


Introduction aux systèmes d'exploitation (IS1)

TP n° 2 : organisation du système de fichiers

Pour ce TP vous devez télécharger l'archive `arborescence_TP2.tar` qui se trouve sur Moodle, dans les documents du cours IS1, puis la désarchiver dans votre répertoire `~/Cours/2019/IS1` grâce à la commande « `tar -xf arborescence_TP2.tar` ».

Modalités de rendu Au cours du TP, vous devez copier dans un fichier texte appelé `reponses_TP2.txt` les commandes utilisées pour répondre aux questions marquées par le symbole  (et seulement à celles-ci), en y joignant le cas échéant le résultat de ces commandes ou une réponse à la question (à voir en fonction de la question). Pensez à indiquer pour chaque réponse le numéro de l'exercice concerné. Vous insérerez dans ce fichier vos nom(s) et prénom(s) (attention à ne pas en oublier si vous travaillez à plusieurs) et vous le déposerez en fin de TP sur Moodle. Pour créer ce fichier, vous pouvez utiliser l'éditeur de texte « emacs ».

Un fichier (ordinaire ou répertoire) peut être désigné de façon non ambiguë en utilisant deux types de *références* (ou *chemins*) :

- une *référence absolue* décrit un chemin dans l'arborescence du système de fichiers depuis sa racine (« / ») jusqu'au fichier ;
- une *référence relative* (au répertoire courant) décrit un tel chemin depuis le répertoire de travail courant de l'utilisateur.



Ces chemins sont représentés par la suite des noms (de base) des répertoires traversés, séparés par le symbole « / », et terminés par le nom de base du fichier. On rappelle que le nom de base du répertoire parent est « .. », et celui du répertoire lui-même est « . ».

En particulier, **une référence relative ne commence jamais par « / »**, mais par un nom de base de répertoire (y compris « . » et « .. »). À l'inverse, une référence absolue commence par « / », ou éventuellement par « ~ » (qui est un synonyme défini par le shell pour le répertoire personnel de l'utilisateur courant) ou « ~machin » (qui est un synonyme défini par le shell pour le répertoire personnel de l'utilisateur *machin*).






Exercice 1 – consulter d'un coup toute une arborescence

1.  Trouver une option de « `ls` » pour lister les fichiers (ordinaires et sous-répertoires) contenus dans le répertoire `~/Cours/2019/IS1/TP2` en déterminant leur type.


Le caractère « * » peut être utilisé dans une ligne de commande pour remplacer n'importe quel morceau de nom (de base, donc sans caractère « / ») de fichier.

2.  En une seule ligne de commande, lister *tous* les fichiers et sous-répertoires situés exactement 3 niveaux en dessous de `~/Cours/2019/IS1/TP2` ; faire en sorte que les répertoires soient seulement nommés, pas explorés (comme les fichiers ordinaires).
3.  Trouver l'option de « `ls` » qui permet de lister, en une seule ligne de commande, *tout* le contenu de l'arborescence de racine `~/Cours/2019/IS1/TP2`.

Exercice 2 – chemin relatif, chemin absolu

1.  En utilisant la commande « `cat` » depuis le répertoire `~/Cours/2019/IS1`, afficher le contenu du (seul) fichier de l'arborescence TP2 dont le nom de base est haddock en utilisant trois références différentes pour le désigner.
2.  Même question depuis le répertoire `~/Cours/2019/IS1/TP2/Duck/CoffreDePicsou`.
3.  Déterminer tous les répertoires de travail dans lesquels la commande `cat ../Personnages/haddock` s'exécute sans erreur (et vérifier).
4.  Depuis le répertoire `~/Cours/2019/IS1/TP2/Toto`, quelle commande permet de lister le contenu du répertoire `~/Cours/2019/IS1/TP2/Tintin/Lieux` ?
5.  Depuis le répertoire `~/Cours/2019/IS1/TP2/Tintin`, en utilisant la commande « `cat` », afficher de deux manières différentes le contenu des fichiers de nom dupont et dupont dans le répertoire `~/Cours/2019/IS1/TP2/Tintin/Personnages`.

Exercice 3 – références et addresses web (URL)

1. Ouvrir un navigateur (firefox, chrome, ...). Dans la barre d'URL, saisir « `file://` » puis la référence absolue de votre répertoire personnel. Qu'affiche le navigateur ? Essayer ensuite de faire de même avec le répertoire personnel de votre voisin. Quel est le résultat ?
2.  Considérer l'URL « `http://www.irif.fr/~chouquet/index.html` ». Quelle est la référence contenue dans cette URL ?

Exercice 4 – fichiers cachés




Malgré les apparences, le répertoire `~/Cours/2019/IS1/TP2/Duck/CoffreDePicsou` contient plusieurs fichiers.

-  Chercher l'option de « `ls` » qui permet de les afficher. Comment s'appellent-ils ?

« du », (**disk usage**) affiche des informations concernant l'espace occupé sur le disque par une arborescence.

Exercice 5 – attention aux excès !

En utilisant les options de « du » décrites dans le manuel,

1. afficher la structure de l'arborescence de racine ~/Cours/2019/IS1/TP2 (i.e. seulement les répertoires, avec leur taille respective) ;
2.  afficher (seulement) la taille de l'arborescence entière, en kilo-octets ;
3.  lister tout le contenu (fichiers ordinaires compris) de l'arborescence ;
4.  écrire une commande permettant de déterminer, avec un affichage aussi lisible que possible, quel sous-répertoire de ~/Cours/2019/IS1/TP2 occupe le plus d'espace sur le disque.

Fonctionnalités utiles du shell Il arrive parfois que l'on ait à utiliser une même commande plusieurs fois, ou que l'on souhaite corriger une commande tapée précédemment. À cette fin, un ensemble de « raccourcis » permet d'accéder à l'historique des commandes. En utilisant les flèches haut et bas, on peut faire défiler les commandes précédentes, de la plus récente à la plus ancienne.

« history » affiche une liste des commandes récentes, chacune précédée d'un numéro. En tapant !*n* où *n* est un numéro dans l'historique, on rappelle la commande correspondante. La composition de touches *Ctrl-r* permet de rechercher une commande dans l'historique qui contient un certain mot.

Avec les flèches gauche et droite, vous pouvez également corriger un détail de chaque ligne de commande, ce qui peut être plus rapide que de retaper une longue commande presque identique. Les compositions de touches *Ctrl-a* et *Ctrl-e* permettent de repositionner le curseur au début ou à la fin de la ligne. *Ctrl-k* permet de supprimer la fin de la ligne (au-delà du curseur).

Autre point utile pour gagner du temps de saisie : la **touche de tabulation** permet de laisser le soin au shell de terminer les noms de fichiers lorsqu'il n'y a pas d'ambiguïté.

Exercice 6 – chercher dans l'historique

En utilisant l'historique, chercher la dernière fois que vous avez exécuté la commande « cat » pour afficher le contenu du fichier dont le nom de base est milou.

Utilisez-la pour afficher à nouveau le contenu du fichier à partir de votre répertoire courant. Si nécessaire, vous éditez donc la commande en ligne, pour corriger la référence relative du fichier.

Exercice 7 – modifier d'un coup toute une arborescence

1. ➤ En une seule ligne de commande, copier `~/Cours/2019/IS1/TP2/Tintin` (et toute l'arborescence en dessous) sous le nom `~/Cours/2019/IS1/TP2/Sauvegarde`.

« diff » (**differences**) compare les fichiers ou répertoires et affiche les différences ligne par ligne, respectivement fichier par fichier.

2. ➤ Vérifier que le fichier dont le nom de base est `milou` a été correctement copié.
3. ➤ En utilisant une option de « diff », vérifier que l'arborescence de `Sauvegarde` est bien une copie de `Tintin`.
4. Supprimer le répertoire `~/Cours/2019/IS1/TP2/Sauvegarde/Lieux/Reels` et vérifier que « diff » détecte la modification.
5. Supprimer toute l'arborescence `~/Cours/2019/IS1/TP2/Toto`, toujours en une seule ligne de commande. Vérifier que `~/Cours/2019/IS1/TP2/Toto` n'existe plus.
6. ➤ Créer, grâce à une option de la commande `mkdir`, toujours en une seule ligne de commande, un répertoire `~/Cours/2019/IS1/TP2/Toto/Tutu/Titi`.

Manipuler des ensembles de noms de fichier

Important : ajouter la ligne suivante dans votre fichier `~/ .bashrc` :

```
export LC_COLLATE=C
```

les jokers ou wildcards permettent de décrire plusieurs références de fichiers en une seule expression : lorsque le shell rencontre un de ces caractères, il l'interprète de toutes les manières possibles correspondant à un nom (de base) de fichier existant.

- « * » représente n'importe quelle suite de zéro, un ou plusieurs symboles pouvant apparaître dans un nom de base (sauf le « . » en début de mot) ;
- « ? » représente exactement un symbole quelconque (toujours sauf le « . » en début de mot) ;
- « [] » représente exactement un symbole parmi l'ensemble de symboles décrit entre les crochets :
 - soit par liste exhaustive de caractères, par exemple `[abz]` ;
 - soit par intervalle (au sens du code binaire) en séparant les bornes de l'intervalle par le symbole « - », par exemple `[a-f]` ou `[2-7]` ;
 - soit par union de listes et d'intervalles, par exemple `[a-fA-FzZ]` ;
 - soit par complément avec « ^ » ou « ! », par exemple `[^wx]` ou `[!a-z]`.

Exercice 8 – filtrer l’affichage

✎ Comment lister (exactement) les noms¹ de fichier dans le répertoire `/usr/bin ...`

1. qui commencent par un `r` ;
2. qui contiennent un `r` ;
3. qui terminent par un `r` ;
4. qui commencent par un `p` et terminent par un `n` ;
5. dont la troisième lettre est un `r` ;
6. qui commencent par un `r` et contiennent exactement 5 caractères ;
7. qui commencent par un `r` et contiennent au moins 5 caractères ;
8. qui commencent par un `r` et contiennent entre 3 et 5 caractères ;
9. qui contiennent un `s` ou un `l` ;
10. qui contiennent un chiffre ;
11. qui contiennent un caractère non alphabétique ;
12. qui contiennent un `r` et un `n` ;
13. qui contiennent un `r` et commencent par une lettre parmi `l, m, n, o, p, q` ;
14. qui contiennent un `r` et ne commencent pas par un `n` ;
15. qui contiennent un `r` et ne commencent pas par une lettre parmi `l, m, n, o, p`.

1. l’option `«-d»` de `«ls»` pourra être utile...