

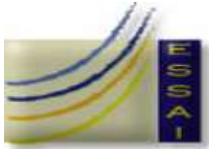


TP 5 Linux

Les commandes Shell (Bash Shell)
Les liens, la redirection et les pipes

Enseignante: MANSOUR Sihem

Contact : sihemmansour@yahoo.fr



Plan du cours



1. Les liens
2. La recherche des fichiers sous Linux
3. Recherche d'un texte dans un fichier: grep
4. Trier des lignes: sort
5. Compter les lignes: wc
6. Supprimer les doublons: uniq
7. Archiver, Compresser et décompresser des fichiers
8. Les flux de redirections
9. Les pipes



Les liens

1. Définition

- Un lien permet de donner plusieurs noms à un même fichier, ou de faire pointer un fichier sur un autre.
- Plutôt que de faire plusieurs copies d'un même fichier pour plusieurs utilisateurs, on peut par exemple permettre à ceux-ci d'accéder à une copie unique, mais depuis des endroits et des noms différents. Pour créer un lien, on utilise la commande **ln**
- Il existe deux types de liens : les liens en dur « lien physique » et les liens symboliques « symbolic links ».



Les liens

2. Liens physiques:

- Un lien physique permet d'ajouter une référence sur un inode.

`$Ln source lien`

- Sous Unix/Linux chaque fichier est en fait référencé dans deux tables :
 - une table d'inode qui contient un numéro, la taille du fichier référencé par cet inode, ses droits d'accès, son propriétaire...etc.
 - et une table catalogue qui est une table de correspondance entre les noms de fichiers et les numéros d'inodes.

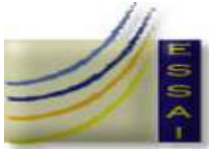


Les liens

- Le lien physique rajoute donc une association dans cette seconde table entre un nom du fichier et un inode. Les droits du fichier ne sont pas modifiés. cad ce type du lien permet d'avoir deux noms de fichiers qui partagent exactement le même contenu, c'est-à-dire le même inode



- **On ne peut pas créer de liens physiques sur des répertoires. Cela ne fonctionne qu'avec les fichiers.**



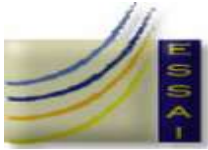
Les liens

- Exemple :
 - On va créer un fichier vide **f1**. pour se faire, taper la commande:

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ touch f1
```

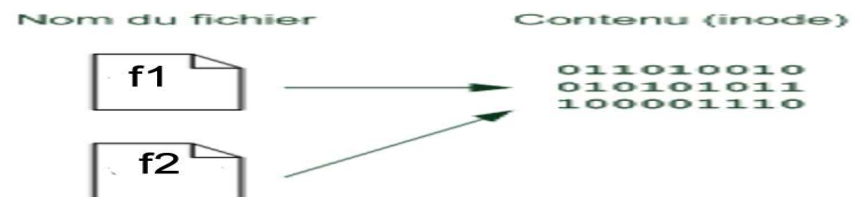
- Maintenant, on va créer un lien physique sur **f1**, qu'on va le nommer **f2**

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ ln f1 f2
```



Les liens

- On a créé deux fichiers qui référencent le même inode (même contenu).



Si on liste le contenu du répertoire où ils existent , on va voir ces informations:

```
administrateur@administrateur-VirtualBox: ~
administrateur@administrateur-VirtualBox:~$ ls -l
total 44
drwxr-xr-x 2 administrateur 4096 sept.  1 20:03 Bureau
drwxr-xr-x 2 administrateur 4096 sept.  1 20:03 Documents
-rw-r--r-- 1 administrateur 8445 sept.  1 19:29 exemples.desktop
-rw-rw-r-- 2 administrateur administrateur  0 sept. 30 09:32 f1
-rw-rw-r-- 2 administrateur administrateur  0 sept. 30 09:32 f2
drwxr-xr-x 2 administrateur administrateur 4096 sept.  1 20:03 Images
```

cette valeur représente le nbre de liens physiques sur l'inode



Les liens



- La seconde colonne de la liste (qui indique « 2 » pour chacun des fichiers) correspond au nombre de fichiers qui partagent le même inode. C'est le seul indice qui vous permet de savoir que quelqu'un a fait un lien physique, mais vous ne pouvez pas savoir lequel. Le seul moyen de vérifier que ces fichiers partagent le même contenu, c'est de faire `ls -li` pour afficher les numéros d'inode correspondants et de vérifier que ces deux fichiers sont associés au même inode.
- Si c'est un dossier, ce nombre indique en revanche le nombre de fichiers à l'intérieur.



Les liens

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ ls -i f1  
37181 f1  
administrateur@administrateur-VirtualBox:~$ ls -i f2  
37181 f2
```



- Si vous supprimez un des deux fichiers, l'autre fichier reste en place et le contenu sera toujours présent sur le disque. L'inode est supprimé uniquement quand plus aucun nom de fichier ne pointe dessus. Cad Il faut supprimer f1 ET f2 pour supprimer le contenu.



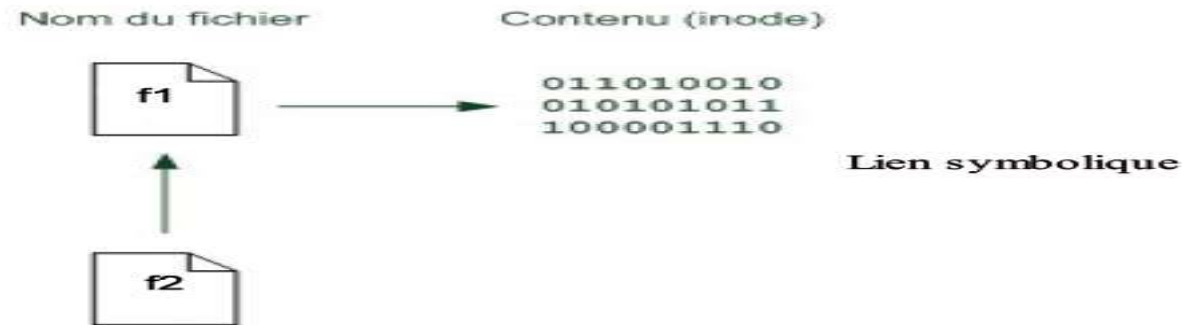
Les liens

3. Liens symboliques

- Les liens symboliques ressemblent plus aux « raccourcis » dont vous avez peut-être l'habitude sous Windows. La plupart du temps, on crée des liens symboliques sous Linux pour faire un raccourci.
- Le principe du lien symbolique est que l'on crée un lien vers un autre nom de fichier. Cette fois, on pointe vers le nom de fichier et non vers l'inode directement .
- Voir le graphique suivant:



Les liens



Exemple:

- Commencer par supprimer le fichier f2 qu' on a créé dans l'exemple précédent.

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ rm f2
```



Les liens

- Créons maintenant un nouveau f2, cette fois sous forme de lien symbolique sur f1. On utilise là encore la commande ln, mais avec le paramètre -s

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ ln -s f1 f2
```

Cette fois, la commande détaillée ls -l sera beaucoup plus précise :

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ ls -l  
total 44  
drwxr-xr-x 2 administrateur administrateur 4096 sept.  1 20:03 Bureau  
drwxr-xr-x 2 administrateur administrateur 4096 sept.  1 20:03 Documents  
-rw-r--r-- 1 administrateur administrateur 8445 sept.  1 19:29 exemples.desktop  
-rw-rw-r-- 1 administrateur administrateur    0 sept. 30 09:32 f1  
lrwxrwxrwx 1 administrateur administrateur    2 sept. 30 11:38 f2 -> f1  
drwxr-xr-x 2 administrateur administrateur 4096 sept.  1 20:03 Images  
drwxr-xr-x 2 administrateur administrateur 4096 sept.  1 20:03 Modèles
```

f2 est un raccourcis sur f1

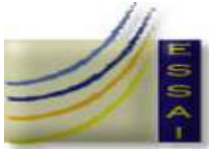


La recherche des fichiers sous Linux



find: une recherche approfondie

- find est la commande de recherche par excellence pour retrouver des fichiers, mais aussi pour effectuer des opérations sur chacun des fichiers trouvés. Elle est très puissante, permet donc de faire beaucoup de choses, et par conséquent... elle est un peu complexe.
- Contrairement à locate, find ne va pas lire dans une base de données mais au contraire parcourir tout votre disque dur.



La recherche des fichiers sous Linux



- **Utilisation basique de la commande find:**

- Recherche à partir du nom

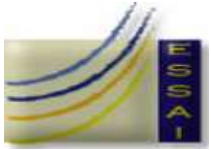
- Exemple:

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find -name "f1"  
./f1
```

Le -name " f1" est un paramètre qui demande de retrouver un fichier qui s'appelle très exactement **f1**



- Comme nous n'avons pas précisé où rechercher, **find** a cherché dans le répertoire dans lequel nous nous trouvions et dans ses sous-répertoires. Le ~



La recherche des fichiers sous Linux



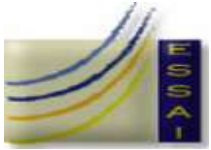
- Maintenant, nous allons préciser le chemin d'accès où on va faire la recherche:

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find /var/log/ -name "sys.log"
```

- Notez que, contrairement à locate, find récupère uniquement la liste des fichiers qui s'appellent exactement comme demandé. Ainsi, s'il existe un fichier nommé syslog2, il ne sera pas listé dans les résultats. Pour qu'il le soit, il faut utiliser le joker :

l'étoile « * »

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find /var/log/ -name "sys.log*"
```



La recherche des fichiers sous Linux



- Si vous voulez chercher un fichier sur tout le disque dur, utilisez le chemin d'accès `/`

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find / -name "sys.log"
```

- **Recherche à partir de la taille:**
- On peut faire des recherche sur les fichiers à partir de leur taille:

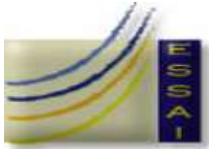
```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find -size +10M
```




La recherche des fichiers sous Linux



- Au lieu de se baser sur le nom, on se base ici sur la taille (-size).
Le +10M indique que l'on recherche des fichiers de plus de 10 Mo. On peut aussi utiliser k pour les Ko, G pour les Go, etc.
- Vous pouvez aussi utiliser un moins « - » à la place du « + » pour obtenir par exemple les fichiers de moins de 10 Mo. Et si vous enlevez le « + », la commande cherchera des fichiers de 10 Mo exactement (ni plus, ni moins).



La recherche des fichiers sous Linux



- **Recherche à partir de la date**
 - Vous êtes sûrs d'avoir accédé à votre rapport au format .odt il y a moins de 7 jours, mais vous n'arrivez pas à le retrouver ?
 - Avec -atime, vous pouvez indiquer le nombre de jours qui vous séparent du dernier accès à un fichier.
 - Exemple:

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find -atime -7
```



La recherche des fichiers sous Linux



- **Recherche uniquement des répertoires ou fichiers:**
 - On peut aussi rechercher uniquement des répertoires ou des fichiers. Utilisez :
 - -type d : pour rechercher uniquement des répertoires (directories) ;
 - -type f : pour rechercher uniquement des fichiers (files).



La recherche des fichiers sous Linux



- Par défaut, find cherche des répertoires ET des fichiers. Ainsi, si vous avez un fichier appelé syslog et un répertoire appelé syslog, les deux résultats seront affichés. Pour obtenir uniquement les répertoires qui s'appellent syslog (et non pas les fichiers), tapez donc :

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find -name "syslog" -type d
```



La recherche des fichiers sous Linux



- **Supprimer les fichiers trouvés**
 - Un des usages les plus courants de find, à part retrouver des fichiers, consiste à les supprimer. Si je veux faire le ménage dans mon home et par exemple supprimer tous mes fichiers « jpg », je vais écrire ceci :

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ find -name "*.jpeg" -delete
```



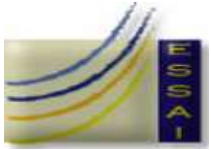
Recherche d'un texte dans un fichier



- grep
 - La recherche d'une chaîne de caractères dans un ou plusieurs fichiers peut se faire à l'aide de la commande **grep**.
 - Cette commande prend en premier paramètre le texte à rechercher, puis la liste des fichiers dans lequel ce texte doit être trouvé :

grep texte fichiers

- Le texte peut être placé entre guillemets si nécessaire (en particulier, s'il contient des espaces ou des caractères interprétés par le shell, comme * et ?)



Recherche d'un texte dans un fichier

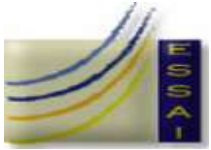


```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ grep fouille cours_linux.html
```

Diagram annotations:

- Callout for `fouille`: texte cherché
- Callout for `cours_linux.html`: le fichier ou on va faire la recherche

- La commande grep possède plusieurs options, exemple:
 - -n: affiche le numéro de la ligne qui contient le texte
 - -l: affiche tous les fichiers qui contiennent le texte
 - -i: pour ignorer la casse
 - -v : inverser la recherche, c'est-à-dire on cherche toutes les données (lignes ou fichiers) qui ne contiennent pas le critère voulu
 - -r : rechercher dans tous les fichiers et sous-dossiers



Trier des lignes

- sort
 - La commande **sort** permet de trier des lignes du fichier.
Remarque: La commande sort ne respecte pas la casse
 - On va créer le fichier text sous notre répertoire personnel qui contient les informations suivantes:
 - Ariana
 - Kelibia
 - Bizerte
 - Sfax
 - Gabes



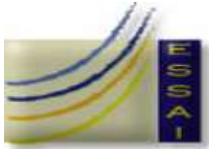
Trier des lignes

- Pour se faire utiliser l'éditeur nano.

```
administrateur@administrateur-VirtualBox: ~  
GNU nano 2.2.6 Fichier : text  
ariana  
kelibia  
Bizerte  
sfax  
gabes
```

- Ensuite, on va exécuter la commande sort sur ce fichier :

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ sort text  
ariana  
Bizerte  
gabes  
kelibia  
sfax  
administrateur@administrateur-VirtualBox:~$
```



Trier des lignes

- La commande sort supporte plusieurs options:
- -o: écrire le résultat de tri dans un autre fichier. En faite cette commande ne modifie pas le fichier, juste elle affiche son contenu trié sur le console. Si on veut mettre le résultat dans un fichier au lieu de l'afficher utiliser l'option -o

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ sort -o text trié text
```

- -r: inverser le tri

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ sort -r text  
sfax  
kelibia  
gabes  
Bizerte  
ariana  
administrateur@administrateur-VirtualBox:~$ █
```



Compter les lignes

- Wc:
 - La commande **wc** signifie word count. C'est donc a priori un compteur de mots mais en fait, on lui trouve plusieurs autres utilités : compter le nombre de lignes (très fréquent) et compter le nombre de caractères. la commande **wc** travaille sur un fichier.
 - Exemple d'utilisation:

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ wc text  
5  5 34 text  
administrateur@administrateur-VirtualBox:~$
```



Compter les lignes

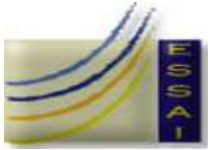
- Ces trois nombres signifient, dans l'ordre :
 - le nombre de lignes.
 - le nombre de mots.
 - le nombre d'octets.
- Cette commande comporte plusieurs options:
 - -l : compter le nombre de lignes:
 - -w : compter le nombre de mots
 - -c : compter le nombre d'octets
 - -m : compter le nombre de caractères

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ wc -l text  
5 text  
administrateur@administrateur-VirtualBox:~$
```



Supprimer les doublons

- uniq
- Parfois, certains fichiers contiennent des lignes en double et on aimerait pouvoir les détecter ou les supprimer. Pour se faire on peut utiliser La commande **uniq**
- Nous devons travailler sur un fichier trié. En effet, la commande **uniq** ne repère que les lignes successives qui sont identiques.
- Je vous propose de créer un fichier doublons contenant les noms suivants :

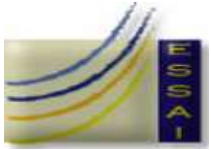


Supprimer les doublons

```
administrateur@administrateur-VirtualBox: ~  
GNU nano 2.2.6      Fichier : doublons  
ali  
ali  
mohamed  
iheb  
iheb  
najia  
samira  
samira
```

- on va utiliser La commande **uniq** , pour afficher le contenu de ce fichier sans doublons

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ uniq doublons  
ali  
mohamed  
iheb  
najia  
samira
```

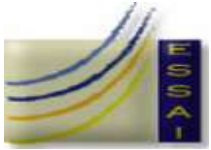


Supprimer les doublons

- Si vous voulez transmettre le résultat de la commande **uniq** à **un fichier** au lieu de l'afficher, vous pouvez procéder de la façon suivante

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ uniq doublons doublonsres
```

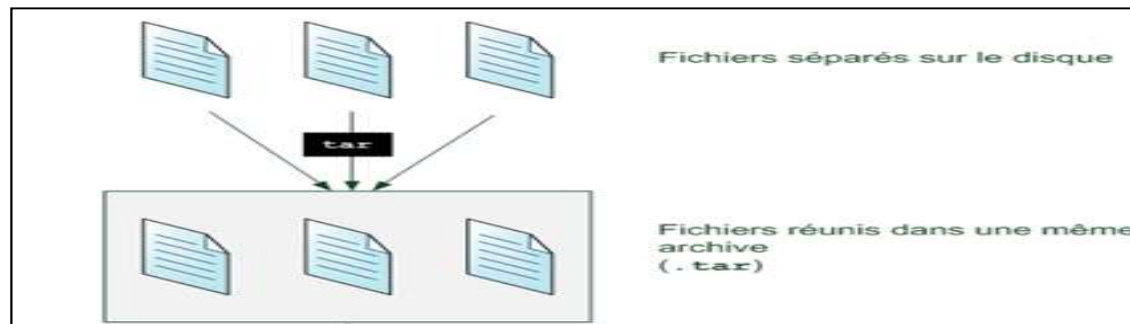
- La commande `uniq` comporte plusieurs options:
 - `-c` : compter le nombre d'occurrences
 - `-d` : afficher uniquement les lignes présentes en double



Archiver et compresser des fichiers sous ubuntu

1. **Assembler des fichiers dans un archive:**

Généralement, on assemble des fichiers dans un archive (un fichier .tar) pour les compresser par la suite. Pour faire l'archivage des fichiers, utiliser la commande tar



- **Consigne:** vaut-mieux mettre les fichiers à archiver dans un dossier avant l'archivage



Archiver et compresser des fichiers sous ubuntu

2. Comment faire l'archivage

- commencer par créer un dossier qui contient tous les fichiers à archiver

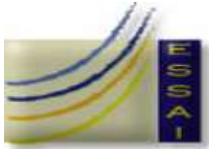
```
essai@essai-VirtualBox:~$ mkdir essai
```

- Accéder à ce dossier et mettre vos fichiers

```
essai@essai-VirtualBox: ~  
essai@essai-VirtualBox:~$ mv f? essai
```

- Créer votre archive

```
essai@essai-VirtualBox:~$ tar -cvf essai.tar essai  
essai/  
essai/f3  
essai/f2  
essai/f1
```



Archiver et compresser des fichiers sous ubuntu



2. Comment faire l'archivage

- Les options utilisées avec la commande tar, sont:
 - -c: signifie créer une archive tar
 - -v: afficher les détails des opérations
 - -f : rassembler l'archivage dans un fichier
- Pour afficher le contenu d'un archivage:

```
essai@essai-VirtualBox:~$ tar -tfessai.tar
essai/
essai/f3
essai/f2
essai/f1
```



Archiver et compresser des fichiers sous ubuntu

2. Comment faire l'archivage

- Extraire des fichier de l'archive
 - On va commencer par supprimer le dossier essai :

```
essai@essai-VirtualBox:~$ ls
Bureau      essai      exemples.desktop  Images  Musique  Téléchargements
Documents   essai.tar  f1~               Modèles Public   Vidéos
```

```
essai@essai-VirtualBox:~$ rm -r essai
```

```
essai@essai-VirtualBox:~$ ls
Bureau      essai.tar      f1~      Modèles  Public  Vidéos
Documents   exemples.desktop  Images  Musique  Téléchargements
```



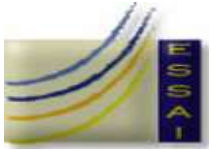
Archiver et compresser des fichiers sous ubuntu



2. Comment faire l'archivage

- Extraire des fichier de l'archive
 - Puis extraire nos fichiers de l'archive

```
essai@essai-VirtualBox:~$ tar -xvf essai.tar
essai/
essai/f3
essai/f2
essai/f1
essai@essai-VirtualBox:~$ ls
Bureau      essai      exemples.desktop  Images  Musique  Téléchargements
Documents  essai.tar  f1~               Modèles Public   Vidéos
```



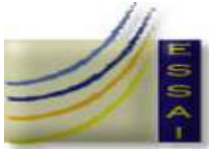
Archiver et compresser des fichiers sous ubuntu

3. Comment compresser l'archivage ou les fichiers

- Pour la compression on a deux programmes qui existent par défaut gzip et bzip2 (plus lent que gzip mais plus efficace), dans notre cours on va voir gzip
- Pour compresser un archive ou un fichier avec gzip, utiliser cette commande

```
essai@essai-VirtualBox:~$ gzip essai.tar
essai@essai-VirtualBox:~$ ls
Bureau  essai  exemples.desktop  Images  Musique  Téléchargements
Documents  essai.tar.gz  f1~  Modèles  Public  Vidéos
essai@essai-VirtualBox:~$
```

- Le fichier résultat est .gz



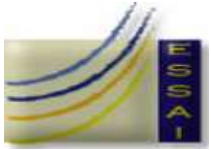
Archiver et compresser des fichiers sous ubuntu



4. Comment décompresser l'archivage

utiliser la commande gunzip

```
essai@essai-VirtualBox:~$ gunzip essai.tar.gz
essai@essai-VirtualBox:~$ ls
Bureau      essai      exemples.desktop  Images  Musique  Téléchargements
Documents  essai.tar  f1~               Modèles  Public   Vidéos
essai@essai-VirtualBox:~$
```



Archiver et compresser des fichiers sous ubuntu

5. Utiliser unrar pour compresser/décompresser des fichier:

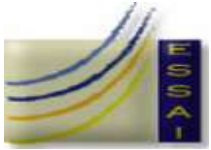
- Commencer par installer unrar

```
essai@essai-VirtualBox:~$ sudo apt-get install unrar
```

- Pour décompresser des fichier rar, utiliser la commande unrar avec l'option e

```
essai@essai-VirtualBox:~$ unrar e essai.rar
```

- Pour compresser, il faut acheter le logiciel

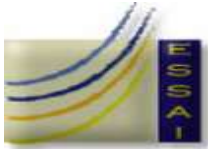


Lancer plusieurs commandes sur la même ligne de commande

- Pour lancer plusieurs commandes sur le même invite, procéder de la façon suivante:

```
Machine  Écran  Périphériques  Aide
Termin  Fichier  Édition  Affichage  Rechercher  Terminal  Aide
essai@essai-VirtualBox: ~
essai@essai-VirtualBox:~$ ls; pwd
```

la commande **ls** est lancée
la première puis **pwd**

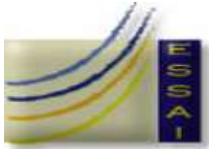


Les flux de redirection

- rediriger le résultat :
- **Rediriger le résultat d'une commande dans un fichier**
 - lorsqu'on tape une commande Shell, le résultat s'affiche par défaut sur la console.
 - Avec Shell, on peut orienter la sortie d'une commande et au lieu de l'afficher sur la console on peut l'enregistrer dans un fichier.

```
essai2012@essai2012-VirtualBox: ~  
essai2012@essai2012-VirtualBox:~$ ls  
Bureau      exemples.desktop  Modèles  Public  Vidéos  
Documents  Images            Musique  Téléchargements  
essai2012@essai2012-VirtualBox:~$
```

Lorsque on tape une commande le résultat par défaut s'affiche sur la console



Les flux de redirection

```
essai2012@essai2012-VirtualBox: ~  
essai2012@essai2012-VirtualBox:~$ ls > t1  
essai2012@essai2012-VirtualBox:~$
```

Avec ce symbole, le resultat de cette commande ne sera plus affiché sur la console mais il va être enregistré dans le fichier t1



L'opérateur **>** diriger le résultat de **ls** vers le fichier **t1**, si ce fichier n'existe pas il va être créé automatiquement et si il existe son contenu va être supprimé et remplacé par le résultat de cette commande



Les flux de redirection

- **rediriger le résultat à la fin du fichier: >>**
 - Le double chevron >> sert lui aussi à rediriger le résultat dans un fichier, mais cette fois à la fin de ce fichier.
 - Avantage : vous ne risquez pas d'écraser le fichier s'il existe déjà. Si le fichier n'existe pas, il sera créé automatiquement.

```
essai2012@essai2012-VirtualBox: ~  
essai2012@essai2012-VirtualBox:~$ ls >> t1
```

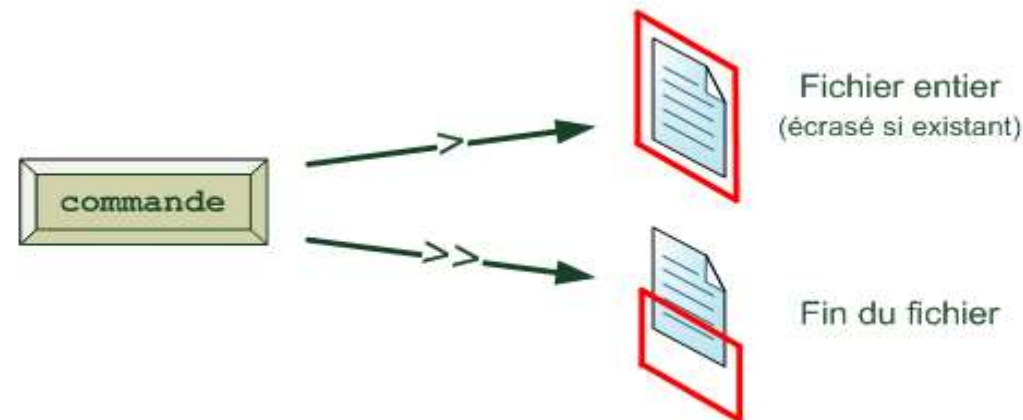
On peut utiliser aussi ce symbole pour dériver le résultat de la commande ls vers le fichier t1

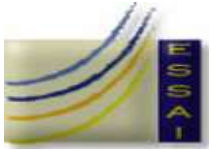


Les flux de redirection

- **Résumé**

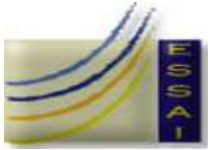
- Nous venons de découvrir deux flux de redirection dans des fichiers :
- `>` : redirige dans un fichier et l'écrase s'il existe déjà ;
- `>>` : redirige à la fin d'un fichier et le crée s'il n'existe pas.





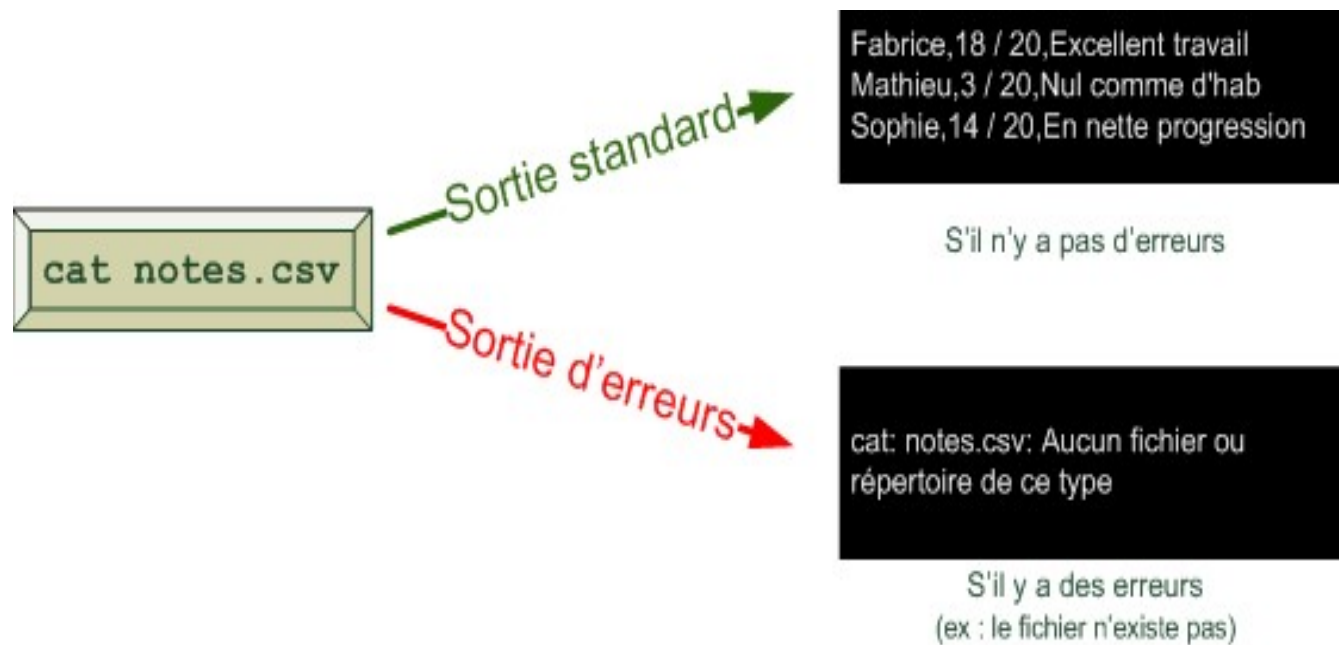
Les flux de redirection

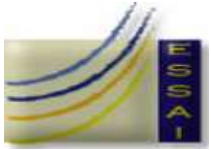
- rediriger les erreurs :
- **introduction**
 - Il faut savoir que toutes les commandes produisent deux flux de données différents.
 - la sortie standard : pour tous les messages (sauf les erreurs) ;
 - la sortie d'erreurs : pour toutes les erreurs.
 - Prenons un exemple: Supposons que vous fassiez un cat du fichier notes.csv pour afficher son contenu. Il y a deux possibilités :
 - si tout va bien, le résultat (le contenu du fichier) s'affiche sur la sortie standard ;
 - s'il y a une erreur, celle-ci s'affiche dans la sortie d'erreurs.



Les flux de redirection

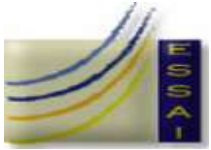
- Voir le graphique suivant:





Les flux de redirection

- **Rediriger l'erreur dans un fichier à part**
 - Si vous voulez enregistrer les erreurs dans un fichier à part pour ne pas les oublier et pour pouvoir les analyser ensuite.
 - Pour se faire, utiliser l'opérateur `2>`
 - Exemple:



Les flux de redirection

```
essai2012@essai2012-VirtualBox: ~  
essai2012@essai2012-VirtualBox:~$ cat fichier_inexistant > res.txt 2> res.log
```

- Il y a deux redirections ici :
 - `> res.txt` : redirige le résultat de la commande (sauf les erreurs) dans le fichier `res.txt`. C'est la sortie standard ;
 - `2> res.log` : redirige les erreurs éventuelles dans le fichier `res.log`. C'est la sortie d'erreurs.
- Vous pouvez vérifier : si `fichier_inexistant` n'a pas été trouvé, l'erreur aura été inscrite dans le fichier `res.log` au lieu d'être affichée dans la console.

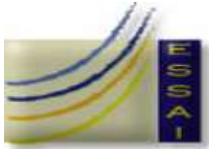


Notez qu'il est aussi possible d'utiliser `2>>`, pour ajouter les erreurs à la fin du fichier.



Les flux de redirection

- **Fusionner les sorties**
 - Si Parfois, on n'a pas envie de séparer les informations dans deux fichiers différents. Heureusement, il est possible de fusionner les sorties dans un seul et même fichier. Comment ?
 - Il faut utiliser le code suivant : `2>&1`.
 - Cela a pour effet de rediriger toute la sortie d'erreurs dans la sortie standard. Traduction pour l'ordinateur : « envoie les erreurs au même endroit que le reste ».



Les flux de redirection

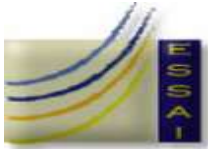
- Exemple:

```
essai2012@essai2012-VirtualBox: ~  
essai2012@essai2012-VirtualBox:~$ cat fichier_inexistant > res.txt 2>&1
```

- On va envoyer les deux sorties (standard et erreur) de cette commande vers le fichier res.txt



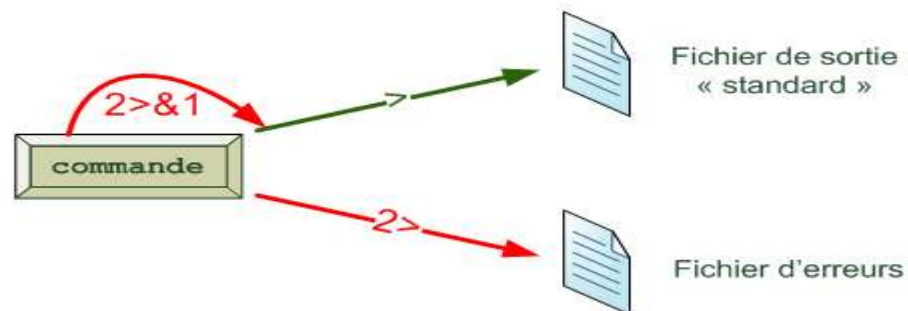
- L'utilisation **>** dans cette commande veut dire crée le fichier res.txt si il n'existe pas et si il existe écrase son contenu et le remplace par le nouveau résultat, toute fois si on veut écrire à la fin de fichier, on doit utiliser **>>**



Les flux de redirection

- **Résumé**

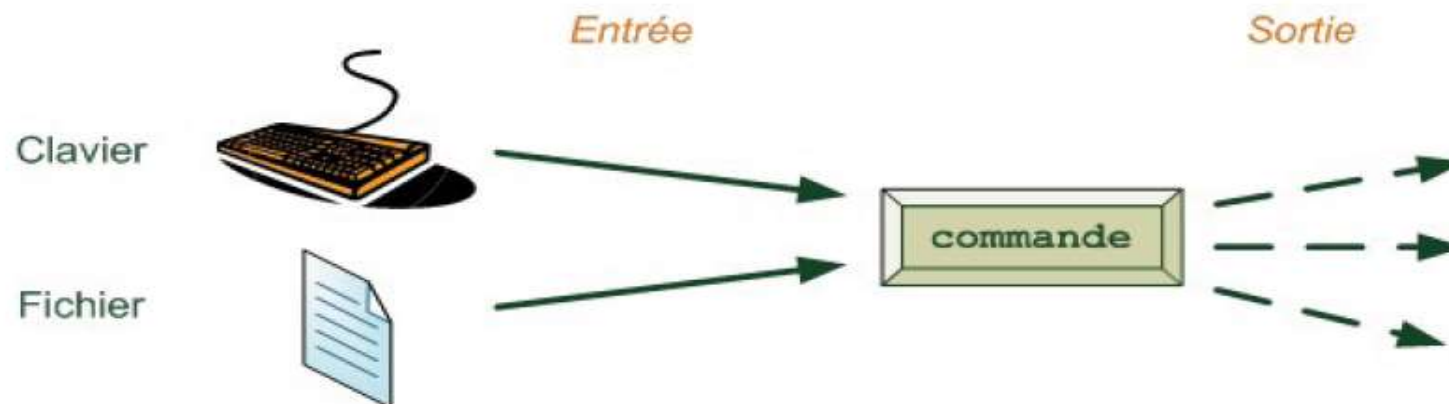
- Nous avons découvert trois symboles :
- `2>` : redirige les erreurs dans un fichier (s'il existe déjà, il sera écrasé) ;
- `2>>` : redirige les erreurs à la fin d'un fichier (s'il n'existe pas, il sera créé)
- `2>&1` : redirige les erreurs au même endroit et de la même façon que la sortie standard.
- Le tout est illustré sur la figure suivante.

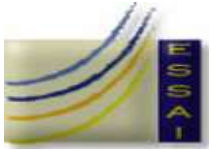




Les flux de redirection

- Lire de puis un fichier ou le clavier :
- **introduction**
 - La rentrée d'une commande Shell peut être prise à partir d'un fichier ou d'une saisie au clavier





Les flux de redirection

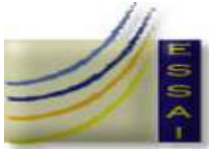
- **Lire depuis un fichier : <**
 - Le chevron ouvrant < permet d'indiquer d'où vient l'entrée qu'on envoie à la commande.

Exemple

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ cat < text
```



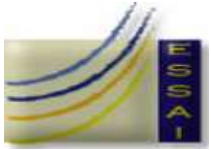
- Vous remarquez bien que il n'y a pas une différence dans le résultat entre la commande **cat text** et **cat < text**



Les flux de redirection



- Mais dans le traitement est différent:
 - Si vous écrivez **cat text**, la commande cat reçoit en entrée le nom du fichier **text** qu'elle doit ensuite se charger d'ouvrir pour afficher son contenu.
 - Si vous écrivez **cat < text**, la commande cat reçoit le contenu de **text** qu'elle se contente simplement d'afficher dans la console. C'est le shell qui se charge d'envoyer le contenu de **text** à la commande cat.



Les flux de redirection

- **Lire depuis le clavier: <<**
 - Le double chevron ouvrant << fait quelque chose d'assez différent : il vous permet d'envoyer un contenu à une commande depuis votre clavier.
 - Exemple:

The screenshot shows a terminal window with the prompt `administrateur@administrateur-VirtualBox: ~`. The command entered is `sort -n << tt`. The double chevron `<<` is highlighted with a green underline, and the `tt` is highlighted with a red underline. Two callout boxes provide explanations:

- A box pointing to `<<` says: "Donner comme entrée à la commande des données à partir du clavier."
- A box pointing to `tt` says: "Ce symbole est utilisé pour arrêter la saisie à partir du clavier. vous pouvez choisir n'importe quel symbole"



Les flux de redirection

```
administrateur@administrateur-VirtualBox:~$ sort -n << tt
> 12
> 4
> 8
> 3
> 1
> tt
```

Vous faites votre saisie

vous terminez la saisie en donnant le symbole

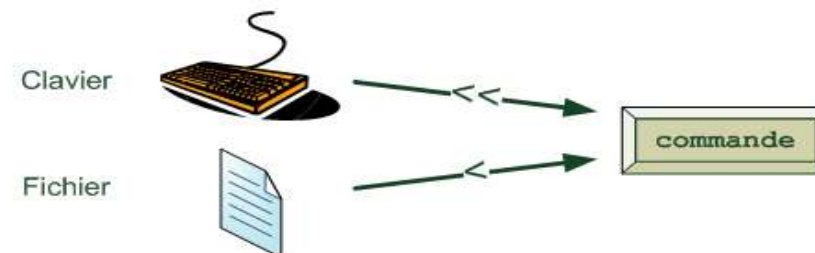


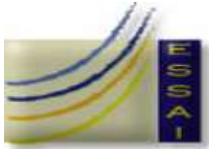
Les flux de redirection

- **Résumé**

- Nous pouvons donc « alimenter » des commandes de deux manières différentes, comme le montre la figure suivante :
- `<` : envoie le contenu d'un fichier à une commande ;
- `<<` : passe la console en mode saisie au clavier, ligne par ligne.

Toutes ces lignes seront envoyées à la commande lorsque le mot-clé de fin aura été écrit.





Les flux de redirection

- Les pipes ou chainage des commandes:
 - Les pipes ou Chainage des commandes c-a-d connecter la sortie d'une commande à l'entrée d'une autre commande.



- Exemple:

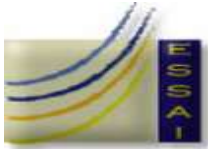
```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ du -h
```



Les flux de redirection

- Cette commande exécutée tout seule. Ne peut pas donner un résultat clair.
- Si on trie le résultat retourné par de cette commande puis l'afficher page par page. Ce dernier ça va être plus clair. Pour se faire, on va utiliser les pipes:

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ du -h | sort -n | more
```



MERCI POUR VOTRE ATTENTION

Ens: MANSOUR Sihem