**Department of Electrical and Computer Engineering**

Homework Assignment No. 02:

**Command Line Processing**

submitted to:

Professor Joseph Picone

ECE 3822: Software Tools for Engineers

Temple University

College of Engineering

1947 North 12th Street

Philadelphia, Pennsylvania 19122

September 3, 2015

prepared by:

Robert Irwin  
Email: robert.irwin@temple.edu

# Problem

In this assignment, the problem is renaming a large amount of files. We will solve this problem two ways: creating a bash script that creates a bash script to rename the files, and using a text editor’s capabilities to rename all the files. Luckily, all the files of interest are formatted identically, so an emacs macro should do the job nicely.

# Approach

To solve this problem with a bash script, the first step was generating a list of all the text files. Once this was done we had to write the bash script that would take this list of files as an argument, and write that file name in the middle of a cp <old file> <new file> command. This list of commands then had to be outputted to another bash script that would run each cp command.

In emacs the approach was much simpler because all of the filenames were formatted identically, i.e. same number of fields and the same delimiters. Therefore, we simply format a zero-padded counter, and use commands like C-e, M-0 C-k, to edit the file names. Then we simply run the macro 199,999 more times because we know there are 200,000 files.

# Results

Here is the list of files that must be renamed. Because there are 200,000, only a subset of the list will be shown.

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Blitch\_Ghislaine/eg\_00.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Blitch\_Ghislaine/eg\_01.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Czachor\_Clair/eg\_00.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Czachor\_Clair/eg\_01.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Fuse\_Yoshiko/eg\_00.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Fuse\_Yoshiko/eg\_01.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Igler\_Roy/eg\_00.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Igler\_Roy/eg\_01.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Loofbourrow\_Marge/eg\_00.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Loofbourrow\_Marge/eg\_01.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Nolting\_Jodi/eg\_00.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Nolting\_Jodi/eg\_01.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Rosek\_Elda/eg\_00.txt

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Rosek\_Elda/eg\_01.txt

Then after the script has run, the script exec.sh has been written. A portion of exec.sh is shown below.

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Blitch\_Ghislaine/eg\_00.txt /Users/tmp/bob\_0000000

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Blitch\_Ghislaine/eg\_01.txt /Users/tmp/bob\_0000001

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Czachor\_Clair/eg\_00.txt /Users/tmp/bob\_0000002

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Czachor\_Clair/eg\_01.txt /Users/tmp/bob\_0000003

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Fuse\_Yoshiko/eg\_00.txt /Users/tmp/bob\_0000004

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Fuse\_Yoshiko/eg\_01.txt /Users/tmp/bob\_0000005

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Igler\_Roy/eg\_00.txt /Users/tmp/bob\_0000006

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Igler\_Roy/eg\_01.txt /Users/tmp/bob\_0000007

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Loofbourrow\_Marge/eg\_00.txt /Users/tmp/bob\_0000008

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Loofbourrow\_Marge/eg\_01.txt /Users/tmp/bob\_0000009

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Nolting\_Jodi/eg\_00.txt /Users/tmp/bob\_0000010

cp /Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Nolting\_Jodi/eg\_01.txt /Users/tmp/bob\_0000011

Now we will take a look at the renamed files done by the emacs macro.

/usr/tmp/bob\_0000000.txt

/usr/tmp/bob\_0000001.txt

/usr/tmp/bob\_0000002.txt

/usr/tmp/bob\_0000003.txt

/usr/tmp/bob\_0000004.txt

/usr/tmp/bob\_0000005.txt

/usr/tmp/bob\_0000006.txt

/usr/tmp/bob\_0000007.txt

/usr/tmp/bob\_0000008.txt

/usr/tmp/bob\_0000009.txt

/usr/tmp/bob\_0000010.txt

/usr/tmp/bob\_0000011.txt

/usr/tmp/bob\_0000012.txt

# Analysis

Here is he code used to create the list of cp commands.

FILES=$1

#count="$(wc -l < $FILES)"

sed 's/^//' $FILES > temp.list

i=0;

FILENAME='temp.list'

lines=`cat $FILENAME`

for x in $lines

do

count=`printf "%07d" $i`

echo "cp $x /Users/tmp/bob\_$count"

i=`expr $i + 1`

done

An important part of this script is the line that reads FILES=$1. In a bash script, FILES will be assigned to the first argument entered on the command line after this script is executed. If you expect two command line arguments the second variable would be $2, and so on. If the amount of arguments is unknown, the $@ option is available.

The next command used is sed. Sed is a text editing command that edits one line of text at a time. The’s’ argument stands for findand replace, the ‘/’ are command delimeters. The argument of sed is of this format: sed ‘option/find/replace/’. This line of code replaces the first character of each line with nothing. This is necessary due to the format of the original files, which is shown below.

./Users/robertirwin/software\_tools/data/book\_00/00000014\_20130204/Blitch\_Ghislaine/eg\_00.txt

Because we are writing a script that creates a bunch of cp commands, our script must write a *valid* cp commands. The cp command expects a valid path and ./path/to/file/ is never a valid path.

The other part of this code is the counter. To create the counter, we set up a for loop that runs once for every filename, and increment the counter each time it runs.

The emacs macro used to rename the files is shown below, along with a description of the command.

C-x C-k C-f %07d //format counter

C-x ( //begin defining macro

C-e

C-b C-b C-b C-b //move cursor on top of '.' in .txt

M-0 C-k //kill from beginning of line up to cursor

/usr/tmp/bob\_ //mace appropriate edts

C-x C-k C-i //insert formatted counter

C-a //move to beginning of line

C-n //move to next line

C-x ) //stop defining macro

M-199999 C-x e //Execute the macro 199,999 more times (we know there are 200,000 files)