

Mémo BASH

Bruno GUILLON

Octobre 2022

Répertoire courant : se situer dans l'arborescence

cmde	signification	commentaires	exemples
pwd	print working directory	affiche le chemin absolu du répertoire courant	<ul style="list-style-type: none">• \$ pwd
cd	change directory	change le répertoire courant	<ul style="list-style-type: none">• \$ cd sous-rép/• \$ cd /chemin/absolu/répo/• \$ cd ~• \$ cd
ls	list	liste des fichiers et/ou le contenu de répertoires	<ul style="list-style-type: none">• \$ ls• \$ ls rép/• \$ ls fichier1 fichier2• \$ ls rép1/ rép2/ fichier1
tree	arbre en Anglais	affiche la structure arborescente d'un répertoire de manière graphique (dans le terminal)	<ul style="list-style-type: none">• \$ tree• \$ tree .• \$ tree /tmp• \$ tree tp/exo{1,2}

Écrire du texte

cmde	signification	fonction	exemples
echo	echo	écrire une ligne	<ul style="list-style-type: none">• \$ echo "Hello world!"• \$ echo *.tex *.pdf• \$ echo tp3/exo{1{, /tmp},2,3}/ tp3/exo1/fichier{1,2,3,4}
printf	print and format	écrire du texte formaté, par défaut sans fin de ligne	<ul style="list-style-type: none">• \$ printf '%s\n' "Hello world!"• \$ printf '%s\n' "Hello world!" "Bonjour monde !"• \$ printf '%d: %s\n' 0 "Hello" 1 "Salut"
tee	la lettre T, prononcée en Anglais, qui ressemble à un tube (caractère) avec bifurcation (et qui est ainsi nommée en plomberie)	écrit chaque ligne de la sortie standard sur des fichiers indiqués, tout en préservant l'écriture en sortie standard (voir bas)	<ul style="list-style-type: none">• \$ echo "Bonjour monde" tee f.out• \$ echo "Hello world" tee -a f.out• \$ ls tee liste-fichiers.out

Redirections

En première approximation, chaque commande s'exécute avec différents flux de texte (associés à un numéro) :

0. une entrée standard, qui peut être lue par la commande (par défaut, ce qui est entré au clavier dans le terminal) ;
1. une sortie standard, dans laquelle la commande écrit des message (par défaut, affiché dans le terminal) ;
2. une erreur standard, dans laquelle la commande écrit des messages d'erreur et d'avertissement (par défaut, également affiché dans le terminal, ce qui donne l'impression que la sortie et l'erreur standard sont mélangés).

Il est toute fois possible de *rediriger* ces flux.

opérateur	fonction	exemples
>	écrit la sortie standard d'une commande dans un fichier indiqué (créé s'il n'existe pas, écrasé sinon)	<ul style="list-style-type: none">• <code>\$ echo "Bonjour monde" >f.out</code>• <code>\$ printf '%s\n' "Hello world!" >f.out</code>• <code>\$ ls >liste-fichiers.out</code>
>>	écrit la sortie standard d'une commande en ajout dans un fichier indiqué (créé s'il n'existe pas, étendu (écriture en ajout à la fin) sinon)	<ul style="list-style-type: none">• <code>\$ echo "Bonjour monde" >>f.out</code>• <code>\$ printf '%s\n' "Hello world!" >>f.out</code>• <code>\$ ls >>liste-fichiers.out</code>
<	lit l'entrée standard depuis un fichier indiqué existant (plutôt que depuis la saisie clavier en terminal)	<ul style="list-style-type: none">• <code>\$ cat <f.in</code>• <code>\$ cat <f.in >f.out #une sorte de copie</code>• <code>\$ bash -i <script.sh</code>
	tube (<i>pipe</i>) : séparateur de commande qui définit l'entrée standard de la commande de droite comme la sortie standard de la commande de gauche	<ul style="list-style-type: none">• <code>\$ cat liste-fichiers.out grep 'tex'</code>• <code>\$ ls tee liste-fichiers.out</code>• <code>\$ pdflatex main.tex grep 'label'</code>
2>	comme > mais redirige cette fois l'erreur standard	<ul style="list-style-type: none">• <code>\$ pdflatex main.tex 2>f.err</code>
2>>	comme >> mais redirige cette fois l'erreur standard	<ul style="list-style-type: none">• <code>\$ pdflatex main.tex 2>>f.err</code>

En fait, chaque flux ayant son numéro, il est possible de rediriger le flux à l'aide de, *e.g.*, `n>` où `n` est le numéro associé au flux. (`>` et `>>` sont des abréviations de `1>` et `1>>`; `<` est une abréviation de `0<`.) Il est même possible de créer de nouveaux flux (avec des numéros libres, comme `3`, `4`, *etc...*). À droite de l'opérateur de redirection, on peut indiquer un fichier (comme dans les exemples plus haut) ou bien, un numéro de flux précédé du caractère `&`. Par exemple :

- `$ echo "Erreur: c'est pas bien" >&2`
- `$ pdflatex main.tex 2>&1 1>sortie.out | grep 'Warning'`

Interagir avec le système de fichier

cmde	signification	fonction	exemples
<code>mkdir</code>	make directory	créer des répertoires	<ul style="list-style-type: none"> • <code>\$ mkdir nouveau/</code> • <code>\$ mkdir -p inexistant/nouveau/</code> • <code>\$ mkdir nouveau1/ nouveau2/</code>
<code>touch</code>	<code>touch</code>	met à jour la date d'accès d'un fichier existant/créer un fichier vide	<ul style="list-style-type: none"> • <code>\$ touch inexistant</code> • <code>\$ touch existant</code> • <code>\$ touch inexistant1 inexistant2</code>
<code>cp</code>	copy	copier un ou des fichiers	<ul style="list-style-type: none"> • <code>\$ cp -iv original copie</code> • <code>\$ cp -iv fichier1 fichier2 fichier3 répertoire/</code> • <code>\$ cp -ivt répertoire/ fichier1 fichier2...</code> • <code>\$ cp -ivr répertoire-original répertoire-copie</code>
<code>mv</code>	move	déplacer/renommer	<ul style="list-style-type: none"> • <code>\$ mv -iv fichier répertoire/</code> • <code>\$ mv -iv ancien-nom nouveau-nom</code> • <code>\$ mv -iv fichier1 fichier2 fichier3 répertoire/</code> • <code>\$ mv -ivt répertoire/ fichier1 fichier2 fichier3</code>
<code>rm</code>	remove	supprimer un fichier	<ul style="list-style-type: none"> • <code>\$ rm -iv fichier</code> • <code>\$ rm -iv fichier1 fichier2</code> • <code>\$ rm -ivr répertoire/</code>
<code>rmdir</code>	remove directory	supprimer un répertoire vide	<ul style="list-style-type: none"> • <code>\$ rmdir -iv répertoire/</code> • <code>\$ rmdir -iv rep1/ rep2/</code> • <code>\$ rmdir -iv répertoire/</code>
<code>ln</code>	link	créer des liens physiques ou symboliques (raccourcis)	<ul style="list-style-type: none"> • <code>\$ ln -iv fichier chemin-alternatif #lien physique</code> • <code>\$ ln -ivs original raccourci-à-créer #lien symbolique</code> • <code>\$ ln -ivs original1 original2 original3 répertoire/</code>

Les commandes `rm`, `cp` et `ln` sont particulièrement dangereuses car elles peuvent entraîner une perte de données. Il est vivement recommandé d'utiliser les options `-i/--interactive` (demande confirmation avant d'écraser/supprimer un fichier), et `-v/--verbose` (affiche les opérations réalisées dans des messages, sur la sortie standard).