

Les commandes du Shell, les métacaractères et les droits sur les fichiers

Septembre 2015

Objectifs

Utiliser le `man`.

Apprendre les commandes de base du shell.

Découvrir les droits sur les fichiers.

1 Le Man

Les commandes qui fournissent de l'aide sont : `man` et `info`.

Ainsi si vous voulez savoir ce que fait la commande `man`, saisissez `man man`.

Pour vous en sortir vous devez apprendre à lire un `man` et notamment savoir lire le synopsis d'une commande.

▷ **Exercice 1 : Utilisation de man**

Décortiquez avec l'enseignant le manuel de `cp`.

Que fait la commande `cat` si elle n'a pas de fichier en arguments ?

Comment afficher les droits d'un répertoire donné et non son contenu avec la commande `ls` ?

Que fait la commande suivante : `ls -ld ..`

2 Utilisation du shell

Le shell est un interpréteur de commande. Il sert d'intermédiaire entre l'utilisateur et le système d'exploitation.

Pour interagir avec la station Linux en ligne de commande, nous ouvrirons une fenêtre console ou terminal. Ce qui est saisi au clavier dans celle-ci est analysé par le shell puis exécuté.

Le premier mot de la ligne doit être une commande suivie éventuellement d'options et d'arguments.

Pour savoir comment le système va interpréter une commande utilisez `which`. Essayez `which ls`.

Astuce : pour ne pas faire des erreurs de saisie et gagner du temps, utilisez la complétion des noms de commande ou de fichier avec la touche TAB suivie de Control d.

3 Utilisation des commandes de base

Dans cette partie du TP vous allez travailler sur le répertoire `Arbo` qui contient un exemple d'arborescence. Celle-ci est composée de fichiers récupérés par un élève de 1^{er} TR sur l'intranet.

Vous pouvez faire tous les tests proposés même si cela provoque des erreurs.

▷ **Exercice 2 : Commandes ls, pwd et chemin absolu/relatif**

Listez le contenu du répertoire `Download` avec `ls` depuis `Arbo`, puis en étant dans celui-ci.

Placez vous à la racine de votre compte (`cd` sans arguments). Vérifiez (`pwd`). Lister à nouveau le contenu de `Download`, en indiquant son chemin en absolu puis en relatif. Pensez à utiliser l'astuce de la partie 2 pour gagner du temps et éviter les erreurs.

▷ **Exercice 3 : Commandes `more`, `less`, `cat` : visualiser des fichiers**

En utilisant l'éditeur de texte `gedit`, créez un fichier premier dans le répertoire `Arbo` contenant une ou deux phrases.

Visualisez ensuite son contenu sans l'éditer avec les commandes suivantes `more`, `cat` et `less`. Quelles différences observez vous ?

▷ **Exercice 4 : Commande `cp`**

1. Placez vous dans le répertoire `1TR/AccSys` (`cd`). Vérifiez ou vous êtes (`pwd`).
2. Trouvez les options de la commande `cp` qui permettent de faire une copie d'un répertoire et de l'ensemble de ses sous-répertoires en une seule ligne de commande.
3. Utilisez les pour copier le répertoire `Arbo` dans `AccSys`. Allez dans le répertoire `Arbo`.
4. Que se passe-t-il si l'on veut faire une copie appelée `BDE/` de son répertoire `Prog/`, et que le répertoire `BDE/` existe déjà ?
5. Que se passe-t-il si vous voulez faire une copie d'un fichier dans un répertoire où il existe déjà un fichier de ce nom ?
6. Est-il possible de taper une ligne du genre de : `cp tp-cesar.zip tp.tar Download/` ? Quel est l'effet de cette ligne ? Quel sera le nom des nouveaux fichiers ?

▷ **Exercice 5 : Commande `mv`**

1. Changez le nom du répertoire `Arbo` source de la copie en `Poubelle` (`mv`).
2. Que se passe-t-il si le répertoire existe déjà ?
3. Comment faire pour à la fois déplacer un fichier et changer son nom ?
4. Renommez en `Introrez0.pdf` et déplacez le fichier `Arbo/IntroL4.pdf` dans le répertoire `Arbo/1TR/Docs`. Vérifiez avec `ls`.
5. Que se passe-t-il si on lance `mv` avec les options `-i` et `-f` ensemble ?

Astuce : ce que vous avez saisi sur la ligne de commande est mémorisé dans un historique. Utilisez les flèches haut et bas pour retrouver vos commandes et évitez de les ressaisir. Pour en savoir plus sur le recours à l'historique `info history`.

▷ **Exercice 6 : Commandes mkdir et rmdir, rm**

1. Vérifiez que vous êtes toujours dans le répertoire Arbo. Créez le répertoire WEI.
2. Créez les répertoires 2TR et 3TR en une seule ligne de commande.
3. Que se passe-t-il si vous tapez : mkdir mkdir ?
4. Vous voulez détruire WEI, ls vous dit qu'il est vide. Cependant, quand vous tapez rm WEI, la machine vous répond : rm: impossible de supprimer «WEI»: est un dossier
Que faut-il faire ? Quelles sont les options à utiliser (attention aux alias) ?
5. A quelle condition est-ce que la ligne de commande mkdir 1TR/Archives peut fonctionner et créer effectivement ce sous-répertoire ?
6. Trouver l'option de mkdir qui permette de créer à la fois un répertoire et un sous-répertoire.
7. Placez vous dans 1TR et utilisez cette option pour créer dans 1TR à la fois le répertoire Examens et un sous répertoire Stats dans Examens/.
8. Quel est l'effet de la commande suivante : rmdir -p Examens/Stats ?
9. Supprimez le répertoire Poubelle et son contenu avec rmdir et rm.

Pensez à vérifier ce qui s'exécute derrière les commandes.

4 Des caractères bien utiles

L'utilisation de caractères spéciaux dans la ligne de commande, notamment pour les noms de fichiers, permet de désigner des ensembles. Voici ces caractères appelés aussi **méta-caractères** : *, ?, [], et ^.

Syntaxe	Signification
*	remplace une chaîne de longueur qq, même vide
?	remplace un seul caractère qq
[ar] [3-8]	un caractère qq de la liste ou de l'intervalle, pas d'espace dans les crochets
[^gte] [^a-z]	n'importe quel caractère sauf ceux de la liste, pas d'espace dans les crochets

C'est le shell qui interprète ces caractères. Pour que ceux-ci soient considérés seulement comme un caractère normal, il faut placer un \ devant.

Il ne faut pas confondre ces méta-caractères permettant de désigner des ensemble de fichiers avec les expressions régulières utilisées par d'autres commandes telles que grep.

Le répertoire de travail est toujours Arbo. Vous avez remarqué que tout est en vrac. Les commandes ci-dessous vont vous permettre de ranger petit à petit ce répertoire.

▷ **Exercice 7 :**

1. Listez les fichiers commençant par tp ou td.
2. Listez les fichiers finissant par .pdf.
3. Déplacez les sujets de TP de 1TR dans le répertoire 1TR/TPs. Pouvez-vous savoir avec le nom du fichier s'il s'agit de la première année, d'un énoncé ?
4. Créez un répertoire Sources dans 1TR/TPs et déplacez y les fichiers matlab (.m).
5. Rangez en une seule commande les fichiers utilisés en langage C (.h et .c) dans Prog.
6. Déplacez au même endroit le fichier makefile.

5 Droits d'accès

5.1 Les fichiers

Un fichier a un propriétaire, un groupe et un ensemble de droits d'accès définissant les possibilités de lecture, écriture et exécution du fichier pour les utilisateurs du système.

Il existe trois ensembles d'utilisateur :

- le **propriétaire** du fichier
- les utilisateurs du même **groupe** de travail que le propriétaire
- les **autres** utilisateurs ayant accès au système

Droits	Signification pour un fichier	Signification pour un répertoire
Lecture (r)	droit de lecture du contenu du fichier	droit de lecture du contenu du répertoire
Écriture (w)	droit de modification du fichier	droit de modif. du contenu du répertoire, suppression et création de fichier
Exécution(x)	droit d'exécuter le fichier	droit de passage dans le répertoire

▷ Exercice 8 :

1. Pouvez vous créer un répertoire sous /home ?
2. Dans le répertoire AccSys, créez un sous-répertoire repdroits. Quels sont les droits par défaut de ces répertoires ?
3. Créez un fichier droit.txt dans le répertoire repdroits. Quels sont ses droits par défaut ?

5.2 Commande chmod

Les droits (ou permissions) d'accès ne peuvent être changés que par le propriétaire du fichier, grâce à la commande chmod. Les options de cette dernière peuvent être numériques ou symboliques.

Valeurs numériques pour chmod

Valeur	Permission	Droits
0	Pas d'accès en lecture, pas d'accès en écriture, pas d'accès en exécution	---
1	Pas d'accès en lecture, pas d'accès en écriture, exécution	--x
2	Pas d'accès en lecture, écriture, pas d'accès en exécution	-w-
3	Pas d'accès en lecture, écriture, exécution	-wx
4	Lecture, pas d'accès en écriture, pas d'accès en exécution	r--
5	Lecture, pas d'accès en écriture, exécution	r-x
6	Lecture, écriture, pas d'accès en exécution	rw-
7	Lecture, écriture, exécution	rwx

Options symboliques pour chmod

Option	Lettre	Signifie
(qui)	u	Utilisateur
(qui)	g	Groupe
(qui)	o	Autre
(qui)	a	Tous (« le monde entier »)
(action)	+	Ajouter des permissions
(action)	-	Retirer des permissions
(action)	=	Fixe les permissions de façon explicite
(permissions)	r	Lecture
(permissions)	w	Ecriture
(permissions)	x	Exécution

▷ **Exercice 9 :**

Nous allons travailler sur le répertoire `Arbo/Prog`.

1. Vérifiez les droits sur celui-ci et que vous pouvez lire son contenu (`r`), y créer un répertoire (`w`), s'y placer (`x`) et afficher le fichier `makefile` (`x`).
2. Enlevez les droits de lecture sur le répertoire `Prog`. Vérifiez votre modification puis refaites les quatre tests ci-dessus.
3. Remettez les droits de lecture et enlevez ceux d'écriture en une seule commande. Vérifiez les droits.
Pouvez vous modifier le fichier `makefile` ? Pouvez vous renommer le fichier `makefile` ?
4. Remettez les droits d'écriture sur le répertoire et enlevez les pour le fichier `makefile`. Refaites les deux test ci-dessus.
5. Trouvez les commandes pour donner les droits suivants à votre répertoire :

	propriétaire			groupe			les autres		
	lecture	écriture	accès	lecture	écriture	accès	lecture	écriture	accès
<i>cas 1</i>	<i>oui</i>	<i>oui</i>	<i>oui</i>	<i>oui</i>	<i>non</i>	<i>oui</i>	<i>non</i>	<i>non</i>	<i>oui</i>
<i>cas 2</i>	<i>oui</i>	<i>non</i>	<i>oui</i>	<i>non</i>	<i>oui</i>	<i>non</i>	<i>non</i>	<i>non</i>	<i>oui</i>
<i>cas 3</i>	<i>non</i>	<i>oui</i>	<i>non</i>	<i>non</i>	<i>non</i>	<i>oui</i>	<i>oui</i>	<i>non</i>	<i>non</i>
<i>cas 4</i>	<i>non</i>	<i>non</i>	<i>oui</i>	<i>oui</i>	<i>non</i>	<i>oui</i>	<i>non</i>	<i>non</i>	<i>non</i>

6. Depuis votre `home`, pour chaque commande de l'exercice précédent :
 - essayez d'accéder au répertoire `repdroits` (`x` sur le `rep`),
 - de lister ce répertoire (`r` sur le `rep`),
 - de modifier le fichier `droit.txt` avec un éditeur quelconque (`w` sur le fichier, `x` sur le `rep`)
 - de renommer le fichier `droit.txt` (`w` du `rep`).
7. Supprimer en une seule commande le `rep` `repdroits`.