# Concurrent Version System

TP n°1 - Hoerdt Mickaël/Quirin Arnaud - Mars 2004

## 1 Exercices

### 1.1 Création de l'environement et de l'entrepot

- Créer un répertoire TP1
- Y créer un répertoire Entrepot
- Créer la variable d'environement CVSROOT en lui donnant la valeur ~/TP1/Entrepot
- Créer l'entrepot avec la commande : *cvs init*
- Quel est le résultat ?

### 1.2 Mise en place des fichiers sources et chargement de l'entrepot

- Copier tous les fichiers de /users/prof/hoerdt/GL/TP1/Othello et les placer dans votre répertoire TP1
- Créer un module "Othello" dans l'entrepot composé de tous ces fichiers avec le message "initialisation du module"
- Quelle sont les modifications dans le répertoire $CVSROOT ?

### 1.3 Manipulations des fichiers gérés par CVS

- Ouvrir une fenêtre "UN" par la commande : xterm -T UN&
- Se placer dans cette fenêtre puis créer le répertoire "dev1"
- Se placer dans ce nouveau répertoire
- Extraire une copie de travail du module Othello
- Modifier le fichier Menu.cc (par exemple en y ajoutant un commentaire
- Valider la modification du fichier par CVS
- Créer un nouveau fichier de nom README
- L'ajouter au module Othello
- Valider cet ajout

### 1.4 Travail à deux sur le même fichier

- Ouvrir une seconde fenêtre "DEUX"
- Se placer dans cette fenêtre
- Créer un répertoire TP1/dev2
- Se positionner dans ce répertoire
- Extraire une copie de travail
- Modifier le fichier Clock.cc
- Valider la modification
- Aller dans la fenêtre "UN"
- La modification a-t-elle été transmise? Pourquoi ?





- Dans cette fenêtre "UN", modifier Clock.cc en rajouter en fin de ligne la ligne "// fin de fichier"
- Valider la modification. Que se passe-t-il ?
- Dans la fenêtre "UN", exécuter la commande "update" pour Clock.cc
- Vérifier le résultat : que contient-il ?
- Valider cette nouvelle version
- Revenir à la fenêtre "DEUX"
- Dans quel état est Clock.cc ? Que faut-il faire pour rétablir la cohérence?
- Après avoir rétabli la cohérence (version identiques) pour les deux fenêtres, faire dans chaque copie de travail une modification différente mais sur la même ligne (par exemple des espaces supplémentaires).
- Valider les modifications pour les deux fenêtres
- Que se passe-t-il ?
- Editer le fichier et observer le résultat
- Choisir une version
- Valider la modification

### 1.5 Les commandes status, history, log

Une aide des commandes peut etre obtenue par : cvs nom_commande –help

- Essayer les commandes "status", "history" et "log" d'abord sans argument puis avec des noms de fichiers
- Expérimenter leurs options
- Utiliser la commande "diff" pour visualiser les modifications apportées aux fichiers entre différentes versions
- Utiliser la commande "annotate" pour afficher toutes les modifications apportées aux fichiers suivant chaque version

### 1.6 Versions des fichiers et versions du logiciel

Les numéros de version des fichiers ne sont pas les numéros de version du logiciel complet. Pour appliquer un numéro de version au logiciel, il faut utiliser le mécanisme de nom symbolique avec "tag".

- Donner le nom symbolique "Version-1-0" à la version d'origine des fichiers
- Donner un autre nom symbolique à la version courante des fichiers
- Créer un répertoire "dev3" sous votre home directory
- Se placer dedans et y extraire les versions originales des fichiers
- Vérifier le résultat (par exemple Clock.cc)