CST8177 – Linux II

More Scripting and sed

Todd Kelley
kelleyt@algonquincollege.com

Today's Topics

- World writeable files on CLS
- More Shell Scripting (Sobel Chapter 27)
  - for loop
  - while loop
  - until loop
- sed stream editor basics

World Writeable Files

- The CLS is a multi-user Linux machine
- World-writeable files are bad… mmmK?
- Check your account on the CLS to make sure you have no file with "w" permission for "other":
  - try the `find` command with `-perm /o+w` see `man find`
- From the CLS sysadmin:

"I might put it into all marking scripts to find files with "other" write permissions and deduct 10% from the current assignment if it finds any."

- You too possibly will some day be a system administrator trying to keep a multi-user system running properly

Shell scripting inventory so far

- shebang line: `#!/bin/sh -u`
- `umask`
- set `PATH`
- set locale (`LC_LANG, LC_COLLATE, LC_CTYPE, LANG`)
- positional parameters (`$#,$*,$0,$1,$2,etc`)
- reading input from the user (`read [-p str] var[s]`)
- test program (`test, [` two names, same thing)
- if statements

Example 1: capitalize.sh

```
#!/bin/sh -u
PATH=/bin:/usr/bin ; export PATH
umask 022
unset LC_ALL                # unset the over-ride variable
LC_COLLATE=en_US.utf8 ; export LC_COLLATE
LC_CTYPE=en_US.utf8 ; export LC_CTYPE
LANG=en_US.utf8
echo "You passed $# arguments, and those are:$*:"
if [ $# -eq 0 ]; then
    echo "You didn't give me much to work with"
else
    echo -n "Here are the arguments capitalized:"
    echo “$*” | tr '[:lower:]' '[:upper:]'
fi
```

Example 2: match.sh

```
#!/bin/sh -u
PATH=/bin:/usr/bin ; export PATH
umask 022
unset LC_ALL                # unset the over-ride variable
LC_COLLATE=en_US.utf8 ; export LC_COLLATE
LC_CTYPE=en_US.utf8 ; export LC_CTYPE
LANG=en_US.utf8
if [ $# -ne 1 ]; then
    echo 1>&2 "$0: Expecting 1 argument; found $# ($*)"
else
    read -p "Enter your string:" userString
    if [ "$userString" = "$1" ]; then
        echo "The string you entered matches the argument"
    else
        echo "The string you entered does not match the argument"
    fi
fi
```

For loop

```
for name [ in word... ] ; do list ; done
```

- `name` is a variable name we make up
- `name` is set to each `word...` in turn, and list is exectuted
- if `[ in word... ]` is omitted, the positional parameters are used instead

For loop example

```
for f in hello how are you today; do
   echo "Operating on $f"
done
```

While loop

```
while command; do
    # this code runs over and over
    # until command has
    # non-zero exit status
done
```

While loop example

```
while read -p "enter a word: “ word; do
    echo "You entered: $word"
done
```

Until loop

- "opposite" of while

```
until [ "$word" = END ]; do
    read –p "Enter a word:“ word
    echo "You entered $word"
done
```

Basic sed

- we'll use sed to read lines from stdin or a file, and write the modified lines to stdout
- we'll concentrate on the forms
  - `sed 's/this/that/'`
    - replace first instance of `this` with `that`
  - `sed '/^#/s/this/that/'`
    - on lines that begin with `#` replace first instance of `this` with `that`
  - `sed 's/this/that/g'`
    - replace all instances of `this` with `that`
  - `sed -e 's/this/that/' -e 's/what/who/'`
    - replace first instance of `this` with `that` and first instance of `what` with `who`

Sed examples

- `echo this | sed 's/this/that/'`

```
that
```

- `echo this and this | sed 's/this/that/'`

```
that and this
```

- `echo this and this | sed 's/this/that/g'`

```
that and that
```

- `echo this and what | sed -e 's/this/that/' -e 's/what/why/'`

```
that and why
```

- `echo this and that | sed -e 's/this/that/' -e 's/that/why/g'`

```
why and why
```