Subversion access

CS 161

Fall 2006

Edits

Thursday, 10/5: Fixed path, replacing `cs161-fa06` with `cs161/fa06`.

1 Setup

Each project group has a group account that exists only to host the group's subversion repository. The account is named `cs161-gN`, where `N` is the group number. We'll use group 0 as a running example; wherever you see 0 as a group number below, replace it with you group's number (one-digit groups are just one character, without a leading zero).

The repository's base is in a subdirectory of the group's home directory called `SVN`. The full path to the repository is

```
/home/cc/cs161/fa06/group/cs161-g0/SVN/
```

Don't forget to replace the zero with your group's number. (See section Section 3 for more on repository URLs.)

2 Subversion basics

You should be familiar with the concept of version control; if not, please ask your partners or come to your TA's office hours to get up to speed. The basic idea is that there is a central repository that keeps the current copy of all files, as well as a full history of changes, and each person working on the project will check out a local copy of the repository. Changes made locally can be checked in to the main repository, and then others will see those changes when they update their local copy.

There is an excellent manual documenting subversion that you can find here: `http://svnbook.red-bean.com/`. Note that the instructional machines have version 1.2 installed, so you'll probably want to look at that version of the manual.

Subversion, or svn, is a cvs replacement that very nearly duplicates cvs's functionality but adds some nice features. Like cvs, it does not require you to

lock a file before editing it, but instead merges different changes when two (or more) people change the same file at the same time.

The basic commands are almost identical to those of cvs. (See Section 3 for what to use as `<url>`.) You can get usage and options with the command `svn help <command>`.

Checkout `svn checkout <url>`, or `svn co <url>`, will check out a copy of the directory specified by `<url>` into the current directory.

Import `svn import <url>` will put the current directory (and all directories/files recursively) into the repository at `<url>`. This directory should not already exist in the repository. Two things to note: first, this command does commit to the repository (unlike `svn add`, which requires a commit afterwards), and second, the svn makes no changes to the current directory (such as adding repository metadata), so you will need to check out the repository in a new location before working on the files.

Update `svn update`, or `svn up`, will bring your local copy up-to-date with the most recent version of the repository, recursively from the current directory. Local changes are preserved, and files are merged if necessary. (There may be a conflict, which will have to be resolved by hand.)

Status `svn stat` is a nice feature of svn that cvs does not have: it will tell you what you have changed locally without updating from the repository.

Add `svn add <file>` will add a file or directory (plus files/directories recursively) to the repository.

Commit `svn commit` will upload your local changes to the repository. Others won't see your changes until you commit. Note that you'll usually have to do an update first to make sure you're up to date.

Diff `svn diff <file>` will show you the changes you've made to `<file>` since updating it.

For more details, use `svn help` or see the manual referenced above. You'll want to look up the procedure for resolving conflicts, as you'll probably have to deal with that at some point.

3 Repository access

Each group member should send their public key to your TA, as described in the project 1 handout. Once you do that, you should have access to the repository using that key. (If you wish to access the repository from another computer/account as well, send another key to your TA for that account; be sure to tell your TA what machine/account it's for.)

You will not have shell access; the only access you have to the group account is the subversion-over-ssh protocol. The svn commands should all work perfectly

if you form the URL correctly. The correct form for group 0's base repository is:

```
svn+ssh://cs161-g0@cory.eecs.berkeley.edu/home/cc/cs161/fa06/group/cs161-g0/SVN/
```

The first field after the protocol is `<groupname>@<host>`; you can use pulsar or quasar or any other host if you prefer. This is followed by the repository's directory, which is the same for all groups except for the group account's name, which will have a different number for each group. I'll use abbreviations `cory` for `cory.eecs.berkeley.edu` and `<path>` for `home/cc/cs161/fa06/group/cs161-g0` below so the URLs here are more readable.

When you create directories in your repository, they will be appended to the end of the path. One directory has been created for you:

```
svn+ssh://cs161-g0@cory/<path>/SVN/project
```

You may put your code, design doc, etc. under this directory (or you may ignore it if you wish and use another naming scheme under the base URL).

There are two ways to get your code initially into the repository. First, you can use `svn import`. To do this, `cd` to the top directory of your code tree (for example, the 'project1' directory you get from untarring the skeleton code) and type:

```
svn import svn+ssh://cs161-g0@cory/<path>/SVN/project/code
```

(Of course, replace `cory`, `<path>`, and the group number with the correct values.) This will check in the current directory (and subdirectories recursively) at the location `project/code` under the base repository. The current directory is not modified to be a local copy of the repository, so you'll have to then go elsewhere and check out a local copy with

```
svn co svn+ssh://cs161-g0@cory/<path>/SVN/project
```

Now there will be a directory called `project` in the directory where you ran this command, which is a local copy of your repository that you can work on.

The second way is to first check out the `project` directory, then copy your code directory into it and do an `svn add`. For example,

```
svn co svn+ssh://cs161-g0@cory/<path>/SVN/project
cd project
cp -r ~/wherever-I-untarred/project1 code
svn add code
```

In this case, the `code` directory is added to the local copy of the repository, so you don't have to checkout again.

This should give you enough to get started. Please look through the documentation for more information, or contact your TA if you have questions.