

**Manuel
d'utilisation
élémentaire
d'Unix sur Sun
et PC-Linux**

Juillet 2006

J.L. Nebut

MANUEL D'UTILISATION ÉLÉMENTAIRE D'UNIX SUR PC-Linux

Jean Louis Nebut
IFSIC
Cours C64

révisé juillet 2006

Contenu

1	INTRODUCTION	3
1.1	Légende	3
1.2	Glossaire	3
2	PRINCIPE DE LA CONFIGURATION	3
3	ORGANISATION ET IDENTIFICATION	4
4	CONNEXION - DÉCONNEXION	5
4.1	Connexion	5
4.2	Correction des fautes de frappes	5
4.3	Établissement, perte ou changement du mot de passe	6
4.4	Déconnexion	6
5	COMMANDES	6
5.1	Quand taper une commande	6
5.2	Et quand rien ne va plus ?	7
5.3	Syntaxe générale des commandes	7
5.4	Noms génériques	7
6	MANIPULATION DES FICHIERS	7
6.1	Liste des fichiers	7
6.2	Création d'un fichier	8
6.3	Destruction d'un fichier	8
6.4	Visualisation d'un fichier	8
6.5	Copie d'un fichier	8
6.6	Renommer un fichier	9
7	LES ÉDITEURS DE TEXTES	9
7.1	Appel de Emacs	9
7.2	Les concepts	9
7.3	Éléments des fenêtres d'Emacs	9
7.4	Déplacement dans le texte	11
7.5	Recherche et remplacement	11
7.6	Modification du texte	11
7.7	Sous-fenêtres	11
7.8	Travail avec plusieurs fichiers	12
7.9	Gestionnaire de fichiers de xemacs	12
7.10	Divers	12

8	SORTIE DE LISTING	13
9	EXEMPLE TYPIQUE DE SESSION	13
9.1	Saisie d'un programme	13
9.2	Compilation	13
9.3	Exécution	14
9.4	Exemple	14
10	DROITS D'ACCÈS	16
10.1	Modes d'accès	16
10.2	Lister les droits d'accès	16
10.3	Modifier les droits d'accès	17
10.4	Remarque	17
11	GROUPEMENT DES COMMANDES	17
11.1	Définition	17
11.2	Les fichiers de démarrage	18
11.3	Paramètres de position	18
12	LES RÉPERTOIRES	18
13	LES SERVICES INTERNET	18
13.1	Le navigateur Firefox	18
13.2	Le courrier électronique : Thunderbird	19
13.2.1	Outil Thunderbird	19
13.2.2	Outil mail/mailx	20
13.3	Votre adresse sur le réseau Internet	21
14	ÉDITEUR DE COMMANDES	21
14.1	Référence à une commande de l'historique	21
14.2	Autres facilités non liées à l'éditeur	21
15	LE SYSTÈME MULTI-FENÊTRES X-WINDOW	22
15.1	Structure d'un bureau	22
15.2	Structure d'une fenêtre	23
15.3	La souris	23
15.3.1	Sélectionner, copier	24
15.3.2	Coller	24
15.3.3	Quand le copier-coller ne marche pas	24
15.4	Les menus sur Linux avec Gnome	24
16	LE SITE WEB DE L'IFSIC	25
	Index26	

1 INTRODUCTION

Cette brochure est destinée à ce que les premiers contacts avec les serveurs SUN et leur système Unix (appelé *Solaris*) et les PC sous Linux se passent dans les meilleures conditions possibles. Malgré son volume qui n'est pas négligeable, seul un infime sous-ensemble des possibilités de Unix/Linux est abordé, celui qui permet de préparer un travail, seul ou à plusieurs, destiné à être ensuite compilé puis exécuté. Il n'est pas possible en particulier de signaler ce qui peut provoquer un fonctionnement anormal du système (on entend par là : un autre fonctionnement que celui qu'on attend) tant les possibilités acceptables par le système (mais non désirées par l'utilisateur) sont nombreuses. Heureusement, les conventions assez uniformes de Unix permettent normalement de ressortir assez rapidement des guêpiers dans lesquels on a pu tomber par inadvertance.

Pour plus de détails, on se reportera au polycopié 75, plus tard. Pour les novices qui ont besoin d'une introduction aux interfaces graphiques et aux fichiers, on se reportera au polycopié 114.

1.1 Légende

Les notations utilisées dans cet ouvrage sont les suivantes :

{ }	L'une ou l'autre des informations entre accolade
[]	Paramètre optionnel
[]*	Paramètre optionnel qui peut être répété et éventuellement omis,
[]+	Paramètre qui peut être répété mais qui doit être présent une fois
RET	Touche retour à la ligne (<i>Enter</i> ou <i>Entrée</i>)
ESC	Touche d'échappement (<i>Echap</i> , <i>Esc</i> ou <i>Escape</i>)
^X	Touche contrôle + touche x (en même temps)
C-x	Touche contrôle + touche x (en même temps), autre notation
M-x	Touche d'échappement puis touche x
COMP	Touche Compose des claviers Sun, et touche <i>menu-windows</i> des PC sous linux, à droite de la barre d'espacement.

La touche *verrouillage majuscule* (le petit cadenas à gauche) permet d'obtenir les majuscules accentuées, la touche COMP permet d'obtenir un caractère par superposition de deux caractères.

Exemples

verrouillage majuscule puis à \longrightarrow Å
COMP puis N puis ~ \longrightarrow Ñ

1.2 Glossaire

nom-de-fichier	<i>file</i>	nom d'un fichier, absolu ou relatif
répertoire, dossier	<i>directory</i>	fichier contenant des (noms de) fichiers
répertoire courant	<i>working_dir</i>	répertoire (ou dossier) sous lequel on travaille
répertoire personnel	<i>\$HOME</i>	répertoire (ou dossier) de départ, d'accueil, propre à l'utilisateur
nom-d'utilisateur	<i>\$LOGNAME</i>	nom enregistré d'un utilisateur
invite	<i>prompt</i>	message indiquant que le système attend une commande
mot de passe	<i>password</i>	mot de passe associé à chaque utilisateur

2 PRINCIPE DE LA CONFIGURATION

L'utilisateur lambda que l'on considère ici est connu du système (en fait, du réseau de l'Université) – son nom est enregistré dans un annuaire – et ne traite que des informations qu'il entre à son clavier ou qui lui sont fournies par d'autres utilisateurs via des fichiers. Le système conserve sur ses disques toutes les informations que l'utilisateur lui demandera de conserver, dans la mesure de la place disponible : l'information (les programmes, les fichiers,...) de l'utilisateur considéré sera toujours *en ligne*.

Le schéma de la figure 1 montre la configuration du réseau de l'Université pour ce qui concerne l'IFSIC (attention, les noms des serveurs ont pu changer).

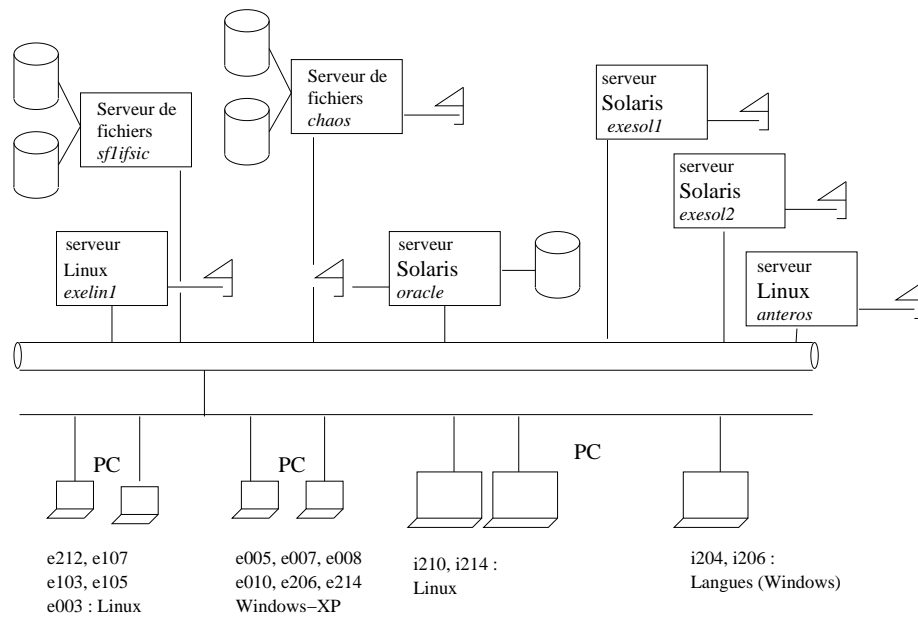


Figure 1: Les SUN et les PC dans le réseau IFSIC

3 ORGANISATION ET IDENTIFICATION

L'unité d'information gérée par le système est le fichier. Un fichier peut contenir n'importe quoi et être éventuellement vide ; selon leur utilisation, on les appelle des répertoires (*directories*) ou des fichiers tout court. Un répertoire est un catalogue de fichiers et contient les noms de ses fichiers. Les outils graphiques *gestionnaires de fichiers* appellent *dossier* un répertoire. Les fichiers en général (c'est-à-dire fichiers et répertoires) ont des attributs comme par exemple les droits d'accès pour le propriétaire, les droits d'accès pour les autres membres du groupe, la taille en caractères,...

L'ossature du système est constituée par une structure arborescente de fichiers et de répertoires. Cette structure est extensible : chaque utilisateur peut ajouter dans son coin de nouvelles branches. Le schéma de la figure 2 montre un fragment de cette ossature. Un rectangle entoure les répertoires, une ellipse, les fichiers.

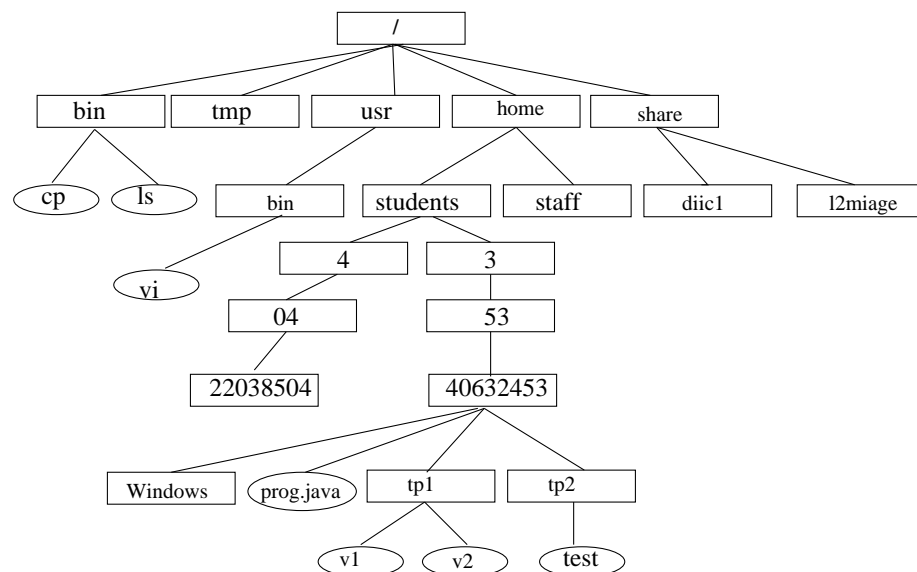


Figure 2: Le système de fichiers Unix à l'IFSIC

/ est le nom du répertoire qui est à la racine de l'arbre ; il contient les noms et caractéristiques de tous les répertoires du niveau administrateur.

Chaque fichier (et répertoire) a un nom unique dans le système : c'est son *nom-de-fichier* (*pathname*) absolu. Le nom absolu du fichier `prog.java` est `/home/students/3/53/40632453/prog.java` : il est formé de tous les noms de répertoires traversés depuis la racine pour l'atteindre, séparés par des `/`.

Le concept de *répertoire courant* (*working directory*) permet de raccourcir la notation d'un nom-de-fichier en utilisant des noms relatifs. Un nom relatif est relatif au répertoire courant. Au début de la session (lors d'une connexion), le répertoire courant est le répertoire personnel ; il est modifié par les commandes adéquates demandées par l'utilisateur (voir au paragraphe 12). Lorsque l'utilisateur Dupont (dont le répertoire personnel est connu du système sous le nom absolu `/home/students/3/53/40632453`) se connecte, il peut nommer en relatif son fichier `prog.java` par `prog.java`, son fichier `v1` par `tp1/v1` : le nom relatif a donc évité l'écriture de `/home/students/3/53/40632453`, c'est-à-dire du nom du répertoire personnel (`$HOME` ou `~`) plus `/`.

Le répertoire courant est noté `.`, et celui du dessus (son père) est noté `..`. Ainsi, si le répertoire courant est `/home/students/3/53/40632453/tp1`, une référence à `test` peut être `../tp2/test`.

Dans un premier temps, l'utilisateur travaillera dans son répertoire personnel (sur l'exemple, cela signifie que les fichiers `prog.java`, `v1`, `v2`, ... sont tous au même niveau) : ce n'est que lorsque le nombre de ses fichiers augmentera trop qu'il les classera sous d'autres répertoires.

Les répertoires qui se trouvent sous `/share` et qui ont un nom de formation contiennent les programmes et les données établis par les enseignants pour les TP.

Le répertoire personnel de chaque utilisateur est accessible depuis *Windows* sur l'unité *H* : ; l'unité *G* : donne accès au répertoire de la formation de l'utilisateur (par exemple, `/share/diic1`).

4 CONNEXION - DÉCONNEXION

4.1 Connexion


Le temps passé lorsqu'un usager est connecté s'appelle une *session*. Les modalités de connexion sont fonctions de la salle où se trouve le terminal (vrai terminal, micro-ordinateur ou station de travail) : ces modalités sont exposées dans un autre document. Notez cependant qu'à la *première* connexion sur un poste de l'Ifsic, on vous demandera peut-être de choisir la configuration du clavier : il faut demander la configuration du *serveur X* et non celle de Gnome.

La *première connexion* au réseau pour les nouveaux étudiants de l'IFSIC se fait sur une station Linux, lors d'une séance encadrée par un enseignant ; les modalités de cette connexion sont données lors de cette séance.

Sur un PC sous Linux, l'invite à se connecter est présente en fond d'écran : l'utilisateur donne alors son nom-d'utilisateur, puis un RET (*Entrée*), puis son sésame (mot de passe).

À partir d'un PC sous *Windows*, on se connecte sur un serveur Unix par la commande **telnet** *nom_du_serveur*. On n'a pas à redonner son nom, car il a déjà été donné pour se connecter au réseau de PC.

Depuis un PC sous Linux, on peut se connecter sur un serveur SUN par la commande **rlogin** *nom_du_serveur* tapé dans une fenêtre de commandes.

On ouvre un « terminal » en choisissant le sous-menu **xterm** du menu **Application-Système** ( Applications, voir figure 12 page 24). L'utilisateur est prêt à travailler en mode commande quand il reçoit l'invite du système qui consiste en le marqueur `>` précédé d'informations définies par les administrateurs. Actuellement, on y voit le nom de l'utilisateur, le nom de la station, et le nom du répertoire courant (figure 9 page 22).

4.2 Correction des fautes de frappes

On peut se tromper lors de la frappe au clavier. Tant qu'on n'a pas tapé le *retour de ligne* (qu'on notera RET par la suite), il est possible de corriger grâce à des caractères spéciaux :

<code>^H</code> (Ctrl+H ou back-space)	annule le dernier caractère frappé
<code>^U</code> (Ctrl+U)	annule tout ce qui a été frappé dans la ligne courante

Rappel : la frappe des caractères accentués non gravés sur le clavier (comme É ou Å) s’obtient en frappant successivement COMP (voir § 1.1), la lettre et l’accent (ou le O pour Å). Sous Linux, on peut aussi frapper la touche de verrouillage en majuscule puis la minuscule accentuée pour obtenir la majuscule accentuée.

Si la touche COMP ne fonctionne pas, vous devez configurer le clavier par le menu **Centre de contrôle GNOME-Clavier-Options de l’agencement-Compose key position** du menu **Bureau**, et ajouter l’outil disponible *Menu is compose*.

4.3 Établissement, perte ou changement du mot de passe

L’obtention du mot de passe se fait par le Web sur <http://portail.univ-rennes1.fr/>, onglet *Sésame*.

Un utilisateur méfiant (ou prudent) changera souvent son mot de passe, mais il faut savoir que s’il perd son mot de passe, personne ne pourra le lui retrouver. Le changement de mot de passe s’effectue de la même façon que l’acquisition ; et si l’on perd son mot de passe, c’est encore vers <http://portail.univ-rennes1.fr/> qu’il faut se tourner.

4.4 Déconnexion

Sur un PC Linux, il faut sélectionner le choix **Clore la session** du menu **Bureau**.

On quitte une connexion Unix ouverte sur un PC sous *Windows* en frappant *exit* ou *logout* après la réception de l’invite, et on se retrouve dans la session *Windows*.

On quitte une connexion Unix ouverte sur un serveur SUN par la même méthode, et on se retrouve dans la session Linux.

5 COMMANDES

Il y a deux façons de demander un service au système (on dit aussi : lui faire exécuter un programme ou une commande, ou activer un logiciel) :

1. par le biais des *commandes* en mode texte,
2. par le biais des clics de la souris en mode graphique ; dans ce cadre, il y a plusieurs façons de faire, selon que les services que l’on demande sont présentés dans des menus, par des icônes sur le plan de travail, ou sont simplement des programmes dans l’arborescence des fichiers.

Ce polycopié ne traite que la première façon, celle que doit maîtriser tout informaticien, même s’il trouve plus commode d’utiliser la deuxième, sensée être « intuitive ». Une brève introduction du mode graphique est donnée page 22¹.

5.1 Quand taper une commande

On tape les commandes en mode texte dans un outil graphique appelé fenêtre shell ou fenêtre de commandes, ou encore fenêtre « terminal » (qui simule un terminal). *xterm* et *gnome-terminal* sont de tels outils, activés à partir du menu général en choisissant les menus **Système** puis **xterm** ou **Terminal** (figure 12 page 24).

L’utilisateur est au *niveau commande* quand apparaît sur une fenêtre shell l’invite décrite au paragraphe 4.1 (voir figure 10 page 23).

L’utilisateur peut alors taper une commande, ce qui correspond à l’activation d’un programme (programme prédéfini dans le système ou programme écrit par lui-même ou un autre usager).

Quand ce programme a fini de s’exécuter, le système envoie l’invite et attend donc une autre commande.

Tant que l’invite n’est pas apparue, l’utilisateur est toujours sous le contrôle du dernier programme lancé. En particulier, ce qu’il peut taper à son terminal sera interprété par ce programme et non pas par le système (par exemple, les commandes ne seront pas reconnues) si le programme fait des lectures au clavier. Si ce n’est pas le cas, l’utilisateur fait de la frappe anticipée (il tape des commandes qui seront exécutées plus tard).

¹Voir aussi le polycopié 114

5.2 Et quand rien ne va plus ?

Si l'utilisateur veut sortir d'une commande en cours d'exécution avant d'avoir reçu l'invite (programme qui boucle indéfiniment, qui ne fait pas ce que l'on veut, etc...) il doit appuyer sur la touche d'interruption qui est la touche `~C` (touche Control + C). Le système envoie alors l'invite. Si cela ne marche pas, l'utilisateur peut détruire la fenêtre dans laquelle s'exécute la commande.

- Si après avoir tapé la commande *pas_une_commande* le système envoie le message

pas_une_commande : command not found

cela veut dire que le système n'a pas trouvé le fichier contenant le programme à exécuter.

- s'il s'agit d'une commande système, l'utilisateur n'a pas tapé le bon nom de commande → vérifier le nom de la commande et recommencer.
- s'il s'agit d'un programme utilisateur cela peut venir également d'une erreur sur le nom (vérifier en faisant *ls* : liste des fichiers (cf paragraphe 6.1)) ou, plus grave, le fichier peut avoir été détruit malencontreusement, ou ne pas avoir été créé (on a oublié de compiler par exemple) ; sous Linux, un programme utilisateur *x* se lance par *./x*.
- Si le système vous indique que la syntaxe de la commande demandée n'est pas correcte, ou si vous voulez connaître les options d'une commande, vous pouvez utiliser la commande :

man nom_commande

qui vous fournira des informations sur la commande indiquée. Lorsqu'un deux-points (:) s'affiche en bas de l'écran, on relance le défilement en tapant un espace. On peut provoquer l'affichage par page du résultat de toute commande ou programme en ajoutant | *less* derrière la commande.

5.3 Syntaxe générale des commandes

Une commande a la forme suivante² :

nom_commande [paramètre de contrôle] [paramètre]**

- le caractère séparant les différents éléments de la commande est le blanc (l'espace) ;
- les *paramètres de contrôle* sont des lettres clés indiquant au système qu'il faut utiliser telle ou telle option de la commande ; ils commencent presque tous par le caractère `-` (signe moins) ou `--` (deux signes moins) ;
- les *paramètres* sont les identificateurs (nom usager, de fichier, de répertoire, ...) des objets sur lesquels la commande va travailler.

5.4 Noms génériques

Dans un paramètre, les caractères `*` et `?` sont des caractères génériques. Le point d'interrogation remplace un caractère quelconque d'un nom de fichier du répertoire courant, et l'étoile remplace un nombre indéterminé (même nul) de caractères quelconques, sauf le point en début de nom. Par exemple :

ls tp1/v? donne les fichiers *tp1/v1* et *tp1/v2*

*ls *.java* donne les fichiers qui ont le suffixe *.java* (sources Java)

6 MANIPULATION DES FICHIERS

Ce chapitre donne quelques commandes relatives à la manipulation des fichiers (par opposition aux répertoires) quand on n'utilise pas l'outil graphique gestionnaire de fichiers.

6.1 Liste des fichiers

Les commandes suivantes donnent la liste des fichiers du répertoire courant :

*ls [nom de fichier]**

*ll [nom de fichier]**

La commande *ls* fournit la liste des fichiers, les noms étant rangés en colonnes ; une barre en fin de nom indique qu'il s'agit d'un répertoire, une étoile qu'il s'agit d'un exécutable. La commande *ll*

²rappel : les crochets indiquent que le paramètre est optionnel, on ne doit pas les taper !

est équivalente à la commande `ls -l` et fournit pour chaque fichier les renseignements décrits dans la figure 3.

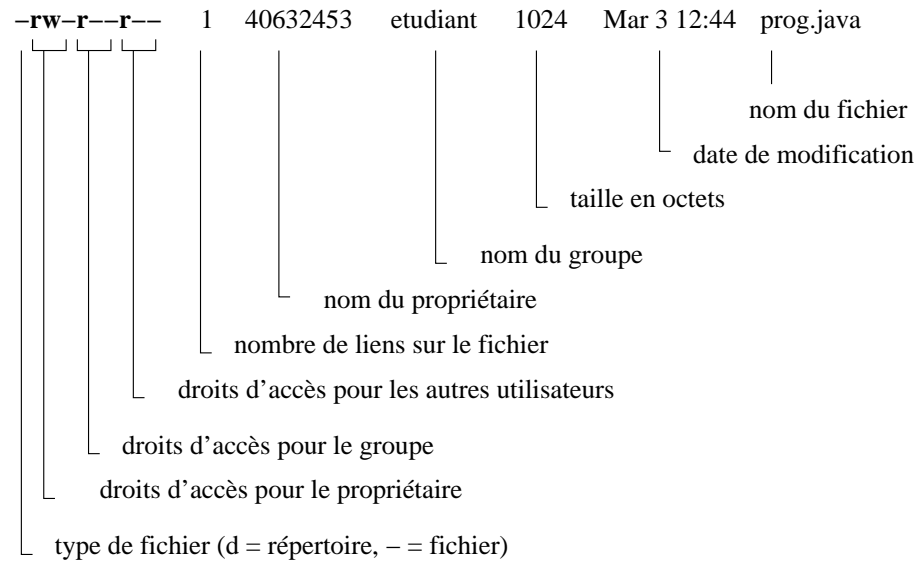


FIG. 3 – Affichage de la commande `ls -l`

6.2 Création d'un fichier

Un fichier vide peut être créé en tapant en début de ligne :

`touch nom-de-fichier`

Attention, si le fichier existe déjà, son précédent contenu est détruit.

6.3 Destruction d'un fichier

La commande suivante détruit un ou plusieurs fichiers :

`rm [nom-de-fichier] +`

La commande peut demander, pour chaque fichier, de confirmer : *y* ou *o* pour détruire, *n* pour conserver.

6.4 Visualisation d'un fichier

La commande suivante affiche totalement le ou les fichiers spécifiés :

`cat [nom-de-fichier] +`

Si le fichier ne contient pas du texte, cette commande est inutilisable. Si la ligne est trop longue pour le terminal, elle se prolonge sur la ligne suivante.

La commande suivante affiche page par page le ou les fichiers :

`less [nom-de-fichier] +`

RET fait passer à la ligne suivante, un espace fait passer à la page suivante, et *q* termine l'affichage.

6.5 Copie d'un fichier

La commande suivante copie le premier fichier nommé dans le second :

`cp nom-de-fichier nom-de-fichier`

Une variante de cette commande permet de copier plusieurs fichiers dans un répertoire donné :

`cp [nom-de-fichier]* repertoire`

6.6 Renommer un fichier

Il est possible de changer le nom ou de déplacer un fichier grâce à la commande :

```
mv ancien-nom-de-fichier nouveau-nom-de-fichier
```

7 LES ÉDITEURS DE TEXTES

L'éditeur de texte est un logiciel de composition de texte ; un texte est composé à partir d'autres textes situés dans des fichiers ou est entré au clavier. L'éditeur de texte ne s'occupe pas de l'édition du texte sur imprimante : voir au paragraphe 8.

Unix comprend plusieurs éditeurs de textes. Celui qui est présenté ici, *Emacs*, a l'avantage de ne pas être un produit typé Unix, et on le trouve également sous *Windows*. Ainsi, vous utiliserez à l'IFSIC *Emacs* quelque soit le type de machine et de système.

Cet éditeur est présent dans deux versions différentes :

1. Emacs version 21.3, qui s'ouvre dans la fenêtre d'appel ;
2. xemacs, qui possède une barre d'icônes et qui s'ouvre dans une nouvelle fenêtre.

7.1 Appel de Emacs

```
Emacs 21    emacs [nom-de-fichier]
xemacs      xemacs [nom-de-fichier] &
```

Comme l'indiquent les méta-crochets, le nom de fichier est facultatif.

Le `&` est nécessaire dans l'appel de **xemacs** si l'on veut conserver l'usage de la fenêtre de commandes. On peut aussi appeler Emacs depuis le menu **Application-Utilitaires-Éditeur**, mais il est préférable (pour les sauvegardes de fichier) de l'appeler depuis le répertoire courant en tapant la commande **xemacs** avec le `&`.

7.2 Les concepts

Emacs permet principalement d'éditer des fichiers. Pour chaque type de fichier (`.e`, `.c`,...), Emacs se met automatiquement dans un *mode* particulier adapté à l'édition de ce type de fichier. Chaque mode redéfinit certaines touches et éventuellement les entrées de la barre de menus. Mais Emacs peut servir à beaucoup d'autres choses : compiler un programme, naviguer dans ses fichiers, consulter des manuels... Pour cela il génère lui-même des fenêtres de texte non éditables (gestionnaire de fichiers, liste des fichiers chargés, résultat d'une compilation, aide, manuels,...) où les touches ne servent plus à rentrer du texte, mais servent à lancer des commandes spécifiques dépendant de ce mode.

Une zone de mémoire où se trouve un texte (contenu d'un fichier ou texte généré par Emacs lui-même) s'appelle un tampon (*buffer*). Les tampons générés par Emacs lui-même ont un nom qui commence et finit par `*` (par exemple `*scratch*` est le tampon visible par défaut quand on lance Emacs sans nom de fichier).

Il est possible d'ouvrir plusieurs fenêtres (appelées *frames*) sur le même tampon ou sur des tampons différents. Elles se manipulent comme des fenêtres ordinaires. Une fenêtre Emacs peut aussi à son tour être subdivisée en sous fenêtres appelées *windows*, qui sont par contre gérées par Emacs et se manipulent différemment.

L'édition (ajout ou destruction de caractères) se fait à l'endroit où se trouve le curseur (appelé *point*). Les opérations de bloc (appelé *region*) se font sur le bloc courant repéré dans Emacs et xemacs par une zone bleutée.

Tout caractère imprimable est inséré devant le curseur. La grande flèche arrière du clavier **Backspace** détruit le caractère à gauche du curseur, le `^D` détruit le caractère sous le curseur.

7.3 Éléments des fenêtres d'Emacs

La figure 4 montre les éléments constitutifs de la fenêtre principale de **xemacs**. Emacs a la même forme mais n'a pas de barre d'icônes ni d'ascenseur.

Une fenêtre (*frame*) se compose de plusieurs éléments :

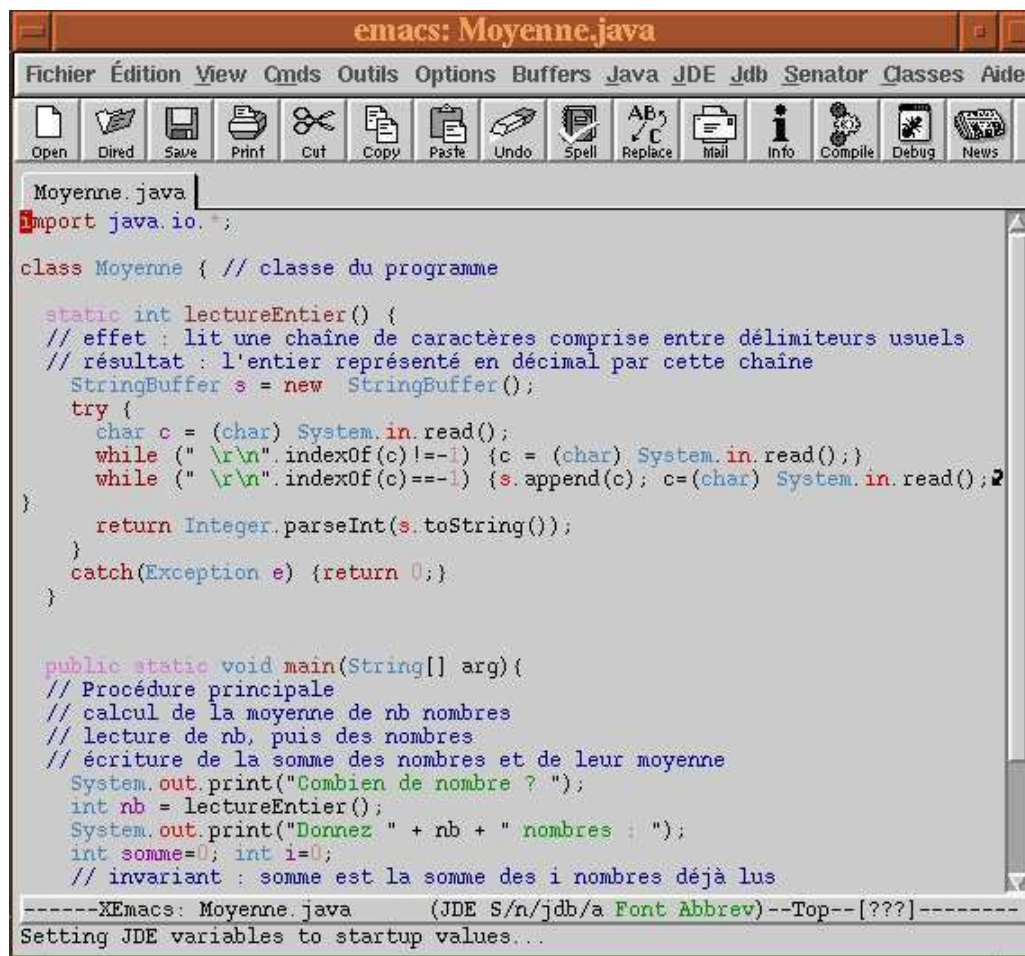


FIG. 4 – Fenêtre principale de xemacs

- une barre de menu en haut. Le menu peut changer selon le mode et offrir des fonctions spécifiques ;
- une ou plusieurs sous-fenêtres (*windows*) affichant chacune une vue d'un tampon (ici, une seule sous-fenêtre) ;
- chaque sous-fenêtre a en bas une *ligne d'état* (*modeline*) qui indique si le tampon a été modifié (**), le nom du tampon ou du fichier (*Moyenne.java*), le mode principal (JDE) et les modes mineurs (Fill). Emacs indique aussi le numéro de ligne où se trouve le curseur, et la position dans le fichier (All ou Top ou 60%). La barre peut changer selon le mode et indiquer éventuellement d'autres renseignements.
- tout en bas, un tampon spécial d'une ligne (appelé *mini-buffer*) sert aux interactions avec l'utilisateur. Cette ligne indique le chargement du mode *JDE* pour Java sur la figure 4. Il faut toujours bien lire ce qui se trouve dans le mini-buffer. Des informations importantes y sont souvent rappelées. De même, lorsqu'une commande a été lancée, Emacs y attend les éventuels arguments. Si vous cliquez ailleurs, cela n'annule pas la commande et le mini-buffer attend toujours, ce qui peut parfois perturber les nouvelles commandes que vous lancez. Dans ce cas, il faut cliquer avec le bouton gauche dans le mini-buffer pour y positionner le curseur et taper *C-g*. Répéter l'opération jusqu'à ce qu'apparaisse le message *Minibuffer window is not active* ou *Quit*.

Par la suite, on donne les commandes d'édition qu'on entre au clavier (ce sont les seules possibilités avec micro-emacs), sachant qu'on peut utiliser les barres de menu ou les icônes bien évidemment.

7.4 Déplacement dans le texte

On peut à tout instant se déplacer d'une position dans le texte avec les flèches ou avec les séquences suivantes ; si on donne ces détails, c'est qu'on peut faire de l'édition de commandes Unix dans le Shell avec ces requêtes.

^B (*backward*) caractère précédent

^F (*forward*) caractère suivant

^P (*previous*) ligne précédente

^N (*next*) ligne suivante

M-f (grand *forward*) mot suivant

M-b (grand *backward*) mot précédent

^A début de ligne

^E fin de ligne

^V page suivante

M-v page précédente (les touches de changement de page du clavier fonctionnent)

M-< début du texte

M-> fin du texte

M-g aller à la ligne *n*, *n* demandé interactivement dans le *mini-buffer*

7.5 Recherche et remplacement

La recherche d'une chaîne commence à la position du curseur.

^S Recherche une chaîne donnée, entre le curseur et la fin du texte. Terminer la chaîne donnée par RET. La recherche est incrémentale

^R Même recherche, en arrière, vers le début du texte.

M-% Remplace toutes les occurrences d'une chaîne par une autre.

Le menu *Replace* de **xemacs** propose la recherche-remplacement avec confirmation sur chaque chaîne trouvée ; on confirme par y ou par un espace ; si on veut tout remplacer, on confirme par !.

7.6 Modification du texte

^O « Ouvre » une ligne (renvoie ligne suivante ce qui suit le curseur)

M-d Détruit entre le curseur et la fin du mot

^K Détruit entre le curseur et la fin de ligne. Supprime la ligne si elle est vide. Suppression d'une ligne non vide : **^A^K^K**

^_ ou **^X u** Défaire ; ce n'est pas une simple bascule.

Pour faire un copier-coller, on sélectionne le texte à la souris ou en posant des marques. On appelle « marque » un point particulier (unique) du texte, situé entre 2 caractères.

On appelle région le texte compris entre la marque et le point à gauche du curseur (ou l'inverse si le curseur est avant la marque) ou entre deux marques. Avec Emacs :

clic à gauche Pose d'une marque en début de région

clic à droite Pose d'une marque en fin de région (qui change alors de couleur)

Dans tous les cas :

^W (*wipe*) couper : détruit le texte compris dans la région

^Y (*yank*) coller : insère le texte qui vient d'être copier ou détruit

7.7 Sous-fenêtres

La fenêtre principale peut être partagée en sous-fenêtres permettant de voir simultanément plusieurs parties d'un texte.

^X 2 Partage l'écran (ou la fenêtre actuelle) en deux (2), avec le même texte

^X 1 Ne laisse à l'écran qu'une (1) fenêtre, celle du curseur.

^X 0 Supprime la fenêtre où se trouve le curseur.

7.8 Travail avec plusieurs fichiers

On peut travailler sur plusieurs textes à la fois, en créant autant de tampons. Un tampon a (en général) le nom d'un fichier de texte, qui sert à l'initialiser, et dans lequel le tampon est sauvegardé par $\wedge X \wedge S$.

$\wedge X \wedge I$ (*insert*) Insère un fichier de texte à gauche du curseur

$\wedge X \wedge F$ (*find file*) Crée un nouveau tampon de nom donné, l'initialise avec le fichier de même nom s'il existe. On peut utiliser la touche TAB comme en Shell pour compléter un nom de fichier dont on n'a tapé que le début. Un deuxième TAB s'il y a ambiguïté crée une sous-fenêtre appelée **Completions** et propose tous les noms de fichiers qui conviennent. On sélectionne avec un clic au milieu sur le nom choisi.

$\wedge X \wedge b$ Le tampon de nom donné devient tampon courant. S'il n'existe pas, il est créé, sans nom de fichier associé (utilisable comme mémoire de manœuvre).

$\wedge X \wedge x$ Passe au tampon suivant, sous micro-emacs (évite de les nommer s'il y en a peu).

$\wedge X \wedge B$ Affiche la liste des tampons. Une astérisque indique les tampons non sauvegardés. Faire $\wedge X \wedge 1$ pour effacer la liste.

$\wedge X \wedge k$ (*kill*) Détruit un tampon (ne doit pas être tampon courant).

$\wedge X \wedge S$ (*save*) Sauvegarde le tampon courant dans son fichier associé

$\wedge X \wedge W$ (*write*) Sauvegarde le tampon courant dans un fichier de nom donné

$\wedge X \wedge R$ (*read*) Initialise le tampon courant par un fichier de nom donné.

7.9 Gestionnaire de fichiers de xemacs

le gestionnaire de fichier de la figure 5 s'ouvre lorsqu'on tape la commande *File-Open...* ou *File-Save as...* et se comporte comme on l'a expliqué pour la commande *C-x C-f* : il affiche les sous-répertoires et les fichiers du répertoire courant. On se déplace dans la hiérarchie de fichiers et/ou on sélectionne un fichier en sélectionnant un nom avec un clic *au milieu* (clic du bouton du milieu de la souris).

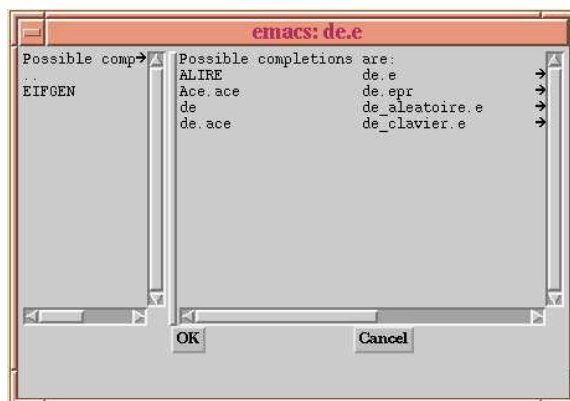


FIG. 5 – Gestionnaire de fichiers de xemacs

7.10 Divers

$\wedge X \wedge S$ sauvegarde le texte dans le fichier associé (utiliser $\wedge X \wedge W$ si l'on veut préciser le nom du fichier)

$\wedge X \wedge C$ quitte l'éditeur. Avant de quitter ou de détruire un tampon non sauvegardé, le système demande : *On quitte malgré tout [o/n] ?* On répond *o* pour ne pas tenir compte des modifications, et *n* pour ne pas détruire le tampon (et avoir la possibilité de le sauvegarder).

M-n ... répète n fois ce qui suit (caractère ou commande)

^G permet d'interrompre toute commande en cours (plusieurs **C-G** de suite peuvent être nécessaires)
^L réinitialise l'écran (en cas de parasites !)
^Q introduit le caractère non imprimable suivant (**^.**, **ESC**,...) dans le texte

8 SORTIE DE LISTING

La sortie d'un listing est faite sur l'imprimante qui se trouve à côté de votre salle de TP. Toutefois, le mode d'emploi peut varier dans le temps, et les commandes qui suivent sont sujettes à modification. Voir à ce sujet les indications qu'on vous donne en début d'année, et surtout les pages du serveur Internet de l'IFSIC.

Trois commandes sont disponibles :

lpr nom-de-fichier envoie le fichier sur l'imprimante locale

lpq donne l'état de la file d'impression : chaque listing en attente y est repéré par un numéro

lprm numéro retire de la file d'impression le listing en attente dont on a fourni le numéro

9 EXEMPLE TYPIQUE DE SESSION

Dans ce paragraphe est succinctement décrit le déroulement de la session d'un usager désireux d'effectuer le classique travail suivant :

saisie d'un programme - compilation - exécution.

On utilise pour cela au moins deux fenêtres :

- une pour faire l'édition du programme,
- l'autre pour faire les compilation et les exécutions.

9.1 Saisie d'un programme

Après s'être connecté, l'utilisateur fait appel au programme d'édition de texte de Unix en lançant l'Emacs de son choix pour effectuer la saisie du texte source de son programme. Le nom choisi pour ce fichier doit impérativement être suffixé par le nom du langage de programmation utilisé (**.java** pour Java). Il s'efforce au cours ou au terme de cette saisie d'obtenir une version « propre » de son programme source, et ce en utilisant les diverses requêtes permettant d'agir sur le tampon mis à sa disposition par l'éditeur. S'estimant satisfait, l'utilisateur sauve le texte, contenu dans le tampon, dans un fichier (par **C-x C-s** ou par les menus) ; si le tampon n'est associé à aucun fichier, on utilise **C-x C-w** (*write*) et le nom du fichier.

Le langage de programmation peut aussi imposer le nom du fichier : par exemple, le compilateur Java cherche la classe *Moyenne* dans un fichier de nom *Moyenne.java*. L'utilisateur s'est ainsi créé un fichier contenant son programme source, fichier dont il peut vérifier l'existence en consultant la liste des fichiers de son répertoire (commande **ls**).

9.2 Compilation

Dans une deuxième fenêtre de type shell, positionnée sur le *répertoire où est sauve le fichier source*, l'utilisateur lance la compilation de son programme.

La syntaxe de la commande de compilation Java est la suivante :

javac Moyenne.java

En cas d'erreurs, bien lire tous les messages, corriger sous emacs, sauve et recommencer la compilation. S'il n'y a pas d'erreurs, la compilation aura produit le fichier exécutable par Java qui s'appelle

Moyenne.class

9.3 Exécution

L'utilisateur est alors en mesure de lancer l'exécution du programme. On ne considère ici que des programmes fonctionnant en mode texte (par opposition au mode graphique avec souris). L'activation d'un programme se fait en l'invoquant sur la ligne de commande, en général dans la fenêtre qui a servi à faire la compilation.

Le plus souvent, un programme utilise des données et fournit des résultats. Pour ce qui est des données, deux cas sont à considérer :

1. l'utilisateur peut transmettre les données directement à partir de sa console, à la demande du programme. Dans ce cas, l'exécution est lancée par la commande :

java nom-du-programme

soit, dans l'exemple du programme *Moyenne.java*, la commande **java Moyenne**.

2. le programme peut aller chercher ses données dans un fichier, fichier que l'utilisateur aura alors, bien sûr, préalablement créé. Dans ce cas, il est nécessaire, lors de l'activation du programme, de spécifier le nom du fichier contenant les données en utilisant la syntaxe suivante :

java nom-du-programme < nom-de-fichier

Le signe « < » prévient le système que les données sont à prendre dans *nom-de-fichier*.

Pour ce qui concerne les résultats, cela fonctionne à peu près comme pour les données. Ainsi, la syntaxe :

java nom_programme > nom-de-fichier

indique au système que les résultats doivent être rangés dans le fichier *nom-de-fichier*.

9.4 Exemple

*Connexion de l'utilisateur Dupont qui
fournit son nom d'utilisateur (dupont)
et son mot de passe*

login : *nom-d'utilisateur*
mot-de-passe

*Création de deux fenêtres de commandes
par **Applications-Système-xterm***

*Construction du programme
source avec l'éditeur Emacs lancé dans une de ces fenêtres*

i214m08[1] > *xemacs Moyenne.java &*

```
import java.io.*;
```

```
class Moyenne { // classe du programme
```

```
    static int lectureEntier() {  
        // effet : lit une chaîne de caractères comprise entre délimiteurs usuels  
        // résultat : l'entier représenté en décimal par cette chaîne  
        StringBuffer s = new StringBuffer();  
        try {  
            char c = (char) System.in.read();  
            while (" \r\n".indexOf(c) != -1) {c = (char) System.in.read();}  
            while (" \r\n".indexOf(c) == -1) {s.append(c); c=(char) System.in.read();}  
            return Integer.parseInt(s.toString());  
        }  
    }
```



```

        catch(Exception e) { return 0;}
    }

    public static void main(String[] arg){
        // Procédure principale
        // calcul de la moyenne de nb nombres
        // lecture de nb, puis des nombres
        // écriture de la somme des nombres et de leur moyenne
        System.out.print("Combien de nombre? ");
        int nb = lectureEntier();
        System.out.print("Donnez " + nb + " nombres : ");
        int somme = 0; int i = 0;
        // invariant : somme est la somme des i nombres déjà lus
        while(i<nb){
            somme = somme + lectureEntier();
            i++;
        }
        System.out.println("\nLe total est : " + somme);
        System.out.println("la moyenne est : " + ((double) somme)/nb);
    }
}

```

*Copie du source dans le fichier
Moyenne.java (sauvegarde)*

C-x C-s

Appel du compilateur dans l'autre fenêtre

i214m08[1] > javac Moyenne.java

*Exécution du programme avec saisie
des données au clavier et affichage
des résultats sur console*

```

i214m08[2] > java Moyenne
Combien de nombres? 5
Donnez 5 nombres : 2 3 5 7 1
Le total est : 18
La moyenne est : 3.6

```

*Construction d'un fichier de
données de nom « donnees » dans un
nouveau tampon de Emacs*

```

C-x C-f donnees
5
2 3 5 7 1
C-x C-s

```

*Exécution du programme avec saisie
des données dans le fichier « donnees »
et affichage des résultats*

```

i214m08[3] > java Moyenne < donnees
Combien de nombres?
Donnez 5 nombres
Le total est : 18
La moyenne est : 3.6

```

*Exécution du programme avec saisie
des données au clavier et stockage
des résultats dans le fichier « resultat »*

```
i214m08[4] > java Moyenne < donnees > resultat  
5  
2 3 5 7 1
```

*Visualisation du résultat de
l'exécution précédente*

```
i214m08[5] > cat resultat  
Combien de nombres ?  
Donnez 5 nombres  
Le total est : 18  
La moyenne est : 3.6
```

Destruction des fichiers utilisés

```
i214m08[6] > rm donnees resultat
```

Déconnexion

Menu de fond d'écran

10 DROITS D'ACCÈS

Ce chapitre donne quelques indications sur les possibilités qu'a un utilisateur pour autoriser ou interdire l'accès de ses fichiers à d'autres utilisateurs.

10.1 Modes d'accès

Il y a trois types d'accès à un fichier noté par une lettre :

- pour le lire noté *r* (*read*)
- pour écrire dedans noté *w* (*write*)
- pour l'exécuter noté *x* (*execute*).

Pour accéder à un fichier, les utilisateurs sont scindés en trois classes :

- le propriétaire du fichier (*User*),
- les autres membres du groupe (*Group*),
- les autres utilisateurs (*Other*).

En combinant les types d'accès et les classes d'utilisateurs, neuf droits d'accès différents existent donc dans Unix.

Malheureusement, l'Université n'a classé les utilisateurs que dans deux groupes :

- le groupe *etudiant* pour *tous* les étudiants de l'Université ;
- le groupe *staff* pour les enseignants, techniciens et administratifs.

10.2 Lister les droits d'accès

Les droits d'accès pour un fichier ou pour un répertoire sont visualisés grâce à la commande **ls -l** ou **ll** (cf § 6.1 page 7). Ils sont constitués de neuf caractères regroupés en trois ensembles de trois caractères :

- le premier ensemble indique les droits d'accès du propriétaire,
- le suivant indique les droits d'accès des autres membres du groupe,
- le dernier indique les droits d'accès des autres utilisateurs.

Chaque ensemble peut contenir les lettres suivantes :

- r* si le fichier ou le répertoire est accessible en lecture
- w* si le fichier ou le répertoire est accessible en écriture
- x* si le fichier peut être exécuté, si le répertoire peut être traversé
- si le droit est interdit

10.3 Modifier les droits d'accès

Les droits d'accès peuvent être modifiés grâce à la commande **chmod** dont la syntaxe est la suivante :

chmod mode [nom-de-fichier]*

mode indique de quelle façon les droits d'accès doivent être modifiés. Il se décompose en :

[qui] [op accès]+

qui indique quelle(s) classe(s) d'utilisateurs est (sont) concernée(s) par cette commande en spécifiant une ou plusieurs lettres parmi les suivantes :

u utilisateur

g groupe

o autres utilisateurs

Toutes les classes sont concernées si on n'en spécifie aucune.

op peut être :

+ s'il faut ajouter des droits d'accès

- s'il faut enlever des droits d'accès

= si on définit de nouveaux droits d'accès

accès est une combinaison quelconque des lettres suivantes :

r accès en lecture,

w accès en écriture,

x accès en exécution ou en traversée.

Exemple :

Retrait pour les autres utilisateurs de tous les droits d'accès :

chmod o-rwx moyenne*

Ajout de droits d'exécution sur un fichier pour toutes les classes d'utilisateurs :

chmod +x moyenne

10.4 Remarque

Depuis la rentrée 2004, les droits sur votre répertoire personnel sont les suivants :

drwx-----

Cela signifie que vous seul avez accès à vos fichiers : votre binôme ou vos enseignants ne peuvent pas y accéder. Pour leur permettre néanmoins un accès, tout en préservant votre intimité, mettez le droit *x* pour tous (le groupe pour votre binôme, les autres pour vos professeurs) par la commande :

cd ; chmod +x .

11 GROUPEMENT DES COMMANDES

11.1 Définition

Il est possible dans Unix d'exécuter une série de commandes contenues dans un fichier. On peut lancer l'exécution de deux manières différentes :

./nom-de-fichier En spécifiant le nom de fichier en tant que commande, cet enchaînement de commande sera exécuté. Le droit d'accès en exécution (*x*) doit exister pour *nom-de-fichier*. Sur les serveurs SUN, le *./* n'est pas nécessaire.

bash [options] nom-de-fichier Appel de la commande **bash** (shell) avec comme paramètres une ou plusieurs options d'exécution et le nom du fichier contenant les commandes. Voici les deux options les plus courantes :

-v écrire les commandes au fur et à mesure de leur exécution telles qu'elles sont lues dans le fichier,

-x écrire les commandes au fur et à mesure de leur exécution telles qu'elles sont exécutées.

11.2 Les fichiers de démarrage

Au moment de la connexion, il y a exécution automatique de plusieurs fichiers :

- le fichier `/etc/profile` du système qui contient la mise en place de l’environnement de votre processus initial sous Linux ;
- le fichier `/etc/bashrc` du système qui contient les abréviations communes pour tous les étudiants (alias) sous Linux ;
- le fichier `.bashrc` du répertoire personnel, s’il existe, qui contient vos commandes initiales personnelles et vos alias sous Linux ;
- le fichier `/etc/csh.login` du système qui contient la mise en place de l’environnement de votre processus initial sur SUN ;
- le fichier `/etc/csh.cshrc` du système qui contient les abréviations communes pour tous les étudiants (alias) sur SUN ;
- le fichier `.tcshrc` du répertoire personnel, s’il existe, qui contient vos abréviations personnelles sur SUN.

11.3 Paramètres de position

On peut paramétrer un fichier de commandes.

Les paramètres effectifs sont transmis de la même façon que les paramètres d’une commande, sous la forme d’une liste d’arguments, dans la commande d’appel :

```
nom-de-fichier liste-d’arguments
bash [option] nom-de-fichier liste-d’arguments
```

Les divers arguments de *liste-d’arguments* sont séparés par un ou plusieurs blancs.

Les paramètres formels présents dans le fichier de commandes sont notés `$i` où `i` est le numéro positionnel du ième paramètre. Ainsi, `$i` fait référence au ième argument de liste-d’arguments.

12 LES RÉPERTOIRES

`pwd` donne le nom du répertoire courant.

`cd [nom-de-répertoire]` effectue un changement de répertoire courant : celui-ci devient le répertoire nommé. Sans paramètre, le répertoire courant revient au répertoire personnel.

`mkdir identificateur` crée un nouveau répertoire sous le répertoire courant.

`rmdir nom-de-répertoire` détruit le répertoire identifié par nom-de-fichier. Ce répertoire doit être vide.

`ls` fournit la liste de tous les fichiers et des répertoires

13 LES SERVICES INTERNET

L’Université met à la disposition de chaque usager une boîte à lettres destinée à stocker les messages que lui envoient d’autres usagers. Cette boîte est un fichier géré par une machine Linux dédiée au courrier au niveau de l’Université. En particulier, ce fichier, pourtant personnel à chaque étudiant, n’est pas dans l’ensemble des fichiers accessibles depuis le réseau de l’IFSIC. Il n’est accessible que par un outil adéquat qui donne accès également à certains forums (les *News* et les *newsgroups*).

13.1 Le navigateur Firefox

On lance le navigateur en tapant la commande

```
firefox &
```

ou en cliquant sur la petite mappemonde du bandeau des menus (voir 11 page 24). Si le navigateur est bien configuré, sa fenêtre s’ouvre, présentant la page d’accueil de l’IFSIC de la figure 6.

Il est possible que le navigateur soit activé automatiquement lors d’une connexion : cela dépend des fichiers de démarrage.




FIG. 6 – Page d'accueil du navigateur sur le Web

13.2 Le courrier électronique : Thunderbird

On peut lire/envoyer du courrier électronique (ou mél) sous *Windows* ou *Unix/X-Window*, par l'outil Thunderbird; sous *Unix* seulement, on peut envoyer du courrier par la commande **mailx** sous *Solaris*, **mail** sous *Linux*.

13.2.1 Outil Thunderbird

La fenêtre du courrier électronique s'obtient entre autre par le menu **Application-Internet-Thunderbird** ce qui ouvre la fenêtre de la figure 7 si votre configuration a été bien engendrée. Il n'y a plus qu'à se laisser guider par les menus et les boutons.

Pour utiliser correctement le mél, il faut avoir conscience de la localisation des fichiers contenant les courriers et détruire au fur et à mesure les méls sans intérêt ou gros. La figure 8 montre que les courriers arrivent en vrac sur le serveur de l'Université. L'opération de lecture du courrier sous *mozilla* est déclenchée automatiquement au lancement du logiciel, et on peut la demander en cliquant sur l'icône  sous le menu **Fichier** (opération **Relever les nouveaux messages**). Elle amène les méls (ou courriels) de l'utilisateur Dupont dans la boîte à lettre de Dupont.

Si le courrier doit être conservé, il faut l'amener dans un classeur de courrier qu'on crée via le menu **Nouveau-Dossier...** (ou **Nouveau sous-dossier...** selon le niveau) du menu contextuel obtenu par un clic droit sur un classeur existant.

Bien entendu, il est raisonnable de prévoir un classement du courrier conservé localement, en créant une structure arborescente de dossiers de courriers analogue à l'arborescence des répertoires du système de fichiers.

Sur la figure 7, on a créé un dossier de nom *12miage* par un clic droit quand *Dossiers Locaux* était sélectionné, sous-menu **New folder...**

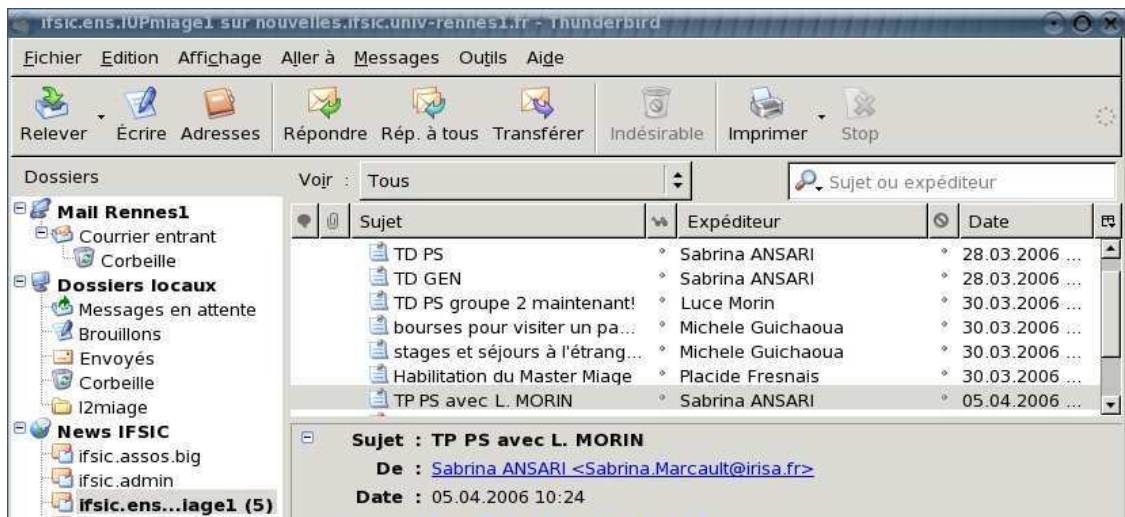


FIG. 7 – Le bandeau du courrier de *Thunderbird*

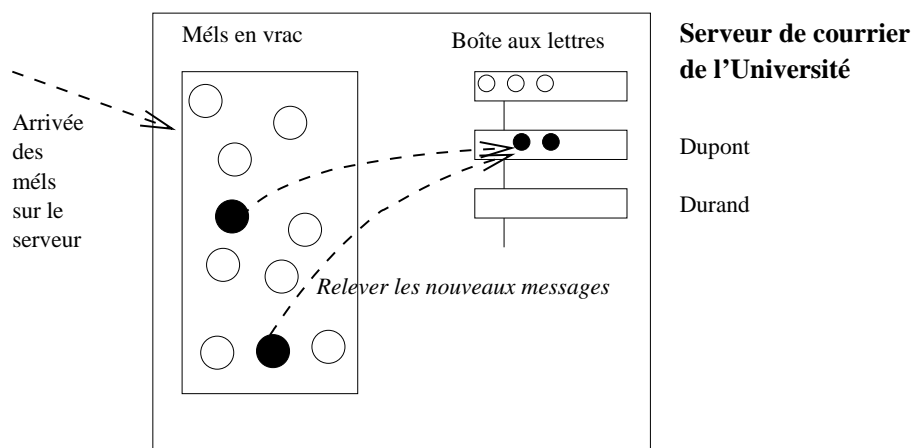


FIG. 8 – Les fichiers impliqués par le courrier

Si le courrier ne doit pas être conservé, alors on le déplace vers la poubelle pour libérer l'espace qu'il occupe sur le serveur de l'Université.

13.2.2 Outil mail/mailx

Envoi de courrier Un usager peut envoyer une lettre à un autre usager sans utiliser **mozilla** par la commande **mail** :

```
mail [nom-d'utilisateur]+
```

Après cette commande, l'utilisateur tape au clavier la lettre à envoyer. Cette dernière doit se terminer par une ligne contenant un point en début de ligne ou par une fin de fichier (^D). Comme c'est une commande Unix comme une autre, on peut utiliser les mécanismes utilisés page 2 : par exemple, pour transmettre le fichier *Moyenne.java* par courrier à l'utilisateur Marie Durand, Dupont tapera

```
mail Marie.Durand@etudiant.univ-rennes1.fr < Moyenne.java
```

Lecture du courrier La structure actuelle du système du courrier ne permet pas d'utiliser la commande **mail** pour lire le courrier.

13.3 Votre adresse sur le réseau Internet

Vos correspondants extérieurs à l'IFSIC peuvent vous envoyer du courrier, et vous pouvez leur en envoyer également, qu'ils étudient en France ou à l'étranger. L'adresse de l'utilisateur Dupont pour un autre étudiant de l'IFSIC est 40632453 ou aussi *Jean-Daniel.Dupont* s'il n'a pas d'homonyme à l'Université. Consultez votre ENT sur <http://portail.univ-rennes1.fr/> pour modifier votre adresse.

Pour l'extérieur, il est connu comme un étudiant de l'université de Rennes, avec un nom comme *etudiant.numéro-etudiant@etudiant.univ-rennes1.fr* ou *prénom.nom@etudiant.univ-rennes1.fr*, soit *Jean-Daniel.Dupont@etudiant.univ-rennes1.fr* : c'est cette adresse qui est reçue par les correspondants extérieurs. Les deux adresses désignent cependant la même boîte aux lettres.

14 ÉDITEUR DE COMMANDES

Le langage de commande utilisé sur PC Linux s'appelle le *bash* ; le langage de commande utilisé sur les serveurs SUN s'appelle le *T-Shell*. Le langage de commande gère un tampon des commandes entrées au clavier, et un éditeur de commandes permet d'engendrer de nouvelles commandes à partir de plus anciennes ou de fragments d'anciennes commandes.

La commande **history** affiche les commandes du tampon. Chaque commande est repérée par un numéro dans le tampon : ce dernier est appelé l'historique des commandes.

Toute référence à l'historique commence par un point d'exclamation!. Après analyse d'une commande comportant un!, le shell affiche la commande engendrée.

14.1 Référence à une commande de l'historique

!<num> rappelle la commande du numéro *num* ;
!! rappelle la dernière commande tapée (= !-1) ;
!x rappelle la dernière commande commençant par la chaîne *x* ;

La frappe seule d'une de ces références a pour effet de reproduire la commande visée, qui est ensuite soumise au shell pour analyse et exécution.

Ainsi, il suffit de taper **!!** pour relancer la dernière commande, et **!e** pour relancer le dernier appel à **emacs**, à condition qu'aucune autre commande commençant par **e** n'ait été tapée depuis.

On peut également utiliser les flèches pour remonter dans l'historique et corriger une ancienne commande avant de la relancer.

14.2 Autres facilités non liées à l'éditeur

Les accolades peuvent servir à factoriser une liste de nom de fichiers : ainsi, **xx{a,b,c}yy** est une abréviation pour la liste **xxayy xxbyy xxcy**.

Le caractère **~**(tilde) placé en début de nom de fichier est une abréviation pour le nom du répertoire personnel. Placé devant un nom d'utilisateur, il est une abréviation du répertoire personnel de cet utilisateur.

Exemple :

```
i214m08[25]> echo ~22038504/tp{3,25}.java  
/home/students/4/04/22038504/tp3.java /home/students/4/04/22038504/tp25.java
```

Lorsqu'un nom de fichier est introduit dans une commande, il est possible de n'en frapper que les premiers caractères, suivis d'un TAB.

Lorsque l'on tape une commande comportant des noms de fichiers avec répertoires, il est possible de suspendre momentanément la frappe du nom et obtenir « pour voir » la liste des fichiers du sous-répertoire correspondant au fragment de nom entré : il suffit pour cela de taper deux TAB (tabulation) sous Linux (convention du *bash*), ou un **^D** sur SUN (convention de T-Shell) après une barre (/). Si l'on s'arrête sur **/xx**, la frappe d'un seul TAB (Linux) ou d'un **^D** (SUN) donne la liste des fichiers du (sous) répertoire commençant par **xx**, sans pour autant que la commande en cours de frappe soit perdue.

15 LE SYSTÈME MULTI-FENÊTRES X-WINDOW

Lors d'une connexion, le logiciel appelé serveur X est installé : il permet de placer sur l'écran plusieurs fenêtres ; dans chacune d'elles il est possible de faire fonctionner une application standard (l'éditeur de texte, le compilateur, un shell, le mail...) ou spécifique à l'utilisateur via une application X. On parle de *bureau* ou de *plan de travail* (structure logique) plutôt que d'*écran* (structure physique) : le serveur X montre sur l'écran un bureau, mais il construit plusieurs bureaux. On change de bureau en sélectionnant l'un deux dans la petite fenêtre située en bas à droite de la barre des tâches de chaque bureau dans la configuration actuelle à l'IFSIC ; la figure 9 page 22 montre la barre des tâches.

15.1 Structure d'un bureau

Plusieurs outils peuvent être mis sur un bureau ; la figure 9 montre un écran sur lequel on trouve un gestionnaire de fichiers (ouvert en cliquant sur l'icône *Dossier personnel*) et une fenêtre de commande *xterm*.

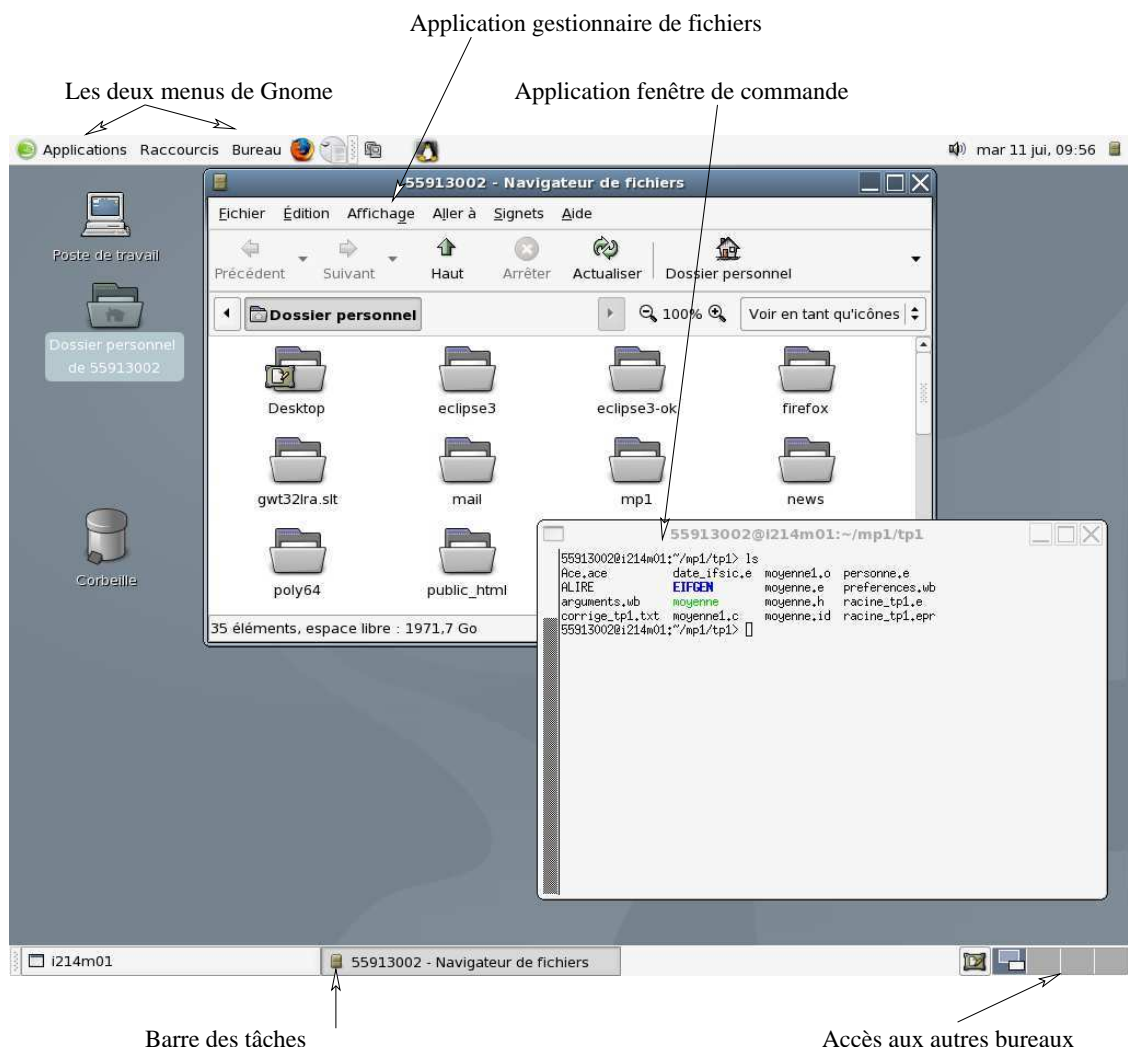


FIG. 9 – Un écran *X-Window* avec le look Gnome sous linux

Plusieurs fenêtres peuvent se chevaucher à l'écran. La souris permet, à l'aide d'un menu, de les déplacer, de modifier leur taille, de les ouvrir, de les fermer, de créer ou supprimer des fenêtres avec leur application ou de changer la fenêtre active, c'est à dire celle qui reçoit les informations en provenance de l'utilisateur : action sur le clavier, la souris.

Attention : les indications de boutons (gauche, milieu ou droit) dans ce qui va suivre ne sont peut-être pas tout-à-fait exactes, car tout est configurable. Vérifiez les actions des boutons au fur et à mesure que vous lisez le polycopié, et corrigez-le si les indications ne sont pas les bonnes dans votre environnement.

15.2 Structure d'une fenêtre

L'habillage d'une fenêtre est réalisé par le gestionnaire de fenêtre choisi lors de la connexion au serveur X. Par défaut, c'est le gestionnaire *Gnome* qui est installé.

Une fenêtre comme celle de la figure 10 est constituée d'un cadre et d'un intérieur :



FIG. 10 – Une fenêtre, son bandeau et son cadre

- le **cadre** est matérialisé par un bandeau supérieur foncé, qui peut contenir diverses informations sur l'outil actif dans la fenêtre, et par une bordure fine qui entoure la fenêtre. Un menu standard, le menu-cadre, est associé au cadre de chaque fenêtre ; il s'active par un clic droit dans le bandeau, ou par un clic gauche dans la case avec un V à gauche du bandeau. Il contient toutes les fonctions de manipulation de la fenêtre. Ces manipulations peuvent se faire aussi en cliquant ou en appuyant sur le cadre.
- l'**intérieur** représente la zone de dialogue de l'application avec l'utilisateur. Il peut être subdivisé en plusieurs sous-fenêtres (par exemple, une barre de menu et une zone de saisie). Chaque sous-fenêtre est elle aussi entourée d'une bordure. En outre, chaque application à l'intérieur d'une (sous-)fenêtre peut disposer d'un menu contextuel qui lui est spécifique.

15.3 La souris

Elle sert à déplacer à l'écran un **pointeur** qui change de forme selon le contexte, c'est à dire selon sa position (fond de l'écran, cadre ou intérieur d'une fenêtre,...), l'application qui le gère et le gestionnaire de fenêtres.

La souris est équipée de trois boutons, qui peuvent être actionnés de diverses manières :

- **clic** : appuyer et relâcher rapidement un bouton. Sert en général à déclencher ou confirmer une action.
- **double clic** : appuyer et relâcher deux fois un bouton. Amplifie l'action d'un simple clic.
- **presser** : appuyer et maintenir enfoncé un bouton. Sert à ouvrir un menu.
- **glisser** : presser un bouton puis déplacer la souris en maintenant le bouton. Sert à déplacer un objet ou sélectionner une portion de texte.

Le bouton du milieu, ou molette, sert aussi à faire défiler le texte d'une fenêtre munie d'un ascenseur.

Certaines de ces actions peuvent, en outre, être combinées avec les touches Majuscule ou Contrôle (Ctrl).

Chaque bouton a, par ailleurs, un rôle spécifique assigné par le gestionnaire de fenêtres ou le serveur *X-window*.

- **bouton gauche** : il sert à sélectionner un objet (le « copier »), un menu, un emplacement.
- **bouton du milieu** : il sert à reproduire une sélection (« coller »).
- **bouton droit** : il sert à ouvrir un menu contextuel, c'est à dire un menu dépendant de la position de la souris et de l'application (ou fenêtre) sous la souris.

Beaucoup d'actions des outils de *X-Window* s'appliquent sur une portion de texte : mots à supprimer ou déplacer, nom de fichier à éditer, etc...

15.3.1 Sélectionner, copier

Avant de réaliser une action il faut indiquer la portion de texte ou les objets concernés, ce qu'on appelle faire une sélection ; pour du texte, glisser avec le bouton gauche (bouton de sélection) depuis la première extrémité de la portion à sélectionner, jusqu'à la deuxième extrémité de la portion.

Il est possible d'utiliser des *raccourcis* pour faire une sélection de texte :

- sélectionner **un mot** : faire un *double-clic* du bouton gauche dans le mot
- sélectionner **une ligne** : faire un *triple-clic* du bouton gauche dans la ligne

La portion de texte sélectionnée change de couleur.

15.3.2 Coller

Une fois une sélection faite, on peut la coller ailleurs (ou copier les objets ailleurs) : un clic sur le bouton du milieu insère la sélection là où est le curseur.

15.3.3 Quand le copier-coller ne marche pas

Il est possible que certaines applications ne connaissent pas les conventions du copier-coller de *X-Window* ; c'est évidemment le cas des applications *Windows*.

Pour faire du copier-coller entre des applications qui n'ont pas les mêmes conventions, la règle est la suivante :

- « copier » selon la règle de l'application dans laquelle on fait la sélection (exemple : sélection à la souris seulement, ou sélection à la souris plus un ^C) ;
- « coller » selon les règles de l'application dans laquelle on fait la copie (exemple : un clic-milieu, ou un ^V).

15.4 Les menus sur Linux avec Gnome


Dans le bureau Gnome installé sur les machines Linux, le menu général s'obtient en cliquant sur  Applications en haut à gauche (ou en bas à gauche) de la barre des menus (selon où elle est placée) c'est un lézard vert, provenant de la distribution *Suse* installée.



FIG. 11 – La barre de menu sous Linux-Gnome

La figure 11 montre la barre de menu standard (vous pouvez facilement la modifier). Le lézard vert est donc le bouton d'accès au menu général, la mappemonde active le navigateur.



FIG. 12 – Menu général **Applications**



FIG. 13 – Menu **Actions**

La figure 12 montre le menu général obtenu par le menu **Applications** (ici, on a sélectionné le sous-menu **Bureautique**) et la figure 13 montre le menu **Actions** dans lequel on a sélectionné la fin de session.

16 LE SITE WEB DE L'IFSIC

Attention, ce paragraphe est d'actualité en juillet 2006, il ne le sera probablement plus dans sa totalité en septembre 2006.

L'IFSIC propose plusieurs sites Web :

- le site vitrine de l'IFSIC, WWW, qui présente l'IFSIC et ses formations,
- le site de l'administration, ADMIN, qui présente les services internes pour les étudiants et les enseignants,
- la passerelle externe, EXT, qui permet de lire son courrier et ses notes en dehors de l'IFSIC,
- le site des enseignants de l'IFSIC, ENSEIGNANT, non accessible,
- les sites pour les enseignants de l'Université, AGE et AGE-DOC, qui n'ont aucun intérêt pour vous,
- le site des administrateurs du réseau, ADMIN-ADMIN, interdit,
- et des sites occasionnels quand le besoin s'en fait sentir.

En général, chaque page de ces sites contient une liste déroulante qui permet de passer d'un site à l'autre (cette liste n'est pas visible sur la figure 6).

L'assistance aux utilisateurs est sur le site ADMIN. C'est là que vous devez aller pour (re)configurer votre compte (mot de passe, configuration Mozilla,...), pour voir les FAQ (Foire Aux Questions), et pour signaler les problèmes dont vous ne trouvez pas la solution dans les FAQ ni dans les questions qui n'ont pas encore eu de réponse.

L'onglet *Assistance aux utilisateurs-Un problème, une question?* vous dirige sur une page d'accès aux réponses aux questions archivées, aux questions en cours (un *ticket* est une question posée par un utilisateur), et qui vous permet de poser votre question ou d'exposer votre problème s'il n'est traité nulle part sur le réseau (par *Créer un nouveau ticket*).

Vous devez créer un ticket si vous constatez une anomalie (machine en panne, plus de papier sur l'imprimante, fichier inaccessible...), mais bien entendu, vous devez vérifier auparavant que personne ne l'a fait avant vous (selon des dates raisonnables).

Index

bash (groupement de commandes), 17

cat (voir un fichier), 8

cd (changer de répertoire), 18

chmod (droits d'accès), 17

commande, 6

courriel, 19

courrier électronique, 19

cp (copier un fichier), 8

droits d'accès, 16

éditeur de commande, 21

emacs (appel de emacs), 9

exécution d'un programme, 14

exit, 6

fichier, 4

 depuis *Windows*, 5

 manipulation, 7

firefox, 18

G :, 5

H :, 5

interrompre (touche d'interruption), 7

invite, 3, 5, 6

Java

 compilation, 13

less (voir un fichier), 8

logout, 6

lpq, 13

lpr (listing), 13

lprm (retirer un listing), 13

ls (liste des fichiers), 7

mail (courrier), 20

man (mode d'emploi), 7

mél, 19

mkdir (faire un répertoire), 18

mv (renommer un fichier), 9

pwd (afficher répertoire courant), 18

répertoire

 courant, 3, 5

 personnel, 3

rm (détruire un fichier), 8

rmdir (détruire un répertoire), 18

Thunderbird, 19