SVN Primer

Adam D. Myers

Introduction

SVN[1] is a collaborative tool where multiple people can work on the same codes or documents. Sort of like Google Docs or GitHub, except that I have more control over it for class. The development of tools using a common user repository is a feature of modern work in large collaborations—it is partly how large surveys such as the SDSS have been able to be so successful.

In this class, all work will be submitted[2] through SVN. For instance, homeworks will be issued before the final lecture of a given week and will be due by uploading code and/or documents to our SVN repository two weeks later on Saturday by 4PM.

Every document or piece of code in the class SVN repository will be available for everyone else in the class to use and to edit[3]. If you are worried about other students copying your homework (and other) submissions, note that a simple "diff" in UNIX, and/or logging via "svn diff", will make it obvious to me if another student has directly copied your submission. In fact, frequently uploading your code to the SVN repository as you write and develop it will make it far harder for your work to be copied than uploading it in a single submission. It is always possible to return an SVN repository to an earlier state, so any unwanted changes can be easily redacted.

But, what is an SVN repository? Basically, it's just a directory to which multiple people have access. Our directory—our SVN repository—sits on my home directory on zulu and is called "Myerszulutrunk/ASTR5160". The directory is special in that it tracks how it changes with time, and logs information on who has made the changes. That way it's possible for any user who has a mirror of the repository to track changes in the repository and to use anything in the repository.

Getting Started

First, check out the repository. Create a directory and change into it (or if you're completely new to UNIX use your home directory). Then, issue this command, which is a checkout of our entire repository:

```
svn co svn+ssh://username@zulu.uwyo.edu/d/users/admyers/Myerszulutrunk/ASTR5160
```

This command checks out (`co`) everything in the ASTR5160 repository and stores it on your local machine under the name ASTR5160. If you look in the directory `ASTR5160/myers/week1` you will even see the original of this document (`svnprimer.pdf`)!

[1] short for *subversion*

[2] Unfortunately, work won't be *graded* through SVN, as FERPA frowns on students seeing each other's grades

[3] including this one, if you'd like to edit it

Common Commands

```
svn up
```

This command updates your local directory to mirror the current copy of the repository. Issue this command *often*. Issue it *after checking out the full repository*. Issue it *before committing any changes to the repository*.

```
svn ci mylocalfile -m "this is what I did"
```

This command commits (`ci`) a file you are working on to the repository. It will be committed to the directory in the repository that mirrors the directory that you are in locally. The `-m` switch commits a comment that will be logged by SVN. *Always include a comment.*

```
svn add anewfileordirectory
```

If the file that you wish to commit doesn't exist in the repository yet (i.e. it's the first time that you've committed it) then you will first have to add it using `svn add`.

```
svn log
```

This command lists all of the changes to the repository. To see recent changes pipe it to `more` at the command line (e.g., `svn log | more`). To see changes somebody specific made, `grep` that person's username at the command line (e.g., `svn log | grep myers`)

```
svn ls
```

This command lists what is in the repository. This is a useful command as you may expect to find a file in the SVN repository when, in fact, you forgot to add or commit that file. This command, then, will allow you to see what is actually in the repository as compared to what is in your local directory.

```
svn diff -r 1:2
```

This command shows the differences in the repository between two changes. In this case, between version 1 and version 2 of the repository. This allows you to track changes that people have made. One use, e.g., would be to see who has added what to a TeX document since you went to bed last night.

If you find yourself using many other different commands, feel free to add them to this document.

The Structure of the ASTR5160 Class Repository

Beneath the parent directory (generically called *the trunk*, and called `Myerzulutrunk/ASTR5160` in our case) I have put a directory

```
ASTR5160/myers
```

This directory is divided into weeks of the course (`week1`, `week2` etc.) I will put any useful PDF and Powerpoint notes (such as this one) into these directories for the relevant week. The same documents will be linked from the course website. You should similarly create a directory that is your surname, and beneath it you will create directories for each week of the course (`week1`, `week2` etc.) in which you will post your homework submissions

```
ASTR5160/yoursurname
ASTR5160/yoursurname/week1
```

Good Practice with SVN

There are various examples of good SVN etiquette to help people share documents and code:

1. always `svn up` before committing anything new to the directory. This way, if somebody else commits something new just before you, then you won't lose track of which version of the directory you're working with.
2. SVN is for storing code and documents, *not large data files*. Large data files will make the repository slow to operate. If you have a large data file, keep it local to your machine and share it with people in other ways.
3. SVN requires good coding practices. *Place comments within the body of your code* (and other people's code) carefully using your initials. So, my code will have many comment lines of the form, e.g., `;;;ADM this line of code does this`. When writing documents collaboratively, make similar comments if you make major changes to the document.
4. when you commit a new document using the `svn ci` command, *always* provide a comment as in `svn ci` *`mylocalfile`* `-m` *`"this is what I did"`*.

Class Rules for SVN

Do not abuse the collaborative power of SVN to plagiarize other student's work. You will learn nothing by copying other people in full. In particular:

1. Do not edit code written by another member of the class without their permission. *Editing other people's code that is placed in any directory that contains their name (i.e.* `ASTR5160/their name/weekx/somecode`*) will be considered grounds for failing the course.*
2. It is permissible to read and to make a copy of any member of the class's code *after* it has been graded as a homework submission...so, in week 2, it will be permissible to raid people's `week1` directory (etc.). But, use other people's code by making a copy of it in your personal directory or linking to it in full. *Do not edit it in their directory. Editing other people's code that is placed in any directory that contains their name (i.e.* `theirname/weekx/somecode`*) will be considered grounds for failing the course.*
3. Provide your own homework solutions. Do not copy each other's work. It is *very* easy for me to check in SVN whether your homework submission greatly resembles another student's submission. *Plagiarizing each other's homework submissions will be considered grounds for failing the course.* Feel free to discuss homework problems and issues but *write your own submissions sitting by yourself.*