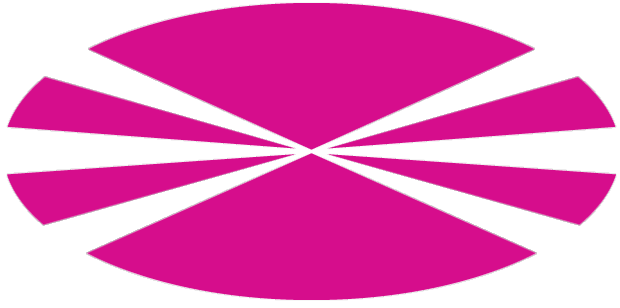
Assignment submission with SVN

Operating Systems

Daniel Valcarce <daniel.valcarce@udc.es>

University of A Coruña



What is a Version Control System?

- Version Control Systems (VCS) manage the changes of compute files.
- It answers the following questions:
  - What changed?
  - Who changed it?
  - When it changed?
  - How it changed?
  - Why it changed?
- Mostly used in software development.
- We avoid creating files such as: `final_assignment.c`, `final_assignment2.c`, `truly_definitive_assignment35.c`, `believe_me_this_is_the_final_one563.c` and so on.

Version Control Systems (VCS)

- VCS are based on repositories that store different versions of your source code.
- Repositories can be either centralised or distributed.
  - Centralised: **SVN**, CVS.
  - Distributed: Git, Mercurial, Bazaar.
- We can revert changes to any previous version.
- We can analyse the history of changes.
- We can compare different versions.

Subversion (SVN)

- We will use Subversion (SVN).
  - Popular and well supported.
  - Based on a centralised repository.
- Official website: `https://subversion.apache.org`
- Official book: `http://svnbook.red-bean.com`
- Although distributed version control has gained widespread adoption (in particular Git), SVN comes in handy for simple tasks.
- In our case, we will use SVN to submit assignments.

SVN commands

Some useful subversion commands:

- `svn checkout <url>`: get a working copy from the repo.
- `svn add <file>`: put file under version control.
- `svn rm <file>`: remove file from version control.
- `svn status`: print the status of the working copy.
- `svn commit`: send locally modified files to the repository.
- `svn update`: update all files with the latest changes.
- `svn revert <file>`: undo local changes of the given file.
- `svn help <command>`: describes the usage of the given command.

Assignment submission

- We provide you with a repository for this course: `https://svn.fic.udc.es/grao2/so/<course_year>/<user>`
- Only one of the members of the group must submit the source code to her/his provided repository.
- For each assignment, you must create the appropriate folder in the repository. The name of the folder must be `PX` where `X` is the number of the assignment. Be careful: **code in the wrong directory will not be graded**.

Submitting your assignment

- First, you must download a working copy of your repository:

  ```
  $ svn checkout
  https://svn.fic.udc.es/grao2/so/<course_year>/<user>
  ```

  This will create a folder named <user> which is your working copy.
- Move your folder PX inside the working copy and put it under version control. **EXAMPLE:** (for a user named Donal Trum, whose login is *d.trum* and tries to submit lab assigment number 1. In his machine, his username is *patito*)

  ```
  $ mv /home/patito/SegundoCurso/SO/P1 /home/patito/d.trum
  $ cd /home/patito/d.trum
  $ svn add P1
  ```

Submitting your assignment

- Now, you must commit your changes in the working copy to the central repository. You need to write a comment for each commit describing the changes you have made. There are two options:
  - Write the comment as an argument of the `svn commit` command:
    ```
    $ svn commit -m "Write here your message"
    ```
  - Or write the comment in your favourite text editor. You can specify which text editor (vim, emacs, nano, gedit...) you want by modifying the `$EDITOR` environment variable.
    ```
    $ EDITOR=nano svn commit
    ```

Happy coding!