Lab Exercise #2 -- C Shell Scripts

This assignment familiarizes you with C shell scripts and related C shell features. You are to complete the experiments below and record the requested information on your worksheet.

For your convenience as you complete this exercise, a list of the built-in C shell commands is available as "~cse410/Labs/lab02.commands".

A. C Shell Dot Files

When it is executed, the C shell checks for a file named ".cshrc" in your home directory. If the shell locates that file, it reads the contents and executes any commands that it finds. This gives you the ability to customize your environment with "personalizations" which are available whenever you access the system.

1) Examine the contents of your ".cshrc" file and briefly describe the actions which occur when the shell processes that file. Note: the "source" command reads and processes the contents of a file.

____

____

____

____

2) Examine the contents of the system-wide initialization file included in your ".cshrc" file ("/soft/sparc/share/lib/cshrc.std") and briefly describe the actions which occur when the shell processes that file.

____

____

____

____

3) Examine the contents of your ".personal" file and briefly describe the actions which occur when the shell processes that file.

____

____

____

4) Examine the contents of your ".aliases" file and briefly describe the actions which occur when the shell processes that file.

____

____

____

B. C Shell Short-cuts

Aliases and the history facility provide a number of convenient short-cuts.

1) Use the built-in command "alias" to display the list of aliases which "csh" currently recognizes, then answer the questions below.

Command(s) equivalent to "cls": ____

Purpose of "cls": ____

Command(s) equivalent to "ll": ____

Purpose of "ll": ____

Command(s) equivalent to "rm": ____

Purpose of "rm": ____

2) Use the built-in command "history" to display the history list maintained by "csh", then answer the questions below.

Purpose of "!!": ____

Purpose of "!8": ____

Purpose of "!-4": ____

Purpose of "!ali": ____

C. C Shell Variables

The shell permits you to assign values to variables and manipulate them using the "set" command and related features.

1) Use the built-in command "set" to display the list of variables which "csh" currently recognizes, then give the current value and purpose of each of the shell variables listed below.

user: ____

cwd: ____

home: ____

shell: ____

history: ____

2) Describe the purpose of the following sequence of commands:

```
cd ~cse410/Labs
set s_list = ‘ls *script*‘
echo $s_list
```

____

____

____

D. C Shell Scripts

A shell script is a sequence of commands stored in a text file so that the commands can be executed, as a group, at a later date. Since some of the built-in shell commands allow you to build selective or repetitive control constructs, shell scripts often can be used instead of C/C++ programs to handle certain tasks.

For example, assume that the file "simple" contains the following lines:

```
   #!/bin/csh -f
   #
   # Display the number of users currently on the system
   #
   echo -n "Number of users on the system:"
   who | wc -l
```

The first line of a shell script identifies the particular shell which should be used to interpret the commands in the shell; subsequent lines are comments (starting with ’#’) or commands to be executed.

1) Examine the "man" entry for "csh" and briefly describe the purpose of the following options to the C shell.

   Purpose of "-f": ____

   Purpose of "-x": ____

After creating a script with an editor, you should set the "execute" bit in the file permissions for the file:

```
   chmod +x simple
```

Now, the script can be executed by typing the name of the file as the first token on the command line. Notice that there are three cases for processing the first command-line token: the token can be the name of a built-in shell command, a pathname for an executable file containing machine language instructions, or a pathname for a shell script (an executable file containing a sequence of commands).

The other tokens entered on the command line are treated as arguments to the shell script. The shell variable "argv[1]" contains the second token from the command line, "argv[2]" contains the third token, and so on.

For convenience, the notation "$1" is equivalent to "$argv[1]".

2) Examine the C shell script named "lab02.script1", then answer the questions below.

   Meaning of "$0": ____

   Meaning of "$#argv": ____

   Meaning of "$argv[*]": ____

   Meaning of "$*": ____

   Meaning of "$argv[2]": ____

   Meaning of "$2": ____

E. C Shell Control Constructs

1) Examine the contents of the C shell script named "lab02.script2", then answer the questions below.

What command is used by the script to display a blank line?

What command is used by the script to display the entire contents of the variable "path"?

What is the purpose of the "foreach" control construct in the script?

What is the purpose of the "-n" option for the "echo" program?

2) Examine the script named "lab02.script3", then answer the questions below.

What is the meaning of the operator "!=" in the script?

What is the meaning of the operator "==" in the script?

What is the purpose of the outermost "if" control construct in the script?

What is the purpose of the innermost "if" control construct in the script?

What is the purpose of the "foreach" construct in the script?

3) Copy the C shell script named "lab02.script4" into your account, then modify it to perform all of the operations described in the comments.