

Redirections et tubes

[Précédent](#)

Chapitre 3. Introduction à la ligne de commande

[Suivant](#)

Redirections et tubes

Encore un mot au sujet des processus

Pour comprendre le principe des redirections et des tubes, ils nous faudra introduire ici une nouvelle notion qui concerne les processus. Chaque processus sous UNIX (y compris les applications graphiques) utilise un minimum de trois descripteurs de fichiers : l'entrée standard, la sortie standard et le canal d'erreur standard. Leurs numéros respectifs sont 0, 1 et 2. En général, ces trois descripteurs sont associés au `Terminal` depuis lequel le processus a été lancé, l'entrée standard étant lue depuis le clavier. Le but des redirections et des tubes est de rediriger ces descripteurs. Les exemples de cette section vous aideront à mieux comprendre.

Redirections

Supposons, par exemple, que vous vouliez connaître la liste des fichiers se terminant par `.png` ^[2] dans le répertoire `images`, et que cette liste soit très longue : il serait donc pertinent de la stocker dans un fichier pour la consulter à loisir ensuite. Vous pouvez alors taper ceci :

```
$ ls images/*.png 1>liste_fichiers
```

Ce qui signifie que la sortie standard de cette commande (1) est redirigée (>) vers le fichier qui a pour nom `liste_fichiers`. Le signe > est l'opérateur de redirection de sortie. Dans le cas où le fichier de redirection n'existerait pas, il serait alors créé. Par contre, s'il existait précédemment, son ancien contenu serait écrasé. Cependant, par défaut, le descripteur redirigé par cet opérateur est la sortie standard, il n'est donc pas nécessaire de le spécifier sur la ligne de commande. Vous pouvez donc écrire plus simplement :

```
$ ls images/*.png >liste_fichiers
```

et le résultat serait exactement le même. Vous pouvez ensuite consulter le fichier à l'aide d'un visualiseur de fichiers texte tel que **less**.

Supposons maintenant que vous vouliez connaître le nombre exact de ces fichiers. Au lieu de compter manuellement, vous pouvez utiliser le bien nommé **wc** (*Word Count*, soit comptage des mots) avec l'option `-l`, qui écrit sur la sortie standard le nombre de lignes du fichier. Pour obtenir le résultat désiré, une solution possible serait la suivante :

```
wc -l 0<liste_fichiers
```

Le signe < est l'opérateur de redirection d'entrée. Le descripteur redirigé par défaut est également celui de l'entrée standard, donc 0. La ligne s'écrit alors simplement :

```
wc -l <liste_fichiers
```

Supposons maintenant que vous vouliez retirer de cette liste toutes les extensions des fichiers puis sauvegarder le résultat dans un autre fichier. L'outil dont vous avez besoin est **sed**, pour *Stream EDitor* (soit éditeur de flux). Il suffit de rediriger l'entrée standard de **sed** vers le fichier `liste_fichiers` et de rediriger sa sortie vers le fichier résultat, par exemple `la_liste` :

```
sed -e 's/\.png$//g' <liste_fichiers >la_liste
```

Il vous sera également possible de consulter à loisir cette nouvelle liste avec un visualiseur.

Il pourrait aussi s'avérer utile de rediriger l'erreur standard. Par exemple, vous voulez savoir quels répertoires dans `/shared` ne vous sont pas accessibles : une solution est de lister récursivement ce répertoire et de rediriger les erreurs vers un fichier, tout en n'affichant pas le canal de sortie standard :

```
ls -R /shared >/dev/null 2>erreurs
```

Ceci signifie que la sortie standard sera redirigée (>) vers `/dev/null`, un fichier spécial dans lequel tout ce qu'on écrit est perdu (et aura comme effet secondaire que la sortie standard ne sera pas affichée) et que le canal d'erreur standard (2) sera redirigé (>) vers le fichier `erreurs`.

Tubes

Les tubes (*pipes* en anglais) sont en quelque sorte une combinaison des redirections d'entrée et de sortie. Leur principe mime en effet celui d'un tube : un processus envoie des données dans le tube par un bout et un autre processus lit les données par l'autre bout. L'opérateur tube est `|`. Reprenons l'exemple de la liste des fichiers `.png` ci-dessus. Supposons que vous vouliez seulement connaître le nombre de fichiers en question sans avoir à stocker la liste dans un fichier temporaire : utilisez alors la commande suivante :

```
ls images/*.png | wc -l
```

ce qui signifie que la sortie standard de la commande **ls** (donc la liste des fichiers) est redirigée vers l'entrée standard de la commande **wc**. Vous obtenez donc le résultat désiré.

Vous pouvez de même construire directement la liste des fichiers « sans les extensions » avec la commande suivante :

```
ls images/*.png | sed -e 's/\.png$//g' >la_liste
```

ou, si vous voulez simplement consulter la liste sans la stocker dans un fichier :

```
ls images/*.png | sed -e 's/\.png$//g' | less
```

Les tubes et les redirections ne sont pas limités à du texte. Ainsi en est-il de la commande suivante, lancée à partir d'un Terminal :

```
xwd -root | convert - ~/mon_bureau.png
```

ce qui effectuera une capture d'écran de votre bureau dans le fichier intitulé `mon_bureau.png` ^[8] dans votre répertoire personnel.

^[2] Il peut vous paraître saugrenu de dire « les fichiers se terminant par `.png` » plutôt que « les images PNG ». Mais, encore une fois, les fichiers sous UNIX n'ont d'extension que par convention : une extension ne détermine en aucun cas le type d'un fichier. Un fichier dont le nom se termine par `.png` peut indifféremment être une image JPEG, un exécutable, un fichier texte ou tout autre chose. Il en est de même sous Windows !

^[8] Oui, ce sera bien une image PNG ! (Le paquetage `ImageMagick` devra néanmoins être installé...)

[Précédent](#)

Motifs d'englobement du shell

[Niveau supérieur](#)

[Sommaire](#)

[Suivant](#)

Le complètement (completion) dans les lignes de commande