

VERSION CONTROL USING MERCURIAL

KONRAD HINSEN

CENTRE DE BIOPHYSIQUE MOLÉCULAIRE (ORLÉANS)

AND

SYNCHROTRON SOLEIL (ST AUBIN)

"FINAL".doc



FINAL.doc!



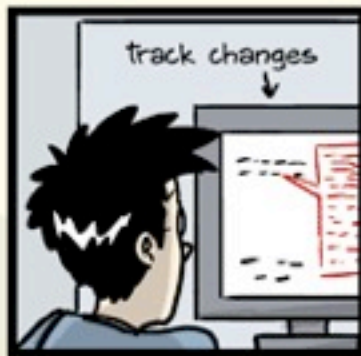
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc



A TIME AXIS FOR FILE CONTENTS

- ▷ Many computer files represent work in progress.
Examples: program source code, scientific publications, theses, ...
- ▷ It is desirable to be able to go back to earlier versions, or identify what has been changed when and by whom.
- ▷ Traceability is particularly important for collaborative work due to the risk of conflicting changes.

A version control system stores consecutive *versions* of each file with

- a time stamp
- the author of the most recent change
- a description of the change provided by its author

VERSION CONTROL IN PRACTICE

- ✻ A *project* consists of several files and directories that belong together. Typically, a project equals a directory.
- ✻ A *repository* is a directory under version control. When it is *initialized*, the version control system creates a database for bookkeeping.
- ✻ When a new version is recorded, a new *changeset* is added to the database.
- ✻ The database is stored in an invisible directory inside the repository (Mercurial: `.hg`). Don't delete it or modify its contents!

POPULAR VERSION CONTROL SYSTEMS

Subversion (SVN): <http://subversion.tigris.org/>

Mercurial: <http://www.selenic.com/mercurial/>

GIT: <http://git-scm.com/>

Bazaar: <http://bazaar-vcs.org/>

Darcs: <http://darcs.net/>

MERCURIAL

Distributed version control system written mostly in Python. Used from the terminal, the command name is **hg**.

Used by several well-known projects:

- Python
- Mozilla
- OpenJDK (Java)
- OpenSolaris

Download from: <http://mercurial.selenic.com/>

Extensive documentation: <http://hgbook.red-bean.com/>

GUIs and IDE integration:

<http://mercurial.selenic.com/wiki/OtherTools>

MINIMAL SETUP (UNIX/MACOS)

Create configuration file

`$HOME/.hgrc` (Unix)

`mercurial.ini` (Windows)

```
[ui]
```

```
username = Konrad Hinsen <konrad.hinsen@cnrs-orleans.fr>
```

```
ignore = ~/.hgignore
```

Create file `$HOME/.hgignore`

```
syntax: glob
```

```
** .o          compiled C/C++ files
```

```
** .pyc        compiled Python modules
```

```
** .pyo        optimized compiled Python modules
```


BASIC COMMANDS

`hg init <directory>`

Initialize a repository.

`hg status`

Show uncommitted changes/additions/removals.

`hg add <file1> <file2> ...`

Add new files to the list of version-controlled files.

`hg commit <file1> <file2> ...`

Commit changes to the listed files.

`hg log`

Show summary of revisions.

`hg serve`

Start Web server for repository inspection.

EXERCISES

- 1) Download a small repository by typing:

```
hg clone http://selenic.com/repo/hello
```

Explore what is in there, modify a file, add a file, and check that all operations have been run successfully.

- 2) Download a real-life repository (pylint, a code analyzer for Python):

```
hg clone http://bitbucket.org/logilab/pylint
```

Then start a Web server:

```
cd pylint
```

```
hg serve
```

and point a browser to <http://localhost:8000/> to explore the contents.

ADVANCED TOPICS

NOT COVERED TODAY, BUT
WORTH KNOWING ABOUT

TEAM DEVELOPMENT

- ☼ People working on a common project need a way to share or exchange modifications to the project's files.
- ☼ Two approaches:
 - A *master repository* is hosted on a Web server (private or public). Project members *push* changes to the servers and *pull* the changes made by others.

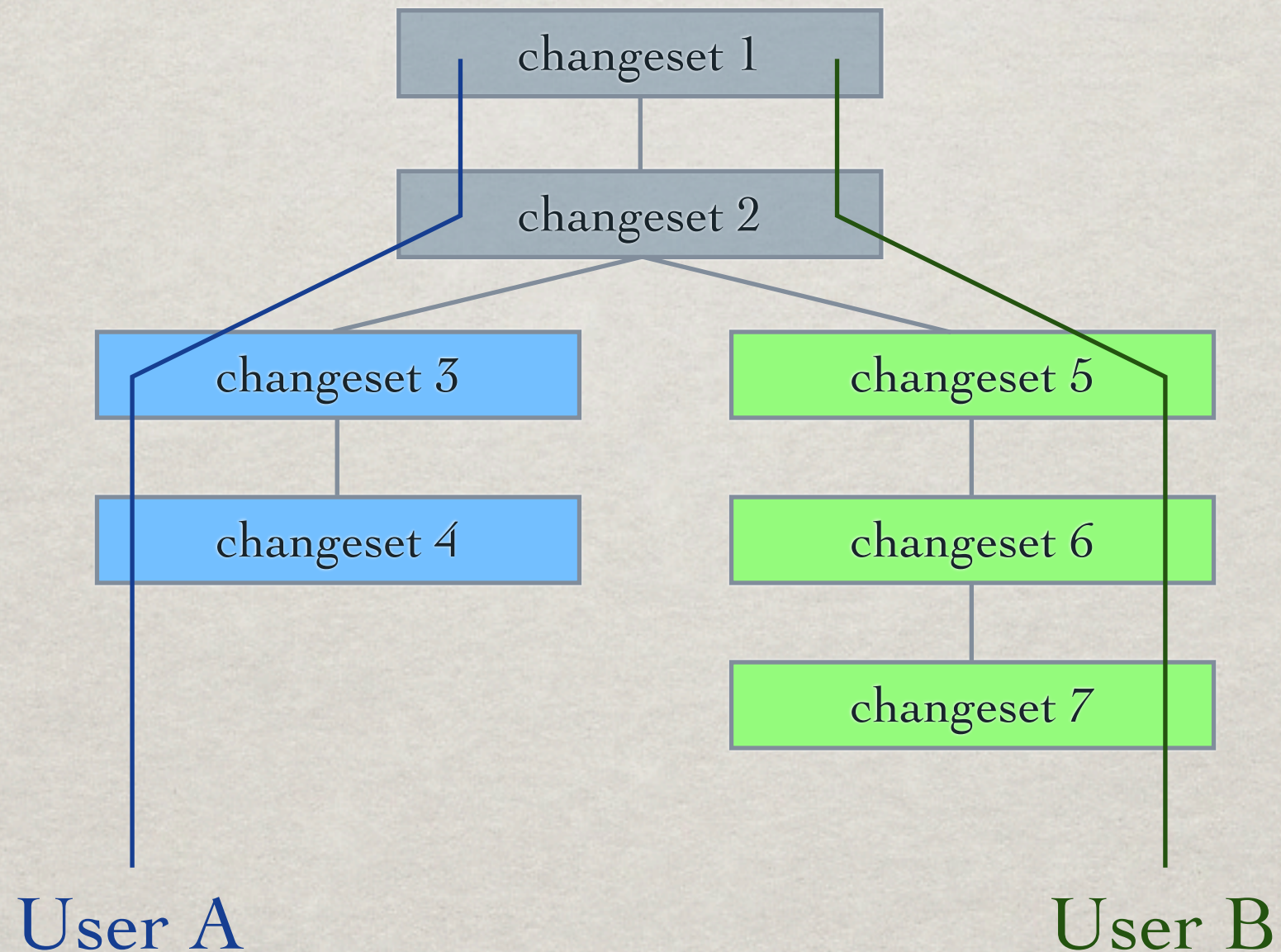
A good hosting service for Mercurial projects is

<http://bitbucket.org/>

- Project members *export* their changes to a file and send that file to other members (e-mail, USB stick, ...) who *import* it.

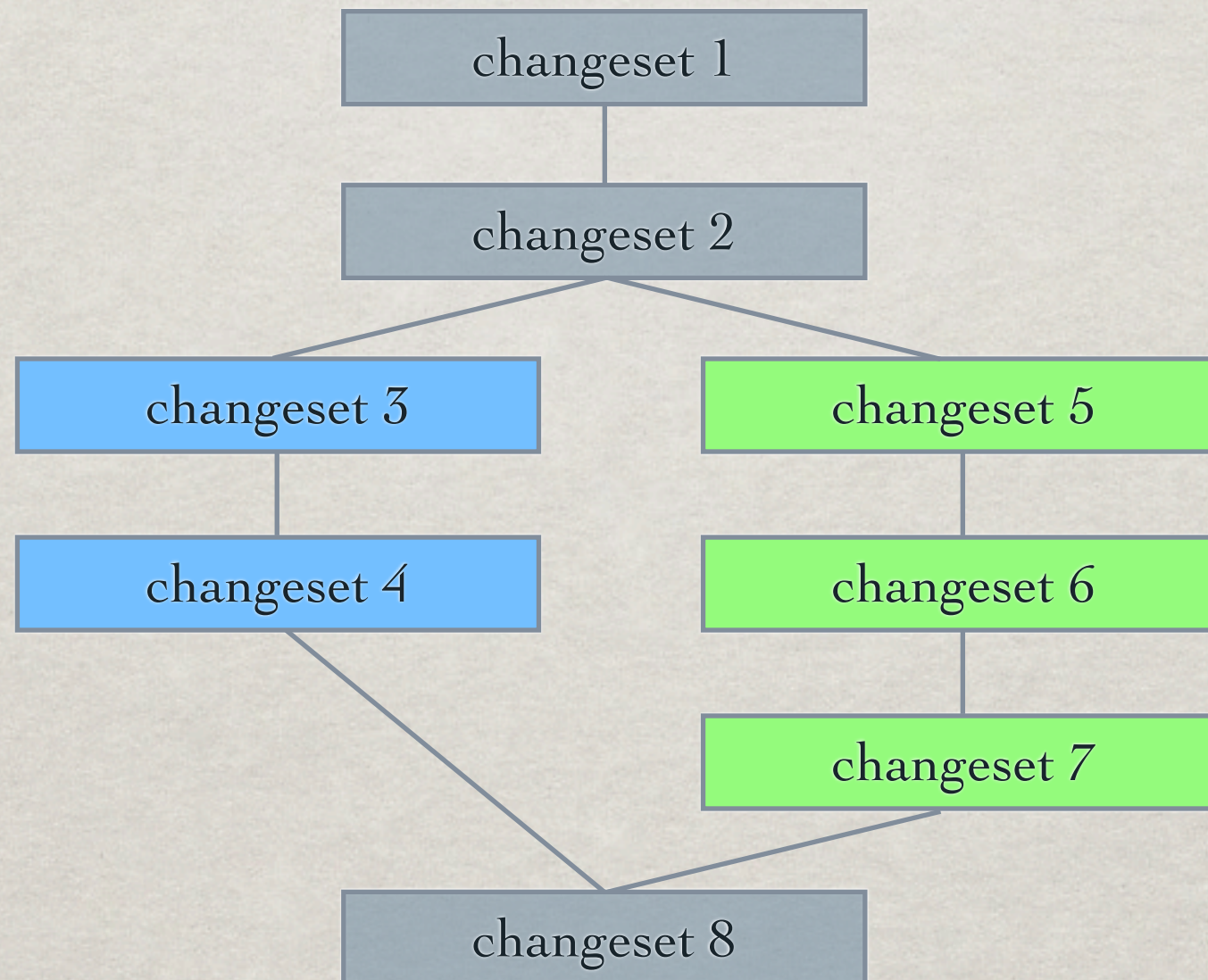
BRANCHES

After two project members share their modifications, the repository looks like this:



MERGING

Branches can be merged. This requires resolving conflicting changes by hand - Mercurial can't read your mind!



THEMATIC BRANCHES

- ✿ A single user can create multiple branches, and give them names.
- ✿ This is used for maintaining several versions (“stable” vs. “development”, ...).
- ✿ It is also used for experimental work that should not disturb mainstream work.