

Notes on `turnin`
whm, August 30, 2006

CSc 372 students please note: This material was written with CSc 352 in mind. The examples below reference the assignment tag `352_1` and file names associated with a first assignment in 352. For the first 372 assignment you'll be using a tag of `372_1` and files like `ftypes.sml`, `to_b.sml`, etc., as specified in the assignment write-up.

`turnin` is a `lectura` command-line utility that copies files from a student's directory into a directory that is accessible to the instructor. The login name of the student submitting the files and the time of submission is recorded.

The first step in using `turnin` is getting the files of interest onto `lectura`. Some ssh clients have an integrated file transfer mechanism. An allegedly secure command-line alternative for Windows is `pscp`, part of the PuTTY package at <http://www.chiark.greenend.org.uk/~sgtatham/putty>.

Here's how I might use `pscp` to copy a couple of Java files from the current directory on a Windows machine to my home directory (`/home/whm`) on `lectura`:

```
> pscp lengths.java mgrep.java whm@lectura.cs.arizona.edu:.
whm@lectura.cs.arizona.edu's password:
lengths.java |      0 kB |    0.4 kB/s | ETA: 00:00:00 | 100%
mgrep.java   |      0 kB |    0.3 kB/s | ETA: 00:00:00 | 100%
>
```

Note that it's the dot at the end of the `pscp` command line that specifies my home directory on `lectura`. Using `whm@lectura.cs.arizona.edu:372` would cause the files to be copied to `/home/whm/372` instead.

Once the files are on `lectura`, run `turnin` to submit them for grading. The general form of `turnin` is this:

```
turnin assignment-tag file1 file2 ... fileN
```

The tag for the first assignment is `352_1`. To turn in your Java programs you'd do this:

```
turnin 352_1 lengths.java mgrep.java eval.java
```

After finishing `bindings.el`, you might then turn it in:

```
turnin 352_1 bindings.el
```

A file that has already been turned in can be overwritten with a newer version by simply turning it in again.

Instead of turning in files a few at a time you might wait until everything is done and name all the files in one long `turnin` command. After we talk about *command substitution*, you'll learn that you can turn in everything like this:

```
turnin 352_1 $(cat /home/cs352/spring05/a1/deliverables)
```

BE SURE that the names of your solutions exactly match the names specified in the assignment. For example, if you use the name `Lengths.java` instead of `lengths.java`, you might end up with a zero for that problem.

The `-ls` option of `turnin` can be used to see what you've currently got turned in, and when it was turned in:

```
$ turnin -ls 352_1
.:
total 56
-rwxrwx--- 1 whm cs352 59 Jan 27 02:24 active
-rw-rw---- 1 whm cs352 159 Jan 27 02:24 bindings.el
-rwxrwx--- 1 whm cs352 84 Jan 27 02:24 deepest
-rw-rw---- 1 whm cs352 801 Jan 27 02:24 eval.java
-rwxrwx--- 1 whm cs352 66 Jan 27 02:24 group
-rw-rw---- 1 whm cs352 496 Jan 27 02:24 lengths.java
-rw-rw---- 1 whm cs352 1820 Jan 27 02:24 mgrep.java
-rwxrwx--- 1 whm cs352 86 Jan 27 02:24 second
-rwxrwx--- 1 whm cs352 27 Jan 27 02:24 trimls
$
```

Make a habit of using `turnin -ls` to double-check that all your solutions have been submitted.

The `-rm` option of `turnin` deletes all your submissions for the assignment:

```
$ turnin -rm 352_1
$ turnin -ls 352_1
$
```

You might use `-rm` to clean things up and start fresh.

To see the `turnin` man page, issue the command `man turnin`.

A `turnin` pitfall

An important thing to understand about `turnin` is that a later submission overwrites an earlier submission of a given file. There is no evidence whatsoever of that earlier submission—it's just gone.

Consider this scenario: Two minutes before the deadline, a student turns in `mgrep.java` and `eval.java`:

```
$ turnin 352_1 eval.java mgrep.java
Turning in:
  eval.java -- ok
  mgrep.java -- ok
All done.
$ turnin -ls 352_1
-rw-rw----  1 whm  cs352           801 Feb  3 23:58 eval.java
-rw-rw----  1 whm  cs352        1820 Feb  3 25:58 mgrep.java
```

A minute past the deadline the student notices that he accidentally inserted a stray character in `eval.java` and that it won't compile. He decides to fix it, turn it in, and hope that the late submission will be accepted. To save time, he uses `C-p` to recall the last `turnin` command and he executes it:

```
$ turnin 352_1 eval.java mgrep.java
Turning in:
  eval.java -- ok
  mgrep.java -- ok
All done.
```

Unfortunately, the reused `turnin` command specified `mgrep.java`, too. The on-time copy of `mgrep.java` is overwritten and now, `mgrep.java` is late, too:

```
$ turnin -ls 352_1
-rw-rw----  1 whm  cs352           801 Feb  4 00:01 eval.java
-rw-rw----  1 whm  cs352        1820 Feb  4 00:01 mgrep.java
```

The moral of the story is this: **If for some reason you turnin a file after a deadline, be sure to not specify extra files that overwrite existing, on-time copies.** Note that turning in a directory overwrites any existing copies of files in that directory. If for some reason you're inclined to turn in a copy of a file after the deadline—perhaps you think you have grounds for an extension—turn in a differently-named copy of the file (or directory) and send mail explaining the situation.