

TP prise en main de SVN

Objectifs

- Comprendre l'utilité d'un logiciel de gestion de version.
- Savoir utiliser SVN de façon basique.

Ce tp est fortement inspiré d'un TP de JC Routier, au FIL/Université Lille 1 :

http://www.fil.univ-lille1.fr/~routier/enseignement/licence/pl/tdtp/tp_svn_partie1.pdf

et http://www.fil.univ-lille1.fr/~routier/enseignement/licence/pl/tdtp/tp_svn_partie2.pdf.

Contexte et préparation Dans ce TP nous allons préparer vos dépôts SVN pour le projet. Nous en profiterons pour faire des manipulations de base avec SVN à la ligne de commande. Vous aurez avec vous les transparents du cours pendant les différentes manipulations. Il vous est fortement recommandé de finir le TP avant de commencer le projet.

1 SVN, principales commandes en mono-utilisateur - 1H15

Pour cette section les deux membres du même binôme seront devant la même machine. Des dépôts SVN ont été créés sur synthe, dont la liste est en annexe. Les dépôts ne seront accessibles en lecture/écriture qu'aux deux personnes du binôme (et aux enseignants). Dans la suite `<nomdudepot>` sera remplacé par votre nom de dépôt.

1.1 Préparation

Allez chercher sur la page web du cours le tgz prévu pour le tp, et extrayez dans un répertoire adéquat. Regardez rapidement le contenu du répertoire `tpsvn` créé.

Premier import Vos dépôts étant vides, il convient de les remplir avec les fichiers du répertoire `tpsvn` :

1. Pour cela, placez-vous un cran au dessus et faire import du répertoire :

```
svn import -m "version init" tpsvn/ svn+ssh://synthe/ima3/<nomdudepot>
```

rentrez votre mot de passe, et observez.
2. Détruisez votre répertoire `tpsvn` qui ne sert plus à rien (`rm -Rf`)

Premier checkout / copie locale

1. Récupérez une copie locale de votre dépôt en faisant :

```
svn checkout svn+ssh://synthe/ima3/<nomdudepot>
```
2. Vérifiez que tout y est.

À partir de maintenant, vous travaillerez dans le répertoire *versionné* s'appelant `<nomdudepot>`.

1.2 Modifications “single user” du dépôt

Modification d’un fichier existant

1. À la racine de votre copie locale, tapez `svn status -uv` et observez.
2. En vous inspirant de la même règle présente dans les autres Makefile, modifiez le Makefile de `libhash/` pour rajouter une règle “clean”.
3. À la racine de votre copie locale, tapez `svn status -uv` et observez.
4. Il s’agit maintenant d’enregistrer les modifications sur le serveur (en français informatique, “commit”) :

```
svn commit -m ‘‘modification du makefile pour hash’’
```

(au passage remarquez que le nom du serveur n’est plus précisé, ni celui du dépôt)

Ajout d’un fichier / d’un répertoire

1. À la racine de votre copie locale, créez un fichier nommé `readme.txt` et mettez les noms du binôme dedans.
2. Statut ?
3. Pour qu’il soit versionné, tapez :

```
svn add readme.txt
```
4. Statut ? Commit (n’oubliez pas le message de log) ! Statut ?
5. Ajoutez puis commitez un répertoire nommé `toto`.

Observation du dépôt La commande `svn status` permet d’observer l’état de la copie locale, la commande `svn log` va nous permettre d’observer plus finement ce qui s’est passé sur le dépôt :

1. Observez le résultat de la commande `svn log dico.english` à partir de la racine du dépôt.
2. Idem avec `readme.txt` et `libhash/Makefile`
3. Observez les différences entre les deux `libhash/Makefile` (révisions 1 et 2) avec la commande :

```
svn diff libhash/Makefile -r1:2
```

Modification de noms (fichiers, répertoire) et déplacements, suppressions

1. Pour changer le nom d’un fichier déjà versionné on utilise `svn mv` :

```
svn mv readme.txt Readme.txt
```
2. Status, commit.
3. Changer de place un fichier :

```
svn mv dico.english toto/
```
4. Regardez le contenu des répertoires, status, commit
5. enfin, on va supprimer le répertoire `toto` :

```
svn rm toto
```

Avant de commiter ¹, placez-vous dans le répertoire `toto` et commitez à partir de ce répertoire (uniquement la destruction du contenu du répertoire, donc). Ensuite, revenez un cran plus haut et commitez la destruction du répertoire.

Autre copie locale Dans un autre endroit de votre compte, refaire un

```
svn checkout svn+ssh://synthe/ima3/<nomdudepot>
```

et vérifiez que cette nouvelle copie locale correspond à la version courante du dépôt, avant de la supprimer (en local).

¹les raisons de cette manip sont assez mystérieuses ...

2 SVN en mode collaboratif - le reste !

Maintenant, chacun des deux membres du binôme doit se loguer sur une machine différentes, et avoir une copie locale du dépôt à jour. **À ce stade du TP, les deux copies locales (sur deux comptes différents) sont identiques.** Nous appelons *dupont* et *durand* les deux membres du binôme. Nous allons travailler sur le fichier `main.c`.

Résolution automatique des conflits

- *dupont* ajoute la déclaration d'une constante symbolique quelconque.
- *durand* fait une impression quelconque dans la fonction `main`. **Avant de faire le moindre commit, il FAUT toujours vérifier que votre programme compile toujours**
- *dupont* commite. Sans problème, normalement
- Observer le message d'erreur lors d'une tentative de commit de *durand*.
- *durand* fait `svn update` (mise à jour). Pendant cette mise à jour, le conflit est résolu (conflit dans des endroits distincts du fichier).
- *durand* commite cette mise à jour. *dupont* récupère cette mise à jour avec `svn update`.

Résolution manuelle des conflits

- Vérifier que *durand* et *dupont* ont bien la même version de copie locale, par exemple en faisant `svn update` et en notant le numéro de version.
- *durand* modifie la valeur de la constante symbolique rajoutée auparavant, et commite (sans problème).
- *dupont* modifie la valeur de la même constante symbolique, avec une autre valeur.
- Observer le message d'erreur lors d'une tentative de commit de *dupont*.
- Pour essayer de résoudre le conflit, *dupont* fait `svn update` (mise à jour), et il peut ensuite visualiser le problème avec `df`, puis faire `p` pour remettre à plus tard la résolution. Ensuite, *dupont* édite le fichier en question, qui a été modifié par `svn`. Les conflits sont délimités par des chevrons :

```
<<<<< .mine
    #define X 42
>>>>> .r4
    # define X 72
```

Le conflit est donc entre la copie locale de *dupont* (`mine`) et la révision 4 du dépôt. *dupont* doit éditer le fichier, enlever les chevrons, ne garder qu'une seule version, sauver. Enfin, il signale le conflit comme résolu :

```
svn resolved main.c
puis commite.
```

- *durand* fait un update pour avoir la même version du fichier.

Avant chaque commit, on fera un update afin de résoudre les conflits en local.

Ménage de fin de TP Supprimer la déclaration de la constante symbolique et le `printf` qui ne sert à rien, puis commiter.

Annexe 1 - Liste des dépôts sur synthe

Voici la liste des dépôts sur synthe :

```
lgonnord@synthe:/var/lib/svn/ima3$ ls
2009_2010      DebieChalaux      KemajouLenormand  PonsRodriguez
AlexandreLeuliet  FeiLiLiu          LambertMairesse   RomerowskiXi
AljaneBangoura   FouaideMabrouk    LelaureYongoua    RukataVandermeersch
AmegavieBonvalet  GouenardKourouma  LoembetMahir      SaadaneThary
BoudjemaChampagne  HamdaneMboup      MoignerTheon      TaiebVincenti
BurtaireGombault  HossieMonteiro    MomboPesqueux     Test
ChretienGasnier   HusseMonier        PetitprezRibreau  WangXing
```

Annexe2 - SVN de polytech, accès extérieur

Les manipulations faites plus haut sont valables uniquement en machine de TP (c'est-à-dire à partir d'une machine du même sous-réseau que **synthe**). Pour récupérer vos fichiers via svn chez vous (sous Linux, je ne fais plus de support windows), il vous faudra faire un tunnel SSH en passant par la machine **portier** ²:

1. étape 1, à ne faire qu'une fois. Éditez `.ssh/config` et mettre les commandes suivantes dans ce fichier :
Host poly
Hostname localhost
Port 2200
User lgonnord
(en remplaçant lgonnord par votre login à polytech)
2. étape 2, à faire à chaque nouveau démarrage de votre machine perso :
`ssh -L 2200:synthe.studserv.deule.net:22 -p2222 lgonnord@portier.polytech-lille.fr`
(en remplaçant lgonnord par votre login à polytech) Cette commande réalise un tunnel ssh entre les machines **synthe** et **portier**. Vous laissez ce terminal ouvert et vous ne travaillez pas dedans.
3. Ensuite, vous pouvez faire comme si poly était un serveur svn local, par exemple :
 - infos sur un repo svn :
`svn info svn+ssh://poly/ima3/<nomdudepot>`
 - checkout :
`svn checkout svn+ssh://poly/ima3/<nomdudepot>`
4. Comment travailler ? Chez vous, vous faites **une unique fois** un checkout vers un répertoire de votre machine :
`svn checkout svn+ssh://poly/ima3/<nomdudepot>`
Ensuite vous travaillez dans ce répertoire (en particulier, vous utilisez le compilateur de votre machine ...). Ensuite, à chaque fois que vous voulez récupérer/enregistrer un changement sur le svn, vous utilisez les commandes `svn up`, `svn commit`, ... en ayant pris soin de bien avoir lancé le tunnel avant.

²parce que portier est la seule machine accessible de l'extérieur en SSH, et uniquement sur le port 2222