

3- INTRODUCTION A L'UTILISATION D'UNIX

3.1- Fichiers et noms

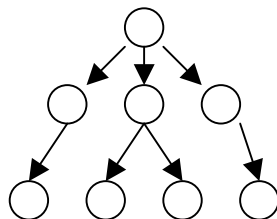
Du point de vue du système d'exploitation, tout ensemble de données que l'utilisateur désire garder en mémoire de masse constitue un *fichier* (*file* en anglais). Exemples : si vous écrivez un programme ou du courrier avec un éditeur de texte, l'éditeur sauvegardera votre travail dans un fichier ; si vous compilez un programme pour produire du code machine exécutable, le programme binaire exécutable résultant sera placé par le compilateur dans un fichier ; si vous dessinez un plan avec un logiciel de dessin, ce dernier produira un fichier pour garder votre plan de façon permanente.

Pour être retrouvé, chaque fichier doit recevoir un nom. Selon les systèmes d'exploitation, des contraintes diverses s'appliquent à la construction d'un nom de fichier. Par exemple UNIX fait la différence entre les majuscules et les minuscules (ainsi MONFICH, monfich et MonFich désignent trois fichiers différents) et interdit certains caractères spéciaux (notamment < et >).

3.2- Répertoires et chemin d'accès

La quasi-totalité des systèmes d'exploitation utilisent un modèle hiérarchique d'organisation des fichiers basé sur la notion de *répertoire* (*directory* en anglais). L'espace disque est ainsi partitionné en répertoires. Le répertoire est l'unité d'organisation. Chaque répertoire peut contenir des fichiers et d'autres répertoires (appelés sous-répertoires). Chaque répertoire est nommé, les noms obéissant en général aux mêmes contraintes que les noms de fichiers.

La structure hiérarchique constituée par les répertoires s'appelle un *arbre*. La structure d'arbre est l'une des structures de données les plus importantes et les plus spécifiques de l'informatique. On peut se représenter ce type d'organisation en pensant par exemple à un arbre généalogique.



3.2.1- Répertoire racine

Le répertoire qui n'appartient à aucun répertoire de rang plus élevé est appelé *répertoire racine*. Il est désigné par un nom standardisé : /

3.2.2- Chemin d'accès

Pour retrouver un fichier dans une arborescence de répertoires, il faut spécifier, outre le nom du fichier, son *chemin d'accès*. Le chemin d'accès est la succession des sous-répertoires qu'il faut traverser pour arriver dans celui où se trouve le fichier. Le chemin d'accès peut être spécifié à partir du répertoire racine (chemin d'accès absolu) ou à partir d'un répertoire dit courant (chemin d'accès relatif).

Exemple de référence de fichier :

/user/info/morellea/exemples/prg.C
chemin d'accès nom du fichier

3.2.3- Répertoire courant

Le *répertoire courant* se définit comme le répertoire par défaut dans lequel le système va chercher vos fichiers si vous ne spécifiez de chemin d'accès. Le répertoire courant est désigné par un nom standardisé : `.`

Exemple (on reprend l'exemple précédent) :

Si le répertoire courant est le répertoire : `/user/info/morellea/exemples/`
alors la simple spécification de `prg.C` suffit au système pour le retrouver.

Si le répertoire courant est le répertoire : `/user/info/morellea/`
alors une référence correcte pourrait être : `./exemples/prg.C`

Au login (i.e. à la connexion ou à l'ouverture d'une session de travail), le répertoire courant est le *répertoire de login*.

La commande UNIX **pwd** (path of working directory) permet d'afficher le nom du répertoire courant ; la commande **cd** (change directory) permet de changer de répertoire courant.

3.2.4- Répertoire père

Si A est un sous-répertoire direct de B, alors B est appelé *répertoire père* de A. Le répertoire père du répertoire courant est désigné par le nom standardisé : `..`

Exemple (on reprend l'exemple précédent) :

Si le répertoire courant est le répertoire : `/user/info/morellea/exemples/perso/an2000/`
alors une référence correcte pourrait être : `../prg.C`

3.2.5- Caractères jokers

Supposons que dans le répertoire courant il y ait 13 fichiers de nom : `tmp`, `tmp1`, `tmp2`, ..., `tmp11`, `tmp12`. Comment faire pour les effacer en une fois sachant qu'il existe une commande **rm** (remove) qui permet de supprimer un fichier ? La solution est l'utilisation de caractères jokers dans l'énoncé du nom du fichier.

Le caractère `?` est un joker qui peut filtrer n'importe quel caractère.

Exemple : **rm tmp?** détruira d'un coup tous les fichiers `tmp1` à `tmp9`

Le caractère `*` est un joker qui peut filtrer une séquence quelconque de caractères, éventuellement vide.

Exemple : **rm tmp*** détruira d'un coup tous les fichiers `tmp`, `tmp1`, ..., `tmp12`

*Remarque : ATTENTION à une commande telle que `rm *` qui détruit immédiatement et irrémédiablement tous les fichiers du répertoire courant !! PRUDENCE donc dans l'utilisation des caractères jokers !*

Les caractères jokers peuvent être utilisés dans tout énoncé de nom de fichier ou de répertoire.

3.3- Principales commandes

Un ordre UNIX est appelé *commande*. Par la suite les commandes sont spécifiées en gras.

Certaines commandes attendent des *paramètres*. Par exemple si on veut supprimer un fichier, le paramètre de la commande de suppression sera évidemment le nom de ce fichier (éventuellement précédé par son chemin d'accès!). Par la suite les paramètres seront désignés par des mots soulignés.

La plupart des commandes peuvent être utilisées telles quelles ou accompagnées d'*options*. Une option est un appendice que l'on ajoute à la commande pour en modifier ou préciser le sens. Par la suite les options seront spécifiées en italique.

Voici un kit de survie pour débuter en UNIX :

3.3.1- Commandes générales

- se connecter (indispensable) **login** identity
- se déconnecter (indispensable) **logout**
- changer de mot de passe (password) **passwd**
- rechercher une commande à partir d'un mot clé **man** *-k* keyword
- obtenir le mode d'emploi d'une commande (man signifie "manual") **man** command

3.3.2- Commandes agissant sur les fichiers

- copier un fichier dans un autre **cp** SourceFile DestinationFile
- effacer un fichier (remove) **rm** file
- renommer un fichier **mv** OldFileName NewFileName
- déplacer un fichier (move) **mv** file directory

3.3.3- Commandes agissant sur les répertoires

- afficher le nom du répertoire courant (path of working directory) **pwd**
- afficher la liste des fichiers et sous-répertoires du répertoire courant (list) **ls**
- afficher la liste détaillée des fichiers et sous-répertoires du répertoire courant : nom, taille, date de création, droits d'accès, ... (list long) **ls -l**
- changer de répertoire courant **cd** path
- créer un nouveau répertoire (make directory) **mkdir** path
- détruire un répertoire (remove directory) **rmdir** path