

Tous les documents relatifs à cet enseignement (transparents de cours d'amphi, supports de cours-TD, sujets de TP, annales d'examen) sont disponibles sur didel

<http://didel.script.univ-paris-diderot.fr/>

à partir du mot-clé IF1.

## 1 Premier contact

Les machines de la salle de TP tournent sous un système de type Unix<sup>1</sup> (FreeBSD) et utilisent l'environnement de bureau léger Xfce. Unix est un système multitâche : on peut utiliser plusieurs logiciels en même temps.

### 1.1 Authentification

Avant de pouvoir travailler il faut *s'authentifier* auprès du système. Pour cela, il faut entrer son nom d'utilisateur (*login*) et son mot de passe (*password*).

Si vous utilisez votre compte pour la première fois, il y a trois cas possibles :

**Vous êtes inscrit administrativement et votre compte est déjà activé.** Dans ce cas, il n'y a rien à faire de spécial. Il vous suffit de rentrer votre identifiant (login) et votre mot de passe (password) lorsque le système d'exploitation est lancé.

**Vous êtes inscrit mais votre compte n'est pas encore activé.** Pour activer votre compte, vous pouvez emprunter la session d'un.e camarade<sup>2</sup>. Rendez-vous alors à l'aide d'un navigateur sur la page

<http://portail.univ-paris-diderot.fr/>

puis suivez le lien `activer son compte`. On vous demande de rentrer les renseignements inscrits sur votre carte d'étudiant (nom, prénom, numéro d'étudiant et date de naissance) afin de vous identifier et de lancer la procédure d'activation de compte. Après deux ou trois minutes, redémarrez votre session pour que l'activation soit effective.

**Vous n'êtes pas encore inscrit (ou en attente de carte).** Vous ne pouvez malheureusement pas vous connecter vous-même : chercher un.e camarade (ayant un accès au système) avec qui travailler.

► **Exercice 1** : Authentifiez-vous auprès du système. Attention : le mot de passe n'apparaît pas en clair : parfois aucun caractère n'apparaît, parfois un rond apparaît quand on presse une touche. Pensez à appuyer sur la touche `Entrée` après votre login et après votre mot de passe.

---

1. ou Unix-like

2. ou bien, sous `Windows`, vous connecter sous le nom d'utilisateur `me`, sans mot de passe, domaine `ENS`

## 1.2 Quelques logiciels

Une fois votre nom d'utilisateur et mot de passe vérifiés, votre session s'ouvre et le gestionnaire de bureau, le programme qui gère l'affichage des menus et des fenêtres (ici KDE) apparaît. Le menu principal (bouton  en bas à gauche) permet de lancer des applications, d'accéder aux outils de paramétrage du système ou encore de fermer la session ou d'éteindre la machine.

La plupart des programmes ainsi que le menu principal disposent d'une entrée « aide » (*help*) permettant d'accéder à l'aide en ligne. Si vous êtes coincé(e), n'hésitez pas à la consulter.

Parmi les logiciels que nous utiliserons, il y a *Emacs* (éditeur de texte), *Firefox* (gestionnaire de fichiers et navigateur web), *Terminal* (fenêtre *shell*).

► **Exercice 2** : Lancez *Firefox*. Déplacez la fenêtre. Changez sa taille. Fermez-la.

► **Exercice 3** : Lancez *Firefox* de nouveau, et consultez la page

`http://didel.script.univ-paris-diderot.fr/`

Cliquez sur le lien Connexion ENT (en haut à droite), puis identifiez-vous (ce sont les mêmes identifiants que pour se connecter à la machine).

Cliquez sur le lien M'inscrire à un cours (dans la colonne à gauche). Dans le champ de saisie de mot-clé, entrez IF1. Sélectionnez ensuite le cours

IF1 - Introduction à l'informatique et la programmation

et inscrivez-vous dans le groupe qui vous correspond.

Inscrivez-vous ensuite dans le cours

DSE - Département de Sciences Exactes

Déconnectez-vous de votre espace didel et reconnectez-vous dessus : des liens vers ces cours apparaissent directement.

Tous les documents liés à l'enseignement IF1 seront disponibles sur le site du cours, ainsi que les annonces importantes. Pensez à le consulter régulièrement. Vous pouvez aussi profiter des forums pour poser des questions ou répondre aux questions des autres.

À chaque connexion ce qui est nouveau par rapport à votre précédente visite est précédé d'un rond rouge.

► **Exercice 4** : Lancez l'éditeur de texte *emacs* et créez un nouveau fichier. Rédigez un (petit) texte vous présentant : vos nom et prénom, votre scolarité passée, vos connaissances en informatique, ce que vous aimeriez être en train de faire à l'instant présent, dans dix ans, etc. Sauvegardez le fichier sous le nom *nom\_prenom.txt*. Ce fichier sera ainsi conservé sur votre espace de stockage ENT et dorénavant accessible depuis l'extérieur.

► **Exercice 5** : Lancez le navigateur web *Firefox*, et pointez-le sur votre « espace numérique de travail » à l'université.

Sous la rubrique Webmail Paris Diderot, créez un nouveau message ayant pour objet

[IF1] fiche de *nom prénom*

copiez-y le texte du fichier que vous avez rédigé et attachez-le en pièce jointe, puis envoyez le tout à vos chargé(e)s de TP et de cours-TD.

Toujours à l'aide de votre navigateur, retrouvez le fichier texte *nom\_prenom.txt*.

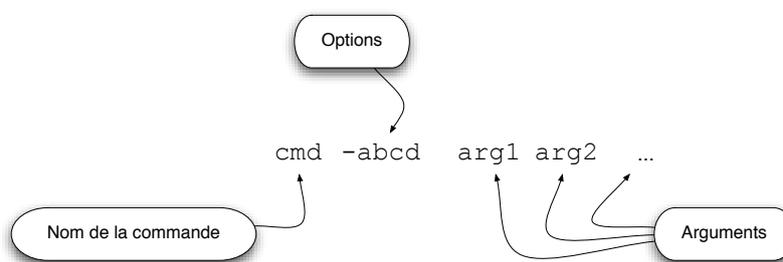
## 2 Premiers pas avec Unix

### 2.1 Le shell Unix

Une autre façon d'interagir avec le système est d'utiliser un *terminal* (ou *console*) dans lequel vous pouvez taper des commandes (qui ne sont rien d'autre que des programmes). Le programme avec lequel vous interagissez pour exécuter les commandes s'appelle le *shell*. Celui-ci se contente d'attendre que vous lui demandiez d'exécuter une commande pour le faire. Vous lui indiquez que vous avez fini de taper une commande en appuyant sur la touche `Entrée` (Enter).

Le shell vous indique qu'il est prêt en affichant une invite (ou *prompt*), en général le caractère dollar (\$) ou supérieur (>), en début de ligne.

Pour affiner le comportement d'une commande, on peut lui adjoindre ce que l'on appelle des *options*, représentées par un tiret suivi d'une ou plusieurs lettre(s). On peut également utiliser certaines commandes avec des *arguments*. La forme générale d'une ligne de commande est souvent (mais pas toujours) la suivante :



**Affichage d'une commande** Si tout se passe bien, la commande soit affiche la réponse à ce qui lui a été demandé, soit n'affiche rien : en effet, la plupart du temps ce qu'on lui demande n'entraîne pas d'affichage (par exemple si on veut renommer un fichier). La commande peut également afficher des messages d'erreur. Dans ce cas, la commande ne s'est pas exécutée correctement et il faut lire le message d'erreur pour comprendre le problème.

**Édition de ligne** Si on se trompe en tapant une commande, et qu'on s'en aperçoit avant d'appuyer sur `Entrée`, on peut utiliser les touches `←` et `→` pour déplacer le curseur à l'endroit où est l'erreur.

**Historique** Si on ne s'aperçoit de l'erreur qu'après avoir démarré la commande, on veut souvent lancer une autre commande corrigée. Au lieu de tout retaper, on peut utiliser la touche `↑`, qui rappelle la commande précédente (puis la commande d'avant, etc., si on appuie plusieurs fois). La touche `↓` permet de redescendre dans l'historique des commandes, vers les commandes les plus récentes.

**Complétion** Lorsqu'on veut taper le nom d'un fichier existant, on peut taper le début de ce nom puis appuyer sur la touche *tabulation* (marquée `Tab` ou `↵`). Le shell insère alors la fin du nom (s'il y a plusieurs possibilités, le shell complète seulement le plus long préfixe commun). La complétion a deux avantages : elle permet de moins taper, et elle assure que le nom complété existe.

### 2.2 La commande man

`man` (*manual*): Le manuel en ligne pour toutes les commandes accessibles depuis le terminal. Il suffit de taper `man cmd` pour accéder à la description complète de la commande `cmd`.

Par exemple, la ligne de commande `man 1s` permet d'obtenir la documentation de la commande `1s`. On peut faire défiler le texte à l'aide de la barre d'espace `␣` et des flèches `↑`/`↓`, et quitter à l'aide de la touche `q` (quit).

Pour chercher un mot dans une page de manuel, par exemple *file* (fichier en anglais), on tape `/file` dans le terminal où a été lancée la commande `man`. La première occurrence du mot est alors mise sur la première ligne. Pour avoir l'occurrence suivante, on utilise la touche `[n]` (next) et ainsi de suite.

- **Exercice 6** : Tapez `man ls` et analysez la structure de la page de manuel. Que fait la commande `ls` ? À quoi sert l'option `-l` ?

## 2.3 Mini introduction aux processus

Un *processus* est un objet dynamique : c'est l'état de l'exécution d'un programme au cours du temps.

- **Exercice 7** (Interrompre un programme, le symbole `&`) :
  1. Ouvrez à nouveau votre fichier texte `nom_prenom.txt` en tapant dans le terminal la ligne de commande : `emacs nom_prenom.txt`
  2. Une fois l'éditeur ouvert, revenez sur le terminal et tapez une commande (par exemple `man man`). Que se passe-t-il ?
  3. Toujours sous le terminal, gardez la touche `[Ctrl]` enfoncée puis pressez et relâchez la touche `[z]` (on dit par abus de langage qu'on utilise la touche `[Ctrl-z]`). Que se passe-t-il ?
  4. Sur le terminal tapez la commande `bg`. Vous pouvez à nouveau utiliser votre fenêtre `emacs`. Fermez-la.
  5. Répétez les étapes 1 et 2, en ajoutant cette fois le caractère `&` en fin de commande (soit par exemple `emacs nom_prenom.txt &`). Voyez-vous une différence ?

**Stopper un processus** Une autre "touche" (en fait une combinaison de touches) à connaître est `[Ctrl-c]` (voir les explications sur `[Ctrl-z]` pour utiliser cette combinaison de touches). Elle stoppe le processus en cours (à condition qu'il n'ait pas été lancé avec le caractère `&`) et le shell affiche une nouvelle invite.

## 2.4 Commandes de manipulation de fichiers et répertoires

**Fichiers** (*file*) Un fichier est une suite de données, représentant par exemple un texte, une image etc. Chaque fichier possède un nom, conventionnellement terminé par un point et une suite de caractères indiquant le type de données qu'il contient (mais cela reste une convention). Par exemple, le fichier qui contient l'énoncé de ce TP s'appelle `tp0.pdf`, et son nom indique qu'il est au format PDF. Les fichiers sources de programmes en Java portent l'extension `.java`.

Les systèmes Unix (comme ceux de la salle TP) font une différence entre majuscules et minuscules : `tp0.pdf`, `TPO.pdf` et `TPO.PDF` désignent trois fichiers différents.

**Répertoires** (*directory*) Sur les systèmes Unix, les fichiers sont organisés sous forme d'un *arbre* (on parle de l'*arborescence* des fichiers) : chaque fichier est stocké dans un *répertoire*<sup>3</sup>, et tout répertoire peut lui-même contenir d'autres répertoires.

**Le répertoire *home*** Le répertoire dit *home* (« maison », *répertoire personnel* ou *répertoire de login*), noté « `~` », est l'endroit où vous pouvez stocker vos fichiers personnels (attention : ce n'est pas `/home/`).

Voici quelques commandes de base pour vous repérer et vous déplacer dans une arborescence de fichiers et pouvoir la modifier :

---

3. Appelé *dossier* par certains.

`pwd` (*print working directory*): Indique dans quel répertoire vous vous trouvez, le *répertoire courant*.

À l'ouverture d'un nouveau terminal, le répertoire courant est toujours le répertoire de login. Chaque terminal possède son propre répertoire courant. Certaines commandes prennent en argument un nom de répertoire. On utilise « . » pour le répertoire courant.

`ls` (*list directory*): Affiche le contenu du répertoire courant, c'est-à-dire la liste de ses fichiers et sous-répertoires.

► **Exercice 8** : Votre répertoire *home* contient-il des fichiers ?

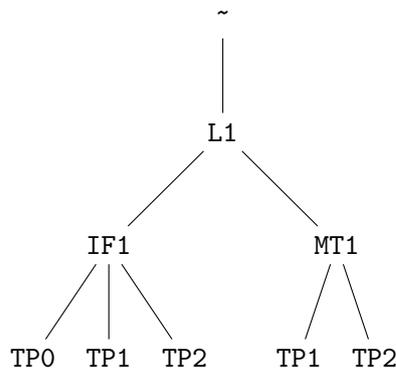
`cd` (*change directory*): Modifie le répertoire courant en lui donnant la valeur de son argument. Utilisée sans argument, cette commande fait du répertoire de login le répertoire courant.

Il est souvent utile de remonter dans l'arborescence de fichiers. On désigne par « .. » le répertoire père du répertoire courant (celui qui est juste au-dessus dans l'arborescence).

► **Exercice 9** : Remontez le plus possible dans l'arborescence. Quel est le nom du répertoire racine (c'est-à-dire celui au-delà duquel on ne peut plus remonter) ?

`mkdir` (*make directory*): Crée un nouveau répertoire dont le nom est donné en argument.

► **Exercice 10** : Créez la sous-arborescence suivante dans votre répertoire de login (les TP0, TP1 et TP2 désignent des répertoires) :



Prenez l'habitude de ranger vos fichiers au bon endroit dans cette sous-arborescence, cela facilitera leurs recherches futures.

Pour désigner un fichier ou un sous-répertoire sans ambiguïté, on peut donner un *chemin* : par exemple le sous-répertoire TP1 de votre répertoire IF1 dans l'arborescence de l'exercice précédent peut être désigné par `~/L1/IF1/TP1`, ce qui permet de le distinguer facilement du répertoire `~/L1/MT1/TP1`. Un chemin est donc une suite de noms de répertoire(s), éventuellement terminée par un nom de fichier, chaque élément étant séparé du suivant par un / (*slash*).

Pour désigner un fichier ou un répertoire de l'arborescence, on peut utiliser un chemin à partir du répertoire courant (par exemple le nom de ce fichier ou ce répertoire s'il se trouve dans le répertoire courant), ou un chemin à partir d'un répertoire qu'on peut désigner de façon non ambiguë : / pour la racine, ~ pour le répertoire de login, .. pour le répertoire père du répertoire courant.

`mv` (*move*): Déplace un fichier. Prend deux arguments : en premier le nom du fichier, en deuxième son nouvel emplacement. Par extension, permet de renommer un fichier, en utilisant en deuxième argument le nouveau nom du fichier. (En fait l'action de cette commande est un peu plus subtile, mais nous ne rentrerons pas dans les détails ici.)

- **Exercice 11** : Déplacez le fichier `nom_prenom.txt` de l'exercice ?? dans le répertoire `~/L1/IF1/TP0`.

`cp` (*copy*): Effectue une copie d'un fichier (dont le nom est donné en premier argument) vers une destination (dont le nom est donné en deuxième argument). Si le deuxième argument est un répertoire, le nom du nouveau fichier est le même que celui de l'ancien (mais pas le chemin qui mène à ce nom).

- **Exercice 12** :

1. Copiez le fichier `~/L1/IF1/TP0/nom_prenom.txt` dans votre répertoire de login en nommant cette copie `presentation` dans la foulée.
2. Déplacez-vous dans le répertoire `~/L1/MT1` et copiez-y le fichier `presentation` créé à la question précédente.

- **Exercice 13** : Quelle option de `cp` permet de faire une copie d'un répertoire (et de toute sa sous-arborescence) ?

`rm` (*remove*): Supprime le fichier dont le nom est donné en argument.

- **Exercice 14** : Supprimez les fichiers `presentation` créés à l'exercice ??.

`rmdir` (*remove directory*): Supprime le répertoire *vide* dont le nom est donné en argument.

- **Exercice 15** :

1. Essayez de supprimer le répertoire `~/L1/MT1` à l'aide de la commande `rmdir`. Que se passe-t-il ?
2. Trouvez l'option de `rm` permettant de supprimer un répertoire non vide.

`cat` (*catenate*): Concatène des fichiers et/ou affiche leur contenu sur la sortie standard.

- **Exercice 16** : Testez la commande `cat` sur vos fichiers textes avec différentes options.

### 3 Premiers pas avec Emacs

Le système d'exploitation que vous utilisez met à votre disposition différents éditeurs de texte. Nous utiliserons *Emacs*, qui est particulièrement bien adapté à la programmation.

Lancez le programme Emacs en tapant `emacs &`

- **Exercice 17** (Utilisation d'Emacs) :

1. À l'aide d'Emacs, créez un fichier `poeme.txt`. Tapez votre poème favori (quelques vers suffiront). Sauvegardez, mais ne fermez pas.
2. Toujours dans le même Emacs, créez un fichier `chanson.txt` où vous taperez les paroles de votre chanson favorite (quelques vers suffiront). Sauvegardez.
3. Revenez au tampon du fichier `poeme.txt`. (*Indication* : regardez dans le menu *Buffers*.)
4. Pouvez-vous voir les deux fichiers en même temps ?

## 4 Manipuler des ensembles de fichiers

### 4.1 Expansion

Il existe un certain nombre de symboles spéciaux appelés *Jokers* ou *wildcards*, qui permettent de désigner plusieurs fichiers à la fois. Il en existe plusieurs, mais les principaux sont :

? : Dans un nom de fichier, représente exactement un symbole.

\* : Dans un nom de fichier, représente n'importe quelle suite de symboles (aucun symbole, un symbole ou plusieurs symboles).

Ainsi, écrire `tp?.pdf` désigne `tp0.pdf`, `tp1.pdf` mais pas `tp12.pdf`, ni `tp-if1.pdf`, et `truc*` désigne `truc`, `trucage`, `trucmuche` mais pas `struck`.

#### ► Exercice 18 :

1. Affichez la liste de tous les fichiers dans le répertoire `/usr/bin` dont le nom commence par `k` et contient exactement 6 caractères.
2. Affichez la liste de tous les fichiers dont l'extension est `jar` dans le répertoire `/usr/local/share/java`.

### 4.2 Archivage

Il peut arriver de vouloir regrouper plusieurs fichiers en un seul, par exemple pour l'envoi de pièces jointes par courrier électronique, ou pour la mise à disposition d'un ensemble de fichiers sur internet. On dispose pour cela d'un outil appelé `tar` (pour *tape archive*, cet outil était auparavant destiné à l'archivage sur bande magnétique). On utilise le suffixe `.tar` pour distinguer les archives.

`tar` (*tape archive*): Archive plusieurs fichiers en un seul ou désarchive, selon les options.

`tar cvf toto.tar fic1 fic2 ...` archive les fichiers `fic1`, `fic2...` dans `toto.tar`

`tar xvf toto.tar` extrait dans le répertoire courant les fichiers contenus dans `toto.tar`

#### ► Exercice 19 :

Créez à l'aide d'Emacs quatre fichiers : `fic1` contenant le mot "Ceci", `fic2` contenant le mot "est", `fic3` contenant le mot "une", et `fic4` contenant le mot "archive". Archivez ces quatre fichiers dans une archive que vous nommerez `test.tar`.

Examinez le contenu de l'archive à l'aide de la commande `cat` par exemple. Que constatez-vous ?

La commande `tar` permet également d'archiver des répertoires entiers directement.