Use Lecture 4 to do the following:

http://www.doc.ic.ac.uk/~wjk/UnixIntro/Lecture4.html

1. Archive the contents of your home directory using `tar`. Compress the tar file with `gzip`. Now uncompress and unarchive the `.tar.gz` file using `cat`, `tar` and `gzip` on one command line.
2. Use `find` to compile a list of all directories in the system, redirecting the output so that the list of directories ends up in a file called `directories.txt` and the list of error messages ends up in a file called `errors.txt`.
3. Try the command `sleep 5`. What does this command do?
4. Run the command in the background using `&`.
5. Run `sleep 15` in the foreground, suspend it with Ctrl-z and then put it into the background with `bg`. Type `jobs`. Type `ps`. Bring the job back into the foreground with `fg`.
6. Run `sleep 15` in the background using `&`, and then use `kill` to terminate the process by its job number. Repeat, except this time kill the process by specifying its PID.
7. Run `sleep 15` in the background using `&`, and then use `kill` to suspend the process. Use `bg` to continue running the process.
8. Startup a number of `sleep 60` processes in the background, and terminate them all at the same time using the `pkill` command.
9. Use `ps`, `w` and `top` to show all processes that are executing.
10. Use `ps -aeH` to display the process hierarchy. Look for the `init` process. See if you can identify important system daemons. Can you also identify your shell and its subprocesses?
11. Combine `ps -fae` with `grep` to show all processes that you are executing, with the exception of the `ps -fae` and `grep` commands.
12. Start a `sleep 300` process running in the background. Log off the server, and log back in again. List all the processes that you are running. What happened to your sleep process? Now repeat, except this time start by running `nohup sleep 300`.
13. Multiple jobs can be issued from the same command line using the operators `;`, `&&` and `||`. Try combining the commands `cat nonexistent` and `echo hello` using each of these operators. Reverse the order of the commands and try again. What are the rules about when the commands will be executed?

14. What does the `xargs` command do? Can you combine it with `find` and `grep` to find yet another way of searching all files in the `/home` subdirectory tree for the word `hello`?
15. What does the `cut` command do? Can you use it together with `w` to produce a list of login names and CPU times corresponding to each active process? Can you now (all on the same command line) use `sort` and `head` or `tail` to find the user whose process is using the most CPU?