

## SGF et commandes Shell

### A LIRE avant de commencer :

Le but de ces TP est de vous familiariser avec le Shell, vous devez vous connecter sous le système d'exploitation Linux. Pour tous les exercices de TP, vous devez utiliser des **commandes** Shell (et non des outils accessibles via l'interface graphique).

Les commandes que vous pouvez utiliser sont décrites dans le document "*commandes.pdf*" à votre disposition sous Moodle. Vous pouvez aussi consulter le manuel en ligne d'une commande en utilisant la commande `man`, citée en cours et décrite dans le document "*commandes.pdf*".

Durant les TP, prenez l'habitude de sauvegarder vos réponses aux exercices (commandes, puis scripts lors des prochaines séances) dans un fichier texte – en les copiant-collant dans un fichier à l'aide d'un éditeur de texte du type nano ou Kate – pour garder trace de votre travail en séance et pouvoir le continuer chez vous ou réviser. Vous pouvez aussi déposer vos fichiers dans le dépôt Moodle prévu à cet effet pour les récupérer depuis une autre machine.

La commande `history` permet d'obtenir la liste des dernières commandes soumises au Shell courant. Vous pouvez soumettre à nouveau une commande de cet historique en tapant `!` suivi de son numéro dans cette liste.

En tapant sur les flèches `↑` `↓` du clavier, il est possible de faire défiler les commandes précédemment soumises au Shell courant, pour éventuellement les soumettre à nouveau.

Si vous n'avez pas terminé en séance, vous **devez** terminer ces exercices. Vous en aurez besoin lors des TD/TP suivants.

Si vous n'avez pas de PC personnel, des cartes d'accès aux salles de TP sont disponibles au secrétariat pour travailler en autonomie.

Si vous avez un PC personnel :

- sous Windows, vous pouvez utiliser des terminaux en ligne ou en émuler (voir la page Moodle du cours)
- sous Linux ou un Mac, vous avez déjà le bon environnement.

Un environnement Linux (ou l'équivalent sous Mac ou en machine virtuelle sous Windows) vous sera utile durant les 3 ans de votre licence)

Enfin, n'oubliez pas que ChatGPT et autres IA ne sont pas forcément vos amis, risquent de vous donner des réponses "hors contexte" et ne réfléchiront pas à votre place durant les contrôles sur table donc n'hésitez pas à travailler sans ces outils et à poser vos questions à votre enseignant de TP si vous êtes bloqué.e...

👁 **Principales commandes utiles pour cette séance :** `cat`, `cd`, `chmod`, `cp`, `cut`, `echo`, `grep`, `head`, `ls`, `mkdir`, `more`, `mv`, `pwd`, `rm`, `rmdir`, `sort`, `tail`, `tr`.

### Exercice 1

Ouvrez un terminal (qui sera identifié par T1 dans la suite de cet énoncé) en utilisant l'interface graphique. Positionnez-vous dans ce terminal T1 et à l'aide de la commande `xterm`, ouvrez un second terminal (identifié par T2).

Dans le terminal T1, tapez la commande `ls`. Est-ce possible ? Pourquoi d'après vous ?

Tapez la commande `exit` dans le terminal T2. Que se passe-t-il ? Pouvez-vous taper la commande `ls` dans le terminal T1 ? Pourquoi ?

Dans le terminal T1, tapez la commande `xterm` suivie du caractère `&` pour créer un nouveau terminal (identifié par T3).

Tapez la commande `ls` dans le terminal T1. Est-ce possible ? Cela est-il aussi possible dans le terminal T3 ? Pourquoi ?

Fermez ce terminal T3.

Depuis le terminal T1, réalisez les actions suivantes :

1. Utilisez la commande `ls` (avec d'éventuelles options) pour visualiser les informations **détaillées** des fichiers se trouvant sous votre répertoire d'accueil.
2. À l'aide de la commande `mkdir`, et en utilisant le **moins** de commandes possibles i.e. la bonne option, créez (dans votre répertoire d'accueil) l'arborescence de fichiers (répertoires) de la figure 1.

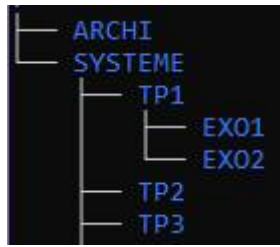


Figure 1

3. Vérifiez, grâce à la commande `tree`, que vous obtenez bien la même arborescence (il est normal que des répertoires Desktop ou Download, faisant partie de votre environnement soient aussi visibles).

## Exercice 2

Grâce à la commande `cd`, placez-vous dans le répertoire *EX01* dans lequel vous allez réaliser cet exercice. Trouvez la commande pour vérifier (en le faisant afficher) que le **chemin d'accès** à votre répertoire courant (ou répertoire de travail) est le bon.

**À noter :** Le répertoire de travail pour cet exercice est le répertoire *EX01*, vous ne devez pas en "bouger" pour réaliser les traitements qui suivent (en d'autres termes, il n'y **aucun besoin de la commande `cd`** pour ces traitements).

1. Trouvez la commande pour lister le détail des informations de tous les fichiers qui se trouvent dans le répertoire *SYSTEME*.
2. Dans un éditeur de textes de votre choix (nano, Kate, atom...), écrivez le texte suivant :

```
Un jour un Coq détourna
Une Perle, qu'il donna
Au beau premier Lapidaire.
```

et sauvegardez-le dans un fichier, sous votre répertoire courant de travail, en lui donnant le nom *exo1.txt*. Votre arborescence doit devenir celle de la figure 2 :

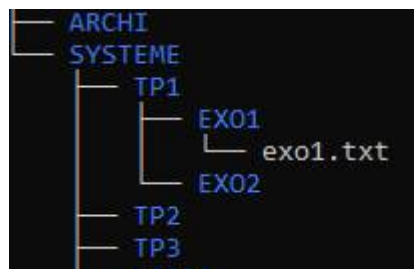


Figure 2

3. À l'aide de la commande `cat`, faites afficher le contenu de ce fichier à l'écran. Corrigez le contenu du fichier si vous constatez que le fichier se termine par une ligne vide.
4. Faites afficher les caractéristiques de ce fichier (taille, propriétaire, droits d'accès, etc.). Quelle taille fait-il ? À qui appartient-il ? Peut-on le lire ? Peut-on y écrire ? À quelle date a-t-il été modifié pour la dernière fois ?
5. Sans utiliser l'éditeur de textes, à l'aide de la commande `echo`, ajoutez le texte suivant en fin du fichier *exo1.txt* :

```
"Je la crois fine, dit-il ;
Mais le moindre grain de mil
Serait bien mieux mon affaire."
...
"A l'époque, on ne parlait pas en $", songea le Coq...
```

Et vérifiez avec la commande `cat` que le contenu de `exo1.txt` est celui espéré.  
Si vous éprouvez des difficultés, lisez la remarque en fin de cet exercice (et le cours).

6. À l'aide de la commande `grep`, faites afficher à l'écran :
  - Toutes les lignes qui contiennent la chaîne "ai".
  - Toutes les lignes qui contiennent la chaîne "il" suivie d'un espace.
  - Toutes les lignes qui contiennent un point.
7. À l'aide de la commande `wc`, faites afficher le nombre de lignes contenues dans ce fichier. Comment peut-on faire pour éviter que le nom de fichier apparaisse aussi en résultat ? *Indice* : en utilisant une autre commande, vue en cours, il faut arriver à conserver (extraire) seulement le nombre de lignes qui apparaît en premier dans le résultat.
8. Sans changer de répertoire, à l'aide de la commande `cp`, copiez ce fichier dans le répertoire `EXO2` en lui donnant le nom `exo1b.txt`. Grâce à la commande `ls`, vérifiez que le répertoire contient bien ce fichier.
9. Toujours sans changer de répertoire, grâce à la commande `sort`, faites afficher le contenu de `exo1b.txt` en classant les lignes par **ordre alphabétique** sur le premier mot.

Vous devez obtenir l'affichage suivant :

```
"A l'époque, on ne parlait pas en $", songea le Coq...
"Je la crois fine, dit-il ;
...
Au beau premier Lapidaire.
Mais le moindre grain de mil
Serait bien mieux mon affaire."
Un jour un Coq détourna
Une Perle, qu'il donna
```

Attention : le résultat de `sort` dépend de la langue de l'environnement - voir remarque en fin du manuel en ligne. Il se peut donc que la commande ne prenne pas en compte les signes de la ponctuation française et que votre résultat diffère légèrement.

10. Sans bouger du répertoire `EXO1` courant, faites de même en les classant dans l'ordre inverse (i.e. par ordre "**décroissant**") sur le **second** mot de chaque ligne. Au lieu d'afficher le résultat, envoyez-le dans un fichier `exo1classe.txt` situé sous le répertoire `TP3`. Faites afficher à l'écran le contenu de ce fichier avant de le supprimer de l'endroit où il se trouve.

Le contenu de `exo1classe.txt` sous `TP3` doit être le suivant :

```
Mais le moindre grain de mil
"Je la crois fine, dit-il ;
"A l'époque on ne parlait pas en $", songea le Coq...
Un jour un Coq detourna
Serait bien mieux mon affaire."
Au beau premier Lapidaire.
Une Perle, qu'il donna
...
```

11. À l'aide de la commande `mv`, toujours sans changer de répertoire courant, **déplacez** le fichier `exo1b.txt`, dans le répertoire `ARCHI`, en lui donnant comme nom `archi-exo1b.txt`. À l'aide d'**une seule** commande, vérifiez que `exo1b.txt` n'est plus dans `EXO2` et que le fichier `archi-exo1b.txt` est bien présent sous `ARCHI`. Puis, supprimez le fichier `archi-exo1b.txt`.

Remarques :

- Le métacaractère `\` annihile l'évaluation du caractère immédiatement suivant, ce qui n'est intéressant que dans le cas où ce caractère est un métacaractère du shell.
- Si un délimiteur `'`, `"` ou ``` est « célibataire » sur sa ligne de commande, alors l'interpréteur de commandes affiche un prompt `>` qui invite l'utilisateur à terminer la commande en cours de frappe. La commande n'est réellement interprétée que lorsque les délimiteurs forment des paires.
- En cas de problème, l'exécution d'une commande peut être interrompue en tapant la combinaison de touches `<Ctrl>C`

### Exercice 3

Placez-vous dans le répertoire *EXO2*. En utilisant une commande, vérifiez que vous êtes dans le bon répertoire.

**À noter :** Le répertoire *EXO2* devient votre répertoire de travail pour cet exercice, vous ne devez pas en "bouger" pour réaliser les traitements qui suivent (en d'autres termes, il n'y a aucun besoin de la commande `cd` pour ces traitements).

Récupérez sous Moodle, le fichier *exo2.tar*. Normalement, ce fichier devrait être téléchargé dans le répertoire "Téléchargement" (ou "Download") qui se trouve sous votre répertoire d'accueil (sinon, trouvez-le).

Sans bouger de votre répertoire de travail *EXO2* et **sans utiliser une interface graphique**, à l'aide de la commande `tar`, **décompressez** le fichier *exo2.tar* pour ajouter la sous-arborescence qui s'y trouve au répertoire *SYSTEME* et obtenir l'arborescence de la figure 3.

Vous pouvez connaître les options de cette commande en faisant : `tar --help`.

Vous devrez utiliser les options : `-xvf` (`x` = extract, `v` = verbose, `f` = file) pour décompresser le fichier (donné en argument de l'option `-f`) en détaillant la liste des fichiers extraits, et `--directory` (ou `-C`) pour le faire vers le répertoire que vous donnerez en argument de cette option.

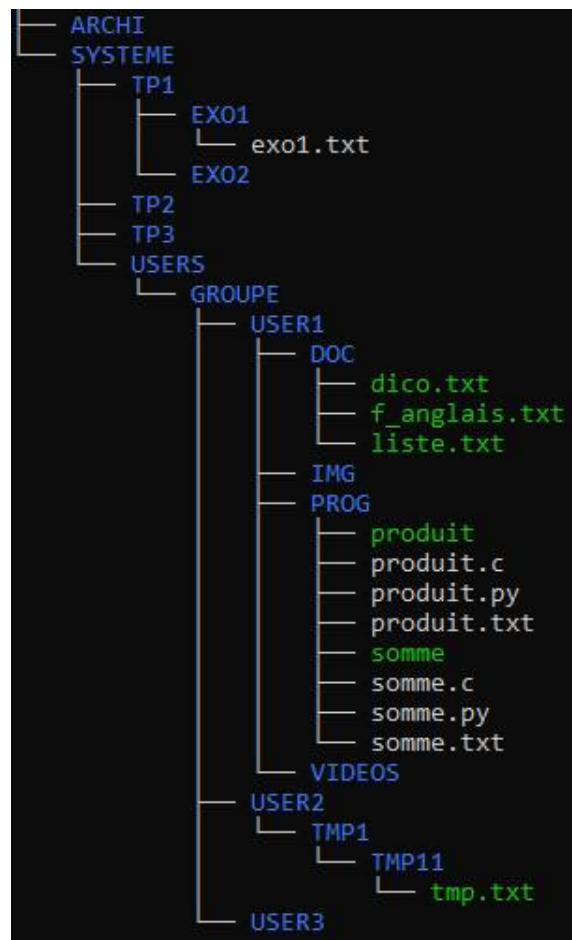


Figure 3

1. Toujours sans bouger du répertoire *EXO2*, vérifiez avec la commande `tree` que vous avez obtenu la bonne arborescence.

2. Faites afficher les caractéristiques détaillées des fichiers qui se trouvent sous le répertoire *PROG* (de la sous-arborescence ajoutée) et :
  - dont le nom comporte la chaîne "somme"
  - ayant une extension *.c*
3. En une commande *cp*, **recopiez** dans votre répertoire de travail tous les fichiers (du même répertoire *PROG*) dont l'extension est *.py*. Vérifiez le résultat.
4. En une commande *mv*, **déplacez** localement le fichier *tmp.txt* qui se trouve dans le répertoire *TMP11* (de la sous-arborescence précédemment ajoutée) en le **renommant** *montmp.txt*.
5. Faites afficher les caractéristiques détaillées de ce fichier *montmp.txt*.
6. Grâce à la commande *chmod*, **enlever le droit d'écriture** du fichier *montmp.txt*.
7. En utilisant un éditeur de textes, essayez d'en modifier le contenu. Que constatez-vous ?
8. En utilisant la commande *rm*, essayez de le supprimer. Que constatez-vous ?
9. En utilisant la commande *echo*, ajoutez le texte "Goodbye World !" en fin de ce fichier. Que constatez-vous ?
10. Rétablissez le droit d'écriture et **supprimez** le fichier *montmp.txt*.

#### Exercice 4

Sous le répertoire *TP1*, créez le répertoire *EXO3* pour obtenir la sous-arborescence de la figure 4.

**À noter :** Le répertoire *EXO3* devient votre répertoire de travail pour cet exercice, vous ne devez pas en "bouger" pour réaliser les traitements qui suivent (en d'autres termes, il n'y **aucun besoin de la commande** *cd* pour ces traitements).

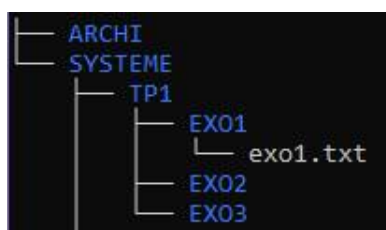


Figure 4

Le fichier *liste.txt*, qui se trouve sous le répertoire *DOC* (de la sous-arborescence ajoutée dans l'exercice précédent), est constitué d'abord de 5 lignes de description puis d'une liste d'étudiants. Chaque ligne relative à un étudiant est constituée de 3 champs : Nom, Prénom et Login qui sont séparés par un espace.

Le but de cet exercice est d'utiliser les **métacaractères des expressions régulières** pour afficher ou trier certaines lignes de ce fichier.

1. Sans bouger du répertoire *EXO3*, faites afficher le contenu du fichier *liste.txt*.
2. Faites afficher la liste des étudiants dont le nom commence par "LA".
3. Faites afficher la liste des étudiants ayant le prénom "LAURENT".
4. Faites afficher la liste des étudiants ayant un login dont le numéro est compris entre 10 et 99.
5. Faites afficher la liste des étudiants ayant un login dont le numéro est inférieur à 100 (on aura intérêt à utiliser l'option *-v* de la commande *grep*). Avant de continuer, vérifiez que toutes les lignes de votre résultat décrivent bien des étudiants. Si ce n'est pas le cas, grâce à quelle commande peut-on ne pas considérer les lignes en trop lors du tri ?
6. Faites afficher cette liste triée par ordre **croissant** de ce numéro.
7. Recopiez localement le fichier *liste.txt*. Faites afficher la liste de tous les étudiants par ordre **décroissant** de numéro de login. Pour éviter que le numéro 100 n'apparaisse juste avant le numéro 10, on conseille de lancer plusieurs commandes successivement, en utilisant l'algorithme suivant :
  - Trier les étudiants dont le numéro comporte *trois chiffres exactement* et écrire le résultat dans un fichier *exo3.txt* local à votre répertoire de travail *EXO3*.
  - Trier les étudiants dont le numéro comporte *deux chiffres exactement*, et les rajouter au fichier précédent.
  - Trier les étudiants dont le numéro comporte *un seul chiffre*, et les rajouter au fichier précédent.
8. Vérifier que le contenu du fichier *exo3.txt* est correct. Puis, quand c'est le cas, supprimez-le.

## Exercice 5

Sous le répertoire *TP1*, créez le répertoire *EXO4* pour obtenir la sous-arborescence de la figure 5.

**À noter :** Le répertoire *EXO4* devient votre répertoire de travail pour cet exercice, vous ne devez pas en "bouger" pour réaliser les traitements qui suivent (en d'autres termes, il n'y **aucun besoin de la commande `cd`** pour ces traitements).

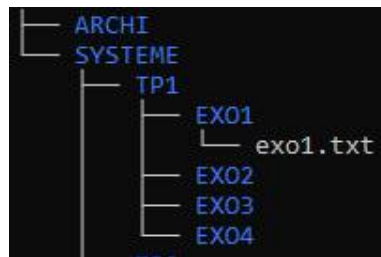


Figure 5

1. Récupérez le fichier *exo4.tar* sous Moodle et décompressez-le, comme dans l'exercice 3, dans le répertoire courant.
2. Écrivez **une** ligne de commande (combinant `echo`, `ls`, `wc` et `grep`) permettant de dénombrer les **fichiers** du répertoire courant (les sous-répertoires du répertoire courant ne doivent donc pas être comptés) dans les quatre catégories données ci-dessous. *Indices :* Comment peut-on distinguer fichiers et répertoires ? Comment peut-on repérer les fichiers cachés ? Il faudra explorer les options `-c` et `-v` de la commande `grep`, et construire chaque commande petit à petit.
  - Fichiers (*cachés ou non*) dont le nom se termine par `.txt` (il y en a 4)
  - Fichiers (*cachés ou non*) dont le nom se termine par `.sh` (il y en a 5)
  - Fichiers *cachés* (il y en a 4)
  - Autres fichiers (il y en a 6)

Chaque résultat devra être affiché sous la forme suivante :

Il y a 4 fichiers (cachés ou non) dont le nom se termine par `.txt`

## Exercice 6

La commande `head` permet de récupérer les premières lignes d'un fichier. La commande `tail` permet de récupérer les lignes situées à la fin d'un fichier

Remplacez-vous sous le répertoire *EXO3*, et combinez – entre autres – les commandes `head` et `tail` pour écrire **une** seule commande qui affichera :

1. Les 20 derniers étudiants de *liste.txt*
2. Les 5 premières lignes de *liste.txt*
3. Les 20 premiers étudiants de *liste.txt*
4. Le nom (et uniquement le nom) de l'étudiant de *liste.txt* ayant le numéro de login le plus petit i.e. le nom qui se trouvera sur la première ligne de la liste, triée par ordre croissant, des étudiants dont le login comprend un seul chiffre
5. Le prénom (et uniquement le prénom) de l'étudiant de *liste.txt* ayant le numéro de login le plus grand

## Exercice 7

**Pour aller plus loin...**

1. Entraînez-vous en refaisant les exercices vus en TD et donnés en exemple dans votre cours.
2. Vous pouvez réviser les commandes de base en vous amusant avec TerminalTutor (<https://www.terminaltutor.com/>) ou GameShell (<https://linuxfr.org/news/gameshell-apprendre-les-rudiments-du-shell-en-s-amusant>).