

Revision Control Systems

CIS320 Individual Project: Skills Session

Monday 12th October 2009 12:00—13:00

1 Using a lab machine

On the lab machines, all the necessary software installation and configuration has been already done. Your home directory on igor has been mounted as the G: drive; darcs has been installed, and the Cygwin GNU tools for Windows are installed and configured. Here is a step-by-step series of instructions for getting started with the software

- 1.1. Start the Cygwin shell: double-click the Cygwin icon on the desktop. A terminal window should open.
- 1.2. We'll need a directory to experiment in. You could use the Explorer shell to create that directory, but you will need to use Cygwin for the following steps anyway, so type

```
mkdir /cygdrive/g/darcstest
```

followed by the Return key into the Cygwin window. This should create a `darcstest` directory in your G: drive: check that you can see it in Explorer.

- 1.3. The Cygwin shell has the notion of the current working directory, which is selected using the 'cd' command. Change the working directory to the directory you just created by entering

```
cd /cygdrive/g/darcstest
```

followed by the Return key into the Cygwin window. The command prompt should change to reflect the new working directory.

- 1.4. To get started with darcs, we need to initialize a repository. At the Cygwin command prompt, type

```
darcs initialize
```

This should create a `_darcs` subdirectory in the `darcstest` directory: check that you can see it. You should usually not worry about the contents of this directory; however, you should feel free to inspect the contents, and how they change, as you go along with the rest of this labsheet.

- 1.5. At the moment, the darcs repository is tracking no files. To add a file to its database, first create a text file (or source code or similar) within the `darcstest` directory, as you would normally, by copying a file from elsewhere or from scratch. Then enter at the Cygwin prompt

```
darcs add filename
```

replacing *filename* with the name of your file, and hit Return. This informs darcs that you wish to track the file, but does not yet confirm that the file that is there is the one you want to keep. To do that, enter

```
darcs record
```

at the prompt. Darcs will ask you questions about the various changes it can see to the working copy: in this case, the addition of a file, and the creation of its contents. Give your e-mail address when prompted (you only need to do this once per repository), answer 'y' to the questions about recording changes, and when prompted for a name for this patch, give a name such as 'initial commit'. Answer 'n' to the question about adding a long comment. At this point, you should have successfully completed the recording of your first patch.

- 1.6. You might well wish to modify your file: maybe to fix a bug, or to add a new feature (if it's source code); to extend a description (if it's a report, or documentation). Make a change to the file you've added, save the new version, then type

```
darcs whatsnew
```

at the prompt. You should see a summary of the changes that you have made. To record them, type

```
darcs record
```

again, and answer the questions asked; name your patch something like 'first changes'. This should record your second patch; you can see your work so far in this repository with

```
darcs changes
```

and `darcs whatsnew` should now tell you that there are no unrecorded changes in the working directory.

- 1.7. You might at some point decide that some experimental modifications to your files are not going to work, and that you want to go back to the last recorded copy in your repository. Make another change to the file you've added, and save the file; then type

```
darcs whatsnew
```

to view the changes, and

```
darcs revert
```

to make the contents of the file in the working directory be the version before those changes. (You may need to reopen the file in whatever editor you are using to see the changed content).

- 1.8. It is possible to generate a complete copy of a repository, including all the metadata and history. You may wish to do this, for example, for backup purposes, or to work on different machines, for example when you are not at Goldsmiths. To do this, first change the current working directory by simply typing

```
cd ~/Desktop
```

followed by Return at the Cygwin prompt. Then, by typing

```
darcs get /cygdrive/g/darcstest
```

a copy of the `darcstest` repository will be created within your current directory. Change to that new directory with

```
cd darcstest
```

and view the changes with

```
darcs changes
```

- 1.9. You can also get copies of repositories over http. For example, to get the repository containing the source files for this skills session, you can issue the commands

```
cd ~/Desktop
```

```
darcs get http://doc.gold.ac.uk/~mas01cr/teaching/cis320/2009-10/revision-control
```

which should produce a `revision-control` directory. Use the `darcs changes` command to inspect the repository that that produces. Visit the same URL in a web browser; what can you see?

2 Working on your laptop

In order to work with darcs on your laptop, it's likely that you will need to install some software and to configure your machine appropriately.

2.1 Software Installation

The only software you should need to install is darcs itself. Darcs is a console program; while Cygwin is a convenient console interface, you can run darcs from `cmd.exe`. Whichever you choose to use, it will be easier if the path to the darcs executable is in your `PATH` shell variable. If you are using Mac OS X or Linux, you should be able to install darcs to a suitable path using your system's package manager.

2.2 Machine configuration

You should be able to connect to the eduroam wireless network within Goldsmiths. When connected to that network, you should also be able to mount your homespace on igor, by connecting to the `\\igor\homes` SMB share.

2.3 Pulling and pushing patches

When you are connected to the eduroam network and have `\\igor\homes` mounted, you should be able to work directly with repositories on igor as though they were local, exactly as in section 1 above. Darcs also supports pulling (but not pushing) patches over http; when you are away from college, you can access repositories that are in the `public_html` subdirectory of your homespace, by using darcs commands such as `get` or `pull` with the URL of your subdirectory as an argument.