

TP MOPO – Utilisation de Subversion

Préambule

Un dépôt SVN a été spécialement créé dans le cadre de ce TP. Il est accessible à tous les étudiants à l'aide de leurs identifiants et mot de passe MOODLE à l'URL suivante :

http://dept-info.univ-fcomte.fr/svn/module_mop2016

Le TP est organisé en deux phases, la première dédiée à l'utilisation de svn en ligne de commande sous un terminal Linux, et la seconde à l'utilisation du client graphique Tortoise SVN sous Windows.

Le contenu initial du dépôt est composé simplement de trois répertoires :

/tp1

 /command_line

 → Destiné à accueillir les travaux exécutés à l'aide de svn sous Linux

 /tortoise

 → Destiné à accueillir les travaux exécutés à l'aide de TortoiseSVN sous Windows

Subversion en ligne de commande

Dans un premier temps, le TP se déroule sous un terminal Linux.

Partie 1 : En binôme

Connexion au dépôt et récupération d'une copie locale

Vérifier la présence de la commande svn dans le PATH à l'aide de la commande « *svn info* ». Se connecter ensuite au dépôt SVN en utilisant le compte d'un des deux membres du binôme, et récupérer une copie locale.

Commandes utiles : svn checkout

Création d'un répertoire correspondant au binôme

Dans « */tp1/command_line* », créer un répertoire à la racine du dépôt, contenant les noms des deux (ou trois en cas de trinôme) étudiants, en minuscule, séparé par des « *_* ».

Par exemple, pour un binôme constitué de Olivier MARTIN et Alexandre DUPONT, créer un répertoire « */tp1/command_line/martin_dupont* ».

Commandes utiles : svn mkdir

Création d'une application basique et ajout du/des fichiers dans le dépôt

A l'intérieur de ce répertoire, créer une application en langage C, qui, dans un premier temps, consistera à déclarer un tableau de 20 éléments de type entier, remplir ce tableau de valeurs aléatoires, et afficher le contenu du tableau.

Déposer ce fichier source sur le dépôt.

Commandes utiles : svn import / svn add

Modification de l'application en rajoutant une fonction et mise à jour du dépôt

Ajout dans cette application une fonction 'Sort', qui dans un premier temps ne contiendra aucune implémentation, et l'appeler à partir de la fonction 'main'. Afficher le tableau avant et après l'appel de celle-ci.

Déposer les modifications sur le dépôt.

Commandes utiles : svn update, svn commit

Partie 2 : Individuellement

Connexion au dépôt et récupération d'une copie locale

Chaque étudiant se connecte sur le dépôt à l'aide de son identifiant propre, et récupère une copie locale du contenu du répertoire précédemment créé.

Chaque étudiant implémente, à l'aide d'un algorithme différent de celui de son binôme, la fonction Sort précédemment créée. Le choix des algorithmes de tri à utiliser est libre. Parmi les plus connus, nous pouvons citer les suivants :

- Bubble sort
- Insertion sort
- Quick sort

Mise à jour du dépôt

Une fois l'algorithme implémenté et fonctionnel, chaque étudiant met à jour le dépôt avec ses modifications.

Des conflits devraient normalement apparaître, étant donné que deux implémentations différentes de la même méthode sont présentes.

Commandes utiles : svn update, svn commit

Résolution des conflits

Pour résoudre le conflit, rajouter un argument à la fonction Sort afin de pouvoir choisir quel algorithme utiliser. Modifier l'application afin de permettre de demander à l'utilisateur quel algorithme de tri il souhaite utiliser pour trier le tableau.

Commandes utiles : svn diff, svn resolve, svn log, svn cat

Mise à jour du dépôt avec la version fusionnée

Redéposer le fichier fusionné sur le dépôt.

Subversion à l'aide de TortoiseSVN

Redémarrer sous Windows et refaire l'exercice à l'aide de TortoiseSVN, en prenant cette fois comme répertoire racine « /tp1/tortoise ».