

Les caractères spéciaux du shell

Un certain nombre de caractères ont une signification spéciale.

Nom de fichiers et de répertoires

6 caractères servent à générer les noms de fichiers ou de répertoires :

- . qui désigne le répertoire courant.
- .. qui désigne le répertoire père, c'est à dire celui qui se trouve immédiatement au-dessus du répertoire courant dans la hiérarchie.
- ~ qui désigne le répertoire d'accueil.
- ? remplace un caractère quelconque.
- * remplace une chaîne de caractères quelconque (*y compris une chaîne vide*).
- [...] remplace les caractères entre crochets, définis par énumération ou par intervalle.

Exemples :

[**Aa**] désigne les caractères **A** ou **a**

[**a-zA-Z**] désigne un caractère alphabétique quelconque.

Les autres caractères spéciaux

- # introduit un commentaire.
- \$ introduit un nom de variable.
- & lance la commande en **background**.
- ; sépare 2 commandes tapées sur une même ligne.
- !! relance la commande précédente.
- !*ch* relance la dernière commande commençant par *ch*.
- !*n* relance la commande numéro *n*.
- ^*ch1*^*ch2* remplace la chaîne *ch1* par la chaîne *ch2* dans la dernière commande.
Exemple : Si la dernière commande est : **cat bidon.txt**
la commande : ^**od**^**do** générera la ligne de commande : **cat bidon.txt**
- < << > >> et | sont des caractères utilisés pour les redirections des entrées-sorties.
- ' délimite une chaîne de caractères contenant des espaces **sans** substitution de variables.
- " délimite une chaîne de caractères contenant des espaces **avec** substitution de variables.
- \ annule la signification du caractère spécial suivant.
- (et) font exécuter les commandes incluses dans un sous-shell.
- { et } regroupent les commandes incluses en une liste.

Redirections des entrées-sorties

Il est possible, sous Unix, de rediriger l'entrée ou la sortie d'une commande :

- < : redirection de l'entrée

Exemple :

grep blabla < essai : la chaîne *blabla* est recherchée dans le fichier *essai*

- **>** : redirection de la sortie

Exemple :

ls > liste : le résultat de la commande *ls* est copié dans le fichier *liste*. Si le fichier n'existe pas, il est créé, sinon son contenu est écrasé.

- **>>** : redirection de la sortie avec concaténation.

Exemple :

ls >> liste2 : si le fichier *liste2* n'existe pas, il est créé, sinon le résultat de la commande *ls* est ajouté à la fin du fichier *liste2*.

- **|** : redirection de la sortie d'une commande sur l'entrée de la commande suivante.

Exemple :

ls | grep *.c : la sortie de la commande *ls* sert d'entrée à la commande *grep*. Ici, on cherche tous les fichiers sources C dans le répertoire courant.