
* Learn how to limit the number of loops
* How to end a loop based on a user input and based on exit status of a given command
* Most common use of while loop is to read a file one line at a time.
* Learn break and continue statements to selectively operate on items in a loop.

A while loop is a loop which repeats a series of commands for as long as a given condition is true.

One thing we can do with while loop is control the number of times it loops

While Loops Part 2

If we try to use a for loop to read a file line by line It will not work

* The for loop will read the file word by word
* To read a file line by line we need to use while loop in conjunction with a read command.
* On the list line of a while loop use the keyword done followed by less than (<) symbol followed by the file you wish to read.

LINE\_NUM=1

While read LINE

do

echo “${LINE\_NUM}: ${LINE}”

((LINE\_NUM++))

Done < /etc/fstab

OUTPUT

1: # /etc/fstab

2: #

3: /dev/mapper/centos-root / xfs defaults 1 1

4: LABEL=boot /boot xfs defaults 1 2

5: /dev/mapper/centos-swap swap swap defaults 0 0

We can also read from the output of a command instead of a file.

To do this pipe the output of a command to while loop.

To modify the output of previous script we can do is to list lines of xfs.

grep xfs /etc/fstab | while read LINE

do

echo “xfs: ${LINE}”

done

IN the previous examples we’re using read command into one variable

* But “Read” Command supports the data it reads into multiple variables.
* Each variable supplied to the read command will store one word or one field of data with any leftover words or fields assigned to the last variable supplied to the read command

Let’s modify previous example a bit

FS\_NUM=1

grep xfs /etc/fstab | while read FS MP REST

do

echo “${FS\_NUM}: file system: ${FS}”

echo “${FS\_NUM}: mount point: ${MP}”

echo “${FS\_NUM}: other data: ${REST}”

((FS\_NUM++)

done