Lab-1a: Introduction to Unix editor “vi” Jun.-6, 2012

http://www.cs.colostate.edu/helpdocs/vi.html

Lab-1b: Introduction to shell-scripting Jun.-6, 2012

Step-0: In this exercise you will create, download, and manipulate a file (meta-data input) that contains a list of ADCP file names. This task requires accessing data at

http://muenchow.cms.udel.edu/ForAllison

1. Access the above directory on the remote server (this could also be any external site);
2. Create a data and process directory on your own
3. Write a script in your program directory (see below) to download the data with wget;
4. Move the downloaded data files from your program directory to the data directory;

Step-1: Scripting:

1. Write, make executable, and execute a c-shell script called step-1.csh that contains
    Line-1: #! /bin/csh
   Line-2: wget –r –nd –l1 –-no-check-certificate http://web_address/file_to_download
    Line-3: echo “Display content of file file.list”
    Line-4: ls –la * >file.list
    Line-5: more file.list
   comment out line-5 as it is not terribly helpful and rewrite as
    Line-5: #more file.list
    Line-5: /usr/bin/nawk ‘END{print NR}’ file.list
2. I just introduced you to a powerful unix scripting tool called “nawk” which processes text strings one line at a time; its executes the file /sw/bin/nawk with a text string ‘END{print NR}’ that is the entire program. The program is executed with the file file.list as input. The program reads one line after another storing the line number into a built-in variable called NR. The END{…} section of the program is executed only after all lines of the file file.list are read in, at which time NR is send to an output device. Adapt to these ideas by adding the following lines (one at a time, you want to study what happens why):
    Line-5: /usr/bin/nawk ‘$9 ~ “.01” {print}’ file.list >file.list01
    Line-6: /usr/bin/nawk ‘END{print “ADCP data bin 01: “,NR} file.list01
    Line-7: /usr/bin/nawk ‘$9 ~ “KS04” {print}’ file.list >file.listKS04
    Line-8: nawk ‘END{print “ADCP data files at KS04: “,NR}’ file.listKS04
    Line-9: echo “#! /bin/csh” >file1.csh
    Line-10: nawk ‘NR == 1 {print “command.here”,$9}’ file.list >>file1.csh
3. The new operators “>” and “>>” are the Unix shell-scripting way to create new files. So, rather than directing output from commands to the (default) screen terminal, these redirects open new files and place the output there. The “>” will over-write any existing files in the current directory that have this name while the “>>” will append the output should that file already exist in the current directory. The above also introduced $x which refers to column-x in the input file, file.list in this case.
4. Please note that in 3. you made a file that potentially constitutes an automatically generated new executable. Please modify Line-10 to generate yourself a shell script file1.csh that would automatically extract header information the data file just downloaded from a web site?
5. Please add 2 lines that make file1.csh executable (Line-11) and execute it (Line-12).