

Étape 1 : Se connecter au système

Afin de pouvoir commencer une *session* de travail, il est nécessaire de se connecter sur le système. Pour cela le système doit pouvoir vous identifier. Chaque utilisateur possède à cet effet un *compte de connexion*. Vous utilisez ces informations depuis le début de l'année.

Un utilisateur connu du système, c'est donc au minimum :

- un **compte** nom d'utilisateur + mot de passe
- un **espace de stockage** en mémoire secondaire (disque) *home directory*
- un **langage d'interprétation des commandes** *shell (bash sous GNU/Linux)*

Pour pouvoir communiquer avec le système un utilisateur a besoin d'interagir avec un *terminal*. Il s'agit juste d'un dispositif permettant d'envoyer (**entrée** vers le système) et de recevoir (**sortie** du système) des données (un couple clavier/écran).

Unix est un système multi-utilisateurs. Historiquement il a donc naturellement toujours permis de connecter simultanément plusieurs terminaux à un ordinateur. Aujourd'hui sur les micro-ordinateurs courants il n'y a que rarement plusieurs terminaux physiques différents. Linux propose donc, en remplacement, des terminaux *virtuels* qui se comportent de la même manière que les terminaux physiques.

En utilisant simultanément les touches `CTRL` + `ALT` + `F1`, `CTRL` + `ALT` + `F2`, etc. l'utilisateur a, par exemple, accès à différents terminaux virtuels à partir desquels il pourra ouvrir des sessions. Certains sont en mode texte (généralement les six premiers), d'autres en mode graphique (généralement le septième).

Sous Unix les informations sont stockées sous forme de *fichiers*. Le fichier `/etc/passwd` contient, par exemple, 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`, `ls`, `cd`, etc.). La commande `cat` permet, par exemple, d'afficher le contenu d'un fichier texte sur le périphérique de sortie de votre terminal.

Le fichier `/etc/passwd` est un fichier *texte*. Dans ce fichier, chaque ligne correspond aux informations liées à un utilisateur de la machine. On peut définir des utilisateurs non pas pour une seule machine, mais pour un ensemble de machines. 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.

Pour les deux premières étapes de ce TP vous allez devoir interagir en mode texte avec le système. Pour basculer de votre session graphique à votre session texte vous devrez utiliser les touches `CTRL` + `ALT` + `F1` et pour retourner à votre session graphique les touches `CTRL` + `ALT` + `F7`.

Q 1. Connectez-vous sur un terminal virtuel en **mode texte**.

Q 2. Saisissez la ligne de commande suivante : `cat /etc/passwd`

Si l'affichage à l'écran d'une commande comporte plus de lignes que ne peut en afficher votre terminal, vous pouvez filtrer l'appel à cette commande par un paginateur (*pager* en anglais). Sur vos stations il y en a au moins 2 différents : `more` et `less`. Pour filtrer une commande il suffit de la terminer par le caractère `|` puis d'ajouter le filtre juste après ce caractère. Par exemple via la commande : `getent passwd | more`.

Q 3. Testez ces 2 paginateurs sur la commande précédente.


Étape 2 : Interagir avec le système

Interpréteur de commandes

Les commandes que vous saisissez sont analysées puis exécutées par un programme particulier : *l'interprète de commandes* ou *shell*. Une commande correspond en fait à l'exécution d'un programme par l'interprète. 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.






Le comportement de base d'un shell est le suivant :




1. afficher une invite de commande¹ : \$
2. attendre l'appui de la touche 
3. interpréter la ligne de commande
4. aller en 1.

Pour sortir de cette boucle infinie, on tape la commande `exit`.

Q 4. Essayez cette commande dans votre session. Que se passe-t-il? Pourquoi?

Q 5. Le reste du TP se fera en mode graphique. Retournez-donc dans votre session graphique puis démarrez un émulateur de terminal graphique. Vous avez plusieurs possibilités pour ça:

- via le choix *Terminal MATÉ*, dans le sous-menu *Outils Systèmes* du menu *Logiciels*;
- directement via le raccourci que vous avez ajouté ( +  + );
- via l'appel au démarreur (accessible par  + ) auquel vous devez préciser le nom de la commande à exécuter (ici `mate-terminal`).

Q 6. Assurez-vous que, dans l'environnement de bureau MATÉ, vous avez bien un raccourci clavier permettant de *Lancer un Terminal* via l'appui simultané sur les touches ,  et .

Commandes

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

Unix est un système multi-utilisateurs, c'est-à-dire qu'il permet à plusieurs sessions d'être ouvertes en même temps. La commande `who` permet de connaître les sessions en cours.

Q 7. Lancez les différentes lignes de commande suivantes et commentez les différents résultats obtenus :

- `who`
- `who -H`
- `who am i`
- `who -y`
- `who --help`

Une commande est, soit un programme qui est stocké dans un fichier (`cat` par exemple est stocké dans `/bin/cat`), soit un ordre que le shell connaît et sait exécuter seul (`exit` par exemple). Les premières sont appelées des *commandes externes* et les secondes des *commandes internes ou intégrées* (*builtins commands* en anglais). L'une des commandes disponibles (`which`) permet de retrouver le fichier correspondant à une commande.

Q 8. Saisissez la ligne de commande suivante `which who`

Q 9. Dans cet exemple `who` est-elle considérée comme une option de la commande `which`? Pourquoi?

Étape 3 : Se documenter

Maîtriser la documentation, c'est-à-dire savoir trouver l'information dont on a besoin, est la condition première à une utilisation correcte d'Unix. Vous devez donc apprendre à manipuler les ressources que GNU/Linux vous offre de ce point de vue.

Une exécution minutieuse et attentive de cet exercice est donc indispensable à la suite de notre enseignement.

UNIX est toujours documenté via un manuel électronique. Il est découpé en chapitres, ou *sections*. Chacune de ces sections contient une *page* décrivant un aspect du système (une commande, un format, etc.). Chaque page est nommée en fonction de ce qu'elle documente. Par ailleurs, chacune des sections contient une introduction (*intro*) et documente un type de fonctionnalité. Les aspects suivants sont documentés dans les sections identifiées par leur numéro :

1. Les programmes et les commandes shells
2. Les appels systèmes (fonctions fournies par le noyau)
3. Les appels de bibliothèques (fonctions fournies par des bibliothèques systèmes)
4. Les fichiers spéciaux
5. Les formats des fichiers

1. on dit aussi le *prompt*

6. Les jeux
7. Les formats des macros
8. Les commandes d'administration du système

Pour visualiser une page il suffit de passer son nom comme argument à la commande **man**. **man** va la chercher dans les sections de manière croissante (de la section 1 à la section 8). Dès qu'une page est trouvée elle vous est présentée.

Par exemple pour avoir des informations sur l'utilisation de la commande *<CMD>*, on pourra consulter le manuel électronique grâce à la commande **man** :

```
man <CMD>
```

L'envoi de cette ligne permet d'obtenir la page du manuel électronique correspondant à la commande nommée *<CMD>*.

De la même manière l'appel de **man man** vous présentera le fonctionnement de la commande **man**.

Toutes les pages du manuel sont structurées de la même manière. Elles comportent toutes des parties distinctes, chacune fournissant une catégorie d'information particulière :

NAME	le nom et une description rapide de la commande
SYNOPSIS	toutes les possibilités de saisies liées à cette commande (syntaxe)
DESCRIPTION	une explication 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
EXAMPLES	des exemples d'appel à cette commande
TIPS	des astuces pour utiliser cette commande.

La commande **man**, recherche, prépare puis formate, la page que vous avez demandée. Ensuite elle transmet cette page à un paginateur (par défaut il s'agit de la commande **less**) avec lequel vous interagissez. Par défaut, **man** prépare la page pour un affichage sur un écran. Elle peut néanmoins formater les pages du manuel pour un autre type d'affichage, comme par exemple un navigateur HTML ou un lecteur PDF.

Avant de pouvoir utiliser le manuel il faut d'abord savoir quoi chercher. Pour cela deux commandes permettent de feuilleter le manuel à la recherche de mots-clés ou de connaître la description rapide d'une commande : **apropos** et **what is**.

Q 10. Lisez la page du manuel de la commande **less**.

Q 11. Pour cette commande trouvez dans la documentation :

- comment on fait pour chercher un mot dans un texte affiché sur le terminal,
- comment on fait pour chercher l'exemplaire² suivant du mot cherché,
- comment on fait pour chercher l'exemplaire précédent du mot cherché,
- comment on fait pour quitter le paginateur,
- à quoi correspond l'utilisation du caractère ?.

Q 12. Lisez la page d'introduction de la section 1 du manuel en exécutant la commande : **man intro**.

Q 13. Lisez les pages du manuel de **apropos**, **what is** puis de **man**. Vous devez être capable d'expliquer à quoi correspondent ces trois commandes.

Q 14. Soyez capable d'expliquer à quoi correspondent les options suivantes de la commande **man** :

- f
- k

Q 15. Quelle ligne de commande permet de lire la page du manuel de **intro** dans le navigateur web Firefox?

Q 16. À l'aide des commandes de documentation cherchez et étudiez une commande permettant d'afficher un calendrier.

Q 17. Quelle ligne de commande permet d'afficher le calendrier :

- de l'année courante?
- de l'année dernière?
- de l'année de votre naissance?

2. On dit aussi l'occurrence suivante.

Q 18. Utilisez la commande d’affichage du calendrier pour déterminer à quelle jour de la semaine correspond votre date de naissance?

Il se peut qu’une commande semble bloquer votre terminal (le prompt ne réapparaît pas). Dans ce cas la commande est en train de lire sur le canal d’entrée du terminal (l’entrée standard) et le fera tant que cette entrée ne sera pas fermée. Pour arrêter une telle commande il faut donc soit provoquer la fermeture du canal d’entrée du terminal, c’est-à-dire prévenir la commande que l’entrée standard est terminée, soit provoquer brutalement l’arrêt du programme.

Dans les terminaux que vous allez manipuler :

- la première solution (fermeture de l’entrée) est accessible en appuyant simultanément sur les touches `Ctrl` et `D` ;
- la seconde solution (arrêt brutale de la commande) est accessible en envoyant un signal particulier (KILL) à la commande, elle arrête la commande sans lui laisser le temps de terminer son travail. Pour cela il suffit d’appuyer simultanément sur les touches `Ctrl` et `C`.

Q 19. Tapez la commande `dd`. Que se passe-t-il?

Q 20. Après avoir repris le contrôle de votre terminal, utilisez le manuel de la commande `dd` pour comprendre ce qui s’est passé. Avez-vous arrêté le programme ou mis fin à la lecture des données?

Q 21. Après avoir lu la page du manuel des commandes `mesg` et `wall`, utilisez cette dernière. Que se passe-t-il?

Q 22. Pour reprendre le contrôle du terminal, devez-vous arrêter le programme ou mettre fin à la lecture des données?

Q 23. À l’aide du manuel et en observant le résultat des lignes de commande suivantes, trouvez ce que font les commandes suivantes :

Commandes	Exemples d’utilisation
<code>cat</code>	<code>cat /etc/passwd</code>
<code>ls</code>	<code>ls</code>
<code>date</code>	<code>date</code>
<code>wc</code>	<code>wc /etc/passwd</code>
<code>cal</code>	<code>cal</code>
<code>which</code>	<code>which cal</code>
<code>touch</code>	<code>touch essai</code>
<code>grep</code>	<code>grep root /etc/passwd</code>
<code>cut</code>	<code>cut -f 3</code>
<code>tr</code>	<code>tr abcdefghijklmnopqrstuvwxyz cdefghijklmnopqrstuvwxyzab</code>
<code>quota</code>	<code>quota -v</code>

La commande `man` ne documente que les commandes externes, pas les commandes internes (celles qui correspondent à des ordres connus du shell). Pour connaître le fonctionnement de ces commandes internes il faut se référer à la page de manuel du shell (dans notre cas `bash`). Le shell offre également une commande qui permet d’obtenir une description du fonctionnement de ce genre de commande : `help`.

Q 24. Utilisez la commande `help` pour lire la documentation des commandes intégrées suivantes et déterminer leur rôle :

- `help`
- `exit`
- `cd`
- `echo`
- `pwd`

Q 25. Quelle commande permet d’afficher le texte `Hello world!` sur votre terminal?

Sur les systèmes GNU/Linux il existe souvent un autre type de documentation accessible via la commande `info`.

Q 26. Utilisez la commande `man info` puis `info info` pour apprendre à manipuler cet outil.

Q 27. Comment fait-on dans `info` pour se positionner sur le prochain lien? Sélectionner un lien?

Q 28. Grâce à la commande `info` déterminez à quoi sert exactement la commande `emacs`.