PHYS 410: Computational Physics Fall 2002 Homework 1 (revised Sept. 13)
Due: Thursday, September 19, 10:00 AM

Problem 1a: Send me (`choptuik@physics.ubc.ca`) an e-mail message from your account on the Physics & Astronomy Department Sun server, `physics.ubc.ca`. Use the short form of the system's "hostname"—i.e. `physics`—for the subject of the message. Include *your* name in the body of the message. Incidentally, note that there is a Unix command `hostname`.

Problem 1b: Send me an e-mail message from your account on the `lnx` machines. Again, use the hostname (one of `lnx1`, `lnx2`, or `lnx3`, as appropriate) for the subject of the message, and include your name in the text of the message.

Problem 1c: Using the `ssh-keygen` command as discussed in class, and described in the *Unix* notes, create (or add to) `~/.ssh/authorized_keys` on *both* your `physics` and `lnx` accounts so that you can `ssh` and `scp` from `physics` to `lnx1, lnx2, lnx3` and vice versa, without being prompted for a password.

Do the remaining problems using your account on the `lnx` *machines (it doesn't matter which one or ones you use). In particular, note that "~" in the following refers to your* `lnx` *home directory, and all file and directory references are to files on those machines. As always, send me mail immediately if you encounter problems using any of the machines.*

IMPORTANT!! Be sure to read and follow all instructions *carefully*, as your grade will depend on certain files being in the right places with the right names.

Problem 2: I have created directories for each of you which you may use to "publish" Web pages (related to this course) via my research group's web server (`http://laplace.physics.ubc.ca`). Your personal directory, which I'll refer to as your Web directory, is `~/public_html`. I have created a "template" homepage named `~phys410/public_html/index.html`. Copy this HTML file into your Web directory (use the same name—`index.html`) and modify the HTML file to reflect your name, address, phone-number etc. Use your text editor of choice (presumably `vi` or `emacs`) to perform the modifications. If you don't feel like publishing any specific piece of information, specify it as "unlisted". Below the horizontal rule (line) in the template file, add suitably labelled links to:

1. The course home page.
2. The instructor's home page.
3. At least 5 Web pages that have something to do with "carbon nanotubes".

Check your work by verifying that you can get to your home page by going to our Course page, clicking on "Student Pages" and then your name. Also, please send me e-mail should the way I have listed your name in the "Student Pages" list need changing.

Problem 3a: I have created a subdirectory named `hw1` (for Homework 1) in your home directory. I have also changed the permissions on this directory so that your work remains private; please do not alter the permissions until after the homework has been graded. In this new directory, create another subdirectory named `a3`. In that directory (`~/hw1/a3`), and using `vi` or `emacs`, create a file named `apple` which contains the following text from *Gravitation*, by Misner, Thorne and Wheeler. Try to duplicate the spacing, line breaks, punctuation etc. as closely as possible.

```
Once upon a time a student lay in a garden under an apple tree reflecting
on the difference between Einstein's and Newton's views about
gravity.  He was startled by the fall of an apple nearby.  As he
looked at the apple, he noticed ants beginning to run along its
surface. His curiousity aroused, he thought to investigate the
principles of navigation followed by an ant.  With his magnifying glass,
he noted one track carefully, and, taking his knife, made a cut in the apple
skin one mm above the track and another cut one mm below it.  He peeled
off the resulting little highway of skin and laid it out on the face
of his book.  The track ran as straight as a laser beam along this
highway.  No more economical path could the ant have found to cover
the ten cm from start to end of that strip of skin.  Any zigs and
zags or even any smooth bend in the path on its way along the apple
peel from starting point to end point would have increased its length.

   ``What a beautiful geodesic'', the student commented.
```

Problem 3b: Create a file in the same directory (`~/hw1/a3`) called `kumquat` which is identical to `apple` except that all occurences of the word "apple" are replaced with "kumquat". Leave a brief note in a file called `README` (again in the same directory) which describes how you created `kumquat` and made the changes.

Problem 4: In a directory `~/hw1/a4`, create a file called `compressed` whose contents are identical to `~phys410/hw1/q4/input` except that all lines which contain *only* spaces have been removed. Leave a brief note in `~/hw1/a4/README` which describes how you solved the problem.

Problem 5: Make the directory ~/hw1/a5. The file /usr/share/dict/words contains a list of "words" (mostly genuine English words—for the purposes of this question, any entry in the file will be deemed a "word"), one per line.

How many words does the file contain? Answer in ~/hw1/a5/README.

In ~/hw1/a5 create files with the following names and contents (words should appear one per line):

1. `7letter` which contains all the seven-character (seven-letter) words in `/usr/share/dict/words` that do *not* begin with a vowel, in alphabetical order. (Define the set of vowels to be 'a', 'A', 'e', 'E', 'i', 'I', 'o', 'O', 'u', 'U', and 'y', 'Y')
2. `14letter` which contains an alphabetical list of all the fourteen-character words in `/usr/share/dict/words` that do *not* contain 'e', 'E', 'u' or 'U'.
3. `rab` which contains all the words in `/usr/share/dict/words` that begin with 'ab' in *reverse* alphabetical order (*Hint:* Use the Unix `sort` command).
4. `5cons` which contains an alphabetical list of all the words in `/usr/share/dict/words` which contain 5 or more *consecutive* consonants. (Define the set of consonants to be the complement of the vowel set used in part 1).

Leave comments in ~/hw1/a5/README which describe how you solved each problem.

Problem 6: Make the directory ~/hw1/a6. Use the plotting program `gnuplot` to produce a plot of $\sin(6x) * \sin(5x) * \cos(4x)$ for $-3 \le x \le 4$. Save your plot as the postscript file `plot.ps` in the directory ~/hw1/a6. *Hint:* Use `gnuplot`'s extensive on-line help: you may find `help plot`, `help postscript` and `help output` useful.

Problem 7: Make the directory ~/hw1/a7. In that directory create a (Bourne)-shell script, `Who`, which executes as follows:

`Who` takes *no* arguments and produces on standard output:

1. The current date and time, as returned by the `date` command
2. A formatted summary of the users that are logged onto `lnx1`, `lnx2` and `lnx3` at the time that the command is issued.

Sample usage and output from an invocation `matt@lnx1.physics.ubc.ca` is:

```
lnx1% Who
Fri Sep  6 09:58:32 PDT 2002

aadutton     matt         phys410

lnx1%
```

If `Who` is invoked with any non-zero number of arguments, it must echo a usage message precisely as follows, then exit:

```
lnx1% Who goes there
usage: Who

       Reports who is logged onto lnx[123].

lnx1%
```

Additional details:

1. User names are to be formatted *left-justified*, in 12-column fields, with a maximum of 6 fields per line.
2. Note that there are single blank lines in `Who`'s canonical output, immediately preceding and following the user listing; similarly, in the usage message there are blank lines preceding and following the line: `Reports who is logged onto lnx[123].`
3. Note that for this command to work properly (i.e. so that execution does *not* result in any prompts for passwords), you will have to complete **Problem 1c** succesfully; if you *cannot* do so, contact the instructor for help.
4. Ensure that your script has execution permission enabled (for yourself at a minimum), so that, provided that your working directory is `~/hw1/a7`, you can execute the script simply by typing `Who`.

Hints:

1. The following Unix commands may be useful: `cut`, `printf`, `sort`, `uniq` and `w`. Refer to the `man` page for each, and note that the use of specific options for certain commands will make implementation of `Who` easier. For the case of `printf`, also refer to the `man` page for the standard `C` library routine of the same name, via `man 3 printf`, for a full description of format strings.
2. Consider the use of command pipelines, as well as output redirection.
3. Consider the use of a temporary file to accumulate information.

Thoroughly test and debug your implementation of `Who`; when you are satisfied that it is working properly, create the directory `~/bin`, and copy the script there. Having done so, and possibly after issuing a `rehash` command (see `man tcsh` for a description of `rehash`, verify that you can then execute `Who` from an arbitrary place in the filesystem, including, at a minimum, `/` (the root of the filesystem), and your home directory. If necessary, modify your script until this final test succeeds—if modifications *are* required, ensure that the versions of `Who` in `~/hw1/a7` and `~/bin` remain identical.