CS301 Compiler Design and Implementation

Prof. Lyn Turbak

Wellesley College

Handout # 7

September 6, 2003

Concurrent Versions System (CVS)

1   Overview

The CS301 filesystem is under control of a version management system named CVS (Concurrent Versions System). CVS is a standard way to control versions of files in software projects.

CVS has some very sophisticated features for allowing multiple users to work on a shared set of files without stepping on each others toes. In CS301, we will be using a very small subset of these features to distribute program files and bug fixes. This will make it easier for me to to post code and bug fixes, and will make it easier for you to download code for your problem sets and to install bug fixes. You should never have to copy files from the ~cs301 directories on puma; instead you will use CVS commands to install the files you need (see Section ?? for details).

2   Installing Your Local CVS-controlled CS301 Filesystem

Puma and all the linux cluster machines have been configured to run CVS. In order to use CVS, you will have to carefully follow the steps below.

1. Set the CVSROOT variable in your shell. If you are configured to run the bash shell, add the following lines to your ~/.bashrc file:

   ```
   export CVSROOT=:local:/usr/network/cs301-cvsroot
   ```

   If you run the csh shell, add the following to your ~/.cshrc file.

   ```
   setenv CVSROOT :local:/usr/network/cs301-cvsroot
   ```

   After this step, log out of your session and log back in so that the CVSROOT variable is set correctly.

2. If you already have a directory named ~/cs301, name it to something else. E.g., execute mv ~/cs301 ~/cs301-old.

3. The CVS-controlled CS301 directories will be installed in a new directory named ~/cs301 within your filesystem. (The installation will automatically create this directory; you should not create it by hand.) You can install the CVS-controlled CS301 directories by executing the following commands in the Linux shell:

   ```
   cd ~
   cvs checkout -d cs301 archive/cs301_fall03/download
   ```

Executing the `cvs checkout` command will create a `~/cs301` directory and will populate it with several subdirectories. As these subdirectories and the files they contain are installed, feedback will be provided on the console as to what is being installed.

When the checkout process finishes, the directory `~/cs301` will now contain your local copy of the CVS-controlled CS301 filesystem. Each CVS-controlled directory in this filesystem contains a subdirectory named `CVS` that contains important administrative information for CVS. **YOU SHOULD NEVER DELETE OR MODIFY ANY DIRECTORY NAMED CVS, NOR THE CONTENTS OF SUCH A DIRECTORY.**

3   Using the CVS-Controlled CS301 Filesystem

Here's the really cool part. Whenever you begin to work on an assignment on any given day, first execute the following commands in a Linux shell:

```
cd ~/cs301
cvs update -d
```

This will automatically update your local copies of the `cs301` filesystem with any modifications, new files, and new directories from the repository. In particular, any files that have been posted since you last worked will be automatically added to your local `cs301` filesystem, and any bug fixes will be automatically installed!

CVS recognizes when you have locally modified one of the CVS-controlled files, and will do its best to merge any modifications from the repository with your local modifications. It usually does a good job. Every once in a while it can't figure out how to merge; see Section **??** below for how to deal with conflicts.

When you do the update, CVS will tell you what it did to each file. It prints out a single letter before each filename that tells the action it performed on that file. Here's what the most common letters mean:

- `U`: This is a new file that CVS installed as part of the update.

- `P` : CVS has installed a patch in a previously installed file (which you might or might not have modified since it was installed).

- `M` : You have modified the file since it was installed, but CVS did not perform any action on this update.

- `C` : An unresolved conflict has been detected between modifications to the repository and your local modifications. See Section **??** below.

- `?` : The file is not under CVS control. (That's OK! You want to have local files that CVS doesn't know about.)

4 Conflicts

When performing `cvs update -d` gives you a conflict (denoted by the letter `C`), it means that CVS could not figure out how to merge modifications to the repository with your local modifications. In whatever spots it can't figure out what to do, it inserts text that looks like:

```
<<<<<<< driver.c
 your local text
=======
 text from the repository
>>>>>>>
```

In this case, you need to manually edit the text between the `<<<<<<<` and `>>>>>>>` to be what is correct. For more details about conflict resolution, consult the online documentation described below. Thankfully, conflicts are rare, and you rarely have to worry about them.

5 Documentation

For more documenation about CVS, consult the on-line manual at

`http://www.cvshome.org/docs/manual/index.html.`

(This is linked from the CS301 home page.)

Note that you are only using a very small subset of the features in the manual. The only CVS commands you should need are:

- `cvs checkout` *You shouldn't need this after the one-time installation.*

- `cvs update`

- `cvs status` *Tells you the status of each CVS-controlled file.*

In particular, you should **never** *try to use any of the following commands in CS301:*

- `cvs add`

- `cvs commit`

- `cvs import`