



# Travaux Pratiques UNIX n°1

## Informatique

— — —

### 1 Les commandes de base

Voici quelques commandes de base sous UNIX:

- **id**: numéros d'identification d'utilisateur et de groupe.
- **tty**: afficher le numéro du terminal qui lance la commande.
- **echo** [<chaine1>] ... [<chainen>]: affiche les *n* arguments.
- **date**: affiche la date.
- **who**: affiche les utilisateurs logés et les numéros de terminaux associés.
- **finger** <utilisateur>: informations sur un utilisateur.
- **ls** [-lisa] [<référence1>] ... [<référencen>]: affiche des informations sur les références spécifiées.
- **passwd**: permet de changer son mot de passe
- **ps**: informations sur les processus
- **top**: informations sur les processus et la mémoire
- **cat** [<fichier1>] ... [<fichier>]: affiche les *n* fichiers dans l'ordre.
- **cp** [<fichier1>] ... [<fichier>] <dest>: copie les *n* fichiers sur la destination.
- **mv** [<fichier1>] ... [<fichier>] <dest>: déplace les *n* références sur la destination.
- **chmod** [droits] [<reference1>] ... [<referencen>]: fixe les droits des références.
- **rm** <fichier1> ... [<fichier>]: efface les fichiers
- **ln** [-s] <fichier1> <fichier2>: crée un lien fichier2 sur fichier1. L' lien est symbolique avec l'option -s.
- **cd** [<repertoire>]: changer le répertoire courant à |repertoire|

- `pwd`: afficher la référence absolue du répertoire courant
- `mkdir <repertoire>`: créer le répertoire
- `rmdir <repertoire>`: supprimer le répertoire
- `man [1-8] <commande>`: manuel d'utilisation de la commande
- `emacs [<fichier1>] ... [<fichiern>]`: éditer les fichiers

► **Exercice 1.**

- *Lancer un nouveau terminal.*
- *Changer votre mot de passe.*
- *Donner le numéro du terminal que vous êtes en train d'utiliser.*
- *Donner votre numéro d'utilisateur. A quels groupes appartenez-vous ?*
- *Aller dans votre racine et en afficher le contenu.*
- *Créer un repertoire `rep1`. Fixer ses droits de manière à ce que personne d'autre que vous ne puisse ni y lire ni y écrire.*
- *Se placer dans `rep1`. Y éditer un nouveau fichier de nom `fic1.txt`. Y mettre son nom. Quitter l'éditeur. Afficher le contenu du fichier.*
- *Changer les droits du fichier de façon à ce que personne ne puisse plus y écrire.*
- *Sans changer de répertoire courant, créer un répertoire de nom `rep2` à la racine de votre répertoire en utilisant des références relatives.*
- *Vérifiez que vous pouvez écrire dans `rep2`. Si ce n'est pas le cas changez les droits de façon à ce qu'on puisse y écrire.*
- *Y copier le fichier `fic1.txt`. Renommer le fichier `fic1.txt` du repertoire courant en `fic2.txt`. Vous n'utiliserez que des références absolues.*
- *Afficher le contenu de `fic1.txt`. Vérifiez les droits de `fic1.txt` et `fic2.txt` en utilisant des références relatives.*
- *Faire un `ls -lisa ~`. Que se passe-t-il ?*
- *Copier `fic1.txt` dans `fic2.txt` du répertoire `rep2`. Que se passe t-il ? Expliquer. Y remédier et recommencer.*
- *Essayer d'effacer `rep1`, `rep2`, `fic1.txt` et `fic2.txt` en une seule commande, à partir du repertoire courant. Que se passe t-il ?*
- *Aller dans le repertoire de nom `tmp` situé à la racine. Effacer `rep1`, `rep2`, `fic1.txt` et `fic2.txt`.*

- Copier `/etc/passwd` dans votre repertoire racine. Créer le lien de nom `passwd2` sur ce fichier. Regarder le contenu de `passwd2`. Le modifier. Regarder le contenu de `passwd`. Que constatez-vous ?
- Comment savoir si un nom de fichier est un lien sur un autre fichier ?
- Effacer `passwd`. Pouvez-vous regarder le contenu de `passwd2` ?
- Créer un lien symbolique de nom `passwd` sur `passwd2`. Regarder le contenu de `passwd2`. Le modifier. Regarder le contenu de `passwd`. Que constatez-vous ?
- Comment savoir si un nom de fichier est un lien symbolique sur un autre fichier ?
- Effacer `passwd2`. Pouvez-vous regarder le contenu de `passwd` ?

## 2 Les masques

Un masque est une suite de caractères qui prend la place d'une autre suite dans une référence.

- l'étoile (\*) prend la place de n'importe quelle suite de caractères (éventuellement de longueur nulle). Par exemple,  $d*f$  désigne les références qui commencent par  $d$  et terminent par  $f$ .
- le point d'interrogation (?) prend la place de n'importe quel caractère. Par exemple,  $d?f$  désigne les références de trois lettres dont la première est un  $d$ , la seconde quelconque et la troisième un  $f$ ,
- les énumérations sont une restriction du ? dans lesquels on spécifie explicitement les caractères autorisés. Par exemple,  $d[atv]f$  désigne les références de trois lettres dont la première est un  $d$ , la seconde est  $a$ ,  $t$  ou  $v$  et la troisième un  $f$ . Pour éviter les énumérations longue il existe une syntaxe qui permet de spécifier des intervalles. Par exemple,  $d[a-m]f$  désigne les références de trois lettres dont la première est un  $d$ , la seconde est comprise entre  $a$  et  $m$  et la troisième un  $f$ .

### ► Exercice 2.

- Se placer dans le repertoire `bin` de la racine du système de fichiers. Faire un `ls` suivi d'un `echo *`. Que constatez vous ?
- Donnez les tailles des fichiers dont le nom commence par  $a$ .
- Donnez les dates de dernière modification des fichiers dont le nom contient  $sh$ .
- Donnez les noms des propriétaires des fichiers de trois lettres qui terminent par  $sh$ .
- Se placer dans le repertoire `sbin` de la racine du système de fichiers.

- *Ecrire une commande qui donne les noms des fichiers qui commencent par une voyelle.*
- *Afficher les noms de fichier qui commencent par une lettre allant de a à m. Afficher les noms des fichiers qui contiennent un chiffre. Afficher les noms des fichiers dont la 4<sup>ème</sup> lettre est a, b, c ou d.*

### 3 Les variables d'environnement

L'interpréteur de commande permet de définir des variables d'environnement, qui sont toutes du types chaîne de caractères. Certaines de ces variables sont exportées (visibles) par les processus lancés à partir du processus dans lequel elles ont été définies. Les principales variables exportées sont:

- HOME: votre répertoire
- USER: votre nom d'utilisateur
- DISPLAY: la référence IP de votre serveur graphique
- TERM: le type de terminal utilisé
- HOSTNAME: le nom de votre machine
- SHELL: programme interpréteur de commandes
- PATH: la liste des répertoires où l'interpreteur va chercher les commandes à exécuter.

La valeur d'une variable s'obtient en préfixant son nom par un \$. Par exemple, `echo $toto` affiche la valeur de la variable `toto`. Exporter `toto` se fait par `export toto`. Fixer la valeur de `toto` se fait par `toto=truc`. La syntaxe `export toto=truc` est un raccourci pour `toto=truc; export toto`. La commande `export` sans argument permet d'afficher toutes les variables exportées.

#### ► Exercice 3.

- *Afficher la valeur de HOME.*
- *Se placer dans tmp. Créer un sous répertoire bin dans votre repertoire, en utilisant la variable HOME.*
- *Se placer dans votre repertoire bin.*
- *Créer une variable d'environnement TEMPORAIRE et lui donner la valeur de la variable PATH. Donner une valeur nulle à la variable PATH.*

- *Editer le fichier toto, y mettre la commande `echo $USER; finger $USER`. Mettre les droits de façon à ce qu'on puisse exécuter toto. Exécuter toto. Que se passe-t-il ?*
- *Rétablir l'ancienne valeur de PATH contenue dans TEMPORAIRE. Exécuter toto. Si toto ne s'exécute toujours pas, c'est que le répertoire courant n'appartient pas à PATH. Le rajouter et relancer toto.*
- *Comment faire pour pouvoir lancer votre commande toto à partir de n'importe quel répertoire ?*

Les guillemets (“”) permettent d'interdire l'interprétation des masques dans les chaînes de caractères. Les cotes (‘ ’) permettent d'interdire l'interprétation des masques et la substitution des variables par leurs valeurs dans les chaînes de caractères. Les anticotes (‘ ‘) permettent de créer une chaîne de caractères à partir d'une commande. La valeur de la chaîne créée est le résultat de l'exécution de la commande.

► **Exercice 4.**

- *Créer un fichier de nom `d*g`.*
- *Afficher tous les noms de fichier du répertoire courant qui commence par `a` et ceux qui contiennent une étoile.*
- *Afficher tous les noms de fichier du répertoire courant qui commence par `a` et qui contiennent une étoile. Indication: le `\` permet de ne pas interpréter un caractère spécial, comme l'étoile.*
- *Créer un répertoire de nom “Mon répertoire”. L'effacer.*
- *Exécutez la commande `echo a' '* ' b`. Expliquez.*
- *Comment faire pour créer un répertoire de nom `$a` ?*
- *Donnez deux façons de créer un répertoire de nom `\`.*

## 4 Les redirections

### 4.1 Les fichiers

Les commandes lisent les données sur l'entrée standard et affiche les résultats sur la sortie standard qui sont par défaut respectivement le clavier et le terminal. Les erreurs sont envoyées sur la sortie erreur standard, par défaut le terminal. Ces entrées/sorties peuvent être redéfinies (redirigées) dans des fichiers par exemple. Pour différencier les deux sorties standard et erreur on leur donne respectivement les numéros 1 et 2. Les opérateurs de sortie avec écrasement sont désignés par `1>` (ou plus simplement `>`) et `2>`. L'opérateur d'entrée est désigné par `<`. Les opérateurs `1>>` (ou `>>`) et `2>>` permettent

les redirections sans écrasement des fichiers de sortie (les données sont ajoutées à la fin du fichier).

- La commande `cat` écrit sur la sortie standard ce qu'elle lit sur l'entrée standard. Taper `cat </etc/passwd >$HOME/mon_passwd`. Que contient le fichier nouvellement créé `mon_passwd` ?
- Comparer les droits de `/etc/passwd` et `mon_passwd`. Copier `/etc/passwd` avec la commande `cp` et comparer. Expliquer.
- Ajouter en fin du fichier `mon_passwd` le fichier `/etc/services`.
- Exécuter `cat <mon_passwd >mon_passwd`. Expliquer.
- Utiliser la commande `cat` pour fusionner tous les fichiers du répertoire `/etc` dans `$HOME/mon_etc` de façon à pouvoir retrouver les erreurs dans le fichier `$HOME/mon_etc_erreur`. Donner la liste des erreurs et expliquer.

## 4.2 Les tubes

Les tubes permettent de rediriger la sortie standard d'une commande vers l'entrée standard d'une autre.

Exemple: la commande `more` permet l'affichage d'un fichier sur la sortie standard avec arrêt du défilement en bas de page. `ls -lR / | more` liste toute l'arborescence du système de fichiers page par page.

Quelques commandes de traitement du texte:

- `wc fichier`: compte les lignes, les mots et les caractères d'un fichier (cf `man wc`).
- `grep chaine fichier`: recherche les lignes contenant une chaîne de caractères donnée dans un texte. (cf `man grep`).
- `cut [-f list] [-d c] fichier`: extrait un champ de chaque ligne d'un fichier, les champs étant séparés par un délimiteur spécifié (cf `man cut`).
- A l'aide de `ps` et `wc`, écrire une commande qui compte le nombre de processus qui tournent sur la machine (cf `man`).
- Exécuter la commande `cat <mon_etc_erreur | cat >mon_etc_erreur`. Qu'est-il arrivé au fichier `mon_etc_erreur` ?
- En utilisant l'option récursive de la commande `ls` (option `-R`) et `grep`, afficher tous les fichiers du système appartenant à `root`.
- La commande `sort` permet de trier son entrée standard sur un champ spécifié. Écrire une commande qui affiche les processus d'un utilisateur donné par ordre croissant de numéro de processus. Écrire une commande qui, donné un nom d'utilisateur et un nom de processus, affiche les numéros des processus de ce nom pour cet utilisateur.

- Ecrire une commande qui tue tous les processus d'un nom donné pour un utilisateur donné.