

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 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`

Q 3. Consultez la version réseau du fichier `/etc/passwd` en utilisant la commande `getent 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 4. 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 le prompt¹ : \$
2. attendre l’appui de la touche 
3. interpréter la ligne de commande
4. aller en 1.

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

Q 5. Essayez cette commande dans votre session. Que se passe-t-il ? Pourquoi ?

Q 6. Le reste du TP se fera en mode graphique. Retournez-donc dans votre session graphique puis démarrez un émulateur de terminal graphique via le choix *Terminal MATÉ*, dans le sous-menu *Outils Systèmes* du menu *Logiciels* ou directement via le raccourci  +  +  que vous avez ajouté lors du TP0.

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. Connectez-vous via le terminal virtuel numéro 7 (la session graphique)

Q 8. 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 9. Saisissez la ligne de commande suivante `which who`

Q 10. 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
6. Les jeux
7. Les formats des macros

1. on dit aussi l’invite de commande

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.

Par défaut la commande **man**, prépare puis formate, la page que vous avez demandée, pour un affichage sur un écran. Ensuite elle utilise un paginateur pour vous la présenter. 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 **whatis**.

Q 11. Lisez la page du manuel de la commande **less**.

Q 12. 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'occurrence suivante du mot cherché,
- comment on fait pour chercher l'occurrence précédente du mot cherché,
- comment on fait pour quitter le paginateur,
- à quoi correspond l'utilisation du caractère ?.

Q 13. Lisez la page d'introduction de la section 1 du manuel en exécutant la commande **man intro**.

Q 14. Lisez les pages du manuel de **man**, **apropos** et **whatis**.

Q 15. Soyez capable d'expliquer à quoi correspondent les commandes **whatis** et **apropos**.

Q 16. Soyez capable d'expliquer à quoi correspondent les options suivantes de la commande **man** :

- f
- k

Q 17. Quelle ligne de commande permet de lire la page du manuel de **intro** dans le navigateur web Firefox?

Q 18. Quelle ligne de commande permet de convertir la page du manuel de **bash** en un fichier nommé **manuel-bash.pdf**. Pour vérifier que votre fichier PDF correspond bien à la documentation du shell **bash** vous pourrez utiliser la commande **evince** qui permet de visualiser un fichier de ce format.

Q 19. À l'aide des commandes de documentation cherchez et étudiez une commande permettant d'afficher un calendrier.

Q 20. Quelle ligne de commande permet d'afficher le calendrier :

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

Q 21. 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.

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 22. Tapez la commande **dd**. Que se passe-t-il?

Q 23. 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 24. Après avoir lu la page du manuel des commandes **mesg** et **wall**, utilisez cette dernière. Que se passe-t-il?

Q 25. Pour reprendre le contrôle du terminal, devez-vous arrêter le programme ou mettre fin à la lecture des données?

Q 26. 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
cat	cat /etc/passwd
date	date
wc	wc /etc/passwd
cal	cal
which	which cal
touch	touch essai
grep	grep root /etc/passwd
cut	cut -d 3
tr	tr abcdefghijklmnopqrstuvwxyz cdefghijklmnopqrstuvwxyzab
quota	quota -v

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 il faut se référer à la page de manuel du shell (dans notre cas **bash**). Le shell offre une commande qui permet d'obtenir une description du fonctionnement de ce genre de commande : **help**.

Q 27. Utilisez la commande **help** pour lire la documentation des commandes intégrées suivantes et déterminer leur rôle:

- **help**
- **exit**
- **cd**
- **echo**
- **pwd**

Q 28. 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 29. Utilisez la commande **man info** puis **info info** pour apprendre à manipuler cet outil.

Q 30. Comment fait-on dans **info** pour se positionner sur le prochain lien? Sélectionner un lien?

Q 31. Grâce à la commande **info** déterminez à quoi sert exactement la commande **emacs**.

Étape 4 : Utiliser les variables d'environnement

Certains aspects de la configuration du système (l'emplacement de certaines bibliothèques, le nom de l'écran à utiliser, etc.) peuvent être nécessaires à l'exécution de certaines commandes externes. Pour rendre ces informations accessibles on doit définir des espaces particuliers (on dit des *variables*) et *exporter* leur contenu dans l'environnement des commandes. On parle alors de *variables d'environnement*. Pour fixer une valeur à une variable d'environnement il suffit d'utiliser la commande intégrée **export** avec en paramètre le nom d'une variable suivi du signe **=** et enfin de la valeur que l'on veut y affecter.

Q 32. La commande **java**, par exemple, consulte le contenu de la variable d'environnement **CLASSPATH** avant son exécution. Cette variable contient une liste de bibliothèques à utiliser. Chaque élément de la liste est séparé du suivant par le caractère **:**. Vous allez manipuler cette variable d'environnement pour illustrer le fonctionnement de celles-ci.

1. Ouvrez un terminal et essayez d'exécuter un des programmes de test contenus dans la librairie `ap.jar` (elle en contient au moins deux : `test.TestAnsi`, `test.TestJasmineTest`). Pour cela ne passez aucune option à la commande `java` en dehors du nom du programme à exécuter (`java test.TestAnsi` par exemple). Que se passe-t-il?
2. Ajoutez la librairie `ap.jar` dans la variable d'environnement `CLASSPATH` en exécutant dans le même terminal la commande suivante :

```
export CLASSPATH=/home/public/ap/ap.jar:.
```

3. Ré-essayez d'exécuter le programme de test. Que se passe-t-il?
4. Fermez votre terminal.
5. Ouvrez un nouveau terminal et ré-essayez d'exécuter le programme de test. Que se passe-t-il?

Q 33. Comme vous l'avez constaté les variables d'environnement ne survivent pas à la fermeture du terminal dans lequel elles ont été fixées. Pour rendre leur définition globale et pérenne il suffit de les assigner dans le fichier `.bashrc` qui est placé dans votre répertoire principal.

1. Modifiez votre fichier `.bashrc` afin d'y ajouter la ligne : `export CLASSPATH=/home/public/ap/ap.jar:.`
2. Démarrez un nouveau terminal.
3. Ré-essayez d'exécuter le programme de test. Que se passe-t-il?
4. Essayez maintenant d'exécuter le programme `TP0` que vous avez créé lors du TP0.

Étape 5 : Aborder l'utilisation de Windows



Windows possède des caractéristiques similaires à UNIX sous bien des aspects abordés dans ce premier TP, même si c'est d'une manière beaucoup moins élaborée. C'est notamment le cas pour ce qui concerne le langage de commandes, la documentation ou les variables d'environnement.

Q 34. Préparez-vous à utiliser Windows :

1. copiez le fichier `ap.jar` dans votre répertoire personnel grâce à la commande :

```
cp /home/public/ap/ap.jar ~
```

2. respirez un grand coup,
3. démarrez une machine virtuelle sous Windows, puis connectez-vous.

Sous Windows il existe, comme sous Linux, de nombreux raccourcis claviers permettant d'effectuer un nombre important de tâches en se passant de l'utilisation de la souris (et donc de gagner du temps). Appuyer en même temps sur la touche  et la touche  permet par exemple d'ouvrir le programme de navigation dans vos fichiers. La [documentation officielle de Windows](#) liste l'intégralité de ces raccourcis claviers liés aux système.

Q 35. En vous aidant de cette documentation, déterminez quelle est la combinaison permettant d'exécuter une commande.

Q 36. Démarrez un terminal Windows en exécutant la commande `CMD`.

Sous Windows il n'y a pas une hiérarchie unique pour tous les supports de stockage, mais une hiérarchie **par** unité de stockage. Chaque unité de stockage, nommée *disque* ou *volume*, est identifiée par une lettre de l'alphabet suivi du caractère `:`. La notion de chemin est *quasiment* la même que sous Unix à la différence près que le séparateur de répertoire n'est pas `/` mais `\`.

La racine du disque `C` est donc identifiée par `C:\`.

Au département, votre répertoire de travail principal, le même sous Unix et sous Windows, est automatiquement rendu disponible comme un *disque* Windows identifié par `E:`. Vous aurez donc accès à toutes vos données quelque soit le système utilisé.

Le terminal Windows permet d'utiliser un langage de commande. Celui-ci est **très** différent et **beaucoup** moins souple ou puissant que le shell Unix. Cependant comme sous Unix sa documentation est disponible en ligne via une commande : `help`.


Q 37. Utilisez la commande `help` dans votre terminal Windows :

1. pour déterminer la commande permettant de connaître le répertoire courant
2. pour déterminer la commande permettant de lister les fichiers contenus dans le répertoire courant
3. pour déterminer la commande permettant de changer le répertoire courant vers `E:\`

Q 38. Sous Windows exporter une variable dans l'environnement se fait via la commande `SET`. Après avoir étudié sa documentation :

1. Essayez d'exécuter le programme de test de la librairie `ap.jar`, comme vous l'avez fait sous Unix. Que se passe-t-il?
2. Fixez la variable `CLASSPATH` à la valeur `E:\ap.jar:.` dans votre terminal Windows,

3. Ré-essayez d'exécuter le programme de test. Que se passe-t-il?
4. Fermez votre terminal Windows, ré-ouvrez en un puis ré-essayer d'exécuter le programme de test. Que se passe-t-il?

Pour fixer une variable d'environnement globalement (c'est-à-dire pour l'intégralité des terminaux et autres programmes Windows) vous devez fixer la variable dans les *Paramètres avancés du système* puis, souvent, quitter votre session et enfin en ré-ouvrir une nouvelle. Pour cela il faut d'abord accéder à la fenêtre des *Propriétés Système* (accessible grâce à  + [Pause]), puis choisir l'item *Paramètres avancés* dans le menu de gauche de cette fenêtre, cliquer ensuite sur le bouton *Variables d'environnement* de la fenêtre qui s'est ouverte et enfin ajouter une *Variable Utilisateur* avec le nom et la valeur qui vous intéresse.

Q 39. Fixez une variable d'environnement globale nommée `CLASSPATH` avec comme valeur `E:\ap.jar:.`, puis ouvrez un nouveau terminal Windows pour essayer d'exécuter le programme de test. Que se passe-t-il?

N'oubliez pas de quitter *toutes* les sessions que vous avez ouvertes lors de votre séance.