

TP 0 : Premier contact

Introduction à l'informatique et à la programmation (IF1)

Semaine du 19 Septembre 2011

Cet énoncé est destiné aux étudiants qui ne suivent pas IS1. Ceux qui suivent IS1 (mentions informatique et mathématiques-informatique) verront ces notions dans cette matière.

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

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

à partir du mot-clé 51IF1IF1.

1 Premier contact

Les machines de la salle de TP tournent sous un système de type Unix¹ (Linux ou FreeBSD) et utilisent le gestionnaire de fenêtres KDE. 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 devez vous connecter sous le nom d'utilisateur me (sans mot de passe). Rendez-vous alors à l'aide d'un navigateur sur la page

`http://portil.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.

1. ou Unix-like

Vous n'êtes pas encore inscrit (ou en attente de carte). Utilisez le compte me provisoirement, en attendant la réception de votre carte. Attention, le compte me ne permet pas de sauvegarder vos fichiers (mais ce n'est pas un problème pour la séance d'aujourd'hui, étant donné qu'il n'y a rien à sauvegarder).

- **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.

1.2 Quelques logiciels

Une fois votre nom d'utilisateur et mot de passe vérifiés, le gestionnaire de bureau, le programme qui gère l'affichage des menus et des fenêtres (ici KDE) apparaît. Le menu principal (bouton K 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 l'ordinateur.

Note : 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 *XEmacs* (éditeur de texte), *Konqueror* (gestionnaire de fichiers et navigateur web), *Konsole* (fenêtre *shell*).

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

- **Exercice 3 :** Lancez Konqueror de nouveau, et consultez la page

`http://didel.script.univ-p_ris-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 bande horizontale). Dans le champ de saisie de mot-clé, entrez IF1 2011. Sélectionnez ensuite le cours

`51IF1IF1 - IF1 2011-2012 - Introduction à l'Informatique`

et inscrivez-vous dans le groupe qui vous correspond.

- **Exercice 4 :** Déconnectez-vous de votre espace didel et reconnectez-vous dessus : un lien vers ce cours apparaît directement.

Tous les documents liés à cet enseignement seront disponibles sur ce site, 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 connection ce qui est nouveau par rapport à votre précédente visite est précédé d'un rond rouge.

- **Exercice 5 :** Lancez l'éditeur de texte *xemacs* 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 10 ans, etc. Sauvegardez le fichier sous le nom `nom_prenom.txt`.

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 du nom du fichier puis appuyer sur la touche *tabulation* (marquée `T b` 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`

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

Par exemple, la ligne de commande `m n ls` permet d'obtenir la documentation de la commande `ls`. 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`.

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 `m n`. La première occurrence du mot est alors mise sur la première ligne. Pour avoir l'occurrence suivante, on utilise la touche `n` et ainsi de suite.

- **Exercice 7** : Tapez `m n 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 8** (Interrompre un programme, le symbole `&`) :

1. Ouvrez à nouveau votre fichier texte `nom_prenom.txt` en tapant dans le terminal la ligne de commande : `xem cs nom_prenom.txt`
2. Une fois l'éditeur ouvert, revenez sur le terminal et tapez une commande (par exemple `m n`). 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 taper la commande `bg`. Vous pouvez à nouveau utiliser votre fenêtre `xem cs`. Fermez-la.
5. Répétez les étapes 1 et 2, en ajoutant cette fois le caractère `&` en fin de commande (soit par exemple `xem cs 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). Cette touche 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 `J v` portent l'extension `.j v`.

Les systèmes Unix (comme ceux de la salle TP) font une différence entre majuscules et minuscules : `tp0.pdf`, `TP0.pdf` et `TP0.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*³, et les répertoires peuvent eux-mêmes 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.

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

`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 9** : 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.

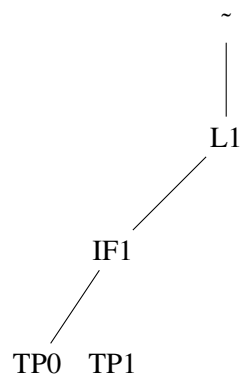
Il est souvent utile de remonter dans l'arborescence de fichiers. On désigne par « `..` » le répertoire père du répertoire courant (celui qui est juste au-dessus dans l'arborescence).

► **Exercice 10** : 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 11** : Créez la sous-arborescence suivante dans votre répertoire de login (les TP* désignent des répertoires) :

3. Appelé *dossier* par certains.



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

- **Exercice 15 :** Supprimez les fichiers présentés à l'exercice 13.

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

- **Exercice 16 :**

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.

3 Premiers pas avec XEmacs

Le système d'exploitation que vous utilisez met à votre disposition différents éditeurs de texte (*kwrite*, *nedit*, *vi*, *XEmacs* etc.). Nous utiliserons *XEmacs*, qui est particulièrement bien adapté à la programmation⁴.

Lancez le programme XEmacs en tapant `xem cs &`

3.1 Utilisation d'XEmacs

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

1. À l'aide d'XEmacs, créez un fichier `poeme.text` ; assurez-vous que le tampon est bien en mode *Text* (regardez en bas de la fenêtre). Tapez votre poème favori (quelques vers suffiront). Sauvegardez, mais ne fermez pas.
2. Toujours dans le même XEmacs, créez un fichier `chanson.text` où vous taperez les paroles de votre chanson favorite (quelques vers suffiront). Sauvegardez.
3. Revenez au tampon du fichier `poeme.text`. (*Indication : regardez dans le menu Buffers.*)
4. Pouvez-vous voir les deux fichiers en même temps ? (*Indication : regardez comment scinder la fenêtre dans le menu View.*)

XEmacs fait des sauvegardes intermédiaires automatiques. Si votre fichier se nomme `mon_fichier`, ces sauvegardes se nommeront `mon_fichier~` ou `#mon_fichier#`.

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.

4. Pour le moment, nous vous imposons d'utiliser *XEmacs*.

* : 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`, `tpi.pdf` mais pas `tp12.pdf`, ni `tp-if1.pdf`, et `truc*` désigne `truc`, `truc ge`, `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 `.j r` dans le répertoire `/usr/sh re/j v`.

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 `t r` (pour *tape archive*, cet outil était auparavant destiné à l'archivage sur bande magnétique). On utilise le suffixe `.t r` pour distinguer les archives.

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

Exemple 1.

`t r cvf toto.t r fic1 fic2 ...` archive les fichiers `fic1`, `fic2`... dans `toto.t r`

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

► **Exercice 19** (Création d'une première archive) :

Créez, à l'aide d'`xem cs` 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.t r`.

Examinez le contenu de l'archive à l'aide des commandes `c t` puis `less`. Que constatez-vous ?

La commande `t r` permet également d'archiver des répertoires entiers directement.