

1 Rappels

- Votre shell par défaut est le `bash`.
- Vous pouvez changer votre mot de passe en utilisant la commande `yppasswd`.
- Durant les TP encadrés, il vous sera demandé d'utiliser l'interface `WindowMaker`.
- La taille de votre répertoire personnel est limitée, vous avez un quota de 50Mo. Cette taille est assez importante cependant, il sera bon de vous inquiéter de temps en temps de la place qui est encore disponible sur votre compte par la commande `quota`. Si vous dépassez la limite, vous ne pourrez plus sauvegarder vos fichiers.

2 Premières manipulations

2.1 Les fichiers

- ① La commande `man <nom-de-commande>` permet d'obtenir une aide en ligne de la commande spécifiée en paramètre. Testez les commandes suivantes à partir d'un terminal :

- ◆ `man cat`
- ◆ `man more`
- ◆ `man less`



QUAND UNE COMMANDE
TU NE CONNAITRAS PAS,
`man` TU UTILISERAS.

- ② Copiez le fichier `/etc/services` dans votre répertoire privé. Visualisez ce fichier à l'aide de la commande `cat`, de la commande `more` puis de la commande `less`.
- ③ Créez deux répertoires `rep1` et `rep2` sous votre répertoire privé et un répertoire `rep3` dans `rep1`. Copiez le fichier `services` dans le répertoire `rep3` sous le nom `essai.txt`.
- ④ Placez-vous dans le répertoire `rep3`. Renommez le fichier `essai.txt` en `essai_bis.txt`. Éditez ce fichier à l'aide d'un éditeur, par exemple `emacs` ou `vi`.
- ⑤ Effacez tous les fichiers et répertoires que vous avez créés pour cet exercice.

2.2 Redirections – communication

- ① Listez le contenu du répertoire `/home` dans un fichier `liste1` et le contenu du répertoire `/usr` dans un fichier `liste2`. Concaténez ces deux fichiers dans un troisième fichier.
- ② Lisez le descriptif de la commande `grep`.
- ③ Listez le contenu du répertoire `/etc`, vous constaterez que le défilement rapide ne permet pas de lire toutes les informations. Listez-le à nouveau de manière à pouvoir le lire pages par pages, éventuellement revenir en arrière, ...
- ④ De la même manière, faites afficher à l'écran les lignes du fichier `/etc/services` qui concernent les services `tcp`.
- ⑤ Détruisez les fichiers `liste1` et `liste2`.

2.3 Le mécanisme d'historique

Le shell gère une liste des dernières commandes exécutées et permet de redemander l'exécution de ces commandes. La commande `history` permet d'obtenir l'historique des dernières commandes avec un numéro d'ordre associé à chaque commande. Les commandes ainsi numérotées peuvent être réexécutées simplement :

- ◆ `!!` permet de réexécuter la dernière commande ;

- ♦ `!n` permet de réexécuter la commande numéro `n` ;
- ♦ `!ch` permet de réexécuter la commande la plus récente commençant par `ch`.

Testez la commande `history` et la réexécution de commandes selon les trois méthodes précisées ci-dessus.

3 Utilisation des terminaux.

3.1 Job Control

Le but de cet exercice est de vous apprendre à manipuler les différentes tâches d'un terminal.

- ① Ouvrez deux terminaux que nous appellerons terminal 1 et terminal 2.
- ② Depuis le terminal 1, lancez en avant plan puis suspendre `xeyes`, lancez en avant plan puis suspendre `xclock`.
- ③ Lancez en avant plan puis suspendre et passer en arrière plan `xclock` depuis le terminal 2.
- ④ Affichez la liste des jobs dans chaque terminal, puis la liste des processus par la commande `ps`.
- ⑤ Utilisez les options de la commande `ps` pour visualiser tous les processus d'un terminal puis tous les procesus de la machine indépendemment de leur propriétaire.
- ⑥ Constatez que le `xeyes` lancé depuis le terminal 1 n'est pas actif puis continuez son exécution en arrière plan.
- ⑦ Tuez le processus `xclock` du terminal 1 par la ligne de commande permettant de tuer un job.
- ⑧ Tuez le processus `xclock` du terminal 2, depuis le terminal 1 par la ligne de commande permettant de tuer un processus.

3.2 Terminaux et entrées/sorties

- ① Créez un répertoire `tp1` à votre racine, placez-vous dans ce répertoire.
- ② Essayez les programmes `dixA` et `dixB` situés dans un répertoire `shell` sur mon compte. Recopiez ensuite ses programmes dans votre répertoire `tp1`.
- ③ Exécutez les programmes `dixA` et `dixB` dans deux terminaux différents. Lancez-les ensuite en tâches de fond dans le même terminal. Que se passe-t-il ? Quel aspect du système est ainsi mis en avant ?
- ④ Utilisez la commande `ps` pour visualiser les processus d'un terminal puis tous les processus de votre machine. Lancez à nouveau les programmes `dixA` et `dixB`, tuez le processus qui affiche les `'b'` au bout de 5 affichages.
- ⑤ La commande `tty` permet d'obtenir le nom du fichier associé à un terminal. Lancez le programme `dixA` dans un terminal (appelons-le `term1`) et effectuez l'affichage des `a` dans un deuxième terminal (appelons-le `term2`). Au même moment, lancez le programme `dixB` dans le terminal `term2` avec affichage des `b` dans le terminal `term1`.



EN UNIX, TOUT EST FICHIER.

4 Quelques commandes

4.1 Aspects multi-utilisateurs

La commande `su` permet de changer l'utilisateur courant, c'est-à-dire de se reconnecter sur la machine sous un nom d'utilisateur différent, de travailler désormais avec les droits de ce nouvel utilisateur.

- ① Essayez cette commande avec votre binôme ou voisin.
- ② Lancez un nouveau terminal, votre binôme peut se connecter depuis ce terminal et effectuer quelques manipulations.
- ③ Vous avez un exemple de deux utilisateurs connectés sur la même machine, visualisez les processus des deux utilisateurs.

4.2 Travail à distance

Un même utilisateur peut également travailler sur plusieurs machines en même temps.

- ① Utilisez la commande `ssh <nom_de_machine>` pour vous connecter sur une machine distante sous votre nom ou `ssh <login>@<nom_de_machine>` pour vous connecter sur une machine distante sous un autre nom.

Le shell interactif de votre terminal est à présent un shell qui s'exécute sur la machine distante. Vous pouvez exécuter un programme sur la machine distante, le « calcul » même s'exécute à distance (*i.e.* vous utilisez le processeur distant) en revanche, l'affichage des résultats se fera dans votre terminal, essayez par exemple `cal`.

- ② Essayez ensuite de lancer `xeyes`. Que se passe-t-il ?
- ③ Vous pouvez utiliser la machine distante pour des commandes « texte ». En revanche, vous ne pouvez ouvrir de nouvelles fenêtres pour deux raisons :
 - (a) vous n'avez pas spécifié à l'application distante où afficher la fenêtre ;
 - (b) vous n'avez pas autorisé les fenêtres distantes à s'afficher sur votre machine.

Il vous faut donc :

- (a) positionner la variable `DISPLAY` à la valeur `<machine-locale> :0` dans votre shell distant par exemple par la commande

```
export DISPLAY=if12:0
```

si vous travaillez sur `if12` ;

- (b) utiliser la commande `xhost +<machine-distante>` sur votre machine locale pour autoriser la machine distante à effectuer des connections à votre serveur X local.

Effectuez la manipulation et réessayez de lancer `xeyes`.



`ssh` vous permet d'établir une connection sécurisée – *i.e.* cryptée – contrairement à d'autres commandes comme `telnet` pour lesquelles toutes les données (y compris le mot de passe) transitent en clair sur le réseau.

4.3 Copies distantes

Il est parfois utile de transférer des fichiers d'une machine à l'autre, d'un compte à l'autre, ... La commande `scp` de la famille de `ssh` vous permet d'effectuer des copies sécurisées. Cette commande accepte de nombreuses options (voir `man`), nous utiliserons pour le moment la forme :

```
scp [-r] [[user@]host1:]file1 [[user@]host2:]file2
```

Le fonctionnement de cette commande est semblable à celui de `cp` excepté qu'il est possible de préciser, pour chaque nom de fichier, l'hôte sur lequel se trouve ce fichier et éventuellement le nom d'utilisateur sous lequel on veut accéder à l'hôte distant. Il est possible qu'un mot de passe soit demandé. L'option `-r` permet d'effectuer des copies récursives.

- ① Copiez dans votre répertoire `tp1` le fichier `tp1.shell` qui se trouve dans le répertoire `tmp` de la machine `1xd1`. Pourquoi ne peut-on effectuer cette copie avec la commande `cp` ?
- ② Créez un fichier `<votre_nom>.txt` et copiez-le dans ce même répertoire.