

**Umeå University**  
Department of Applied Physics and Electronics

**Linux as Development Environment 7.5 ECTS**  
**5EL142 HT-16**

**Assignment 3**

Submitted 2016-09-24  
Author: Lorenz Gerber, 20161202-2033 (lorenzottogerber@gmail.com)  
Instructor: Sven Rönnbäck

## 1 Step-by-step description of CVS exercise

1. In a sh shell, setup CVS locally:

```
> mkdir ~/cvs
> export CVSROOT=/home/myusername/cvs
> ~/cvs/cvs init
```

2. Create a new directory with some files

```
> mkdir ~/project_import
> cd ~/project_import
> touch test1.txt
> touch test2.txt
```

3. Import files into a new CVS repository and checkout a working copy

```
> cvs import -m 'initial source import' new_project LOGE start
> cd ~/
> rm -r ~/project_import
> cvs checkout new_project
```

4. Make some changes to the file(s) in the working copy and inspect the difference to the repository

```
> cd ~/new_project
> ls > test1.txt
> cvs status
> cvs diff test1.txt
```

5. Commit the changes in the working copy into the repository

```
> cvs commit -m 'added some text'
```

## 2 Step-by-step description of SVN exercise

1. Checkout working copy of remote repository

```
> mkdir ~/working_copy_1
> cd ~/working_copy_1
> svn co svn://130.239.163.12/lab4 . --username lab4
> mkdir lorenz_gerber
> echo "Testing SVN Repositories" > lorenz_gerber/lg.txt
> svn add lorenz_gerber
> svn commit -m 'adding my stuff'
```

2. Add name to users file

2(4)

```
> echo "Lorenz Gerber 2016-09-24" >> users
> svn commit -m 'added name to users file'
```

### 3. Test some commands

```
> svn update
> svn help log
> svn log -l 3 -v
> svn --diff users
> svn info
Path: .
Working Copy Root Path: /home/xxxxxxx/working_copy_1
URL: svn://130.239.163.12/labbb4
Relative URL: ^/
Repository Root: svn://130.239.163.12/labbb4
Repository UUID: 218399d0-4e29-49cd-b0e8-479a866bfbe8
Revision: 2371
Node Kind: directory
Schedule: normal
Last Changed Author: labbb4
Last Changed Rev: 2371
Last Changed Date: 2016-09-24 10:30:25 +0200 (lör, 24 sep 2016)
> svn help commit
```

### 4. Checkout second working copy of remote repository

```
> mkdir ~/working_copy_2
> cd ~/working_copy_2
> svn co svn://130.239.163.12/labbb4 . --username labbb4
```

### 5. Create a conflict by changing the same file in both working copies. Committing and resolving conflict

```
> echo "add some text" >> ~/working_copy_1/lorenz_gerber/lg.txt
> echo "add some other text" >> ~/working_copy_2/lorenz_gerber/lg.txt
> cd ~/working_copy_2
> svn update
> svn commit -m 'added some text'
> svn update
> cd ~/working_copy_1
> svn update
Updating '.':
C    lorenz_gerber/lg.txt
Updated to revision 2372.
Summary of conflicts:
  Text conflicts: 1
```

```

Conflict discovered in file 'lorenz_gerber/lg.txt'.
Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,
        (mc) my side of conflict, (tc) their side of conflict,
        (s) show all options: df
--- lorenz_gerber/lg.txt.r2372- THEIRS
+++ lorenz_gerber/lg.txt- MERGED
@@ -1,2 +1,7 @@
    Testing SVN Repositories
+<<<<<<< .mine
+add some other text
+||||||| .r2371
+=====
    add some text
+>>>>>>> .r2372
Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,
        (r) mark resolved, (mc) my side of conflict,
        (tc) their side of conflict, (s) show all options:

svn update
Updating '.':
C    lorenz_gerber/lg.txt
Updated to revision 2372.
Summary of conflicts:
    Text conflicts: 1
Conflict discovered in file 'lorenz_gerber/lg.txt'.
Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,
        (mc) my side of conflict, (tc) their side of conflict,
        (s) show all options: df
--- lorenz_gerber/lg.txt.r2372- THEIRS
+++ lorenz_gerber/lg.txt- MERGED
@@ -1,2 +1,7 @@
    Testing SVN Repositories
+<<<<<<< .mine
+add some other text
+||||||| .r2371
+=====
    add some text
+>>>>>>> .r2372
Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,
        (r) mark resolved, (mc) my side of conflict,
        (tc) their side of conflict, (s) show all options: m
Merging 'lorenz_gerber/lg.txt'.
Conflicting section found during merge:
(1) their version (at line 2)      |(2) your version (at line 2)
-----+-----
add some text                      |add some other text
-----+-----
Select: (1) use their version, (2) use your version,
        (12) their version first, then yours,

```

```

(2l) your version first, then theirs,
(e1) edit their version and use the result,
(e2) edit your version and use the result,
(eb) edit both versions and use the result,
(p) postpone this conflicting section leaving conflict markers,
(a) abort file merge and return to main menu: eb
Merge of 'lorenz_gerber/lg.txt' completed.
Select: (p) postpone, (df) show diff, (e) edit file, (m) merge,
        (r) mark resolved, (mc) my side of conflict,
        (tc) their side of conflict, (s) show all options: r
Resolved conflicted state of 'lorenz_gerber/lg.txt'
Summary of conflicts:
  Text conflicts: 0 remaining (and 1 already resolved)
> svn status
M      lorenz_gerber/lg.txt
> svn commit -m 'conflict resolved'
> svn update
Updating '.':
At revision 2373.

```

### 3 Description Conflict Resolution in SVN exercise

Two scenarios were tested:

1. edit same file, on different lines
2. edit same file, same line

In the first case the files were opened with an editor. Sequential committing and updating the individual working copies worked without a conflict. Please note that this case is not shown in the above step-by-step description.

For the second case, non-identical text was appended to the same file. The first working copy was committed and updated without obtaining a conflict message. Then, before trying to commit the edit in working copy 2, `svn update` was run which resulted in a 'conflict resolution dialog' as shown in the step-by-step description above. It was decided to edit both files. Some changes were made and saved to the local copy. On exiting the editor, the 'conflict resolution dialog' became active again, no 'mark resolved' was chose which resulted in return to the command prompt. The working copy contained now the local conflict resolution. Applying `svn commit -m 'conflict resolution'` committed the edits to the remote respository and after another `svn update`, the working copy was confirmed to be clear by running `svn status`. This case is the one shown as step-by-step description in the previous section.