

# TP2 - MEDEV

Carito Guziolowski

Département MATH-INFO, École Centrale de Nantes

21 novembre 2019

## 1 Utilisation de SVN avec la ligne de commandes

**Objectif :** Explorer quelques opérations de base en utilisant subversion (svn) pour le contrôle de version de fichiers.

Un identifiant pour l'exercice 5 vous sera attribué au début de cet exercice. L'idéal est de travailler avec la ligne de commandes dans un OS Linux ou Mac. Pour ceux qui souhaitent rester sur Windows vous pourriez installer et travailler avec Tortoise (voir slides du cours).

1. Allez dans votre répertoire personnel, créez un sous-répertoire appelé `versionControl` et positionnez vous (commande `cd`) dans ce répertoire de travail. Créer une copie de travail locale du référentiel SVN à l'URL `https://subversion.ec-nantes.fr/svn/eleves/MEDEV`. Utilisez votre login et mot de passe ECN pour pouvoir avoir accès à ce référentiel. Regardez les fichiers.

### Solution

```
cd
mkdir versionControl
cd versionControl
svn co https://subversion.ec--nantes.fr/svn/eleves/MEDEV
```

2. Examinez l'historique des sous-versions pour voir comment le référentiel a changé avec chaque révision.

### Solution

```
svn log
```

3. Créez un nouveau fichier dans votre copie de travail du référentiel. Appelez cela "besoinsXX" où XX est le numéro qui vous a été attribué au début de cet exercice, par exemple : besoins03 Mettez le texte suivant dans votre fichier :

```
Sécurité
Aventure
Liberté
Créativité
Expression
Acceptation
```

puis enregistrez le fichier. Ajoutez-le à la liste des fichiers suivis par subversion.

### Solution

```
svn add besoinsXX
```

- Utilisez la commande `svn diff` pour voir comment Subversion enregistre vos modifications. Faire un commit des vos modifications dans le référentiel, en utilisant un message de journal pour décrire ce que vous avez fait.

#### Solution

```
svn update
svn commit -m ``message``
```

- Pour le dernier exercice, vous devrez travailler en groupes de deux ou trois. Choisissez l'un des fichiers "besoins" ajoutés par un membre de votre groupe aux exercices 3 et 4. Chaque membre du groupe doit ensuite ajouter une nouvelle ligne au fichier et éditer la même partie de ce fichier, par exemple, vous pourriez changer le texte dans la troisième ligne. Qu'est-ce qui se passe lorsque vous essayez tous d'envoyer vos changements au référentiel ? Pouvez-vous résoudre le conflit afin que vous puissiez tous exécuter le `commit` avec succès et que la nouvelle ligne qui appartient à chaque membre soit incluse ?

## 2 Utilisation de GIT - *Branching* en locale

**Objectif :** Explorer quelques opérations de base concernant les branches dans un travail en local en utilisant GIT.

Dans la figure 1 il sont présentés des états du système GIT que vous devriez obtenir en exécutant des commandes spécifiques en GIT.

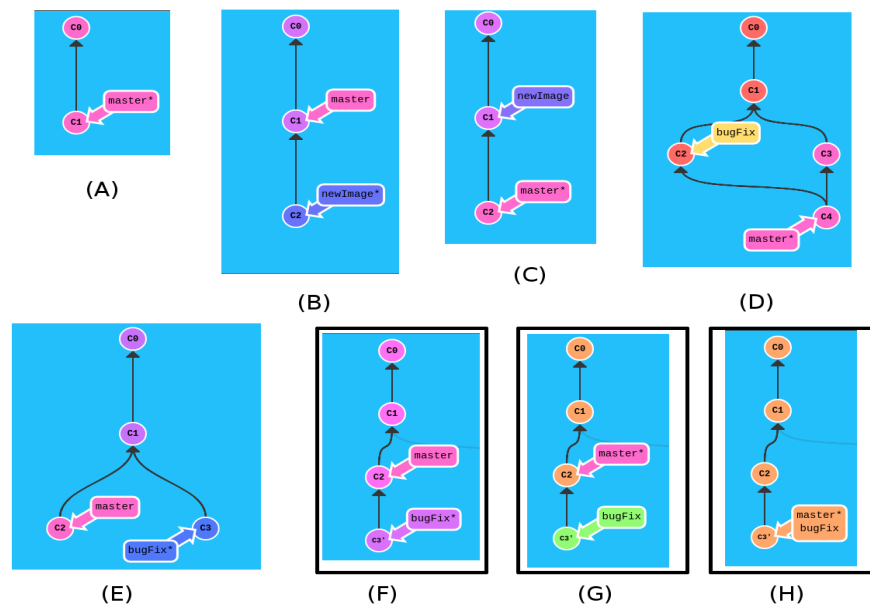


FIGURE 1 – Diagrammes représentant l'état d'un système GIT. Le \* représente le focus actuel de l'état en GIT.

- Créer un dossier GIT en locale
- Ajouter un fichier dans la branche master et modifiez le en exécutant 2 commits (C0-C1), comme indiqué dans la figure 1 (A).

#### Solution

```
cguziolo] vi my_file.txt; git add my_file.txt
cguziolo] git commit -am "1st_commit_C0"; vi my_file.txt
cguziolo] git commit -am "2nd_commit_C1"
```

3. Créez une branche `newImage`, au niveau du commit "C1", et faites un commit ("C2") dans la branche `newImage` pour obtenir l'état illustré dans la figure 1 (B).

**Solution**

```
cguziolo] checkout -b newImage; vi my_file.txt
cguziolo] git commit -am "Commit_C2"
```

4. De façon alternative à (3), vous pourriez créer à partir des instructions écrites dans (2) l'état décrit dans la figure 1 (C). Hint : regarder la documentation de la commande `git branch`

**Solution**

```
cguziolo] git branch newImage; vi my_file.txt
cguziolo] git commit -am "Commit_C2"
```

5. Créer les étapes pour obtenir la configuration illustrée dans la figure 1 (D). Vous pourriez recommencer dans un autre dossier avec `git init` si nécessaire

**Solution**

```
cguziolo] (.. step 2) git checkout -b bugFix
cguziolo] git commit -am my_file.txt;
cguziolo] git checkout master; git commit my_file.txt;
cguziolo] git merge bugFix
```

6. Créer les étapes pour obtenir la configuration illustrée dans figure 1 (E). Faites de tests et dessiner la configuration du système si nous exécutons `git rebase master` (figure 1 (F)); dessiner la configuration si nous exécutons ensuite la commande `git checkout master` (figure 1 (G)) et finalement `git rebase bugFix` (figure 1 (H)).
7. Pour aller plus loin dans cette notion : <https://learngitbranching.js.org/>