

## Feuille de travaux pratiques n° 2

### Gestion de version

Cet exercice se divise en deux parties : durant la première partie, chaque binôme va travailler isolément pour faire évoluer le logiciel `hello` permettant de dire bonjour en plusieurs langues. Dans la deuxième partie, l'ensemble des binômes vont travailler sur le même programme.

**Pour la première partie, on prendra soin de relancer l'ordinateur utilisé afin de s'assurer de l'absence de serveur « `svnserve` » lancé en démon par un autre utilisateur.**

**Première partie.** Dans un premier temps, chaque binôme va travailler avec un dépôt *subversion* local. Durant la deuxième partie de la séance, tous les binômes devront travailler avec un dépôt commun dont l'adresse sera donnée par l'enseignant responsable du TP.

1. Récupérer l'archive `hello.tar.gz` sur *madoc* et la décompresser localement ;
2. Créer un dépôt *subversion* local dans `$HOME/svnroot` ;
3. Importer le code de `hello` dans le dépôt. À partir de ce point, on travaillera dans une arborescence dont les fichiers sont contrôlés par *subversion* ;
4. Modifier le logiciel `hello` de façon à ce que la fonction `afficher_aide()` soit définie dans un fichier `messages.c`. Faire prendre en compte tous les changements par *subversion* et identifier le nouveau programme comme une *release* par le tag `release_1_0` ;
5. Ajouter un répertoire `doc/` contenant une petite documentation sur l'utilisation de `hello` et la liste des langues supportées dans un fichier `hello-doc.txt`. Faire prendre en compte cette nouvelle version par *subversion* ;
6. Renommer le répertoire `src/` en `hello/` et faire un *commit* des changements. Créer une nouvelle release `release_1_1`.

#### Deuxième partie.

1. Créer une copie locale du programme `hello` se trouvant dans le dépôt dont l'adresse est donnée par l'enseignant de TP ;
2. Choisir une ou plusieurs langues dans la table 1 et modifier `hello` pour qu'il les supporte en plus de celles déjà reconnues. Des conflits avec les autres binômes devraient apparaître. Gérer ces conflits de façon à ce que le programme final fonctionne comme initialement en gérant l'ensemble des langues données dans la table 1 ;
3. Une fois l'ensemble des conflits résolus, désigner l'un des binômes pour faire une *release* (ajout d'un tag) qui servira de base dans la suite (i.e. on fera un *update* de cette version avant de continuer) ;
4. Ajouter une fonctionnalité au choix au programme en gérant entre les binômes concernés les éventuels conflits.

TABLE 1 – « Bonjour » dans différentes langues

Espagnol	<i>Hola</i>	Lithuanien	<i>Sveikas</i>
Maori	<i>Tena koe</i>	Somali	<i>Maalim wanaqsan</i>
Suédois	<i>Hallå</i>	Thaï	<i>Sawatdi</i>
Yoruba	<i>E kú àárò</i>	Zoulou	<i>Sawubona</i>