14. Command name aliasing, .Cshrc and Tar

Lab 14.1 - Alias

Both csh and ksh provide command name aliasing, to allow you to rename commands. Aliasing can save a lot of keystrokes if you must frequently issue the same lengthy command. The alias command requires two pieces of information: The command you wish to alias, and the alias you wish to use to refer to it.

To alias the "history" command to "hi" you could use the following command:

```
alias cl cat days
```

This defines a new command, "cl". When you type "cl", the computer acts
just as if you had typed "cat .login". Try it out.

```
alias cl
```

to see what "cl" is defined to be. You can also type

```
alias
```

"alias" by itself gives you a list of all the currently-defined aliases
in csh.

To get rid of an alias, the command is "unalias". Type

```
unalias cl
```

If you subsequently type "cl", you should get some sort of error message, and "alias cl" will produce no result.

Now, "alias" typed from the keyboard is not really all that useful. It will disappear once the session is over and you log out and will not be there the next time you log in.
What is helpful is a more permanent alias. You can make an alias "permanent" by putting the alias command in your .cshrc file. Since all the commands in the .cshrc file are executed every time csh is started up, you then have your newly-defined command available every time you log in.

It is likely there are already some aliases in your .cshrc file. A very common one is "alias ls ls -sF". This makes "ls" shorthand for a longer version of itself, which prints the size of a file, along with a
special character that tells what type of file it is ("/", meaning directory, and "*", meaning executable, are the most useful of these).

Putting aliases in .cshrc should be done with care, since modifying such an important file can be dangerous. After all, what if you destroy the file? Also, if you don't know how to use an editor, how to you get the alias out of the file?

One solution to this is to make a backup of .cshrc, by typing

```
cp .cshrc cshrc.bak
```

for example. Then if things get messed up, you can just type "cp cshrc.bak .cshrc", log out, and all will be well.

This command will then be aliased the next time you login. To make the changes you have made effective in your current shell you must first source your .cshrc file by typing:

```
source .cshrc
```

NB: You must be in the directory that contains your .cshrc file or else specify the correct path to your .cshrc file.

Lab 14.2 - A note on using your .cshrc file

When you execute commands from a command prompt, you are doing so from a *shell* environment.

What this basically means, is that you are using a program that creates an environment for you to interact with the operating system.

In UNIX, the program which allows you to interact through a command prompt is called a *shell*. There are several different kinds of shells for use on a UNIX system, each with it's own special abilities, bells, and whistles. However, for many UNIX flavors (including SUN and SGI), the default shell for a new user account is the ***C Shell***.

When you logon to the system, there are several commands that are executed to initialize your session on the machine. The ideas of hidden files (those files or directories whose name begins with a period ".") was mentioned in the section entitled "Moving around the file system". Hidden files contain settings information for programs, files, and shell environments.

What this means is that some programs can be customized by placing commands or parameters within their associated hidden files.

It is possible to place comments in your ".cshrc" file. Comments begin with the pound ("#") character.

You can use the ".cshrc" file to create *aliases* for commands that you use alot, or for commands whose syntax can be complex and confusing.

The above ".cshrc" file example shows many aliases:

To create an alias, simply open your ".cshrc" file in a text editor, go to the end of the file and

on a new line make an alias entry. If an alias section already exists in your ".cshrc" file, place your new alias entry on a line by itself anywhere within that section.

Lab 14.3 – TAR

tar stands for Tape Archive. It was originally designed for tape backups, but it is used to create a tar file anywhere on the filesystem. Tar creates one "tar file" (also known as a "tarball") out of several files and directories. A tar file isn't compressed, it's just a heap of files assembled together in "one container". So, the tar file will take up the same amount of space as all the individual files combined, plus a little extra. A tar file can be compressed by using gzip or bzip2.

Here are some examples:

tar –xvf example.tar – *Extract the contents of example.tar and display the files as they are extracted.*

Tar –cf backup.tar /home/ftp/pub – *Create a tar file named backup.tar from the contents of the directory /home/ftp/pub*

tar –tvf example.tar - *list contents of example.tar to the screen*

Lab 14.3.1 - compress

This reduces the size of a file, thus freeing valuable disk space. For example, type

```
ls -l science.txt
```

and note the size of the file. Then to compress **science.txt**, type

```
compress science.txt
```

This will compress the file and place it in a file called **science.txt.Z**

To see the change in size, type **ls -l** again.

To uncomress the file, use the uncompress command.

```
uncompress science.txt.Z
```