

Lunix Activities

1. Log on a Linux machine or connect to one from a Windows machine. Enter your login (user name) and password at relevant prompts.
2. Enter these commands at the Linux prompt, and try to interpret the output. Ask questions and don't be afraid to experiment (as a normal user you cannot do much harm):

- o echo hello world ←
- o passwd ←
- o date ←
- o hostname ←
- o arch ←
- o uname -a ←
- o dmesg | more ← (you may need to press q to quit)
- o uptime ←
- o who am i ←
- o who ←
- o id ←
- o last ←
- o finger ←
- o w ←
- o top ← (you may need to press q to quit)
- o echo \$SHELL ←
- o echo {con,pre}{sent,fer}{s,ed} ←
- o man "automatic door" ←
- o man ls ← (you may need to press q to quit)
- o man who ← (you may need to press q to quit)
- o who can tell me why i got divorced ←
- o lost ←
- o clear ←
- o cal 2000 ←
- o cal 9 1752 ← (do you notice anything unusual?)
- o bc -l ← (type quit ← or press Ctrl-d to quit)
- o echo 5+4 | bc -l ←
- o yes please ← (you may need to press Ctrl-c to quit)
- o time sleep 5 ←
- o history ←

2. Try the following command sequence:

- o cd
- o pwd
- o ls -al
- o cd .
- o pwd (where did that get you?)

- o `cd ..`
 - o `pwd`
 - o `ls -al`
 - o `cd ..`
 - o `pwd`
 - o `ls -al`
 - o `cd ..`
 - o `pwd` (what happens now)
 - o `cd /etc`
 - o `ls -al | more`
 - o `cat passwd`
 - o `cd -`
 - o `pwd`
3. Continue to explore the filesystem tree using `cd`, `ls`, `pwd` and `cat`. Look in `/bin`, `/usr/bin`, `/sbin`, `/tmp` and `/boot`. What do you see?
 4. Explore `/dev`. Can you identify what devices are available? Which are character-oriented and which are block-oriented? Can you identify your `tty` (terminal) device (typing `who am i` might help); who is the owner of your `tty` (use `ls -l`)?
 5. Explore `/proc`. Display the contents of the files `interrupts`, `devices`, `cpuinfo`, `meminfo` and `uptime` using `cat`. Can you see why we say `/proc` is a pseudo-filesystem which allows access to kernel data structures?
 6. Change to the home directory of another user directly, using `cd ~username`.
 7. Change back into your home directory.
 8. Make subdirectories called `work` and `play`.
 9. Delete the subdirectory called `work`.
 10. Copy the file `/etc/passwd` into your home directory.
 11. Move it into the subdirectory `play`.
 12. Change into subdirectory `play` and create a symbolic link called `terminal` that points to your `tty` device. What happens if you try to make a hard link to the `tty` device?
 13. What is the difference between listing the contents of directory `play` with `ls -l` and `ls -L`?
 14. Create a file called `hello.txt` that contains the words "hello world". Can you use "cp" using "terminal" as the source file to achieve the same effect?
 15. Copy `hello.txt` to `terminal`. What happens?
 16. Imagine you were working on a system and someone accidentally deleted the `ls` command (`/bin/ls`). How could you get a list of the files in the current directory? Try it.
 17. How would you create and then delete a file called "\$SHELL"? Try it.
 18. How would you create and then delete a file that begins with the symbol #? Try it.
 19. How would you create and then delete a file that begins with the symbol -? Try it.
 20. What is the output of the command: `echo {con,pre}{sent,fer}{s,ed}?`
Now, from your home directory, copy `/etc/passwd` and `/etc/group` into your home directory in one command given that you can only type `/etc` once.
 21. Still in your home directory, copy the entire directory `play` to a directory called `work`, preserving the symbolic link.
 22. Delete the `work` directory and its contents with one command. Accept no complaints or queries.

23. Change into a directory that does not belong to you and try to delete all the files (avoid `/proc` or `/dev`, just in case!)
24. Experiment with the options on the `ls` command. What do the `d`, `i`, `R` and `F` options do?
25. Describe three different ways of setting the permissions on a file or directory to `r--r--r--`. Create a file and see if this works.
26. Team up with a partner. Copy `/bin/sh` to your home directory. Type "`chmod +s sh`". Check the permissions on `sh` in the directory listing. Now ask your partner to change into your home directory and run the program `./sh`. Ask them to run the `id` command. What's happened? Your partner can type `exit` to return to their shell.
27. What would happen if the system administrator created a `sh` file in this way? Why is it sometimes necessary for a system administrator to use this feature using programs other than `sh`?
28. Delete `sh` from your home directory (or at least to do a `chmod -s sh`).
29. Modify the permissions on your home directory to make it completely private. Check that your partner can't access your directory. Now put the permissions back to how they were.
30. Type `umask 000` and then create a file called `world.txt` containing the words "hello world". Look at the permissions on the file. What's happened? Now type `umask 022` and create a file called `world2.txt`. When might this feature be useful?
31. Create a file called "hello.txt" in your home directory using the command `cat -u > hello.txt`. Ask your partner to change into your home directory and run `tail -f hello.txt`. Now type several lines into `hello.txt`. What appears on your partner's screen?
32. Use `find` to display the names of all files in the `/home` subdirectory tree. Can you do this without displaying errors for files you can't read?
33. Use `find` to display the names of all files in the system that are bigger than 1MB.
34. Use `find` and `file` to display all files in the `/home` subdirectory tree, as well as a guess at what sort of a file they are. Do this in two different ways.
35. Use `grep` to isolate the line in `/etc/passwd` that contains your login details.
36. Use `find` and `grep` and `sort` to display a sorted list of all files in the `/home` subdirectory tree that contain the word `hello` somewhere inside them.
37. Use `locate` to find all filenames that contain the word `emacs`. Can you combine this with `grep` to avoid displaying all filenames containing the word `lib`?
38. Create a file containing some lines that you think would match the regular expression: `(^[0-9]{1,5}[a-zA-Z]+$)|none` and some lines that you think would not match. Use `egrep` to see if your intuition is correct.
39. Archive the contents of your home directory (including any subdirectories) using `tar` and `cpio`. Compress the `tar` archive with `compress`, and the `cpio` archive with `gzip`. Now extract their contents.
40. On Linux systems, the file `/dev/urandom` is a constantly generated random stream of characters. Can you use this file with `od` to printout a random decimal number?
41. Type `mount` (with no parameters) and try to interpret the output.