

Les entrées-sorties standards

Lors de l'exécution d'une commande, un processus est créé. Celui-ci va alors ouvrir trois flux :

stdin l'entrée standard, par défaut le clavier, identifiée par l'entier **0** (descripteur)

stdout la sortie standard, par défaut l'écran, identifiée par l'entier **1**

stderr la sortie d'erreur standard, par défaut L'écran, identifiée par l'entier **2**

Les redirections

Il est possible de rediriger les flux d'entrée-sortie au moyen d'opérateurs spécifiques :

- > redirection de la sortie standard (par exemple dans un **fichier**)
- < redirection de l'entrée standard
- >> redirection de la sortie standard avec concaténation
- > & redirection des sorties standard et d'erreur
- >! redirection avec écrasement de fichier
- | redirection de la sortie standard vers **l'entrée standard** (pipe)

Redirection des entrées/sorties

Redirection de l'entrée

Exemple

- `cat < fichier`

afficher le contenu du fichier

- `sort < participants.txt`

L'entrée standard de sort est prise dans le fichier indiqué.

Redirection des entrées/sorties

Redirection de la sortie

Exemple

- `ls > résultats`

Créer un fichier nommé résultats contenant le contenu du répertoire courant

- `ls >> resultats`

Ajouter **à la fin du fichier** résultats le contenu du répertoire courant

- `ls ~/saddam/* > ~/gwb/weapons_mass_destruction.txt`
- `cat obiwan_kenobi.txt > starwars_biographies.txt`
- `cat han_solo.txt >> starwars_biographies.txt`

Redirection des entrées/sorties

Redirection entrée et sortie:


Exemple

- `cat < fichier >> résultat` → ajouter au fichier **résultat** le contenu de **fichier** après l'avoir récupérer avec cat,

Redirection vers des tubes (pipes): |

Les pipes Unix sont très utiles pour rediriger la sortie standard d'une commande vers l'entrée standard d'une autre commande.

Exemple:

- `ls | sort` → afficher le contenu du répertoire courant trié
- `cat *.log | grep -i error | sort`
- `grep -ri error . | grep -v "ignored" | sort u`  `> serious_errors.log`
- `cat /home/*/homework.txt | grep mark | more`

Il s'agit d'une des fonctionnalités les plus puissantes des shells Unix!

Liens Symboliques et Physiques

Un lien est un type spécial de fichier qui permet à plusieurs noms de fichiers de faire référence au même fichier sur le disque.(lien physique “dur” vs lien symbolique)

Liens Symboliques:

- Un lien symbolique est un fichier spécial qui est juste une référence au nom d'un autre (fichier ou répertoire).
- Utile pour simplifier et réduire l'utilisation du disque quand deux fichiers ont le même contenu.

Exemple:


biographie_anakin_skywalker -> biographie_darth_vador

- Comment distinguer les liens symboliques:

ls l : affiche -> et le fichier référencé par le lien

GNU ls affiche les liens avec **une couleur** différente

Création de liens Symboliques:

- Pour créer un lien symbolique (même ordre que dans cp):
`ln -s nom_fichier nom_lien`
- Pour créer un lien vers un fichier dans **un autre répertoire**, avec le même nom:
`ln -s ../LISEZ_MOI.txt`
- Pour créer plusieurs liens d'un coup dans un dossier donné:
`ln  fich1 fich2 fich3 ... Rep`
- Pour supprimer un lien:
`rm nom_lien`

Biensûr, cela ne supprime pas le fichier référencé par le lien!

Création de liens Physiques:

- Un *lien physique* vers un fichier est un fichier ordinaire, avec exactement le même contenu physique
- Bien qu'ils économisent toujours de la place, les liens physiques sont indiscernables des fichiers d'origine.
- Si vous supprimez le fichier d'origine, cela n'affecte pas le contenu du lien physique. Le contenu est supprimé quand il n'y a plus aucun fichier (lien physique) qui y fait référence.

Par défaut, **ln** crée des *liens physiques*

ln [options] fichier_source lien

Exemples :

```
$ ln /home/yannick/cours.pdf /home/yannick/projet/cours.pdf
```

le fichier cours.pdf du répertoire projet du répertoire yannick est un lien vers le fichier cours.pdf du répertoire yannick

```
$ ln -s /home/yannick/cours.pdf /home/yannick/projet/cours.pdf
```

idem avec un lien symbolique