TP 0 : Prise de contact

Informatique Fondamentale (IF121)

22–26 septembre 2003

1 Introduction

Le but de ces séances de TP est de mettre en pratique les notions introduites lors des cours/TD. La première notion a acquérir est une connaissance élémentaire des ordinateurs et du système d'exploitation associé. C'est l'objectif de ce TP.

Les sujets de TP et autres documents seront distribués au fur et à mesure. En principe, ils seront également placés sur le web, à l'adresse suivante :
`http://para.inria.fr/~peskine/enseignement/deug-2003/index.html`

Les disques des salles de TP sont régulièrement effacés. Aussi il n'est pas possible de conserver vos fichiers après la fin du TP. Si vous souhaitez garder une trace de votre travail, vous pouvez **apporter une disquette**. Si vous disposez d'une adresse électronique, vous pouvez aussi y envoyer les fichiers que vous avez écrits.

2 Découverte de l'ordinateur

2.1 Le matériel

Les ordinateurs sur lesquels vous allez travailler sont constitués d'une unité centrale et de périphériques. Par analogie avec le corps humain, l'unité centrale peut être vue comme le cerveau de l'ordinateur alors que les périphériques sont le reste du corps.

Ainsi, l'unité centrale effectue les tâches de traitement de l'information. Pour cela, elle est composée d'un processeur qui effectue diverses opérations, de mémoire vive qui stocke des informations temporaires et d'un disque dur qui stocke de l'information même lorsque l'ordinateur est éteint. Enfin, l'unité centrale contient divers composants lui permettant de dialoguer avec les périphériques.

Les périphériques sont variés mais permettent tous de communiquer avec l'utilisateur. On peut classifier ceux-ci en périphériques d'entrée, de sortie ou d'entrée et sortie. Ainsi, le moniteur (écran), les hauts parleurs et l'imprimante sont des périphériques de sortie, leur rôle étant d'envoyer à l'utilisateur des données (images, texte imprimé, son). Inversement, la souris et le clavier recueillent des informations auprès de l'utilisateur : ce sont des périphériques d'entrée. Le lecteur de disquette et le lecteur/graveur de CD-ROM sont des périphériques d'entrée et de sortie.

2.2 Le système d'exploitation

Un système d'exploitation est la couche située entre les applications (traitement de texte, jeux...) et l'unité centrale et ses périphériques. Les ordinateurs de la salle de TP tournent sous FreeBSD, une variante d'Unix. Vous aurez essentiellement affaire à l'interface KDE.

Une fois l'ordinateur allumé, vous devez vous identifier avant de pouvoir travailler. La machine demande un nom d'utilisateur (`login:`) et un mot de passe (`password:`). Dans la salle de TP, il y a un compte commun appelé `me` sans mot de passe.

Une fois que vous vous êtes identifié, le gestionnaire de fenêtre, KDE, se lance. Plusieurs icônes apparaissent. KDE, à l'image de *Windows*, propose une interface conviviale à l'aide de menus, de sous-menus et d'icônes pour lancer des applications.

2.3 Fichier, répertoire

Fichiers Un fichier est un suite de caractères réunis en une même entité et représentant une donnée. Cette donnée peut être par exemple une image, un son, un programme Java. L'ensemble des fichiers est stocké sur le disque dur, y compris lorsque l'ordinateur est éteint. Un fichier peut être également stocké sur une disquette, un CD-ROM...

Chaque fichier possède un nom, généralement constitué de deux parties, la *base* et l'*extension*, séparé par un point. L'extension permet de savoir de quel type de fichier il s'agit, alors que la base nous renseigne sur le contenu du fichier. Ainsi le fichier qui contient ce document s'appelle `tp00.pdf` : il contient le TP0 et est au format `pdf`. J'aurais pu lui donner n'importe quel autre nom mais il aurait alors été plus difficile de savoir de quoi il s'agissait. En général, la base et l'extension sont une suite de lettres et de chiffres (on peut utiliser d'autres caractères, mais il peuvent poser des problèmes avec certains logiciels). Notez que les systèmes Unix (comme ceux de la salle TP) font une différence entre majuscules et minuscules. Ainsi `tp00.pdf`, `TP00.pdf` et `TP00.PDF` désignent trois fichiers différents.

Voici quelques extensions fréquemment rencontrées :

Extension	Type de contenu
`.txt`	Texte simple (sans mise en forme)
`.pdf`	Texte formaté (visualisable et imprimable)
`.html`	Page web
`.java`	Programme Java
`.class`	Programme Java compilé

Répertoires Les fichiers doivent être organisés afin de pouvoir les retrouver facilement. Cette organisation est faite de façon arborescente, ce qui permet de retrouver facilement le fichier que l'on recherche. Chaque fichier prend place dans un *répertoire*. Un répertoire contient donc des fichiers. Il peut également contenir d'autres répertoires (on parle alors de *sous-répertoire*). De façon imagée, on peut voir le disque dur comme un meuble, les répertoires comme des dossiers et les fichiers comme des feuilles. Un dossier peut contenir des feuilles et aussi d'autres dossiers.

Un répertoire important est le répertoire *maison* dans lequel vous êtes au démarrage de l'ordinateur. C'est dans ce répertoire ou ses sous-répertoires que vous devez enregistrer vos données. Sous Unix, le répertoire maison est souvent abrégé en `~` (caractère tilde).

3 Les logiciels

3.1 L'interface KDE

KDE est un système multitâche et multifenêtre : on peut utiliser plusieurs logiciels en même temps, et chacun s'affiche dans sa fenêtre. Parmi les logiciels que nous utiliserons, il y a *kwrite* (éditeur de texte), *konqueror* (gestionnaire de fichiers et navigateur web), *konsole* (fenêtre *shell*), etc.

Remarque : il y a aussi des logiciels qui ne s'affichent pas dans une fenêtre. Par exemple, *kwin* est la partie de KDE qui gère, entre autres, les menus et les icônes en bas de l'écran. Il y a aussi beaucoup de logiciels qui ne sont pas directement visibles par l'utilisateur, comme *init* (le logiciel de démarrage, dont le rôle est de lancer les autres) ou *inetd* (logiciel qui ne s'occupe pas de l'utilisateur qui est devant la machine mais des connections réseau)...

3.2 Le *shell* Unix

Vous aurez besoin de taper des commandes Unix pour plusieurs raisons : lancer un éditeur de texte pour saisir vos programmes, compiler puis exécuter vos programmes, les sauvegarder sur une disquette, etc. Pour cela, lancez une fenêtre *shell* (sous KDE, cliquez sur l'icône représentant un coquillage devant un écran d'ordinateur).

Le *shell* permet d'exécuter immédiatement des commandes en tapant leur nom, et éventuellement en précisant sur quoi elles doivent agir. Lorsqu'on le lance, il affiche une *invite*, par exemple :

```
bash-2.04$
```

On peut alors taper une commande, et appuyer sur la touche « entrée » pour l'exécuter :

`bash-2.04$ ls -l` Entrée

Commandes du *shell*

Voici quelques commandes utiles et leur syntaxe :

`pwd`	affiche le chemin (nom complet) du répertoire courant
`ls -l`	afficher la liste des fichiers dans le répertoire courant
`cp` *`fichier-original nom-de-la-copie`*	faire une copie d'un fichier
`mv` *`ancien-nom nouveau-nom`*	renommer un fichier
`rm` *`fichier`*	supprimer définitivement un fichier
`mkdir` *`nom-de-répertoire`*	créer un nouveau répertoire dans le répertoire courant
`cd` *`répertoire`*	changer de répertoire courant
`cd ..`	aller au répertoire parent du répertoire courant
`rmdir` *`répertoire`*	supprimer un répertoire vide
`grep` *`mot fichier1 fichier2 ...`*	chercher un mot dans un ou plusieurs fichiers

Il y a bien sûr beaucoup d'autres commandes... (environ 5000 sur la machine où ce texte a été tapé).

Quelques aides pour taper les commandes

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 **flèches gauche et droite** 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 **flèche vers le haut**, qui rappelle la commande précédente (puis la commande d'avant, etc., si on appuie plusieurs fois).

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 Tab ou $\overleftarrow{\rightarrow}$). 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.

Interrompre une commande Une autre touche à connaître est `Ctrl+C` (contrôle-C : enfoncer la touche `Ctrl`, puis appuyer ponctuellement sur `C`, et relâcher `Ctrl`). Cette touche interrompt le *shell* ; elle annule la commande en cours, et le *shell* affiche une nouvelle invite.

3.3 Utilisation de *kwrite*

Le principal logiciel à connaître est l'éditeur de textes *kwrite*. Un éditeur de texte permet d'écrire du texte (des programmes en Java par exemple), de les enregistrer (afin de les compiler ou de les compléter ultérieurement), de les imprimer, *etc.*

Lancez *kwrite* et regardez dans le menu les commandes proposées. Les plus importantes sont dans le menu *Fichier* (*Nouveau* pour créer un fichier, *Ouvrir* pour éditer un fichier existant, *Enregistrer* pour écrire le fichier modifié sur le disque dur).

kwrite peut afficher les programmes Java en couleurs. Pour cela, utiliser l'entrée *Coloration syntaxique* dans le menu *Configuration*, puis choisir *Java*.

Un fichier contenant du code source Java doit avoir l'extension `java`, c'est-à-dire que son nom se termine par `.java`. De plus, il est d'usage que le nom du fichier soit identique au nom de la classe, et commence donc par une majuscule. Ceci donne, par exemple, un fichier appelé `Coucou.java` contenant le code source de la classe `Coucou`.

3.4 Utilisation des disquettes

Les disques des salles de TP sont régulièrement effacés. Aussi il n'est pas possible de conserver vos fichiers après la fin du TP. Si vous souhaitez garder une trace de votre travail, vous pouvez **apporter une disquette**.

Bref guide d'utilisation des disquettes (commandes du *shell*) :

`mdir a:`	affiche la liste des fichiers sur la disquette
`mcopy` *`toto`* `a:`	copie le fichier *`toto`* du disque dur vers la disquette
`mcopy a:`*`toto`* `.`	copie le fichier *`toto`* de la disquette vers le disque dur
`mcopy` *`toto`* `a:`	copie le fichier *`toto`* du disque dur vers la disquette
`mdel a:`*`toto`*	efface le fichier *`toto`* de la disquette

3.5 Courrier électronique

Une autre possibilité est d'envoyer vos fichiers par e-mail (bien sûr, ceci nécessite que vous disposiez d'une adresse électronique). Si vous passez par un webmail, utilisez le navigateur `konqueror` (icône représentant un globe terrestre dans une moitié de roue de gouvernail). Sinon, utilisez le logiciel de messagerie `kmail` (icône contenant une enveloppe et un « E »).

Le logiciel `konqueror` est aussi un gestionnaire de fichiers : il permet de voir de manipuler les fichiers stockés sur le disque de l'ordinateur.

3.6 Développement Java

L'objectif des TP sera d'écrire des programmes dans le langage Java. Une fois une version du programme conçue et écrite, voici la marche à suivre :

- Ne pas oublier d'**enregistrer** le fichier (ici `Coucou.java`) dans l'éditeur.
- **Compiler** le programme, c'est-à-dire produire du code pour la machine virtuelle Java à partir du code source. La commande est la suivante :

  ```
  javac Coucou.java
  ```
- Si le compilateur Java n'a pas signalé d'erreur, **exécuter** le programme en tapant la commande suivante :

  ```
  java Coucou
  ```

S'il y a des erreurs signalées, utiliser `kwrite` pour les corriger, et recommencer le cycle compilation-exécution. Rappel : dans le *shell*, on peut utiliser la flèche vers le haut pour rappeler les commandes précédentes.

4 Exercice : manipulations de base

1. Par la méthode de votre choix (`konqueror` ou *shell*), créez un sous-répertoire du répertoire maison à votre nom.
2. Si ce n'est pas déjà fait, lancer un *shell*.
3. Affichez le répertoire courant à l'aide de la commande `pwd`. En principe, il s'agit du répertoire maison.
4. À l'aide de la commande `cd`, placez-vous dans le répertoire créé à l'étape 1.
5. Vérifiez que vous êtes dans le bon répertoire à l'aide de la commande `pwd`. Comparez la sortie de la commande `pwd` avec l'étape 3.
6. Retournez dans le répertoire maison.
7. Placez-vous à nouveau dans le répertoire à votre nom. Combien de touches suffisent ? (sans utiliser la souris...)
8. Dans le sous-répertoire à votre nom, créez un fichier appelé `Coucou.java` et ayant le contenu suivant :

```
import fr.jussieu.script.Deug;
class Coucou {
    static void main (String[] args) {
        Deug.println("Bonjour.");
    }
}
```

9. En utilisant le *shell*, créez une copie du fichier `Coucou.java` portant le nom `AffichageDeCoucou.java`.
10. Toujours dans le *shell*, créez une troisième copie appelée `AffichageDeBonjour.java`.
11. Le fichier `Coucou.java` est un programme Java qui affiche le texte « Bonjour. ». À votre avis, comment modifier ce programme pour qu'il affiche « Coucou ! » ? Effectuez cette modification.
12. Faites un peu de ménage : effacez les fichiers `AffichageDeCoucou.java` et `AffichageDeBonjour.java`.

5 Exercices : programmes Java

5.1 Afficher « Bonjour »

Taper ce programme dans un fichier nommé `Bonjour.java`. Lancer `javac` pour le compiler, puis `java` pour l'exécuter.

```
import fr.jussieu.script.Deug;
class Coucou {
    static void main (String[] args) {
        Deug.println("Bonjour.");
    }
}
```

5.2 Puissance entière

Taper ce programme dans un fichier nommé `Puissance.java`. Lancer `javac` pour le compiler, puis `java` pour l'exécuter.

Lorsque le programme demande des nombres, les taper et appuyer sur « entrée ».

```
/* Calcul de la puissance entière d'un réel */
import fr.jussieu.script.Deug;
class Puissance {
    static float puissance (float r, int n) {
        int i;
        float p = 1;
        for (i=1; i<=n; i++) {
            p = p * r;
        }
        return p;
    }

    static void main (String[] args) {
        Deug.println("Calcul d'une puissance entière.");
        Deug.println("Entrez un nombre réel :");
        float a = Deug.readFloat();
        Deug.println("A élever à la puissance :");
        int n = Deug.readInt();
        if (n < 0) {
            Deug.println("La puissance doit être positive.");
        } else {
            Deug.println("" + a + " puissance " + n + " = " + puissance (a,n));
        }
    }
}
```

5.3 Un programme amusant

Taper ce programme dans un fichier nommé `Quine.java`. Lancer `javac` pour le compiler, puis `java` pour l'exécuter. Qu'affiche-t-il ?

```
import fr.jussieu.script.Deug; class Quine{static void main(String[]a){char q=34;
String t="import fr.jussieu.script.Deug; class Quine{static void main(String[]a){
char q=34;String t=;Deug.println(Deug.subString(t,0,90)+q+t+q+Deug.subString(t,90
,161));}}";Deug.println(Deug.subString(t,0,90)+q+t+q+Deug.subString(t,90,161));}}
```

6 Ressources

- `http://www.enseignement.polytechnique.fr/profs/informatique/Jean-Jacques.Levy/init0/a1.pdf` « L'écho du A » : ce qui se passe du moment où on appuie sur la touche « A » jusqu'au moment où il apparaît à l'écran.
- `http://www.eleves.ens.fr/tuteurs/` Unix très bien expliqué ; recommandé même si certaines parties sont spécifiques à l'ENS. Lire notamment le glossaire.
- `http://www.grbb.polymtl.ca/syst_info/cours_unix.html` Bien, mais un peu vieux.
- `http://www.kde.org/fr/index.php` Le site francophone de KDE.

En outre, Unix dispose d'une documentation en ligne assez fournie. La commande `man` affiche le manuel d'une commande, par exemple « `man ls` » L'aide en ligne de KDE est accessible par le menu « Aide ».