

Plus de commandes et git

TP 1

Objectifs

- ★ Utiliser Git.
- ★ Manipuler de nouvelles commandes.
- ★ Composer des commandes.

Rendu

- ⇒ Le TP est à rendre en mettant vos fichiers dans votre dépôt *github*.
- ⇒ Les fichiers doivent être ajoutés au commit, comités et enfin pushés sur le serveur

```
git add fichiers_ajoutés_ou_modifiés_pendant_le_tp
git commit -m''un message''
git push
```

- ⇒ Les fichier doivent être comités au fur et à mesure du TP

Les fichier à rendre vont nous servir à savoir si vous avez compris, n'hésitez pas à y ajouter des explications.

1 Git

Git est un gestionnaire de version décentralisé le plus populaire actuellement. Git trouve ses origines dans le développement du noyau Linux et a été fondé par Linus Torvalds. Il est aujourd'hui utilisé par un grand nombre de projet open source, comme par exemple Android, Eclipse ...

Pour illustrer l'utilisation du gestionnaire de version git, nous allons utiliser le serveur *github.com*. Il est tout à fait possible d'utiliser d'autre serveur git comme par exemple *gitlab.org*, *framagit.org* ou en auto-hébergement. Git en soit n'a pas besoin de serveur pour fonctionner, l'idée est de gérer l'évolution de fichiers et répertoires. Le serveur permet de pouvoir échanger facilement les fichiers gérés entre plusieurs ordinateurs.

Préparation : Créer un compte sur GitHub.com

- Créez un compte sur <https://github.com>. Transmettre votre login à votre enseignant.
- Vous allez recevoir une invitation *github/classroom* pour ce TP.

Préparation : Créer un répertoire de travail

- Créer un répertoire *TP1* dans votre répertoire de travail habituel.
- Pour chaque exercices créer un répertoire différent si besoin.
- Les réponses pour les exercices sur git sont à écrire dans un fichier *git.txt*

Exercice 1 Travailler seul.e avec Git (TP1/ex1)

1. Nous allons commencer par configurer Git en vous identifiant. Cela n'est à faire qu'une seule fois.

```
git config --global user.name "votre_login"
git config --global user.email "votre_mail"
git config --global core.editor emacs # ou votre éditeur préféré
git config --global color.ui auto
git config --global alias.lola 'log --graph --decorate --oneline --all'
git config --global alias.lop 'log --pretty=format:"%h - %an, %ar : %s"'
```

2. Créer un répertoire de dépôt git, c'est à dire créer un répertoire normalement, puis s'y déplacer et enfin utiliser la commande

```
git init
```

qui va donc initialiser le répertoire courant comme étant maintenant sous le contrôle d'un gestionnaire de version.
Quels répertoires ont été créés ?

3. Créer un fichier dans ce répertoire.

4. Tapez la commande

```
git status
```

et comprendre le message.

5. Tapez la commande

```
git add votrefichier
```

, quel est maintenant le statut de votre fichier ?

6. Vous pouvez maintenant commiter vos changements

```
git commit -m"Un superMessage"
```

7. Quel est maintenant le statut de votre répertoire ?

8. Expliquer maintenat le résultat de la commande

```
git log -p
```

9. Modifier maintenant votre fichier et comprendre le résultat de la commande

```
git diff
```

10. Vous allez maintenant lier votre dépôt local à un dépôt sur le serveur `https://github.com`.

— Créer un repository dans votre compte github.

— Dans votre répertoire appliquer la procédure décrite dans github qui correspond à votre cas.

Github met une sécurité en place pour pusher dans un *repository*. Il faut générer un token qui est un mot de passe temporaire. Pour cela, dans vos settings → Developer Settings → Personal Access Token → Tokens(classic), générer un nouveau token et cocher repo. Enregistrer bien ce token il ne sera plus visible par la suite.

11. Que voyez-vous dans votre dépôt sur le serveur github ?

12. Procédure pour mettre à jour les modifications sur le serveur :

— Modifier votre fichier.

— Ajouter le fichier pour mettre à jour ce qui sera commiter

```
git add filename
```

— Commiter votre fichier

```
git commit -m"Un message"
```

— Pusher les commits.

```
git push
```

Vérifier que vos commits sont bien arrivés sur le serveur.

Exercice 2 Travailler à plusieurs (TP1/ex2)

1. Aller dans un autre répertoire que la question précédente.
2. Clonez le répertoire `https://github.com/Polytech-Sorbonne-OPI-2024/tp1_ex2`

```
$ git clone https://github.com/Polytech-Sorbonne-OPI-2024/tp1_ex2.git
```

Quels répertoires ont été créés ?

3. Déplacez vous dans le répertoire versionné, que retourne la commande `git status`.
4. Créer un fichier `votre_nom.txt` où `votre_nom` est votre nom ou votre login.
5. Ajouter votre fichier sur le dépôt distant (voir la procédure question précédente).
6. Que se passe-t-il lors du push ?
7. Mettre à jour votre copie locale (le répertoire que vous avez cloner) avec la commande `git pull`.
Il y a 2 stratégies pour mettre à jour votre dépôt local :
 - `merge` : fusionner vos modifications avec celles du dépôt distant, le dernier commit à plusieurs passé.
 - `rebase` : applique les commits précédents et merge les modifications. Le dernier commit à 1 seul passé.
8. Il faut souvent configurer `git pull` en choisissant la stratégie d'application des modifications.
9. Réessayer de *pusher* (comme beaucoup d'élèves vont pusher en même temps il va falloir mettre à jour plusieurs fois votre copie locale avant de pouvoir pusher).
10. Modifier la première ligne de votre fichier et celle de votre voisin.
11. Commiter (*add/commit*) vos changements et envoyer les sur le serveur (*push*).
12. Que se passe-t-il ?
13. Régler le conflit sur votre fichier (choisissez la ou les modifications que vous voulez garder), commiter et pusher.
14. Regarder l'historique des changements avec la commande `git lola` et `git lop` ou l'interface graphique `gitg` ou sur github.

2 Tout compris ?

Écrire les réponses à ces questions dans un fichiers à pusher dans votre dépôt git créer à partir de cette invitation : Classroom invite

Exercice 3 Savez-vous répondre à toutes ses questions ?

Question 1

Quel caractère permet de lancer un programme en arrière plan ?

Question 2

Comment s'appelle le répertoire racine sous GNU/Linux ?

Question 3

Quelles commandes permettent toujours de retourner à la racine de son compte ?

- | | |
|---|---|
| <input type="radio"/> <code>cd</code> | <input type="radio"/> <code>cd ~</code> |
| <input type="radio"/> <code>cd .</code> | <input type="radio"/> <code>cd \\${HOME}</code> |
| <input type="radio"/> <code>cd ../</code> | <input type="radio"/> <code>cd \\${PWD}</code> |

Question 4

Soit l'arborescence de la figure 1, on suppose que l'on se trouve dans le répertoire TP, quel sera l'affichage de la commande `ls ~/*` ?

Question 5

Dans l'arborescence de la figure 1, on suppose que l'on se trouve dans le répertoire TP2. Quelle commande permet de copier les fichiers `ex2.c` et `IMG216.jpg` dans le répertoire TP2.

Question 6

Vous venez de compiler votre programme C et vous avez généré l'exécutable `a.out`. Que devez-vous faire pour lancer le programme, de n'importe où sans avoir besoin de préfixe ?

- ☐ Ajouter le chemin du répertoire qui contient `a.out` dans la variable `PATH`.
- ☐ Changer les droits d'accès au fichier `a.out`.
- ☐ Il n'y a rien à faire.
- ☐ Copier `a.out` dans le répertoire `/bin/`

Question 7

Quel est le résultat de la commande `echo 'OPI c'est utile.' | wc -l` ?

Question 8

Quelle est la meilleure façon de ne pas afficher les messages d'erreurs d'une commande dans le terminal.

Question 9

Que cherche-t-on à savoir avec la commande :

`ps -ef | grep -c evince`

Question 10

Soit le résultat de la commande `ls -l` suivant :

```
-rw-----. 1 cecile panda 9 16 oct. 14:20 file1
```

Quelles commandes permettent d'ajouter les droits en lecture et en écriture uniquement au groupe panda sur le fichier `file1` ?

- ☐ `chmod g+r,g+w file1`
- ☐ `chmod 660 file1`
- ☐ `chmod o+rw file1`
- ☐ `chmod a+rw file1`
- ☐ `chmod 777 file1`

Question 11

Soit un fichier csv ci-dessous (les champs sont séparés par des tabulations)

3810335 M.	Anes	ABDOU	Inscrit	
3224931 M.	Thomas	AUBERTIER	Non	
2004782 Mme	Cassandra	DELPLANQUE	Inscrit	
3831627 M.	Niccolo	DZIRI	Inscrit	
3493926 M.	Mohamed	ghali	ES-SADEL	Inscrit
3284840 M.	Héline	FAUCHEU	Non	
3590176 Mme	Ilona	GEDEON-AMROUCHE	Inscrit	
4035745 Mme	Sara	GRISSIA	Non	
4210027 Mme	Chadha	HASSINE	Non	
3673118 M.	Francis	JEGOU	Non	

Quel est le résultat de la commande ?

```
cut -f1,5 ../../../../liste_main3_2023_2024.csv | grep Non
```

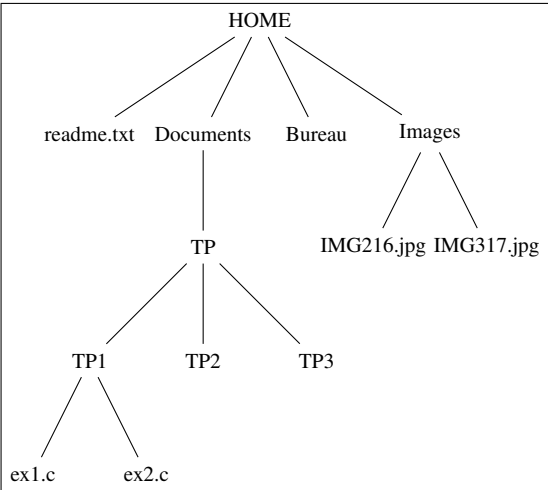


FIGURE 1 –