

Lecture #9 – Programming the C and TC Shells (Chapter 10)

- Reading user input

The `$<` variable is used to read a line from standard input up to but not including the newline character.

```
% echo "What is your name?"
% set name = $<
Rich
% echo $name
Rich
```

Note: You can also use alternative syntax (i.e. `set name = `head -1``)

- Conditional Statements

One line form:

```
if (expression) command
```

Example:

```
if ($#argv == 0) echo "There are no args"
```

Syntax:

```
if (expression) then
    commands
else if (expression) then
    commands
else
    commands
endif
```

Example :

```
set num = $argv[1]

if ($num < 0) then
    @ class = 0
else if (0 <= $num && $num < 100) then
    @ class = 1
endif
```

File testing with C-shell:

```
-d      file is a directory
-e      file exists
-f      file is a plain file
```

- Switch (Case) statements

Syntax:

```
switch (test)
case pattern:
    commands
    breaksw
```

```
default:
    commands
    breaksw
endsw
```

Example:

```
switch ($argv[1])
case [yY][eE][sS]:
    echo "argument one is yes"
    breaksw
case [nN][oO]:
    echo "argument one is no"
    breaksw
default:
    echo "argument one is neither yes or no"
    breaksw
endsw
```

- Foreach Loops

Syntax:

```
foreach loop-index (args-list)
    commands
end
```

Examples:

```
foreach arg ( $argv[*] )
    echo $arg
end
```

```
foreach file (`ls`)
    echo $file
end
```

Note: Remember break and continue

- While Loops

Syntax:

```
while (expression)
    commands
end
```

Example:

```
set limit = $argv[1]
set index = 1
set sum = 0

while ($index <= $limit)
    @ sum += $index
    @ index++
end
```

- Jumps

goto label
where label is labelname: on a line

onintr label
is goto when interrupt key is pressed

- Example #1

```
#!/bin/csh

#
# rmc.csh: Prompt user if removing more than $limit files
#

set limit = 2
set recursive = no
set force = no
set args =

# Parse command line args for "-r" and "-f"

while ($#argv > 0)

    switch ($1)
        case -*r*:
            set recursive = yes
```

```
        breaksw;
    endsw

    switch ($1)
        case -*f*:
            set force = yes
            breaksw;
    endsw

    switch ($1)
        case -*:
            set args = "$args $1"
            breaksw;
        default:
            break;          # filename
    endsw

    shift
end

# Check for filename at end of options
if ($#argv < 1) then
    echo "usage: $0 <options> <file(s)>"
    exit 1
endif

# Count files to be deleted
if ($recursive == "no") then
    set count = `ls -dC $* | wc -w`
else
    set count = `find $* -print | wc -l`
endif

# Exit if none found
if ($count == 0) then
    echo "No files found"
    exit 1
endif

# Go ahead and delete in under limit or force
if ($count < $limit || $force == "yes") then
    rm $args $*
else
    # Ask user for permission

    set done = "false"
```

```
while ($done != "true")

    echo "remove $count entries (y/n)?"
    set OK = $<

    switch ($OK)
        case [yY]:
            rm -f $args $*
            break
        case [nN]:
            echo "Nothing removed"
            exit 0
            breaksw
        default:
            breaksw
    endsw
end
endif
```

- Example #2

```
#!/bin/csh

#
# Perform the sum of the command line args
#

@ sum = 0

# Loop over the command line args

foreach arg ($*)

    # Check current arg for numeric
    expr $arg + 0 >&/dev/null

    if ($status != 0) then
        echo "$arg is not numeric: ignoring"
        continue
    endif

    # Accumulate the sum
    @ sum = $sum + $arg
end
```

```
# Print the results
echo "Sum of $* = $sum"
```

- Example #3

```
#!/bin/csh

#
# rmempty.csh: Remove any empty directories starting with command arg
#

# Check for proper usage

if ($#argv != 1) then
    echo "usage: $0 <starting dir>"
    exit 1
endif

set start = $1

# Use find to get a list of directories
foreach dir (`find "$start" -type d -depth -print`)

    echo "checking $dir"

    # Check number of files in directory
    @ count = `ls "$dir" | wc -l`

    # If we find some, then loop
    if ($count >= 1) then
        continue
    endif

    # Perform deletion and check status
    echo " deleting $dir"
    rmdir "$dir"

    if ($status != 0) then
        echo "Removing dir $dir failed: $status"
        exit $status
    endif
end
```