CS 35L Software Construction Laboratory (Lab2-B)
Wed, Jan 18, 2012, Ver 1.1

Shell Scripting (Continued)

More Linux Commands to learn:
  tr   -- transliterate files with a pattern
  sort -- sort lines of text files
  head -- display first lines of a file
  tail -- display the last part of a file
  comm -- select or reject lines common to two files
  cmp -- compare two files byte by byte
  ln   -- make links
  grep -- print lines matching a pattern

How to execute a shell script?
    Two options:
    1) Make the file executable by adding "+x" attribute (chmod +x file_name)
    2) Call the bash interpreter directly (bash file_name)

Running in the background
      & at the end of the command/line of code
      Shell doesn't wait for the command to finish if the program is running
    in the background.

Shell parameters
    The first parameter to the shell is known as $1, the second as $2, etc.
    The collection of ALL parameters is known as $*.
    Sample code 4:
        #! /bin/bash
        printf "the first parameter is: %s\n" $1
        printf "the second parameter is: %s\n" $2
        printf "echo the collection of ALL parameters is: %s\n" $*
    Note:
    There is something *WRONG* within the piece of code shown above.

More Linux Commands
    grep: [g]lobal [r]egular [e]xpression [p]print
          -- print lines matching a pattern
          http://www.panix.com/~elflord/unix/grep.html
          eg 1. cat file_1.txt | grep set
                print out lines with sring "set" in file_1.txt
          eg 2. ls -l | grep 'o'
                print out files or directories whose name contains character o
          eg 3. ps ax | grep chrome
                print out processes whose name contains the string "chrome"

    sed -- Read and modify the input line by line
         http://en.wikipedia.org/wiki/Sed
         http://www.grymoire.com/Unix/Sed.html
         Search and replace using sed

         option: -n, --quiet, --silent
             suppress automatic printing of pattern space

```
     Pick out line using line number
         eg 1. cat sample.txt | sed -n 1p
               print out the same line
         eg 2. cat sample.txt | sed -n 1~2p
                                 (first~step)
               print all the odd-numbered lines in the input stream
     Search and replace
         NOTE: The input could be a file or standard input (stdin)
         eg 1. sed s/bad/good/ < sample.txt (the content of a file as stdin)
         eg 2. sed s/bad/good/ sample.txt (the file name as a parameter)
         eg 3. cat sample.txt | sed s/bad/good/ (using pipeline)
         NOTE: By doing this, it only replace the first occurence
         NOTE: Global replacement
               sed s/bad/good/g  -- make changes to every occurence
     NOTE: sed in Mac OS's behaviour is really different

cmp  -- Compare two files byte by byte
     option: -s  --quiet  --silent
             Output nothing; yield exit status only.
             Exit status is 0 if inputs are the same,
                           1 if different,
                           2 if trouble.
             NOTE: exit code can be accessed via two approaches
                   1) echo $?
                   2) in shell script, use 'if clause'
```