

INF203 - Travaux pratiques, séance 1

Arborescence, droits d'accès, interpréteur de commandes, shell

On rappelle l'adresse du site web de l'UE, sur laquelle sont régulièrement publiés des documents utiles (attention au **s** de **https**) :

<https://inf203.gricad-pages.univ-grenoble-alpes.fr>

Consultez le document « Consignes importantes pour les TPs » sur ce site avant de commencer le TP. Si ce n'est pas déjà fait, suivez les instructions pour installer l'environnement de travail Visual Studio Code.

Démarrage d'une session MobaXterm et connexion au système Unix

(Si votre machine personnelle est correctement configurée, vous pouvez aussi télécharger les fichiers du TP sur le site web de l'UE et décompresser l'archive (le fichier zip) dans votre répertoire de travail sur votre machine.)

Pendant tout le semestre, vous n'utiliserez pas directement le poste qui est devant vous (PC sous Windows), mais vous travaillerez dans un environnement *Unix*, qui s'exécute sur un ordinateur distant appelé **turing**.

Pour accéder à l'environnement Unix :

1. Ouvrez une session Windows en saisissant une première fois vos identifiants.
2. Démarrez le programme MobaXterm (ignorez les éventuels avertissements de sécurité).
3. Cliquez sur le menu « Sessions », puis « New session ».
4. L'application vous demande « Choose a session type », choisissez « SSH ».
5. Dans la case « Remote host », entrez l'adresse du serveur : **im2ag-turing.univ-grenoble-alpes.fr**
6. Cochez la case « Specify username », et entrez votre identifiant **uga** (en général : nom + 1ère lettre du prénom). Cliquez sur OK.
7. Entrez votre mot de passe. (*Note : pour des raisons de sécurité, rien ne s'affiche à l'écran quand vous tapez votre mot de passe.*) Vous pouvez le mémoriser (mais il faudra définir un autre mot de passe, utile si vous changez de machine).

Une console de commande doit s'afficher sur l'écran, avec une invite de la forme **im2ag-turing:~\$**

Pour modifier vos fichiers, vous avez principalement deux possibilités :

- lancer le logiciel gedit ou nedit en ligne de commande
- ouvrir les fichiers en parcourant l'arborescence située à gauche de la fenêtre (choisir Autosave la première fois que vous enregistrez un fichier)

1 Premières commandes Unix

L'objectif de cette première partie est de se familiariser avec les commandes vues en Cours et en TD, rappelées ici :

echo	affiche ses arguments	mkdir	crée un nouveau répertoire
man	affiche le manuel d'une commande	cp	copie des fichiers ou des répertoires
cd	change de répertoire courant	mv	déplace et/ou renomme un fichier ou un répertoire
ls	liste le contenu d'un répertoire	rm	supprime des fichiers
pwd	affiche le nom du répertoire courant	rmdir	supprime un répertoire vide
cat	affiche le contenu d'un fichier	touch	met à jour la date d'un fichier (le crée si nécessaire)

[a] Que se passe-t-il si l'utilisateur tape un nom de commande inexistante, comme par exemple **xyz32** ? ■

Arborescence et chemins

Vous êtes initialement dans votre répertoire principal.

Remontez dans le répertoire parent et observez son contenu.

Remontez à nouveau dans le répertoire parent et observez également son contenu.

[b] D'après ces observations, expliquez comment sont organisés les répertoires principaux des étudiants du DLST sur **turing**.

Donnez deux chemins, l'un absolu et l'autre relatif, permettant d'accéder au répertoire principal de votre binôme à partir du vôtre. ■

Organisons maintenant notre espace de travail :

— Revenez dans votre répertoire principal.

— Créez un répertoire **INF203**. Placez vous-y.

— Copiez le répertoire **/Public/203_INF_Public/TP1** et son contenu dans le répertoire courant, à l'aide de la commande **cp** et de son option **-R**.

(Alternativement, faites en sorte que les fichiers téléchargés soient placés dans le répertoire **~/INF203/TP1** sur votre machine. Il vous faudra peut-être créer ou renommer des répertoires, déplacer des fichiers...)

[c] Vous répondrez à cette question au fur et à mesure du TP : il s'agit de dessiner l'évolution de l'arborescence de fichiers de racine **INF203** lors de chaque création ou suppression de fichiers et de répertoires (pour les suppressions, vous barrez au crayon).

Pour commencer ce dessin, vous écrivez donc simplement **INF203** au milieu de la page, en-dessous duquel vous placez **TP1** et vous laissez suffisamment de place en dessous (environ 1/2 page).

Attention : ne dessinez pas l'arborescence du répertoire **bottedefoin**, qui servira dans un exercice complémentaire plus loin. Vous comprendrez vite pourquoi... ■

Effectuez les opérations suivantes :

— entrez dans le répertoire **TP1** ;

— supprimez le fichier **a.supprimer** ;

— créez une copie de **fich1** portant le nom **copie-fich1** ;

— renommez cette copie en **copie2-fich1** ;

— créez un répertoire **monrep** ;

— déplacez le fichier **fich2** dans **monrep**.

[d] Comment vérifiez-vous que ces manipulations se sont déroulées comme prévu ? ■

Essayez de supprimer le répertoire **monrep**.

En cas de message d'erreur, expliquez le problème, et recherchez la solution dans le manuel de **rm**.

[e] Comment faites-vous pour supprimer **monrep** en une seule commande ? ■

Quelques facilités d'écriture des commandes

Pour ne pas taper les commandes intégralement

La touche  ("Tabulation") permet dans certains cas de compléter un nom automatiquement :

[programmes] Placez-vous dans le répertoire **~/INF203/TP1/programmes** et tapez successivement sur les touches : **ls -l pg** 

[f] Que se passe-t-il ?

Tapez une seconde fois sur . Que voyez-vous sur l'écran ?

Quel est le plus court préfixe de **donneespgcd** qu'il faut saisir avant  pour que ce nom de fichier soit complété sans ambiguïté ? ■

Dupliquez **donneespgcd** en un fichier nommé **donquichotte**, et refaites l'expérience ci-dessus.

[g] Quel est le plus court préfixe désormais nécessaire ? ■

Par ailleurs, certaines commandes sont utilisées souvent. On retrouve les commandes précédentes avec les touches “flèches”.

[h] Quelle était la quatrième précédente commande que vous aviez exécutée ? ■

Métacaractères

Utilisez les caractères spéciaux `?` ou `*` pour :

- Copier `pgcd1.c` et `pgcd1.p` dans le répertoire `Prog1`.
- Copier `progquiboucle.p`, `pgcd1.p` et `pgcd2.p` dans le répertoire `Pascal`.
- Copier `pgcd1.c`, `pgcd2.c`, `pgcd1.p` et `pgcd2.p` dans le répertoire `Pgcd`.
- Effacer tous les fichiers dont le nom *se termine par* les caractères `pgcd`.
- Lister tous les fichiers dont le nom *contient* les caractères `qui`.

[TP1] Créez un répertoire `Etoile`, placez-vous dans ce répertoire, et exécutez la commande

```
touch LLL
```

Ceci a pour effet de créer un fichier `LLL` vide : vérifiez-le. Si votre répertoire courant comporte autre chose que le fichier `LLL`, vous vous êtes trompés quelque part, recommencez.

[i] Pour savoir ce que vaut `*` maintenant, exécutez :

```
echo *
```

Que signifie le message obtenu si vous exécutez la commande

```
cp *
```

? ■

[j] Répondez de nouveau à la question précédente dans les cas suivants :

1. le répertoire courant contient uniquement les fichiers `LLL` et `MMM`.
2. le répertoire courant contient uniquement les fichiers `LLL`, `MMM` et `NNN`.
3. le répertoire courant contient uniquement les fichiers `LLL`, `MMM`, `NNN` et un répertoire `AAA`.
4. le répertoire courant contient uniquement les fichiers `LLL`, `MMM`, `NNN` et un répertoire `ZZZ`.

Expliquez à chaque fois ce qui se produit. ■

Exercice complémentaire :

Une aiguille dans une botte de foin

Effectuez les opérations suivantes dans l'ordre, sans changer de répertoire en cours de route. Il est possible que certaines opérations affichent des messages d'erreur, vous pouvez les ignorer.

- Créer quatre répertoires portant respectivement les noms `g`, `u`, `e`, `s`.
- Copier le fichier `le` du sous-répertoire `gui` du répertoire `ille` dans le répertoire `ll`.
- Déplacer tout le contenu de tous les sous-répertoires du répertoire `uil` dans le répertoire courant.
- Créer un fichier `ai` dans le répertoire `s`.
- Déplacer dans le répertoire `ig` tous les fichiers et répertoires dont le nom commence par `u`.
- Déplacer dans `aigu` tous les fichiers et répertoires dont le nom se termine par `g` ou contient un `e`.
- Supprimer tous les fichiers et répertoires dont le nom comporte 3 lettres ou plus et leur contenu.
- Déplacez dans le répertoire courant tout le contenu des sous-répertoires du répertoire courant.
- Supprimez tous les répertoires vides.

Si vous avez effectué toutes les étapes correctement, la commande `ls` doit afficher le mot `aiguilles` (avec quelques espaces au milieu). Si ce n'est pas le cas, supprimez votre répertoire `bottedefoin`, recopiez à nouveau celui de `/Public/203_INF_Public/TP1` et recommencez la manœuvre.

[k] Affichez le contenu de tous les fichiers du répertoire courant. Que lisez-vous ? ■

2 Fichiers de commandes

Dans cette seconde partie, nous découvrons les *fichiers de commande* (ou *shell-scripts*). Ils regroupent une suite de commandes, que l'on pourra exécuter en une seule fois, en exécutant le script lui-même.

[~/INF203/TP1/scripts] Ouvrez le fichier **sauvegarde.sh** dans l'éditeur de texte de MobaXterm.

Observez le contenu de **sauvegarde.sh** et essayez de prévoir ce qui se passera lors de son exécution.

[l] Essayez ensuite d'exécuter ce fichier en saisissant dans le terminal son nom précédé par les caractères point et slash (**./sauvegarde.sh**).

Quel message d'erreur obtenez-vous ? Notez-le. ■

Exécutez la commande **ls -l sauvegarde.sh**. Parmi les différents champs du résultat, on trouve le nom du fichier, sa date de création, sa taille et ses droits d'accès.

[m] Notez soigneusement ces 4 informations. Où retrouvez-vous la raison du message d'erreur précédent ? ■

Donnez-vous les droits d'exécution sur ce fichier en utilisant la commande **chmod**, et vérifiez que les droits d'accès sur ce fichier ont bien changé.

Exécutez maintenant à nouveau **./sauvegarde.sh**.

[n] Quel est maintenant votre répertoire de travail ? Pourquoi ce répertoire n'est-il pas le répertoire **sauve_TP1** ? ■

Arguments de la ligne de commande

[scripts] Lisez le texte du programme **args.sh**, et exécutez les commandes suivantes :

```
./args.sh 12 abc x
./args.sh inf 203
./args.sh inf203
```

[o] Que représentent les expressions **\$1**, **\$2**, **\$3** et **\$#** ? Et que représente l'expression **\$*** ? (modifiez **args.sh** pour le savoir !) ■

Observez maintenant le contenu du fichier **installeTP.sh**.

[p] Comment utilisez-vous ce script pour recopier le répertoire **/Public/203_INF_Public/TP1bis** dans votre répertoire **INF203** ? ■

Conservez précieusement ce script, vous l'utiliserez au début de chaque séance pour récupérer les fichiers du TP courant. Faites-en une copie dans **~/INF203** pour plus de commodité. Si vous travaillez en local, modifiez-le pour plutôt récupérer les fichiers depuis votre répertoire de téléchargement habituel.

Écriture d'un fichier de commandes

Écrire un script **sauve_fichiers.sh** dont le comportement est le suivant : si on tape

```
./sauve_fichiers.sh rep suffixe motif
```

— on suppose que le répertoire courant contient un répertoire de nom **rep** ;

— un répertoire dont le nom est **rep** suivi de **suffixe** est créé (dans le répertoire courant) ;

— tous les fichiers (pas les répertoires) de **rep** dont le nom contient le **motif** sont copiés dans le nouveau répertoire.

[q] Joignez le texte de votre script à votre compte-rendu. ■

3 La commande de la semaine : grep

[~/INF203/TP1] La commande **grep** permet de rechercher (et afficher) les lignes d'un fichier contenant une chaîne de caractères donnée. Essayez par exemple la commande suivante : **grep** **Candide** **Candide_chapitre1.txt**, qui recherche la chaîne “Candide” dans le fichier **Candide_chapitre1.txt**.

L'option **-c** de **grep** permet de n'afficher que le nombre total de lignes contenant la chaîne recherchée (et non les lignes elle-mêmes). L'option **-v** de **grep** permet d'inverser la recherche, c'est-à-dire de n'afficher que les lignes ne contenant pas la chaîne donnée.

Expérimentez avec grep, en exécutant par exemple les commandes suivantes :

```
grep "mm" Candide_chapitre1.txt
grep "mmm" Candide_chapitre1.txt
grep "Ce" Candide_chapitre1.txt
grep -v "a" Candide_chapitre1.txt
```

[r] Combien de lignes contiennent le mot “Candide” dans ce fichier ? Et le mot “Cunegonde” ? ■