Introduction to Unix

May 10, 2009

Exercise: Unix File System Hierarchy

Note: The "#" and "$" characters before commands represents your system prompt and is not part of the command itself. "#" indicates a command issued as root while "$" indicates a command issued as a normal user.

Be careful in this exercise. Running as root (the Unix administrative account) means that you can easily damage your system, so we ask that you log out of your root account and log in as your own user account instead.

To logout, type

```
# exit
```

If you are unsure of how to proceed ask your instructor or assistants for help before continuing.

The first command that we are going to use is *man*, this is short for "man"ual. Read about each command to see the range of options that exist.

Many of the basic commands we'll be practicing are built in as part of your shell environment (that is you won't find a binary/program file for *cd*). To read about commands like *cp, cd, ls, mv, rm* in more detail you can just type:

```
$ man builtin
```

And, for a command like *ls* you can type:

```
$ man ls
```

And, even for a built-in command you can just type "man commandName", or something like:

```
$ man cd
```

and this will open the "builtin" man page for you.

If you have problems exiting from "man" press the "q" key. Also, you can use the keyboard arrows to move around in the descriptions.

As we move around directories an extremely useful command is *pwd*, which return the working directory name you are in. So, if you get lost just type:

```
$ pwd
```

We'll do this from time to time as we use directory commands.

Simplified Map of Unix Directory Tree

```
/                  ("root directory")
/etc               (contains configuration files)
/etc/rc.d          (contains system startup scripts)
/root              (user root's home directory)
/tmp               (place to store TeMPorary files)
```

/usr	(contains the majority of user utilities, applications, home directories) **cont.==>**
/usr/home	(home directories for users on the system*)
/usr/local/etc	(contains third-party configuration files and startup scripts*)
/var	(multi-purpose log, temporary, transient, and spool files location)

*Different from Linux

For details on the (almost) complete directory tree under Unix/Lunx type "`man hier`"

Command Glossary

cd	Change Directory
ls	LiSt files
mkdir	MaKe DIRectory
mv	MoVe files
pwd	Print Working Directory
rm	ReMove files
rmdir	ReMove DIRectory
touch	Update date on file/Create new empty file if none exists

Note: There are some special files on UNIX, which are `'.'` and `'..'`:

- `'.'` - "this" directory (the one you are "in" as reported by the `pwd` command
- `'..'` - the parent directory, i.e.: the one above

Now we are ready to practice a bit with the commands:

```
$ cd /
$ pwd
$ ls
$ ls -la
$ cd /tmp
$ cd ..
$ pwd
$ cd tmp
```

What's going on here? If you don't understand, ask.

```
$ cd               (take you back to your home directory)
$ pwd
$ touch text.txt
$ cp text.txt new.txt
$ mv text.txt new.txt
```

What's happening now? If prompted to overwrite, respond "y". Note that "inst" is the name of the user account you created in the first exercise.

Now watch what happens if you try to copy a file on itself.

```
$ cp new.txt /home/inst/.
$ cd ../../home/inst
$ pwd               (where are you now?)
$ cd                (to return to our home directory)
$ cp new.txt new.txt.bak
```

The tab key makes life much easier. Now type:

```
$ cd
$ mkdir tmp
$ mv new.* tmp/.
$ ls
```

Finally, we are going to remove the directory that contains the two archives.

```
$ cd tmp
$ ls
$ rm *
$ cd ..
$ rmdir tmp
```

You can force this using a command like this:

```
$ rm -rf tmp
```

The use of "rm -rf" is **very dangerous!**, and, naturally, very useful. For example, if you are "root" and you type "rm -rf /*" this would be the end of your server. This commands says "remove, forcibly and recursively, everything" - Or, if you start in the root directory (/), remove all files and directories *without asking* on the entire server. If you want to use "rm -rf *" always take a deep breath and check where you are first (really, do this!):

```
$ pwd
```

First this says in what directory you are. If you are mistaken, then you have the opportunity to not remove files that you might really need.