

UNIX Stuff for 372
William H. Mitchell
Last Revised: February 7, 2015 at 3:45pm

I've talked to a few students that are having trouble dealing with some of the mechanics of UNIX. Here are a few things that may help.

My old 352 slides

A set of my 352 UNIX slides is in <http://cs.arizona.edu/~whm/352/unix.sli.pdf>. Those slides represent a basic set of things that I think are important/useful/handy to know when working with UNIX. Whether you've had 352 or not, I bet you'll learn at least a few things if you just flip through those slides. Additional 352 stuff is in that same directory.

Dead simple, low-tech backup

Here's a one-line, low-tech backup that you can run on `lectura`, in your `a2` directory:

```
$ pr *.hs | mail -s 372 your-netid@email.arizona.edu
```

The `pr` command writes the content of each of your `.hs` files in turn to standard output, and that output is piped into `mail`, a command line mailer. (Try `man mail`.) You'll get a message with the subject "372". For a more descriptive subject you could use `-s "372 a2 backup"` instead, but note that quotes are used in that case to get the spaces embedded in the string. `pr` generates some page headers that you'll have to hack out if you need to recover files, but your source code will all be there.

Like any backup procedure, try some recoveries for practice to possibly discover omissions or flaws in the procedure.

Package that low-tech backup in a script!

The easier it is to do a backup the more likely you are to do it.

Put that "pipeline" with `pr` above into a file. I like short names for commands I'm going to run frequently, so I'll suggest the name `bk`. Here's what that file should look like, as shown by `cat`:

```
$ cat bk
pr *.hs | mail -s "372 a2 backup" whm@email.arizona.edu
```

We could run it right now with `bash bk` but let's make it executable, to save a little typing:

```
$ chmod +x bk
```

We should now be able to run `bk` by simply typing `./bk`. Depending on how your path is configured you might be able to type just `bk`. (For more details on `bk` vs. `./bk`, search for `PATH` in my 352 slides and also see <http://cs.arizona.edu/~whm/352/Spring05/dotornot.pdf>.)

You're now able to ship off a copy of your `.hs` files with 3-5 keystrokes. If you want to send your backups to two locations, maybe a Yahoo mail account, too, just duplicate that line and change the email address, or see slide 88 in my UNIX set to learn about the `for` statement in `bash`.

Things like Dropbox and Google Drive give you instantaneous backup and that's great but I really hate to pay for the same real estate twice, so I supplement those with some low-tech backups. If Dropbox were to have a catastrophic failure, I'm sure they'd put out a well-written note expressing their regret and a rededication to ensuring data integrity but I doubt they'd be mailing any checks to help say how sorry they are that all your files are gone.

~/ .snapshot has periodic snapshots of all your files on lectura

Let's look at my ~/ .snapshot directory on lectura. We'll use `-1` (dash-one) to force single-column output:

```
$ ls -1 ~/.snapshot
hourly.0
hourly.1
hourly.2
...
nightly.0
nightly.1
...
weekly.0
weekly.1
...
```

Each of those is a directory that's a "copy" of my directory tree on lectura at those particular times. You can see that the frequency of retention decreases for older copies but the latest snapshots are four hours apart. If you want to see and/or recover a file from some point in the past, just `cd` into the appropriate snapshot directory and copy the file out. Example:

```
$ cd ~/.snapshot/hourly.2/372/a2
$ ls -1
[...lots...]
$ cat join.hs
--this version compiles!
x = 1
$ cp join.hs ~/372/a2/join-compiles.hs
```

See <http://faq.cs.arizona.edu/index.php?action=artikel&cat=5&id=58> and <http://www.cs.arizona.edu/computing/accounts/snapshots.html> for more details on ~/ .snapshot. See also slide 55 in my UNIX set.

What does tilde (~) mean in a path?

bash, like many shells, interprets a path like ~/xyz to mean "xyz in my home directory".

The `echo` command can be used to see what a command line looks like after the shell does its various expansions and substitutions on a command line. Here

```
$ echo ~/.ghci
/home/whm/.ghci
$ echo ~/372/a2/cpfx.hs
/home/whm/372/a2/cpfx.hs
```

Tilde substitution done only if the tilde is at the start of a line:

```
$ echo ~/x~x/y~
/home/whm/x~x/y~
```

Try `echo ~postgres/x`.

See slide 62+ in my UNIX set for more on tilde expansion.

Although unrelated to ~, here are some examples of exploring with `echo`:

```
$ echo $HOME
/home/whm
```

```

$ echo $PATH
/home/whm/sbin:/home/whm/3bin:/home/whm/bin:/usr/local/bin:...lots
more...
$ echo $RANDOM $RANDOM
2344 16797
$ echo *.hs
join-compiles.hs join.hs x.hs

```

See slide 82+ in my UNIX set for more on variables (\$...); 64+ covers wildcards.

A little more with symlinks

The a2 writeup recommends creating this symlink in your assignment 2 directory,

```
$ ln -s /cs/www/classes/cs372/spring15/a2 .
```

so you can then run the tester with `a2/tester`. Here's a symlink that lets you run the tester with `./t` (or maybe just `t`—see `dotornot.pdf` above.)

```
$ ln -s a2/tester t
```

That creates a symlink named `t` that uses the `a2` symlink.

For future assignments you might do something more general, like this for `a3`:

```

$ cd ~/372
$ ln -s /cs/www/classes/cs372/spring15 www
$ cd a3
$ ln -s ../www/a3 a3
$ ln -s a3/tester t

```

That creates a symlink `www` in `~/372` that references the root directory for the 372 materials parked on the web. Then, in `a3`, you make a symlink named `a3` that uses the `~/372/www` symlink, saving the trouble of having to type out `/cs/www/...`. In turn, the symlink `t` uses the `a3` symlink—triple indirection!

See slide 58+ in my UNIX set for more on symlinks.

In WinSCP use **Commands > Keep Remote Directory up to Date...** (!)

If I had a dollar for every time I've seen a student drag a file between WinSCP windows to copy the latest from their machine to `lectura`, I might be able to pay for a class trip to Hawaii for Spring Break.

Instead of that per-save dragging, just get the source and target directories open in WinSCP and do **Commands > Keep Remote Directory up to Date...** It'll ask if you want to perform full synchronization of the remote directory first. You do, unless you've been editing some files directly on `lectura`. (If that's the case, just get the latest, greatest versions back to your Windows machine before you activate this automatic synchronization.)

WinSCP has good help about this facility.

Cyberduck is a WinSCP equivalent for the Mac that I've heard good things about but have never experimented with.

Remote editing

I typically use Emacs' remote editing facilities for editing files on `lectura` from my Mac.

Students last Spring seemed to have good luck using http://wbond.net/sublime_packages/sftp for remote editing with Sublime. I've got a dim memory of some remote editing package for Sublime having a race condition that

would occasionally result in another user getting your code when they tried to open their file but I can't find any trace of that of that incident now. If you should see anything remotely like that (no pun intended!), let me know ASAP.

If you'd rather type just `lec` instead of `lectura.cs.arizona.edu`...

On OS X, Linux and Windows, host names are looked up in `/etc/hosts` before consulting DNS. This lets you add a short name for a host, like `lec` for `lectura.cs.arizona.edu`. Here's the line for `lectura` from my `/etc/hosts`:

```
192.12.69.186  lec
```

It'd be an unusual situation for the IP address for `lectura` to be changed but if it ever should, you'd need to update that `/etc/hosts` entry.

I use `sudo vi /etc/hosts` to edit that file on my Mac but you can use any editor.

Windows has a corresponding file but it's a longer story. Here's a link for what looks to be a pretty good how-to: http://www.rackspace.com/knowledge_center/article/how-do-i-modify-my-hosts-file

Once you've done this, just about everywhere you used to type `lectura.cs.arizona.edu` you can type just `lec` instead. That works in PuTTY, WinSCP, Emacs, on the command line, and lots more.

If you're doing web development, a handy addition to the `127.0.0.1` entry is just an "l" (L).

```
127.0.0.1      localhost l
```

With that you can hit `l://...` instead of `localhost://...`