David Wells

10/22/10

CS270

Assignment 3

**Source**

**p1.pl**

#!/usr/bin/perl

$FILE=$ARGV[0]; #pass in file as command line arg

open (FILE);

use Scalar::Util qw(looks\_like\_number); #funtion to detect if a character is a number or not

@file\_name=<FILE>; #reads all file names from file and stores them into an array

$num\_files=scalar @file\_name; #stores number of file names in num\_files

print "\n";

for($i=0;$i<$num\_files;$i++) #for how many file names that are in the file

{

$numeric\_val = substr(@file\_name[$i],0, 1); #takes first char off of a file name

$k=1;

if(looks\_like\_number($numeric\_val)) #if the first char is a number then remove the number

{

while(looks\_like\_number($numeric\_val))

{

$num\_length=$k-1; #adjusts for 1 less than total numbers plus exit condition

$numeric\_val = substr(@file\_name[$i],0, $k);

$k++;

}

@file\_name[$i] = substr(@file\_name[$i], $num\_length);

}

else #if first char is NOT a number then add one

{

$num=$i+1; #allows numbering to start with "01" instead of "00"

if($i<9)

{

@file\_name[$i]= "0" . $num . " " . @file\_name[$i]; #concatonation of 0+number+file name

}

else

{

@file\_name[$i]= $num . " " . @file\_name[$i]; #if it is the tenth file then dont concatonate a 0 before number

}

}

print "@file\_name[$i]\n"; #print out result

}

**P2.pl**

#!/usr/bin/perl

$numargs = $#ARGV + 1; #setting numargs to be number of command line arguments

for ($i=0;$i<$numargs;$i++) #for how many files on command line

{

$ext = substr($ARGV[$i], -6); #remove last 6 chars from file name

if($ext eq ".ascii") #if 6 chars are ".ascii"

{

$file\_name=substr($ARGV[$i], 0, -6); #strip extension off

$new\_file=$file\_name . ".txt"; #add new .txt extension

rename($ARGV[$i], $new\_file); #rename old file with new file

}

}

**Results P1.pl**

Contents of text.txt:

01Now.txt

44the time.cpp

09to add

12numbers

33this file.txt

12is crazy

1num

Execution:

**-bash-3.2$ ./p1.pl text.txt**

Now.txt

the time.cpp

to add

numbers

this file.txt

is crazy

num

**Results P1.pl (cont)**

contents of text.txt:

Now.txt

the time.cpp

to add

numbers

this file.txt

is crazy

num

Execution:

-**bash-3.2$ ./p1.pl text.txt**

01 Now.txt

02 the time.cpp

03 to add

04 numbers

05 this file.txt

06 is crazy

07 num

**Results P2.pl (cont)**

ls

Ass3.tar file.ascii letters.txt lop.ascii p1.pl p2.pl p2.pl~ text.txt text.txt~

-bash-3.2$ ./p2.pl file.ascii lop.ascii

-bash-3.2$ ls

Ass3.tar file.txt letters.txt lop.txt p1.pl p2.pl p2.pl~ text.txt text.txt~

**Programming log**

Total duration 2 ½ hours:

*File I/O -5 minutes*

*Learning to use string manipulation -15 minutes*

*Implementing logic -1 1/2 hours*

*Testing- ½ hour*

*Miscellaneous (naming files, making test file, ect)-10 minutes*

I found Perl to be a very easy language and think it more intuitive than c++. Of course, there is less power in Perl. One obvious task with both assignments was to be able to string manipulate. Basically both assignments were to rename strings. Starting with p1.pl, which was an assignment to add or remove numbers at the beginning of a file name, I first started off with just simple file I/O in order to read the command line argument. Once I had this working, I had to go on the internet to search for Perl syntax on manipulating strings. In c++ this would be more difficult because you would be shifting array elements, but in Perl all I had to do was concatenate or remove characters and assign that new text to a variable. My file had to have an if statement in order to detect if the file already has a number in front of it and if so it needed to remove it. p2.pl was much easier. The task for p2.pl was to rename a file passed as a command line argument. Because I was already familiar with string manipulation, It was easier for me to rename the file with a new extension. Once I had this working I would print out the result. One I could see the result was what I wanted, I looked up the syntax for renaming a file. I threw this command in my program so that it would use my result (that I earlier printed for testing purposes) to actually rename the file given.