Homework 2: CS114 Unix Tools, Fall 2007

Due electronically (via CMS) on Friday 5 October 2007, 11:59PM

1 Configuring your shell

In this section, you will customize your shell environment. The first part of this problem is choosing which file to edit. The second part is editing it to produce the changes you desire.

1.1 Choosing the file to edit

By default, you are assigned the `tcsh` shell. You may have chosen to file a ticket to be switched to another shell - that's fine, but you will get less help in doing this problem. If you are using `tcsh` or `csh`, the file to edit is `~/.cshrc`. If you are using `bash`, you'll want to edit `~/.bashrc`. I realize that the top of these files says `DO NOT CHANGE THIS FILE` - just ignore this[1]. Your changes should all be appended to the end of the file.

1.2 Customizing your shell environment

Change your shell prompt. This involves changing the shell variable `$prompt` (for C shell variants) or `$PS1` (for `bash`). You can change the prompt to whatever you wish; look at the manpages for the shells to see the format for the prompt variables.

Set the value of the `$EDITOR` environment variable to point to `nano` (or whatever the UNIX editor you like is). Note that the value of `$ EDITOR` needs to be the full absolute path to the editor program[2].

1.3 What to submit

Whichever file you modified, copy it to a file called `shell-init.sh` and add a final extra line indicating what the original name of the file was. Submit `shell-init.sh` in CMS, by the due date.

[1] The shell configuration files are yours to edit as you please. The system administrators in the CSUG lab have tried to make everyone's life easier by providing a set of files with "hooks", and encouraging users to modify those hooks rather than the original files. Since the structure of this will vary from system to system and this is a class in general UNIX tools, I'm encouraging you to just edit the .cshrc/.bashrc anyway. If you wish to follow the directives in `/usr/share/skel/CUSTOMIZATION`, feel free.

[2] If you're curious, this is important because sometimes programs will automatically start a text editor for you, and if you don't specify the `$EDITOR` variable, the editor chosen will generally be `vi`.

2 Written problems

This section consists of a number of questions. You should answer each of the questions, and submit all the answers in an ASCII text file called `written.txt` in CMS, by the due date.

1. Give one command that will create a file called `'" *` (this filename is four characters long: single-quote double-quote space asterisk). There are several ways to do this, any will do.

2. Most of the directories directly under the root directory (i.e. `/bin`, `/usr`, etc.) are set to be readable and executable by anyone. One of these directories is also writeable by anyone - which one? Which of these directories is not even readable by anyone but the owner?

3. Consider the files in `/usr/bin` that are two characters long. Now consider the 10 files among these that are alphabetically last. Give a one-line command that will write these 10 files, in reverse alphabetical order, to a file called `usr-bin-last`. For example, if there were 15 files named `aa, bb, cc, dd, ee, ff, gg, hh, ii, jj, kk, ll, mm, nn, oo`, the command should write a file containing `oo nn mm ll kk jj ii hh gg ff` (one name per line). Programs that may be of use: `ls, head, tail, sort`.

4. Consider the following session on `empire.csuglab.cornell.edu`:

   ```
   [someone@empire ~]% jobs
   [1] - Suspended yes > /dev/null
   [2] + Suspended vi temp.txt
   [someone@empire ~]% sort -u verylarge.txt > file &
   [3] 16186
   ```

   This session now has three jobs associated with it.

   What is the status of each of these jobs: stopped, running in the foreground, or running in the background? For any non-running processes, how would you make them resume running in the foreground? How would you remove process 2 without letting it run any further?