CSE 303, Winter 2006, Assignment 1
Due: Monday 16 January, 9:00AM[1]

Last updated: January 3

You will get experience using the Linux bash shell, using emacs, and writing a short script.

1. (Exploring commands) First run the command `script problem1`. Then run at least 75 *different* commands using at least 12 different programs. Then run the `exit` command.

   - Only commands that succeed (do not print an error) count.
   - For this problem, two commands are *different* if they use different programs and/or different options, but *not* just different filenames. (Examples: `ls` and `ls -a` are different but `ls foo` and `ls bar` are the same.)

2. (Exploring command-line editing) Suppose you mean to type `cat foo`, but you accidentally type `cta foo`. You notice this when the cursor is to the right of the second "`o`" (you haven't hit return yet).

   - Describe keystroke-by-keystroke how to turn the command-line into `cat foo` without typing any letters. Give at least two solutions.
   - Suppose your previous command was `cat food`. Give a third solution that uses this useful fact.

   Write your solution in a file called `problem2`.

3. (Commands and output) Use each of the following commands such that "hello" (without the quotes, and nothing more) is printed (on standard out, and nothing is printed on standard error). You can precede your commands with other commands (e.g., to create a file) and/or pass options to your commands.

   `echo`, `cat`, `ls`, `grep`, `!!`

   Hint: The last one is tricky. Put a script that does nothing in a file with a particular name.

   In a text file called `problem3`, describe your solution, including each command you use and a *very brief* explanation of it.

4. (Aliases)

   - Using an alias, make it so that `cta` works just like `cat`. For example `cta foo` would print the contents of file `foo`.
   - Make an alias `lsdd` (stands for "list double-dots") such that typing `lsdd` at the prompt does the following:
     - It prints every file or directory in the current directory that has *two or more* dots in its name. If a file starts with "." and has one or more other dots, it *should* be printed.
     - It prints only one file name per line and only the files' names.
     - For directories, it prints the directory name, *not* its contents.
     - Note that the "parent directory" (`..`) will always be printed.

     Hint: Use two options (see man ls) and two file-patterns (one for hidden files and one for others).

   Put your solution in a file called `myaliases`. Running `source myaliases` should successfully add the aliases to the shell.

5. (Short script) Write a `bash` script in a file `mycat` that takes 3 or more arguments (let's call them f1, f2, f3, ..., fn) and works as follows:

[1] Yes, this is a holiday, but there's little reason you cannot complete the assignment by the end of the preceding week.

- It treats all arguments as filenames.
- If fewer than three arguments are given, it prints an appropriate error message on stderr and exits.
- If a file (or directory) named f1 or f2 exists, it prints "Error: file exists" on stderr and exits.
- Else it concatenates the contents of f3, ..., fn into file f1. It should not print any error messages from this (for example, if some file does not exist or is a directory). Instead, such error messages should be saved in file f2.
- Note: Put filenames in double-quotes in case they contain "funny characters".

Note: Sample solution is 15 lines. Longer is fine within reason.
Hints: shift, `$@`, -lt, -a

6. (**Extra Credit**) Note: Remember the course policy on extra credit.
   - Make an alias `lsdd2` that is like `lsdd` except that it does *not* print the parent-directory (`..`). Hint: Pipe to grep.
   - Make a script `mycat2` that is like `mycat` except it *detects duplicates*, i.e., it puts the contents of a file in the output at most once. Hint: Use loops to output one file at a time (using `>>` to append) and an array to store what files have already been output.

Assessment: Your solutions should be:

- Correct shell scripts that run on `attu.cs.washington.edu`
- In good style, including indentation and line breaks
- Of reasonable size

Turn-in Instructions Use the `turnin` command (man turnin) for course cse303 and project hw1. In particular, type:

```
turnin -ccse303 -phw1 problem1 problem2 problem3 myaliases mycat
```

from a directory containing your solution. If you use a late-day (see the syllabus) use the project hw1late instead of hw1. If you do the extra credit, put `lsdd2` in `myaliases` and also turnin `mycat2`.