

L3 – Cours de Système – Travaux dirigés

Installation de Nachos sous CVS

Emmanuel Agullo, Eddy Caron, Nicolas Veyrat Charvillon

Vendredi 17 février 2006

1 Création du serveur CVS

Vous vous appelez Dupont et Durand. C'est Dupont qui héberge le serveur CVS.

Dupont choisit de l'installer dans le répertoire

```
/home/dupont/
```

Il récupère une copie du serveur tout installé nachos-scratch-cvsroot.tgz à un des deux endroits suivants

```
/home/eagullo/teach/systeme/arch  
http://graal.ens-lyon.fr/~eagullo/systeme/nachos/
```

et la sauve sous /home/dupont/cvsrootnachos.tgz. Il suffit ensuite de l'ouvrir pour restaurer le serveur en état de marche.

```
cd /home/dupont/  
tar xfvz cvsrootnachos.tgz
```

La racine du serveur est /home/dupont/cvsrootnachos.

Chacun des partenaires ajoute à son fichier .bashrc (ou dans \${HOME}/etc/shell/bashrc.d/80_Cvs si vous avez installé Soft Élèves) les lignes suivantes.

```
export CVSROOT=/home/dupont/cvsrootnachos  
export CVSUMASK=000  
export CVSEEDITOR=pico # Par exemple...
```

Il faut ensuite recharger le fichier de configuration correspondant. Lors de votre prochaine entrée sous le système, ce sera fait automatiquement à l'initialisation.

```
source ~/.bashrc
```

Maintenant, il faut que les deux partenaires puissent écrire dans ce répertoire et y créer des fichiers. D'abord une méthode simple.

```
chmod -R ug+rwX,o-rwx $CVSROOT
```

L'inconvénient est que tous les membres du groupe Unix (la plupart des élèves de l'école!) peuvent aussi y accéder, voire y modifier des fichiers... Les ACL (*Access Control Lists*) n'étant plus disponibles, la seule manière de sécuriser réellement ces données, est de créer un nouveau groupe

Unix en allant sur la page de l'intranet <http://intranet.ens-lyon.fr> dans la rubrique Outils. Mettre les logins des personnes du binôme dans ce groupe.

On va ensuite mettre la racine `cvrootnachos` dans ce groupe.

```
chgrp nom_du_groupe cvrootnachos
```

On fixe enfin les permissions sur cette racine.

```
chmod ug+rx,o-rwx cvrootnachos
```

Il n'est *a priori* pas nécessaire de fixer ce groupe et ces permissions sur tous les fichiers de ce répertoire.

Maintenant, tout est prêt pour commencer à travailler en binôme.

2 Utilisation de CVS

Maintenant, on a créé le *Repository* (là où les versions à jour des fichiers sont sauvegardées). Pour faire des modifications, un utilisateur doit récupérer une copie des fichiers, les modifier, puis les mettre à jour dans le Repository.

Durand a décidé de travailler dans un répertoire personnel.

```
cd /home/durand/MIM1/Systeme/  
mkdir Nachos  
cd Nachos
```

Si vous avez peur que le binôme d'à côté copie votre travail, il faut interdire aux autres personnes de lire ce répertoire.

```
chmod og-rwx ~/Nachos
```

Exportez depuis le serveur CVS la version courante du projet.

```
cv co nachos # co = check-out
```

Cela reconstruit au point d'exportation toute la hiérarchie de racine `nachos`.

Pour voir quelle est la version courante, faire

```
cv st # st = status
```

Promenez-vous dans l'arborescence pour aller modifier un fichier. Par exemple

```
cd nachos/code/test
```

et modifiez le fichier `halt.c`.

```
cat >> halt.c
```

```
/* This is a strange comment... */
```

```
^D # Control-D
```

Vérifiez avec `cv st` que la modification a bien été vue par CVS (*locally modified*). Par contre, votre binôme ne la voit pas encore.

Maintenant, enregistrez cette modification.

```
cvcs ci          # ci = check-in. On peut aussi dire cvcs commit
```

Bon sang mais c'est bien sûr ! Vous en êtes à la version 1.2... Incroyable, non ? Que votre binôme fasse `cvcs co` ou `cvcs up` chez lui, et il verra la nouvelle version.

De plus, si par hasard il était lui aussi en train de travailler sur `halt.c`, CVS essaie de fusionner sa version et votre version de manière "raisonnable". En général, ce n'est pas trop mal. En cas de conflit insoluble, CVS vous met un message et annote le fichier. Faites un essai !

Maintenant, détruisez (par erreur !) le fichier `halt.c` : "Zut !" (ou toute autre interjection adaptée). Vous pouvez remettre à jour votre copie locale à partir du serveur CVS en tapant tout simplement

```
cvcs up          # up = update
```

De manière plus générale, à chaque fois que vous commencez à travailler, n'oubliez pas de faire

```
cvcs up
```

pour mettre à jour votre copie locale à partir de la dernière version déposée sous le serveur. N'utilisez `cvcs co` que pour créer une nouvelle copie. Vous pouvez avoir plusieurs copies différemment modifiées à différents endroits de votre arborescence.

3 Quelques conseils...

Ne faites des `cvcs ci` que lorsque votre partie est suffisamment testée. Sinon, votre binôme sera complètement perdu ! L'invariant doit être : la version disponible sous CVS fonctionne.

Prenez le temps de mettre des commentaires significatifs dans le *log*. Cela vous aidera beaucoup dans les phases d'intégration.

Placez votre `cvcsrootnachos` dans une autre partie de votre arborescence que la partie dans laquelle vous travaillez habituellement. L'idéal serait même qu'elle soit sur un autre disque, d'ailleurs. Prenez aussi le temps de faire des sauvegardes de temps en temps, on ne sait jamais...

```
mkdir ~/Sauvegardes
cd $CVSROOT
tar cvfz ~/Sauvegardes/nachos.tar.gz .
```

Un petit résumé des commandes principales de CVS, fait par Brice Goglin (merci encore !), se trouve à l'adresse suivante :

<http://graal.ens-lyon.fr/~eagullo/systeme/arch/cvcs.txt>