



**R 1.04**

**2023 - 2024**

# **Introduction aux systèmes d'exploitation et à leur fonctionnement**

**TD N°1  
« UNIX premier contact »**



**ANNE Jean-François**  
**D'après le TD de F. BOURDON**

Le but de ce TD est de se familiariser avec les systèmes d'exploitation et avec leur fonctionnement.

# « UNIX premier contact »

## Notions vues dans ce TD :

Session, interprète de commandes, commandes, répertoire de travail, éditeur de texte.

Nombre de séance de 2h prévu pour faire ce TD : 1.

PS : Les parties correspondant à du travail à faire sont toutes en italiques ; le restant étant du complément au cours.

## 1. Connexion au système et ouverture de sessions de travail

Avant de pouvoir travailler, il est nécessaire de se connecter sur le système. Chaque utilisateur possède un « compte de connexion » auquel est associé un « mot de passe ».

 En utilisant les touches *CTRL+ALT+F1*, *CTRL+ALT+F2*, *CTRL+ALT+F3* ... *CTRL+ALT+F8* l'utilisateur dispose de quatre écrans virtuels à partir desquels il pourra ouvrir quatre sessions. En vous mettant à deux étudiants par machine, ouvrez chacun une session en parallèle. Pour revenir à la session multi-fenêtres tapez *CTRL+ALT+F7*.

Un utilisateur connu du système, c'est donc :

- Un compte = nom d'utilisateur + mot de passe,
- Un espace de stockage en mémoire secondaire (disque) ⇨ « home directory »,
- Un environnement d'interprétation ⇨ « Shell » (bash sous Linux),
- + d'autres données, que l'on verra au fur et à mesure du cours.

Les informations sont stockées sous forme de « fichiers ». Le fichier « */etc/passwd* » contient la liste des utilisateurs du système avec les informations les concernant.

On manipule le système et les informations contenues dans des fichiers, au moyen de commandes (*passwd*, *cat*, *cal*, *date* ...). La commande « **cat** » permet d'afficher le contenu d'un fichier texte.

 Taper la ligne de commande suivante (« *prompt>* » correspond à l'invite du Shell, c'est-à-dire la chaîne affichée par le Shell pour inviter l'utilisateur à entrer une commande) :

***prompt> cat /etc/passwd (Enter)***

Dans ce fichier, chaque ligne correspond aux informations liées à un utilisateur de la machine. Ce sont les utilisateurs système qui sont listés. On peut définir des utilisateurs non pas pour une seule machine, mais pour un ensemble (pool). Pour cela, un fichier « */etc/passwd* » sera placé sur une machine de référence appelée « serveur » ; on parle alors pour ce fichier d'une ressource partagée.

Si l'affichage à l'écran d'une commande comporte plus de lignes que ne peut en afficher votre terminal, vous pouvez faire suivre l'appel à cette commande de « *| more* ».

 Testez-le sur la commande précédente, puis sur la commande suivante :

**prompt> ls /bin (Enter)**

---

## **2. Interprète de commandes**

Les commandes que vous tapez sont analysées et exécutées par un interprète de commandes appelé « Shell ». Il en existe plusieurs types différents qui sont équivalents. On en choisira un par défaut pour la session de travail, en positionnant le dernier champ de la ligne correspondante à cet utilisateur dans le fichier « /etc/passwd ».

Une commande correspond en fait à l'exécution d'un programme dans l'interprète (Shell). Elle peut :

- Avoir besoin d'informations pour s'exécuter,
- Produire ou transformer de l'information dans un fichier,
- Produire un message d'erreur en cas de mauvaise utilisation de cette dernière.

 La commande « **passwd** » permet de changer de mot de passe. Pour cela, après vous être connecté, lancer cette commande en tapant :

**prompt> passwd (Enter)**

Le comportement d'un Shell est le suivant :

1. Afficher le prompt (invite)
2. Attendre Enter (touche entrée)
3. Interpréter la ligne de commande
4. Aller en 1.

 Pour sortir de cette boucle infinie, on tape la commande « **exit** ». Essayer cette commande dans votre session ouverte avec CTRL+ALT+F1. Que se passe-t-il ?

---

## **3. Commandes**

Parmi les informations dont peut avoir besoin une commande, on trouve les options et les paramètres. C'est ainsi qu'une commande peut être lancée seule, avec une ou plusieurs options, un ou plusieurs paramètres ou les deux (options et paramètres).

Unix est un système multi-utilisateur, c'est-à-dire qu'il permet d'ouvrir plusieurs sessions en même temps. La commande « **who** » permet de connaître les sessions en cours.

 Lancer les différentes lignes de commande suivantes et commenter les différents résultats obtenus :

**prompt> who (Enter)**

**prompt> who -H (Enter)**

**prompt> whoami (Enter)**

---

prompt> who -y (Enter)

prompt> who --help (Enter)

Une commande est un programme qui est stocké sous la forme d'un fichier. L'une d'entre elles (« which ») permet de retrouver le fichier correspondant.

 Taper la ligne de commande suivante :

prompt> which who (Enter)

 Dans cette exemple « who » est-elle considérée comme une option de la commande « which » ? Pourquoi ?

 Pour avoir des informations sur l'utilisation d'une commande, on pourra utiliser le manuel électronique (« man ») en tapant la ligne de commande suivante :

prompt> man commande\_choisie (Enter)

Où « commande\_choisie » correspondant à la commande pour laquelle on veut de l'information.

On trouve en général les informations suivantes :

**NAME** : le nom de la commande dont la fonction est brièvement décrite ;

**SYNOPSIS** : toutes les possibilités de saisies liées à cette commande (syntaxe) ;

**DESCRIPTION** : une explication courte des conséquences de la commande ;

**FILES** : les fichiers modifiés par la commande ou nécessaires au moment de la saisie ;

**OPTIONS** : la liste des différentes options de cette commande ;

**SEE ALSO** : les références croisées vers d'autres commandes proches ;

**DIAGNOSTICS** : des explications sur les messages d'erreur ;

**RETURN VALUES** : ce que renvoie la commande ;

**BUGS** : des problèmes connus de cette commande ;

**EXEMPLES** : des exemples d'appel à cette commande ;

**TIPS** : des astuces pour utiliser cette commande.

Pour sortir du manuel, il suffit de taper la lettre « q » (comme « quit »). La commande « man » étant une commande presque comme les autres, elle possède sa propre entrée dans le manuel.

 Tapez :

prompt> man man (Enter)

---

#### 4. La commande cp

1. Est-il possible de faire une copie d'un répertoire et de l'ensemble de ses sous-répertoires en une seule ligne de commande, sans d'abord faire une copie de chacun des fichiers l'un après l'autre ?
2. Que se passe-t-il si l'on veut faire une copie appelée tentative/ de son répertoire essai/, et que le répertoire tentative/ existe déjà ?
3. Que se passe-t-il si vous voulez faire une copie d'un fichier dans un répertoire où il existe déjà un fichier de ce nom ?
4. Est-il possible de taper une ligne du genre de :

- cp fichier1 fichier2 fichier3 repertoire/
- Quel est l'effet de cette ligne ?

### **5. La commande mv**

1. La syntaxe de mv est mv fichier1 fichier2; fichier1 est renommé en fichier2. Mais que se passe-t-il si fichier2 existe déjà ?
2. Supposez que vous avez chez vous l'arborescence que voici, et que vous vous trouvez dans ~/Rapport/docs/.
  - Comment faites-vous pour y déplacer Rapport.tex qui se trouve dans le répertoire père ?
  - Comment faites-vous ensuite pour déplacer ce fichier à son emplacement d'origine ?
3. Comment renommer un répertoire bla en blo ? Que se passe-t-il si le répertoire blo existe déjà ?
4. Comment faire pour à la fois déplacer un fichier et changer son nom ? Que se passe-t-il si il existe déjà un fichier de ce nom dans le répertoire d'arrivée ?
5. Mêmes questions avec un répertoire.
6. Que se passe-t-il si on lance mv avec les options -i et -f ensemble ?

### **6. Quelques commandes utiles**

 *A l'aide du manuel (« man ») et en observant le résultat des lignes de commande suivantes, expliquer ce que font les commandes « cat », « date », « wc », « cal », « which », « touch » et « grep ».*

```
prompt> cat /etc/passwd (Enter)
prompt> date (Enter)
prompt> wc /etc/passwd (Enter)
prompt> wc -l /etc/passwd (Enter)
prompt> cal (Enter)
prompt> cal -j (Enter)
prompt> which cal (Enter)
prompt> touch essai (Enter)
prompt> grep spool /etc/passwd (Enter)
```

Il se peut qu'une commande ne rende pas la main (le prompt « *prompt>* »). Dans ce cas il faut provoquer l'arrêt du programme correspondant en tapant au clavier « **CTRL C** ».

 *Taper la commande suivante :*

```
prompt> dd (Enter)
```

 *Que se passe-t-il ? Après avoir repris la main, utiliser le manuel (« **dd** ») pour comprendre ce qui s'est passé.*

### **7. Répertoire de travail**

Votre compte de connexion est un espace de travail personnel. Certaines commandes, dont « pwd », « ls », « touch », « mkdir », « cd » et « rm », permettent d'explorer cet espace de travail et de le modifier



A l'aide du manuel (« man ») et **en exécutant et en analysant** le résultat de chacune des lignes de commande suivantes, préciser le rôle de chacune de ces commandes :

```
prompt> pwd (Enter)
prompt> ls (Enter)
prompt> ls -l (Enter)
prompt> ls -a (Enter)
prompt> ls -al (Enter)
prompt> rm essai (Enter)
prompt> ls -l (Enter)
prompt> mkdir SE_essais (Enter)
prompt> ls -l (Enter)
prompt> cd SE_essais (Enter)
prompt> pwd (Enter)
prompt> ls (Enter)
prompt> touch toto (Enter)
prompt> ls -l (Enter)
prompt> cd (Enter)
prompt> pwd (Enter)
prompt> ls -l (Enter)
prompt> cd /etc (Enter)
prompt> pwd (Enter)
prompt> ls -l (Enter)
prompt> which cd (Enter)
prompt> which date (Enter)
prompt> cd /bin (Enter)
prompt> pwd (Enter)
prompt> ls -l (Enter)
prompt> cd (Enter)
prompt> pwd (Enter)
prompt> ls -l (Enter)
```

---

## 8. Les éditeurs de texte

Le rôle d'un éditeur de texte est de mettre des informations sous forme textuelle dans un fichier afin de les conserver dans le temps. La commande « **cat** » permet une première approche simplifiée de ce service.



Entrez et commentez la suite de lignes de commande suivantes :

```
prompt> cat > fich1 (Enter)
entrer le texte (Enter)
que vous voulez conserver (Enter)
dans ce fichier (Enter)
...
CTRL D
prompt>
prompt> ls -l
prompt> cat fich1
```

La commande « cat », comme beaucoup de commandes, peut avoir besoin d'informations « en entrée » pour travailler ; elle peut produire « en sortie » des informations.

Dans cette exemple « > fich1 » indique à la commande « cat » que les informations qu'elle produira ne seront pas affichées à l'écran, mais redirigées dans le fichier « fich1 ».

 Dans le deuxième appel à « cat » quel le statut du mot « fich1 » ? Expliquer.

Des applications plus sophistiquées que « cat » permettent de manipuler des fichiers de texte : ce sont des éditeurs de texte (vi, nano, emacs, xemacs, ...).

## 9. Les fichiers texte

### a) Exercice 1 (nano, less, ls)

1. Créez un fichier `premiertexte` contenant une ou deux phrases.
2. Visualisez le contenu de `premiertexte` sans l'éditer.
3. Quelle est la taille de `premiertexte` ?
4. Éditez `PREMIERTEXTE`. Que constatez-vous ?

### b) Exercice 2 (cp, ls, mv)

1. Faites une copie de `premiertexte` appelée `double`.
2. Comparez leurs tailles.
3. Renommez `double` en `introduction`.
4. Quelle différence y a-t-il entre
5. `mv double introduction`  
et  
`cp double introduction`

### c) Exercice 3 (mkdir, mv, cp, ls, cd)

1. Créez un répertoire `essai/`.
2. Déplacez `introduction` dans `essai/`.

3. Faites une copie de `premiertexte` appelée `copie`, et placez-la également dans `essai/`.
4. Affichez une liste de ce que contient `essai/`.

**d) Exercice 4 (rmdir, cd, rm)**

1. Essayez de détruire `essai/`. Que se passe-t-il ? Que faut-il faire pour détruire un répertoire ?
2. Détruisez tout ce que contient `essai/`.
3. Détruisez `essai/`.

## **II. Webographie :**

- <https://perkamon.traduc.org/>
- <https://www.geeksforgeeks.org/linux-commands/>
- <https://www.malekal.com/liste-des-commandes-linux/#Commandes sur le systeme de fichiers>
- <https://www.cours-gratuit.com/unix-linux/exercices-sur-les-commandes-unix-de-base-pour-debutant>
- <https://www.cours-gratuit.com/unix-linux/exercices-unix-sur-cp-et-mv>
-