



Licence ASRALL

Principes et utilisation Linux

Étienne ANDRÉ

Basé sur des documents de Giulio MANZONETTO (Université Sorbonne Paris Nord)

www.loria.science/andre/enseignement/Linux/



Version : 6 octobre 2021

Table des matières

1	Système de fichiers Linux	3
2	Utilisation de <code>find</code>	8
3	Droits, redirections et liens	9
4	Manipulation de fichiers textes	12
5	Commandes réseaux	15
6	Exercices de révisions	16
7	Gestion de processus & variables d'environnement	18
8	Introduction à la programmation de scripts shell	21
9	Utilisation de <code>sed</code>	23

TP 1

Système de fichiers Linux



Exercice 1.1 (Système de fichiers). Ouvrir un terminal (menu : accessoires → terminal). Les exercices suivants sont à faire en utilisant exclusivement le terminal.



Question 1 : Quel est votre répertoire personnel?



Question 2 : En utilisant la commande `cd`, déplacez-vous dans le répertoire `/tmp`




Question 3 : Créer dans votre répertoire de travail le sous-arbre suivant, en utilisant `touch` et `mkdir` (éventuellement `rm` et `rmdir` pour faire des corrections, et `ls` pour lister).


```
.
|-- coursLinux
|   |-- Linux1.txt
|   |-- Linux2.txt
|   \-- Linux3.txt
|-- alire
|   |-- LPASRALL
|   |   |-- TPLinux.pdf
|   |   |-- TD.pdf
|   |   |-- manuel.html
|   |   \-- reseaux.html
|   |-- notes.txt
|   |-- readme
|   |-- répertoire
|   \-- EdT.pdf
|-- films
|   |-- film1.mp4
|   |-- film2.mp4
|   |-- film3.mp4
|   |-- film4.mp4
|   |-- film5.mp4
|   |-- film6.mp4
|   |-- film7.mp4
|   |-- film8.mp4
|   |-- film9.mp4
|   |-- film10.mp4
|   \-- film11.mp4
```

```
| |-- film12.mp4
| |-- film13.mp4
| |-- film14.mp4
| |-- film15.mp4
| |-- film16.mp4
| |-- film17.mp4
| |-- film18.mp4
| |-- film19.mp4
| \-- film20.mp4
|-- WKW.png
|-- Nolan.jpg
\-- 2046.avi
```


 **Question 4 :** Pour chacun des fichiers suivants, donner son parcours absolu.

1. coursLinux
2. 2046.avi
3. readme
4. reseaux.html


 **Question 5 :** Se déplacer dans le répertoire LPASRALL


 **Question 6 :** Pour chacun des fichiers suivants, donner son parcours relatif au répertoire de travail.

1. coursLinux
2. 2046.avi
3. readme
4. reseaux.html

 **Question 7 :** Pour chacun des fichiers suivants, donner leur parcours relatif à votre répertoire personnel.


1. coursLinux
2. reseaux.html

 **Question 8 :** Copier le fichier WKW.png dans votre répertoire personnel.

 **Question 9 :** Créer les répertoires dir1, dir2 et dir3 dans votre répertoire personnel.

 **Question 10 :** Déplacer le répertoire dir2 dans le répertoire dir1


 **Question 11 :** Renommer le répertoire dir2 en mydirectory


 **Question 12 :** Copier le répertoire coursLinux dans le répertoire dir3


 **Question 13 :** Exécuter en étant dans dir3 la commande suivante :


```
rmdir coursLinux
```

Quel est l'effet de cette commande? Pourquoi? Quelle est la bonne commande à utiliser pour effacer le répertoire?


 **Question 14 :** Déplacer le fichier `WKW.png` dans votre répertoire personnel en le renommant `toto`.


 **Question 15 :** Refaire l'exercice `Q1.1`, cette fois-ci en minimisant le nombre de commandes. On peut le faire en seulement 2 commandes!


 **Exercice 1.2 (Commandes de base).** Pour chacune des questions suivantes, préciser la ou les commandes nécessaires à son exécution et donner le résultat obtenu.


 **Question 1 :** Lancer un terminal (un interpréteur de commandes). Quel est votre répertoire personnel?

 **Question 2 :** Dans votre répertoire personnel, créer un répertoire nommé `tp_intro_sys`


 **Question 3 :** Afficher le contenu du répertoire que vous venez de créer.


 **Question 4 :** À l'aide de la commande `touch`, créer un fichier nommé `toto`. Quel est la taille de ce fichier?


 **Question 5 :** Déplacer le fichier `toto` dans le répertoire `tp_intro_sys`.


 **Question 6 :** Copier le fichier `toto` dans votre répertoire personnel en le renommant `toto.copie`


 **Question 7 :** Effacer le fichier `toto.copie`


 **Question 8 :** Aller vers le répertoire `tp_intro_sys`. Quel est maintenant votre répertoire de travail? Et votre répertoire personnel?


 **Question 9 :** Placez-vous dans le répertoire `/etc`.

 **Question 10 :** Créer dans le répertoire `/etc` un fichier vide `essai`. Justifier le résultat obtenu.

 **Question 11 :** Donner *trois* commandes différentes pour revenir dans votre répertoire personnel.

 **Question 12 :** Créer un répertoire `tp1sys` dans le répertoire `tp_intro_sys`. Copier le fichier `toto` dans ce répertoire.

 **Question 13 :** Donner une commande qui permet de dupliquer le répertoire `tp_intro_sys` vers `tp_intro_sys_copie`.

 **Exercice 1.3 (Interprétation des commandes).** Se placer dans votre répertoire personnel, et y revenir après chaque question.

Pour chacune des commandes suivantes, expliquer le résultat de son exécution, et proposer une meilleure alternative si possible.

 **Question 1 :**

```
cd ../..
```

 **Question 2 :**

```
ls .
```

 **Question 3 :**

```
cp /etc/passwd .
```

 **Question 4 :**

```
cp /etc/hosts ./tp_intro_sys/..
```

 **Question 5 :**


```
ls ../../../../../../etc
```


 **Question 6 :**


```
ls /home/./usr/./bin/
```





Exercice 1.4 (Utilisation des méta-caractères). Se placer dans votre répertoire personnel pour exécuter les commandes suivantes. *L'usage de la commande `find` n'est pas demandé.*

 **Question 1 :** Copier dans le répertoire `tp_intro_sys` tous les fichiers qui se trouvent dans `/usr/include` dont le nom commence par « a » et qui ont l'extension « .h ».

 **Question 2 :** Effacer du répertoire `tp_intro_sys` les fichiers ayant l'extension `.h`

 **Question 3 :** Afficher la liste de tous les fichiers du répertoire `/usr/include` qui ont une extension et dont le nom, sans compter l'extension, comporte exactement 4 caractères.


 **Question 4 :** Copier dans `tp_intro_sys` les fichiers qui se trouvent dans `/usr/include` et dont le nom commence par « i », « u » ou « t ».

 **Question 5 :** Créer dans le répertoire `tp_intro_sys` les fichiers `1a.txt`, `1b.txt`, `1c.txt` et `1d.txt`


 **Question 6 :** Exécuter la commande suivante :

```
rmdir tp_intro_sys
```


Quel est le résultat de cette commande?


 **Question 7 :** Vider le répertoire `tp_intro_sys_copie` créé à la question 1.2.

 **Question 8 :** Effacer le répertoire `tp_intro_sys_copie`.

 **Question 9 :** Expliquer ce que fait la commande suivante :

```
ls *
```

 **Question 10 :** Comment lister *récursivement* en utilisant la commande `ls` (avec une option) tous les répertoires dans le répertoire courant ainsi que tous leurs sous-répertoires (de façon récursive)?

 **Question 11 :** Expliquer ce que fait la commande suivante :

```
touch ~/a{1..3}{d..g}.txt
```

TP 2

Utilisation de `find`



Exercice 2.1 (Recherche de fichiers). La commande `find` sera utilisée pour réaliser l'ensemble des questions suivantes.



Question 1 : Trouver tous les fichiers qui sont dans la sous-arborescence `/usr`, dont le nom commence par `std` et qui ont l'extension `.h`.



Question 2 : Afficher les informations détaillées (c'est-à-dire `ls -l`) sur les fichiers trouvés dans la question précédente.



Question 3 : Copier ces fichiers dans `tp1sys`.



Question 4 : Afficher les informations détaillées (c'est-à-dire `ls -l`) sur tous les répertoires qui sont dans l'arborescence de votre répertoire personnel.



Question 5 : Trouver tous les fichiers qui ont une taille de 0 octet et qui ont été créés aujourd'hui.



Question 6 : Trouver tous les fichiers dans l'arborescence de votre répertoire personnel qui sont antérieurs au fichier `toto`.



Question 7 : Trouver tous les fichiers dans l'arborescence de votre répertoire personnel qui ont été modifiés depuis moins de 24 heures, et dont la taille est supérieure à 1 Kio.



Question 8 : Expliquer la différence entre les deux commandes suivantes :

```
ls -l ~/*.txt
```

```
find ~ -name "*.txt" -exec ls -l {} \;
```

TP 3

Droits, redirections et liens



Exercice 3.1 (Droits, utilisateurs et groupes).



Question 1 : À l'aide de la commande `cat` et des opérateurs de redirection, créer un fichier `carte_visite.txt` dans votre répertoire personnel. Ce fichier doit contenir sur trois lignes les informations suivantes : votre nom, prénom et adresse électronique. (Il doit être rempli en ligne de commande et non pas avec un éditeur graphique, ni même `vi`.)



Question 2 : Afficher le contenu de ce fichier.



Question 3 : À l'aide des commandes `id` et `ls -l`, déterminer si votre voisin peut lire ou éditer ce fichier.



Question 4 : Que fait la commande `whoami` ? Quelle est l'utilité d'une telle commande ?



Question 5 : Copier le fichier `carte_visite.txt` dans les répertoires `/tmp` et `/etc`. Justifier les résultats obtenus.



Question 6 : Quels sont les droits d'accès du fichier `carte_visite.txt` ?



Question 7 : Donner aux utilisateurs qui appartiennent au même groupe que vous le droit de lire et de modifier ce fichier. Exprimer ces droits de deux manières : code octal/approche symbolique.



Question 8 : À l'aide de la commande `umask`, changer les droits par défaut sur vos fichiers pour qu'ils soient en lecture/écriture exclusives pour vous seulement. Exprimer les nouveaux droits d'accès de deux manières : code octal et approche symbolique.



Question 9 : Vérifier à nouveau les droits associés au fichier `carte_visite.txt`. Justifier votre réponse.




Question 10 : Donner une commande qui permet de trouver tous les fichiers dans votre compte qui sont accessibles par les autres utilisateurs en mode écriture.



Question 11 : Faire une copie du fichier `carte_visite.txt` en le renommant `carte_visite.copie`. Quels sont les droits associés au nouveau fichier ? Justifier.



Question 12 : Remettre `umask` à sa valeur usuelle.


 **Question 13 :** Créer un répertoire nommé « secret » tel que vous ne puissiez pas lister son contenu mais puissiez lire les fichiers qui y sont placés. Créer un fichier `film.mp4` dans ce répertoire. On devra alors obtenir :

```
$ ls secret
ls: secret: Permission denied
$ ls secret/film.mp4
secret/film.mp4
```





Exercice 3.2 (Groupes).


 **Question 1 :** Lister tous les groupes auxquels *vous* appartenez.

 **Question 2 :** Lister tous les groupes présents sur le système.

 **Question 3 :** Lister tous les utilisateurs appartenant au groupe `sudo`.


 **Question 4 :** Ajouter un nouveau groupe `musique` au système.

 **Question 5 :** Ajouter votre login au groupe `musique`. Vérifier que vous appartenez bien à ce groupe.


 **Question 6 :** Supprimer le groupe `musique`. Vérifier que vous n'appartenez plus à ce groupe.





Exercice 3.3 (Opérateurs de redirection entrée/sortie).


 **Question 1 :** Donner une commande qui permet de remplacer le contenu du fichier `carte_visite.txt` avec les noms des fichiers qui se trouvent dans le répertoire `/usr/include` et qui ont l'extension `.h` et dont le nom est composé de trois caractères seulement.


 **Question 2 :** Créer une copie de ce fichier sous le nom `f3.copie`

 **Question 3 :** Donner une commande qui permet d'ajouter à la fin du fichier `carte_visite.txt` la ligne `*** fin de fichier ***`

 **Question 4 :** Utiliser la commande `diff` pour comparer les deux fichiers `carte_visite.txt` et `f3.copie`.

 **Question 5 :** Fusionner les deux fichiers `carte_visite.txt` et `f3.copie` en un seul nouveau fichier.

 **Question 6 :** Utiliser la commande `wc` pour donner le nombre de lignes dans chacun des trois fichiers.

 **Question 7 :** Executer la commande `mkdir` sur le répertoire `secret` avec redirection de la sortie d'erreur standard sur un fichier `erreur.txt` dans votre répertoire personnel.



Exercice 3.4 (Manipulation de liens).



Question 1 : Créer un lien physique `carte_visite.phy` qui pointe vers le fichier `carte_visite.txt`. Comment peut-on reconnaître que `carte_visite.phy` est un lien physique?



Question 2 : Créer un lien symbolique `carte_visite.sym` qui pointe vers le fichier `carte_visite.txt`.



Question 3 : En utilisant la commande `man ls` chercher comment visualiser le numéro `i-node` d'un fichier. Ensuite, comparer les numéros `i-node` des trois fichiers créés précédemment. Que constate-t-on?



Question 4 : Ajouter, en utilisant la commande `cat`, une ligne à la fin du fichier `carte_visite.txt`. Afficher le contenu de `~/carte_visite.txt` et de `~/carte_visite.phy`. Que constate-t-on?



Question 5 : En utilisant la commande `ls -l`, afficher les détails des trois fichiers `carte_visite`. Que constate-t-on?



Question 6 : En utilisant la commande `file`, afficher la nature des trois fichiers `carte_visite`. Commenter.



Question 7 : Effacer le fichier `carte_visite.phy`. Quel est l'effet sur les autres fichiers?



Question 8 : Copier le fichier `carte_visite.txt` dans `carte_visite.copie` puis effacer `carte_visite.txt`. Quel est l'effet sur le fichier `carte_visite.sym`? Afficher la nature de `carte_visite.sym`.



Question 9 : Créer un lien physique `exp` qui pointe vers le fichier `carte_visite.sym`. Commenter.



Question 10 : Copier `carte_visite.copie` en `carte_visite.txt`. Quel est l'effet sur `carte_visite.sym`?

TP 4

Manipulation de fichiers textes

Attention L'usage de `sed` n'est *pas* demandé pour ce TP. (Ce sera l'objet d'un futur TP.)



Exercice 4.1 (Utilisation des filtres). Le fichier `/etc/passwd` contient des informations sur les utilisateurs déclarés dans le système :

```
1 Username:passwd:UID:GID:GECOS:home:shell
```

où `shell` est l'interpréteur de commande de l'utilisateur.



Question 1 : Afficher le contenu du fichier `/etc/passwd`.



Question 2 : Afficher le contenu du fichier `/etc/passwd` en mode page par page.



Question 3 : Donner le nombre de mots contenus dans le fichier `/etc/passwd`



Question 4 : Afficher les trois premières lignes du fichier `/etc/passwd`



Question 5 : Afficher les deux dernières lignes de ce même fichier.



Question 6 : Afficher la liste de tous les interpréteurs de commandes déclarés dans ce fichier (c'est-à-dire la valeur du dernier champ de chaque ligne).



Question 7 : Afficher le contenu du fichier en remplaçant le séparateur « : » par le séparateur « \t »



Exercice 4.2 (Composition de commandes simples).



Question 1 : Afficher l'avant-dernière ligne du fichier `/etc/passwd`




Question 2 : Donner une commande qui renvoie le nombre de fichiers qui ont l'extension `.txt` et qui se trouvent dans l'arborescence de votre répertoire personnel.




Question 3 : Donner une commande qui renvoie le nombre de sous-répertoires présents (immédiatement, sans récursivité) dans le répertoire courant.




Question 4 : Afficher le nombre d'interpréteurs de commandes **différents** déclarés dans le fichier `/etc/passwd`.


 **Question 5 :** Donner une commande qui renvoie le nombre de fichiers accessibles en écriture pour les autres utilisateurs dans l'arborescence du répertoire courant.


 **Exercice 4.3 (Manipulation des groupes).** Le fichier `/etc/group` contient des informations sur les groupes du système :


```
1 ...
2 adm:x:4:syslog,martin
3 tty:x:5:syslog
4 disk:x:6:
5 lp:x:7:
6 wireshark:x:1001:martin
7 ...
```


où le premier champ (séparés par « : ») contient le nom du groupe, et le dernier champ contient la liste des utilisateurs de ce groupe (séparés par « , »).


 **Question 1 :** Donner le nombre de lignes de ce fichier.


 **Question 2 :** Donner le nombre de groupes différents dans ce fichier.

 **Question 3 :** Donner le nombre de groupes auxquels appartient l'utilisateur `votre_login` (c'est-à-dire vous).

 **Question 4 :** Donner le nombre de groupes auxquels appartiennent *plusieurs* utilisateurs (c'est-à-dire > 1).

 **Question 5 :** Lister le nom des groupes (et seulement leurs noms) auxquels appartiennent *plusieurs* utilisateurs (c'est-à-dire > 1).


 **Question 6 :** Donner le nombre lettres différentes utilisées comme deuxième champ de chaque ligne (la réponse sera très probablement 1, car vous n'aurez probablement que des `x` en 2^e champ).


 **Exercice 4.4 (Manipulation d'un annuaire).** En utilisant la commande `wget`, télécharger le fichier `annuaire.txt` dans votre répertoire courant.


```
wget
https://www.loria.science/andre/enseignement/Linux/annuaire.txt
--user-agent="Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:26.0)
Gecko/20100101 Firefox/26.0"
# le --user-agent empêche l'hébergeur de croire que nous sommes
un robot !
```


Chaque entrée dans ce fichier donne le prénom, le nom et l'adresse électronique d'une personne :


```
1 Quentin:Tarantino:Quentin.Tarantino@gmail.com
2 Kar-Wai:Wong:wkw@af.hkbu.edu.hk
3 Julia:Ducournau:Julia.Ducournau@kazak.fr
4 Kiyoshi:Kurosawa:Kiyoshi.Kurosawa@mail.yahoo.co.jp
5 Kar-Wai:Wong:wongkarwai@riseup.net
6 Jean-Luc:Godard:jeanluc@lescahiers.fr
7 Kar-Wai:Wong:wkw@af.hkbu.edu.hk
```

 **Question 1 :** Donner deux commandes différentes pour afficher la première adresse électronique dans le fichier `annuaire.txt`.

 **Question 2 :** Afficher la liste des adresses électroniques (et seulement celles-ci) triée par l'ordre alphabétique *des prénoms*.

 **Question 3 :** Afficher la liste des adresses électroniques (et seulement celles-ci) triée par l'ordre alphabétique inversé des noms.

 **Question 4 :** Donner une commande qui permet d'afficher les adresses électroniques (et seulement celles-ci) de l'utilisateur Wong (ce qui est à comprendre simplement comme une ligne qui contient « Wong »).

 **Question 5 :** Donner une commande qui renvoie le nombre d'adresses électroniques différentes de l'utilisateur qui a comme nom Wong (là encore, c'est à comprendre simplement comme une ligne qui contient « Wong »).

TP 5

Commandes réseaux



Exercice 5.1 (Commandes réseaux).



Question 1 : À l'aide de la commande `hostname` afficher le nom symbolique de votre machine.



Question 2 : Essayer d'attribuer `MyPC` comme nom symbolique de votre machine. Justifier la réponse obtenue.



Question 3 : À l'aide de la commande `ifconfig`, relever l'adresse IP de l'interface réseau `eno1` de votre machine et son adresse MAC. Communiquer votre adresse IP à votre voisin.



Question 4 : Utiliser la commande `ping` pour envoyer un (et seulement un) paquet à la machine de votre voisin. Quelles sont les informations retournées par cette commande?



Question 5 : Utiliser `ping` pour tester l'accès à la machine `www.linuxmint.com`. Comparer les résultats avec ceux obtenus dans la question précédente.



Question 6 : Mêmes questions pour `dbs.com.sg`. Comparer.



Question 7 : Utiliser la commande `traceroute` pour donner le chemin entre votre machine et la machine de votre voisin. Puis entre votre machine et la machine `iut.charlemagne.nancy`



Question 8 : À l'aide de la commande `nslookup` retrouver si possible les adresses IP des trois machines : `www.univ-lorraine.fr`, `www.nancy.fr` et `iut.charlemagne.nancy`

TP 6

Exercices de révisions



Exercice 6.1 (révision).



Question 1 : Se placer à la racine de l'arborescence du système de fichiers.



Question 2 : Créer dans votre répertoire personnel un fichier `CommandesLinux.md`



Question 3 : Créer dans votre répertoire personnel un sous-répertoire `scripts`



Question 4 : Comparer les droits d'accès de `CommandesLinux.md` et `scripts`. Pourquoi sont ils différents? En déduire la valeur de `umask`



Question 5 : Déplacer le fichier `CommandesLinux.md` dans le répertoire `scripts`



Question 6 : Écrire dans le fichier `CommandesLinux.md` la phrase « `echo Linux rocks!` »



Question 7 : Donner au fichier `CommandesLinux.md` le droit d'être exécuté par l'utilisateur, les membres du groupe et les autres. Ajouter ce droit sans modifier les autres. Ensuite, exécuter le fichier. Que se passe-t-il?



Question 8 : Donner une commande qui affiche le nombre de fichiers dans l'arborescence enracinée dans votre répertoire de travail qui sont exécutables pour l'utilisateur et qui ont une taille inférieure à 10 kibi-octets.



Question 9 : Donner une commande qui affiche tous les fichiers dans l'arborescence de votre répertoire personnel qui ont été créés il y a moins de 24 heures ou qui ont une taille supérieure à 10 kibi-octets, triés alphabétiquement et en remplaçant le séparateur « `/` » par « `\` »




Question 10 : Extraire du fichier `annuaire.txt` l'adresse électronique du réalisateur qui précède immédiatement la ligne contenant `Julia`.



Question 11 : Dans une adresse électronique, la sous-chaîne qui suit le dernier point donne le nom du domaine de premier niveau de l'adresse (par exemple `jp` est le domaine de premier niveau de l'adresse `Kiyoshi.Kurosawa@mail.yahoo.co.jp`). Donner une commande qui renvoie le nombre de domaines de premier niveau différents mentionnés dans le fichier `annuaire.txt` (qui a été introduit dans le TP 4.4).

Note : la commande `rev` permet d'inverser une chaîne de caractères (caractère par caractère).

 **Question 12 :** Entrer dans le répertoire `scripts` et effacer le fichier `annuaire.txt` en utilisant son parcours relatif. (Utiliser une commande d'une seule ligne.)

TP 7

Gestion de processus & variables d'environnement



Exercice 7.1 (Gestion de processus).



Question 1 : Exécuter l'application `gedit`. Pouvez-vous exécuter une autre commande dans le même shell? Justifier votre réponse.



Question 2 : Suspendre l'application `gedit` en appuyant sur les touches `<ctrl>+Z`. Peut-on exécuter une autre commande dans le même shell? Justifier.



Question 3 : Reprendre l'exécution de l'application suspendue en premier plan.



Question 4 : Suspendre à nouveau l'application puis reprendre son exécution en tâche de fond. Peut-on exécuter une autre commande dans le même shell? Justifier.



Question 5 : Donner une commande qui permet d'arrêter l'application graphique exécutée.



Question 6 : Exécuter l'application `emacs` comme tâche de fond. Fermer la fenêtre de l'interpréteur de commandes à partir de laquelle l'application est lancée. Quel est l'effet de cette action sur l'application `emacs`? Justifier votre réponse.



Question 7 : Dans un nouveau terminal, exécuter la commande suivante :

```
nohup emacs
```

Quel est le processus père de `emacs`? Ensuite, arrêter le processus du shell. Quel est l'effet sur l'application `emacs`? Quel est le (nouveau) processus père de l'application `emacs`?



Question 8 : Afficher en temps réel les informations sur l'ensemble de processus exécutés sur votre machine.



Question 9 : En utilisant la commande `ps`, afficher les informations détaillées sur tous vos processus.



Question 10 : Lancer à nouveau l'application `gedit` en tâche de fond. Donner une commande qui renvoie uniquement le PPID du processus qui exécute cette application.



Exercice 7.2 (Les signaux).



Question 1 : Donner une commande pour afficher les signaux définis dans le système.



Question 2 : Lancer à nouveau l'application `gedit` en tâche de fond. À l'aide de la commande `kill` envoyer les signaux adéquats pour suspendre, reprendre puis arrêter le processus `gedit`.



Question 3 : Dans le shell courant et à l'aide de la commande `trap` remplacer le traitement par défaut du signal d'interruption `SIGINT` par une commande qui affiche le message « Signal `SIGINT` désactivé ». Tester.



Question 4 : Rétablir le traitement par défaut du signal `SIGINT`.



Exercice 7.3 (Variables d'environnement).



Question 1 : Afficher l'ensemble des variables d'environnements définies dans votre shell avec leurs valeurs associées.



Question 2 : Donner une commande qui renvoie la liste des noms des variables d'environnement *sans les valeurs associées*. La liste doit être triée par ordre alphabétique.



Question 3 : Afficher la valeur associée à la variable `PATH`. Quel est le rôle de cette variable ?



Question 4 : Donner une commande qui renvoie le nombre de répertoires déclarés dans la variable `PATH`.



Question 5 : À l'aide de la commande `which`, localiser le compilateur `gcc`.



Question 6 : À l'aide de la commande `which`, essayer de localiser la commande `ifconfig`. Justifier le résultat obtenu.



Question 7 : À l'aide de la commande `alias`, renommer la commande `ls` pour que le résultat de son appel soit équivalent à la commande `ls -l`.



Exercice 7.4 (Affichage et interprétation de variables). Dans un shell, exécuter la commande suivante :

```
X="ls -l"
```

Exécuter ensuite les commandes suivantes et, pour chacune, justifier le résultat obtenu :



Question 1 :

```
echo X
```



Question 2 :

```
echo $X
```

 **Question 3 :**

```
echo '$X' # ici, ' indique le guillemet droit
```

 **Question 4 :**

```
echo "$X"
```

 **Question 5 :**

```
echo '$X' # ici, ' indique le guillemet oblique
```



Exercice 7.5 (Portée des variables). Dans un shell bash, taper les commandes suivantes et justifier les affichages obtenus :

```
X1=3
Y1=10
Z1=4
export Y1
env | grep X1=
echo $X1
echo $x1
env | grep Y1=
unset Y1
export X1
bash
env | grep X1=
echo $Z1
exit
echo $Z1
```

TP 8

Introduction à la programmation de scripts shell



Exercice 8.1 (Premier script shell).



Question 1 : Écrire un script `bonjour.bash` qui :

1. affiche le message « Bonjour » suivi de la liste des noms passés comme paramètres et un point d'exclamation,
2. affiche « La date d'aujourd'hui est » suivi de la date (commande : `date`).

Exemple : l'appel `bonjour.bash Yann Lori` affiche le message suivant (la date peut avoir un format légèrement différent)

```
> bonjour.bash Yann Lori
Bonjour Yann Lori !
Aujourd'hui, nous sommes le mercredi 8 septembre 2021, 17:09:23
(UTC+0200)
```



Question 2 : Écrire un script `salut.bash` qui affiche les messages suivants :

1. « Le script `<nom_du_script>` » est exécuté par le processus `PID`
2. « Salut à `x` personnes », où `x` est le nombre de noms passés comme paramètres.
 - S'il y a au moins un paramètre, rajoute « dont la première est `<premier nom>` ».
 - S'il y en a d'autres, rajoute aussi « et les autres sont `<noms des autres>` ».

Exemple : l'appel `salut.bash Yann Lori Anna` affiche le message

```
> salut.bash Yann Lori Anna
Le script salut.bash est exécuté par le processus 6536
Salut à 3 personnes, dont la première est Yann et les autres sont
Lori Anna
```



Exercice 8.2 (Gestion d'un drapeau). Un *drapeau* est un simple fichier texte qui peut contenir soit la valeur 1, soit la valeur 0. On dit que le drapeau est *positionné* si le fichier contient la valeur 1. On voudrait développer une commande nommée `flag` qui a le fonctionnement suivant :

- `flag <nom_du_drapeau>` : retourne l'état actuel du drapeau. Si le fichier correspondant n'existe pas, il affiche un message d'erreur.
- `flag <nom_du_drapeau> on` : positionne le drapeau (c.-à-d. met dans le fichier correspondant la valeur 1). Si le fichier n'existe pas, la commande doit le créer.

- `flag <nom_du_drapeau> off` : enlève le drapeau (c.-à-d. met dans le fichier correspondant la valeur 0). Si le fichier n'existe pas, la commande doit le créer.
- `flag <nom_du_drapeau> flop` : inverse l'état du drapeau. Si le fichier correspondant n'existe pas, la commande affiche un message d'erreur.



Exercice 8.3 (Poubelle). Développer une commande nommée `poubelle` qui permet de transférer les fichiers à effacer dans un répertoire nommé `trash`. La syntaxe de cette commande est la suivante :

1. `poubelle f1 f2 f3 ... fn` a pour effet de transférer les fichiers de `f1` à `fn` dans le répertoire `trash`.
2. `poubelle -f` a pour effet d'effacer le contenu du répertoire `.trash`.
3. `poubelle` a pour effet d'afficher un message d'aide décrivant la syntaxe correcte de la commande.



Exercice 8.4 (Gestion de sauvegarde).



Question 1 : Écrire un programme shell nommé `saveTxt` qui permet de copier dans un répertoire nommé `~/ .backup` tous les fichiers qui se trouvent dans l'arborescence du répertoire personnel et qui se terminent par le suffixe `.txt`. Si le répertoire `.backup` n'existe pas, alors la commande doit le créer.



Question 2 : Modifier le programme précédent afin de ne pas écraser les fichiers existant dans le répertoire `.backup`.

Suggestion : utiliser la commande `basename` pour récupérer le nom d'un fichier à partir de son parcours absolu/relatif.

TP 9

Utilisation de sed

Tous les exercices de ce TP doivent être traités exclusivement avec la commande `sed`, sauf mention contraire.



Exercice 9.1 (exemples simples).



Question 1 : Proposer une commande qui va remplacer toutes les occurrences de `iOS` par `Android` dans un fichier donné (on peut prendre `iOS.txt` comme exemple).

Comparer avec et sans l'option `g`.



Question 2 : Même question en ne remplaçant que la 2^e occurrence de `iOS` sur chaque ligne.



Exercice 9.2 (liste de films). On considère ici le fichier `films.txt`, reproduit ici :

```
1 In the Mood for Love ; Wong Kar-wai ; 2000
2 Dunkerque ; Christopher Nolan ; 2017
3 Drive My Car ; Ryūsuke Hamaguchi ; 2021
4 Casablanca ; Michael Curtiz ; 1942
5 Les Garçons de Fengkuei ; Hou Hsiao-hsien ; 1983
6
7 Taxi Driver ; Martin Scorsese ; 1976
8
9 Tropical Malady ; Apichatpong Weerasethakul ; 2004
10 Kill Bill : Volume 2 ; Quentin Tarantino ; 2004
11 300 ; Zack Snyder ; 2006
12 Grave ; Julia Ducournau ; 2016
13 Interstellar ; Christopher Nolan ; 2014
```



Question 1 : Remplacer toutes les occurrences de `« ; »` par `« , »`.



Question 2 : Supprimer tous les `« ; »` du fichier.



Question 3 : Supprimer les années, en utilisant une solution supprimant les séquences de 4 chiffres.



Question 4 : Remplacer la 2^e occurrence de `« ; »` de chaque ligne par `« en »`.





Question 5 : Supprimer les lignes 2 à 4 du fichier.




Question 6 : Supprimer toutes les lignes, sauf les lignes 2 à 4 du fichier.

 **Question 7 :** Supprimer les lignes mentionnant un film de Christopher Nolan.


 **Question 8 :** Supprimer les lignes possédant un « W ».


 **Question 9 :** Supprimer les lignes possédant un « H », un « K », un « W », ou un « Z ».


 **Question 10 :** Supprimer les lignes commençant par une lettre majuscule.

 **Question 11 :** Supprimer les lignes terminant par « 2004 ».

 **Question 12 :** Supprimer la première ligne contenant « Martin » et toutes les lignes qui suivent jusqu'à la fin du fichier.


 **Question 13 :** Supprimer les lignes blanches du fichier.

 **Question 14 :** Ajouter « Film : » devant chaque ligne du fichier.

 **Question 15 :** Ajouter « Film : » devant chaque ligne non-vide du fichier.




Exercice 9.3.

 **Question 1 :** À l'aide de `sed` et exceptionnellement `find`, trouver l'ensemble des fichiers d'extension `.txt` présents sur l'ordinateur, et les afficher *sans* leur extension `.txt`.




Exercice 9.4 (Une histoire de correspondance).

 **Question 1 :** Télécharger le fichier `lettre-alfred-de-musset-canular.txt`

Il s'agit d'une lettre faussement attribuée au romancier Alfred de Musset à destination de l'écrivaine George Sand dans la seconde moitié du XIX^e siècle.

Proposer avec `sed` une commande qui supprime une ligne sur deux à partir de la 6^e ligne. C'est-à-dire que les lignes conservées doivent être les lignes 1, 2, 3, 4, 5, 7, 9...

 **Question 2 :** Télécharger le fichier `correspondance-alfred-de-musset-george-sand.txt`

Il s'agit d'une correspondance cette fois probablement authentique entre Alfred de Musset et George Sand (seconde moitié du XIX^e siècle).

Proposer avec `sed` une commande qui ne garde que le premier mot de chaque ligne.

Attention : un mot est séparé d'un autre par une espace, ou un autre symbole typographique (penser au tiret -).