

# Chapitre 6

## Environnement LINUX et éditeur Emacs

### 6.1. Introduction

L'ordinateur utilisé durant les travaux pratiques est un PC bi-processeur, le système d'exploitation est LINUX avec une interface graphique KDE. L'utilisateur se connecte au PC via une station de travail qui a été préalablement 'booté' (booted) sur le PC. Ces stations de travail ne disposent pas de disques propres et sont donc très silencieuses. Il ne faut en principe *jamais* éteindre une station de travail, sauf lorsqu'il faut la rebooter (par exemple lors de l'arrêt du PC).

Chaque utilisateur a un nom unique qui l'identifie à l'ordinateur ainsi qu'un mot de passe, qui lui seront communiqués lors de la première séance de TP. Pour se connecter ("login"), il faut entrer son nom d'utilisateur et ensuite son mot de passe (qu'il est vivement recommandé de modifier par la commande `passwd`).

### 6.2. Les interfaces

Les ordinateurs modernes offrent aux utilisateurs des interfaces graphiques (Graphical User Interface ou GUI) sous forme de petits icônes intuitifs. Pour un usage avancé, il est cependant indispensable d'apprendre également à utiliser l'interface de commande en lignes (Command Line Interface ou CLI), qui nécessite d'entrer manuellement au clavier de nombreux termes archaïques, basés sur l'anglais et certainement moins intuitifs que les icônes du GUI. Le CLI permet de réaliser beaucoup plus de commandes que le GUI et avec un peu d'habitude, il est aussi beaucoup plus efficace.

Le GUI est en effet basé sur le CLI. Quand on double-clique sur un icône, on ne fait qu'exécuter cette

commande. Pour visualiser celle-ci, il faut cliquer sur l'icône avec le bouton droit de la souris, sélectionner 'Properties' et la commande devrait apparaître sous le bouton "Execute". La même commande peut-être exécutée en ouvrant un terminal et en y entrant le nom de la commande suivi de "retour chariot" (carriage return, CR), souvent marquée par "Retour" ou "Entrée" sur le clavier.

Pour ouvrir un terminal (ou console), il faut cliquer sur l'icône correspondant ou cliquer sur un des boutons de la souris et sélectionner la commande appropriée. Quand le prompt, qui est dans notre cas `[votrelogin@aurora ~]$`, est affiché, on peut entrer les commandes.

On peut ouvrir plusieurs consoles en même temps. Pour travailler dans une console, il faut positionner la souris dans la fenêtre et cliquer sur le bouton gauche. Ceci est vrai en fait pour toute fenêtre, par exemple celle ouverte par un éditeur. Dans le coin droit supérieur d'une fenêtre, il y a deux boutons : le bouton carré agrandit la fenêtre pour remplir l'écran (ou la réduire à la taille normale) ; le bouton avec un point met la fenêtre en forme d'icône, la taille normale est retrouvée en cliquant le bouton gauche sur l'icône. Le bouton dans le coin supérieur gauche conduit à un menu qui permet de changer la taille de la fenêtre, de la déplacer ou de la fermer.

## 6.3. Organisation des données

Sous LINUX, les données (programmes, documents, dots) sont stockées dans des fichiers ("files"), qui se trouvent eux-mêmes dans des répertoires ("directory").

Un répertoire peut contenir d'autres répertoires. Les répertoires sont ainsi organisés en arbre. A la base se trouve la racine (root), désignée simplement par / (slash). /u ou /home contient les répertoires pour les utilisateurs. /bin contient des fichiers de commande de base du système, /usr d'autres fichiers de commande et de configuration, etc. /tmp désigne un répertoire où sont déposés les fichiers temporaires, /work désigne un espace de travail.

Le caractère . désigne le répertoire courant (où l'on se trouve actuellement), .. le répertoire juste au-dessus (le parent), ~ le répertoire principal de l'utilisateur (son "home").

On peut désigner un répertoire par son adresse relative à celui où l'on se trouve (dir1, ../../dir1, etc), par son adresse relative au home (~src/dir1) ou par son adresse absolue qui commence toujours par / (/tmp/guest/dir1, /usr/local/bin, etc).

Chaque fichier et chaque répertoire a un nom. Ce nom ne doit pas comporter de caractères spéciaux à part \_ (l'underscore) et . (le point). LINUX distingue entre majuscules et minuscules, donc file et File sont des fichiers différents. Des extensions communes désignent des fichiers du même type : par exemple les fichiers dont les noms se terminent par .f90 sont des programmes écrits en Fortran 90, par .cpp des programmes en C++, par .o des fichiers objets (contenant la traduction des programmes en langage machine) créés par un compilateur. On distingue des fichiers textes, qui sont lisibles par l'utilisateur, et des fichiers binaires, lisibles par l'ordinateur.

Le nom complet d'un fichier se compose de l'adresse absolue du répertoire dans lequel se trouve ce fichier, suivie du nom local (/home/guest/dir1/fich, /tmp/guest/dir1/prog.dat, etc).

Les commandes de base sont des mots de quelques lettres, généralement des abréviations : par exemple `ls` vient de `list` (le contenu d'un répertoire). Les commandes ont souvent une ou plusieurs options, indiquées par `-option` : par exemple `ls -t` donne la liste des fichiers en fonction de la date de leurs dernières modifications. Les commandes doivent être introduites après le prompt, et exécutées en frappant sur la touche CR.

Pour changer les contenus d'un fichier texte, on utilise un éditeur de texte. Ceci permet de créer un fichier texte, de le remplir, de le modifier ou de le détruire. L'éditeur fourni avec LINUX est `vi` ("view"), l'éditeur `emacs` est aussi souvent disponible.

## 6.4. Commandes Linux de base

`fich` représente un nom de fichier et `dir` un nom de répertoire

**ls (fich)** "list" : pour avoir une liste des fichiers d'un répertoire, ainsi que des sous-répertoires. Le nom `fich` peut être un nom de fichier ou de répertoire. Une `*` remplace n'importe quel caractère ou chaîne de caractères. Il y a plusieurs options pour `ls` :

`-a` "all" : pour voir tous les fichiers, y compris ceux dont le nom commence avec `.` (fichiers cachés)

`-l` "long" : liste de fichiers avec tous leurs détails (type, date de création, ...)

`-t` "time" : liste de fichiers ordonnés selon la date de la dernière modification.

On peut combiner plusieurs options : `ls -lt` fournit la liste des fichiers avec leurs détails et ordonnés selon la date de la dernière modification

`ls a*.f90` affiche tous les fichiers dont le nom commence par `a` et se termine par `.f90`

**pwd** "print working directory" : affiche le nom du répertoire courant

**cd (dir)** "change directory" :

`cd dir` change le répertoire courant vers le répertoire `dir` qui doit être un sous-répertoire du répertoire courant

`cd /dir` change le répertoire courant vers le répertoire dont l'adresse absolue est `/dir`

`cd ..` change le répertoire courant à celui juste au-dessus

`cd` fait repasser au répertoire principal

**mkdir dir** "make directory" : crée le répertoire `dir`

**rmdir dir** "remove directory" : détruit le répertoire `dir`, qui doit avoir été préalablement vidé

**cp fich1 fich2** "copy" : crée une copie du fichier `fich1` nommée `fich2`

`cp -i fich1 fich2` demande une confirmation si `fich2` existe déjà

`cp fich dir` crée une copie de `fich` dans le répertoire `dir`, avec le même nom

`cp /dir/fich .` crée dans le répertoire courant une copie du fichier `/dir/fich` avec le même nom

`cp /dir/*.f90 .` copie dans le répertoire courant tous les fichiers du répertoire `/dir` dont le nom se termine par `.f90`

**mv fich1 fich2** "move" : change le nom `fich1` en `fich2`

`mv fich dir` fait passer le fichier `fich` dans le répertoire `dir`

**rm fich** "remove" : détruit le fichier `fich`. Il n'y a pas moyen de récupérer le fichier après sa destruction

`rm -i fich` détruit le fichier `fich`, en demandant une confirmation avant

**rm \*** détruit tous les fichiers du répertoire

**more fich** : fait défiler à l'écran **fich**, page par page

**head fich** : affiche le début du fichier **fich**  
**head -n N fich** affiche les *N* premières lignes du fichier **fich**

**tail fich** : affiche la fin du fichier **fich**  
**tail -n N fich** affiche les *N* dernières lignes du fichier **fich**

**cat fich** "concatenate" : fait défiler à l'écran **fich**  
**cat fich1 > fich2** écrit le contenu du fichier **fich1** dans le fichier **fich2**. Si **fich2** existe déjà, son contenu est perdu  
**cat fich1 >> fich2** ajoute le contenu du fichier **fich1** à la suite du fichier **fich2**

**diff fich1 fich2** : affiche les différences entre les fichiers **fich1** et **fich2**

**touch fich** : si **fich** existe déjà, met à jour sa date ; si **fich** n'existe pas, crée un fichier vide de ce nom

**man** "manual" : l'aide pour commande

**f90** compilateur Fortran : **f90 -o prog prog.f** compile le programme Fortran **prog.f** et met l'exécutable dans le fichier **prog**  
Pour exécuter un programme, il suffit d'entrer son nom (par exemple **prog**)  
Si les données d'entrée standard (unité 5) sont dans un fichier **data**, et si on veut les sorties standards (unité 6) dans un fichier **out**, il faut utiliser la commande **prog < data > out**

**date** : donne la date et l'heure

**cal (month) (year)** : **cal 1 2001** - donne le calendrier du mois de janvier 2001  
**cal 2001** - donne le calendrier de l'année 2001

**time** : **time command** - donne le temps d'exécution de **command**

**ps** "process status" : liste les processus en cours

**CRTL** : touche CTRL (habituellement en bas à gauche et à droite du clavier)  
CTRL **c** : arrête le process en cours  
CTRL **z** : suspend le process en cours. Celui-ci peut être réactivé en background par la commande **bg** (permettant l'exécution d'autres commandes), ou en foreground par la commande **fg** (la fenêtre n'accepte alors pas d'autres commandes)

**kill** : **kill -9 N** - arrête le process numéro **N**

**telnet** : **telnet aurora.univ-rennes1.fr** permet de se connecter sur l'ordinateur **aurora.univ-rennes1.fr**

**ftp** "file transfer protocole" : **ftp aurora.univ-rennes1.fr** - permet de transférer des fichiers à travers le réseau

Les commandes de base LINUX de changement de répertoire **cd dir** et d'impression de liste de fichiers **ls** peuvent être activées à l'adresse visitée. Pour changer de répertoire local (c-à-d sur la machine dont on a lancé la commande **ftp**), il faut utiliser la commande **lcd dir**

Pour déposer un fichier : **put fich**

Pour prendre la copie d'un fichier : **get fich**

Pour transférer des fichiers en binaire, entrer **bin** avant **get** ou **put**

Pour transférer plusieurs fichiers à la fois : **mget fich** ou **mput fich**

Pour éviter de devoir confirmer le transfert de chaque fichier, entrer **prompt** avant **mget** ou **mput**

Pour économiser de l'espace disque et diminuer les temps de transfert par Internet, on peut compresser un groupe de fichiers dans un fichier archive dans le format tar. La commande

```
tar -cvf name.tar file.names
```

produit un fichier archive du nom de name.tar comprenant tous les fichiers file.names. L'argument c signifie créer, v signifie verbose et f spécifie le nom des fichiers comprimés. Celui-ci peut-être une suite explicite de fichiers, ou un directory. Dans ce dernier cas, tous les fichiers, y compris ceux des sous-répertoires, seront comprimés et pourront être récupérés avec la structure en arborescence originale par la commande

```
tar -xvf name.tar
```

(x pour extract). Si un seul fichier particulier, file, doit être récupéré, la commande est

```
tar -xvf name.tar file
```

Pour comprimer encore plus le fichier d'archive, on peut le 'zipper'

```
tar -cvzf name.tar.gz
```

Les commandes de décompression correspondantes sont

```
tar -xvzf name.tar.gz
```

```
tar -xvzf name.tar file
```

## 6.5. Commandes de l'éditeur Emacs

`emacs fich` : éditer le fichier `fich`.

`CRTL-x CTRL-c` : quitte emacs

`CTRL x s fich1` : sauvegarde "fich1" sur le disque dur

`CTRL-a` : déplace le curseur au début de la ligne

`CTRL-e` : déplace le curseur à la fin de la ligne

`CTRL-k` : tue à partir du curseur jusqu'à la fin de la ligne et la place dans une zone mémoire "tampon"

`CTRL-y` : colle les contenus de la zone mémoire "tampon" à l'endroit où se trouve le curseur

`CTRL-g` : annule la commande sur la ligne de commande

`CTRL-x u` : annule la dernière commande

`CTRL-x 2` : divise la fenêtre en 2

`CTRL-x 0` : détruit la fenêtre où se trouve le curseur

`CTRL-s` : cherche un mot vers l'avant du fichier

`CTRL-r` : cherche un mot vers l'arrière du fichier

`CTRL-s, CTRL-s` : répète la commande de recherche

`CTRL-u n command` : répète n fois command

Entrée : quitte la procédure de recherche

Esc % : remplace un mot par un autre (suite de la commande sous la barre noire du buffer)

SPC : remplace l'occurrence, puis passe à l'occurrence suivante

, : remplace l'occurrence, sans passer à l'occurrence suivante

DEL (Suppr) : passe au suivant sans remplacer l'occurrence

^ : revient à l'occurrence précédente

! : remplace toutes les autres occurrences restantes

ESC : quitte la procédure de remplacement