

OUTILS SHELL

STDIN, STDOUT, STDERR

La philosophie d'Unix est, lorsqu'on fait un outil, d'au lieu de faire un gros outil qui fait tout, de faire des petits outils qui font une seule chose mais qui la font bien, et de les combiner ensemble.

Pour gérer cette communication entre outils, chaque processus a, dans son environnement, 3 variables qui sont définies :

- l'**entrée standard** (STDIN), où il va lire ses informations ;
- la **sortie standard** (STDOUT), où il va écrire ;
- la **sortie d'erreur standard** (STDERR), où les messages d'erreurs vont s'écrire.

Par défaut, les 3 variables sont dirigées vers la console. Par exemple, la commande « cat nomFichier » affiche le contenu du fichier sur la sortie standard, c'est à dire dans la console. Et la commande « cat » tout court, sans argument, attend que l'utilisateur entre des informations sur l'entrée standard (la console) pour les renvoyer vers la sortie standard (toujours la console).

On peut faire communiquer des outils en redirigeant une ou plusieurs de ces variables vers un autre processus, par exemple en redirigeant la sortie standard d'un processus vers l'entrée standard d'un autre. Il est aussi possible de les rediriger vers un fichier pour un traitement ultérieur. Ces redirections sont combinables entre elles, et il est possible ainsi de faire une longue ligne de commande qui enchaîne les commandes les unes après les autres.

command1 | command2

Redirige la sortie standard de *command1* vers l'entrée standard de *command2*. Par exemple, « ls | cat » prend le résultat de *ls* et l'envoie comme entrée de *cat*.

command1 > file

Redirige la sortie standard de *command1* dans le fichier *file*. Attention, le contenu du fichier *file* est écrasé et remplacé par le résultat de la commande *command1*.

command1 >> file

Redirige la sortie standard de *command1* vers la fin du fichier *file*. Le contenu du fichier n'est pas écrasé, le résultat de la commande est ajouté à la fin du fichier.

command1 < file

Redirige l'entrée standard de *command1* vers le fichier *file*. C'est comme si le fichier *file* était donné en entrée de la commande *command1*.

Outils

Voici quelques outils de la ligne de commande qui vous seront très utiles, notamment pour rechercher de l'information dans un répertoire projet rempli de code source par exemple.

N'oubliez pas la commande « *man* » vu au début de ce module.

history

Retourne la liste des commandes que vous avez tapé dans la ligne de commande depuis très longtemps (la taille du cache est fixée dans le shell). Très pratique pour retrouver une ancienne commande.

!!

Est remplacé par la dernière commande tapée. Par exemple, « *sudo !!* » est très pratique lorsqu'on a oublié de faire un *sudo* sur une commande...

strings *fichierBinaire*

Affiche toutes les chaînes de caractère trouvables dans un fichier binaire (executable).

find

Permet de rechercher un ou des fichiers, suivant différents critères (notamment le nom avec -name). Il est aussi possible d'associer une commande à réaliser pour chacun des noms de fichiers trouvés, avec -exec.

grep

Permet de rechercher un motif dans un document. -i permet de faire une recherche sans prendre en compte la casse (pas de différence en les majuscules et les minuscules). -r permet de faire la recherche de manière récursive, dans une hiérarchie de répertoires.

head

Affiche le début d'un fichier.

tail

Affiche la fin d'un fichier.

wc

Permet de compter des items, comme des mots, ou des lignes (avec -l).

sort

Permet de classer les items.