

# MEUMEU

Meumeu Est Un Manuel sur Emacs et Unix



---

François GUILLAUME

3 juin 2009

---

## Remerciements

Ce manuel a été écrit à l'intention des personnes qui découvrent l'environnement Unix et qui peuvent être en quête de repère. C'est donc en essayant de répondre aux questions que les novices sous Unix peuvent se poser que ce manuel a été rédigé.

Je tiens à remercier les  $\beta$  testeurs de ce manuel ainsi que de ses exemples. Pour leur retours qui je l'espère auront permis de rendre ce manuel le plus compréhensible possible.

# Table des matières

<b>1</b>	<b>Prolègomène</b>	<b>5</b>
1.1	Définitions . . . . .	5
1.2	Le .emacs . . . . .	8
1.3	Les fichiers de sauvegarde . . . . .	9
1.4	Nécessaire de survie . . . . .	9
1.5	Organisation de ce manuel . . . . .	10
<b>2</b>	<b>Emacs : un éditeur de texte</b>	<b>11</b>
2.1	Déplacements . . . . .	11
2.2	Sélectionner . . . . .	14
2.3	Remplacer . . . . .	15
2.4	Les rectangles . . . . .	16
2.5	Recherches sophistiquées : Les expressions régulières . . . . .	16
2.6	Exemples d'utilisation . . . . .	17
<b>3</b>	<b>Emacs un gestionnaire de fichiers</b>	<b>19</b>
3.1	Ouverture de dossiers . . . . .	19
3.2	Gestion des Buffers . . . . .	20
3.3	Déplacement au travers des dossiers . . . . .	21

3.4	Disposition des buffers . . . . .	22
3.5	Recherche d'un fichier . . . . .	22
3.6	Opérations sur les fichiers . . . . .	23
3.7	Autres fonctions utiles . . . . .	24
3.7.1	Droits d'un fichier . . . . .	24
3.7.2	Lien symbolique . . . . .	24
3.7.3	Compression . . . . .	25
3.8	Travailler sur un groupe de fichiers . . . . .	25
3.9	Exercices . . . . .	26
<b>4</b>	<b>Shell, scripts et autres outils</b>	<b>29</b>
4.1	Le mode shell . . . . .	29
4.2	Quelques éléments du langage shell . . . . .	29
4.3	Les scripts . . . . .	31
4.4	Autres outils . . . . .	31
4.5	Exercices . . . . .	32
<b>5</b>	<b>Programmation</b>	<b>33</b>
5.1	Fonctionnalités avec Fortran 90 . . . . .	33
5.2	programmation avancée . . . . .	34
5.2.1	makefile . . . . .	35
5.2.2	Tags . . . . .	36
5.2.3	CVS . . . . .	36
5.3	Un exemple . . . . .	37
<b>6</b>	<b>Emacs et ESS</b>	<b>39</b>
6.1	Mode SAS . . . . .	39
6.2	R et Emacs . . . . .	39

## TABLE DES MATIÈRES

---

6.3	Exercice . . . . .	41
<b>7</b>	<b>L<sup>A</sup>T<sub>E</sub>X</b>	<b>43</b>
7.1	Rudiments de L <sup>A</sup> T <sub>E</sub> X . . . . .	43
7.2	Mode L <sup>A</sup> T <sub>E</sub> X . . . . .	44
7.3	Mode Postscript . . . . .	44
7.4	Exemple . . . . .	45
<b>8</b>	<b>Gadgets d'Emacs</b>	<b>47</b>
8.1	ftp et telnet . . . . .	47
8.2	L'agenda et les calculatrices . . . . .	47
8.3	Lecteur de mail . . . . .	48
8.4	Jeux . . . . .	48
8.5	Dessin . . . . .	49
8.6	Le psychanalyste . . . . .	49



# Prolègomène

Ce manuel a été rédigé afin de donner un aperçu de ce qu’Emacs peut apporter dans le travail de tous les jours à la SGQA. Ce manuel n’a pas pour but de bouleverser les pratiques de chacun, il peut éventuellement permettre de découvrir un bien bel outil et quelques méthodes de travail, voire vous aider dans la vie de tous les jours.

Une mise en garde est toutefois nécessaire, l’apprentissage d’Emacs **n’est pas facile**, on doit donc être prêt à faire des efforts pour maîtriser la bête. Néanmoins, une fois qu’une certaine habitude du monstre est acquise, s’en passer peut devenir chose difficile.

## 1.1 Définitions

Nous allons tout d’abord définir ici des notions essentielles à la compréhension de ce manuel.

### Interface graphique

Emacs comporte une interface graphique<sup>1</sup>, ainsi à l’appel de la commande `emacs` dans un shell, vous obtenez un écran ressemblant à celui de la figure 1.1.

Les informations figurant sur la page d’accueil sont : le statut de ce logiciel “libre”<sup>2</sup>, les moyens de trouver des informations sur Emacs et enfin la version ainsi que le type de machine/système pour lequel Emacs a été compilé.

On peut identifier plusieurs parties dans l’écran d’emacs 1.1.

---

1. contrairement à vi, l’autre grand éditeur de texte sous unix  
2. Son créateur étant le créateur du projet GNU et de la licence libre : Richard Stallman [http://fr.wikipedia.org/wiki/Richard\\_Stallman](http://fr.wikipedia.org/wiki/Richard_Stallman)

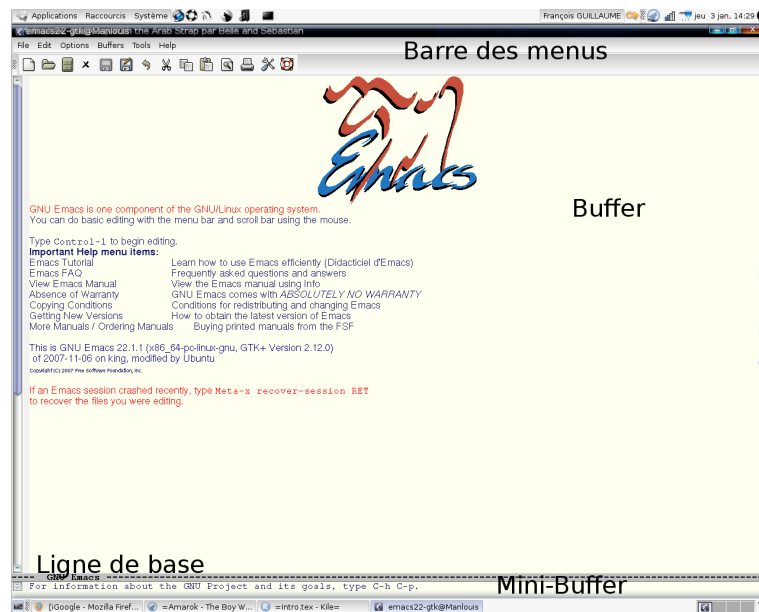


FIGURE 1.1 – Écran d’accueil d’Emacs

- La fenêtre principale, contenant dans notre exemple le logo et les informations sur emacs est appelée “Buffer”, c’est dans cette partie que nous travaillerons.
- En haut de l’interface, on a la “barre de menus” contenant sur sa première ligne les menus Fichier, Edition etc. et sur la seconde ligne des icônes permettant d’effectuer des tâches courantes (création de fichier, ouverture, enregistrement etc.). Ces menus se déroulent à l’aide d’un clic et permettent d’accéder aux fonctions les plus courantes.
- En bas du buffer principal, se situe la “ligne de base” qui donne des renseignements divers sur : l’état du buffer, la date et l’heure, les droits d’écriture sur le buffer etc.
- Enfin, juste en dessous de cette ligne de base, se situe le “Mini-buffer”, qui sert principalement à exécuter des commandes. Il s’active dès qu’un raccourci clavier a été exécuté, et attend que vous répondiez à ses demandes.

L’un des atouts d’emacs est sa gestion de multiple buffers. Ainsi vous n’avez pas à ouvrir emacs autant de fois que vous ouvrez de fichiers, il vous suffit d’ouvrir Emacs une seule fois et de naviguer ensuite entre les buffers. Nous reviendrons plus en détail sur la gestion de multiples buffers dans le chapitre 2 de ce manuel.

## Mode d’Emacs

Emacs est un programme qui regroupe une infinité de fonctions et programmes. Puisque tous ces programmes ne sont pas utiles en toute occasion, Emacs est structuré en Mode.

Un mode est une sélection de fonctions utiles dans un contexte particulier. Par exemple, si vous éditez un fichier Fortran 90, Emacs passera directement en mode Fortran-90 et proposera



## CHAPITRE 1. PROLÈGOMÈNE

des fonctions telles que l'indentation du code, la compilation du code source, la coupure des phrases longues selon les conventions du langage etc ... En fonction du contexte l'interface d'Emacs évoluera donc énormément.

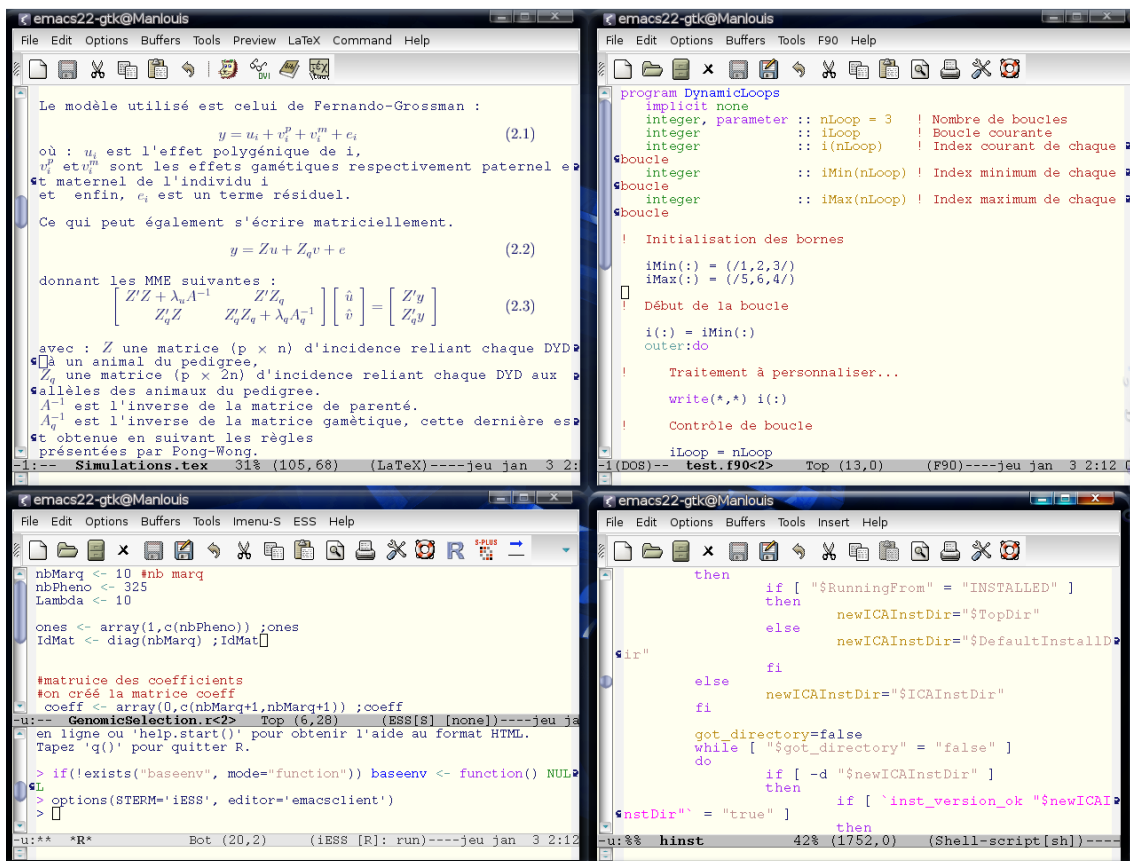


FIGURE 1.2 – Illustration des différents modes d’Emacs. En haut à gauche, mode  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , à droite le mode Fortran90, En bas à gauche, mode ESS (pour R) et à droite le mode script shell

Comme le montre la figure 1.2, l’interface d’Emacs va donc évoluer en fonction du type de fichier à traiter. Ainsi, on notera que les barres de menus ne sont pas toutes les mêmes et que chacune des lignes de base indique le mode dans lequel Emacs se situe.

### Les raccourcis clavier et leur syntaxe

Emacs a recours à deux touches très régulièrement, il s’agit des touches “Control” et “Meta”.

La touche **Control** correspond à la touche Ctrl du clavier, elle est abrégée dans les manuels emacs par **C**.

La touche **Meta** correspond quant à elle à la touche Alt du clavier, son abréviation courante est **M**.

Les raccourcis clavier présentés dans les pages d'aides ainsi que dans le présent manuel suivent une convention assez simple. Lorsque deux touches doivent être tapées simultanément un “-” sépare ces deux touches. Dans le cas contraire, les deux touches doivent être tapées successivement.

Ainsi :

- la commande C-x signifie “appuyer sur Ctrl et x simultanément”
- C-x o signifie quant à lui “appuyer sur Ctrl et x simultanément puis sur la touche o”
- C-x C-s signifie enfin “appuyer sur Ctrl et x simultanément puis appuyer sur Ctrl et s simultanément”

Attention, la casse est importante g et G n’auront pas le même effet ! Quand, dans un raccourci clavier, une lettre figure en majuscule, on considère implicitement que vous appuierez sur la touche maj et la lettre.

Les raccourcis clavier figurent sur les menus déroulant de la barre des menus, on peut ainsi les apprendre plus facilement avec le temps. Il existe également une carte , la “Quick Ref Card” récapitulant les raccourcis les plus courants sous emacs, celle-ci est présente en annexe.

## La complétion

La complétion est un mécanisme qui permet de déduire la fin d’une chaîne de caractères à partir de ses premiers éléments . Pour le mettre en oeuvre, il suffit de taper quelques caractères puis d’appuyer sur la touche de tabulation.

1. Dans le cas, où aucun doute n’est possible la chaîne de caractères est complétée entièrement.
2. Si un doute est permis la chaîne est complétée jusqu’au premier caractère sur lequel différentes alternatives existent, et vous devez compléter manuellement.
3. Si vous n’êtes pas en mesure de compléter, un nouvel appui sur la touche tabulation, ouvrira un buffer dans lequel figureront l’ensemble des complétions possibles.

Cette fonctionnalité est très pratique dans la vie de tous les jours, prenez donc l’habitude de l’utiliser. Nous verrons quelques exemples de mise en oeuvre pratique de la complétion dans les chapitres suivants.

## 1.2 Le .emacs

Comme beaucoup de programmes sous unix, emacs utilise un fichier de configuration. Celui-ci s’appelle “.emacs”. Ce fichier est un fichier texte, contenant des instructions en LISP (le langage de programmation d’emacs), qui permet de configurer emacs à votre goût (couleurs, options, etc...). Il existe de nombreux sites sur internet présentant des exemples de .emacs .<sup>3</sup> Dans

---

3. <http://www.dotemacs.de/>

les parties à suivre nous utiliserons un script qui vous permettra d’avoir un .emacs totalement fonctionnel (cf exercice chapitre 1).

Si vous avez un peu de curiosité, vous pouvez lire le fichier .emacs. Les commentaires sont précédés du signe “;”, les instructions sont des fonctions délimitées par des parenthèses, permettant de définir des variables e.g. (set-background-color "ivory"), de charger des fichiers de code lisp conférant de nouvelles fonctionnalités à Emacs e.g. (load "/usr/share/emacs22/site-lisp/ess/lisp/ess-site.el") ou activant au démarrage de Emacs des fonctions optionnelles (mouse-wheel-mode 1).

Ces codes peuvent être soit retrouvés sur internet, ou bien générés par Emacs lui-même, via l’interface graphique (Options, customize emacs).

### 1.3 Les fichiers de sauvegarde

Emacs a pour habitude de réaliser une copie de sauvegarde de chacun des fichiers qu’il ouvre et modifie, cette copie a le même nom que le fichier original suffixé par le signe “~”. Cette sauvegarde est très pratique pour revenir à la dernière version fonctionnelle d’un programme que vous pensiez “améliorer”, ou même pour comparer deux versions consécutives d’un même fichier (et afficher par exemple les lignes de code auxquelles vous avez ajouté vos “améliorations”).

Pour ceux qui préfèrent ne pas conserver de version de sauvegarde, ou ceux qui, au contraire, voudraient conserver les 5 dernières versions d’un fichier, une commande lisp peut-être ajoutée dans le .emacs, pour tenir compte de vos préférences.

Lors d’un crash d’Emacs<sup>4</sup>, un fichier de récupération est créé, il est préfixé et suffixé du signe #. Ce dernier n’est pas utilisable tel que et doit être réouvert à l’aide d’emacs. Vous récupèrerez ainsi les buffers dans leur état lors du crash.

### 1.4 Nécessaire de survie

Avant de vous lancer dans les plus belles prouesses emacsiennes, envisageons le pire, c’est à dire qu’en lieu et place de prouesses, vous obtenez plutôt une mauvaise symphonie de super-marché composée des bips d’erreurs de votre terminal !<sup>5</sup>

Deux raccourcis sont donc à apprendre absolument avant toute tentative de lancer Emacs.

**Annuler une commande** : C-g

---

4. Ça peut arriver, mais cela reste aussi fréquent que le passage de la “Comète de Encke”

5. Ça peut aussi arriver

Il s'agit ici de désactiver le mini-buffer qui dans le cas contraire attend de votre part de nouvelles instructions.

**Annuler la dernière modification de texte :** `C-_` ou `C-x u`

Des appuis successifs annuleront une à une les dernières modifications.

Enfin, Emacs possède une aide pléthorique (tutoriels, FAQ, Aide) à laquelle vous pourrez accéder via le menu Help, ou la commande `C-h`. N'hésitez donc pas à vous référer à cette aide si besoin.

## 1.5 Organisation de ce manuel

Armés de ces définitions, nous allons maintenant présenter différents aspect du logiciel emacs et les accompagner d'exemples pratiques.

Les trois premiers chapitres de ce manuel présenteront les bases pour éditer du texte , naviguer dans les dossiers, et exécuter des commandes shell via emacs.

Les chapitres suivants présenteront des modes d'emacs moins courants mais néanmoins utiles (Logiciel statistiques, programmation de langages compilés, PAO avec  $\text{\LaTeX}$ ) . Une présentation succinctes de outils utilisés sera faite avant la présentation du Mode Emacs. Des exercices d'applications seront là pour illustrer l'utilisation possible de ces modes.

Le dernier chapitre présentera des outils plus gadgets, parfois totalement inutiles...donc totalement indispensables.

# Emacs : un éditeur de texte

D'après Wikipedia <sup>1</sup> :

“Emacs est une famille d'éditeurs de texte disposant d'un ensemble extensible de fonctionnalités et qui est très populaire parmi les programmeurs et plus généralement les personnes ayant des compétences techniques sur les ordinateurs.”

Emacs est donc avant tout un **éditeur de texte**. Dans ce chapitre, nous allons présenter les fonctionnalités les plus courantes pour éditer du texte avec Emacs.

## 2.1 Déplacements

### Déplacements simples

Il existe plusieurs manières de se déplacer dans le buffer d'emacs.

1. À l'aide de la souris
2. À l'aide des flèches de direction
3. À l'aide des lettres du clavier

Nous ne ferons pas l'affront de détailler l'usage de la souris qui est très intuitif, mais, il faut le rappeler, pas forcément productif. En effet, l'usage exclusif du clavier peut permettre de gagner en efficacité.

Les déplacements du curseur se font à l'aide des flèches directionnelles. Un moyen de se déplacer plus rapidement est d'employer les touches directionnelles avec la touche CTRL. Ainsi, les déplacements vers la droite et vers la gauche se feront mot à mot <sup>2</sup>. De même, les déplacements vers le haut ou vers le bas se feront paragraphe par paragraphe.

---

1. <http://fr.wikipedia.org/wiki/Emacs>

2. Dans la pratique, c'est souvent ce type de déplacement que vous utiliserez le plus souvent

Commande	Equivalent	Déplacement
→	C-f	Un caractère à droite
←	C-b	Un caractère à gauche
↑	C-p	Une ligne en haut
↓	C-n	Une ligne en bas
↖	C-a	Début de la ligne
Fin	C-e	Fin de la ligne
Pgup	C-v	Une page en haut
PgDwn	M-v	Une page en bas
C-→	M-f	Un mot vers la droite
C-←	M-b	Un mot vers la gauche
C-↑	M-a	Un paragraphe vers le haut
C-↓	M-e	Un paragraphe vers le bas
C-↖	M-<	Début du texte
C-Fin	M->	Fin du texte

TABLE 2.1 – Récapitulatif des raccourcis clavier de déplacements

Les touches Home et Fin permettent de se déplacer respectivement au début et à la fin d'une ligne. La combinaison de ces touches avec Ctrl permettra de se déplacer respectivement au début du buffer et à la fin de celui-ci.

À titre anecdotique, les déplacements peuvent également se faire à l'aide de certaines touches clavier. Cet emploi peut s'avérer utile notamment si l'on veut utiliser emacs via telnet (dans ce cas de figure les touches directionnelles ne sont plus actives !)

## Déplacements moins simples

On est parfois amené à vouloir se déplacer de façon très précise dans un fichier texte. Par exemple pour atteindre une ligne particulière qui poserait problème ou encore pour contrôler qu'un fichier texte est conforme à un format de lecture à colonnes fixes.

La majorité des fonctions emacs peut être répétée. Il en va de même pour les déplacements. Pour se déplacer de 8 colonnes vers la droite par exemple, on répète 8 fois la fonction C-f. La fonction de répétition est appelée par la commande C-u suivie du nombre de répétitions puis la commande à effectuer plusieurs fois. Ainsi, pour se déplacer en colonne 8 d'un fichier, il faudra effectuer la séquence suivante :

- Passage en début de ligne en position colonne 0      C-a
- C-u 8 C-f ou C-u 8 →

La fonction de répétition pourra s'appliquer de même pour passer n lignes, n paragraphes<sup>3</sup> etc.

3. Ou bien pour mettre en forme vos lignes de commentaires dans vos programmes avec une ligne de n fois le caractère "\*" par exemple

Commande	Commande complète	Action
C-x r m Enter		Pose un bookmark à l'endroit où se trouve le point
C-x r m <nom> Enter	M-x bookmark-set	Pose un bookmark et lui donne un nom
	M-x bookmark-delete <nom>	Détruit le bookmark <nom>
C-x r b <nom>	M-x bookmark-jump	Aller au bookmark <nom>
C-x r l	M-x bookmark-list	Lister tous les bookmarks
	M-x bookmark-file	Sauve les bookmarks

TABLE 2.2 – Liste des commandes relatives aux bookmarks.

Une autre fonction utile est de pouvoir atteindre une ligne précise. Ceci est rendu possible par la séquence suivante :

- M-g g suivie du numéro de la ligne à atteindre ou **M-x goto-line**

Enfin, il est fréquent qu'au sein d'un fichier (notamment un gros programme) on soit amené à aller régulièrement aux mêmes endroits dans le fichier, pour cela on pourra utiliser des marque-pages, ce qui consiste à donner un nom à une position du fichier pour pouvoir s'y retrouver facilement. Les commandes relatives aux marque-pages sont décrites dans le tableau 2.2.

## Recherches simples

La recherche d'une expression, mot *etc...* est une tâche très courante. La recherche se fait avec emacs à l'aide de commande C-s. Le minibuffer est alors activé et vous êtes invités à entrer le mot à rechercher. Au fur et à mesure que vous tapez le mot à rechercher, les occurrences de l'expression à rechercher sont mises en surbrillance dans le buffer. Pour déplacer le curseur d'occurrence en occurrence, il suffit de taper sur C-s. La commande C-r permet de faire une recherche mais à l'envers (i.e. en remontant dans le texte). Ainsi, on peut facilement revenir sur une occurrence.

Si la chaîne de caractères à rechercher n'est pas discriminante, on peut compléter celle-ci pendant la recherche. Si vous avez fait une erreur dans la chaîne de caractère, vous pouvez également effacer (à l'aide de la touche backspace) les derniers caractères du mot à trouver. Attention, si vous aviez commencé à faire défiler les occurrences du mot que vous souhaitez corriger, emacs va d'abord revenir au point de départ de la recherche avant que la touche Backspace ne fasse effet.

Il arrive souvent que l'on soit amené à faire la recherche des mêmes mots. La fonction de recherche permet de rappeler les dernières expressions recherchées. Ainsi, pour rappeler l'avant dernière expression recherchée, il suffit de :

1. Entrer dans le mode recherche (C-s)
2. Rappeler la dernière expression à l'aide des touches M-p

```

570 5088 1069 66 FR2920781829 FRUS02266008 FR2997064123 2
571 100 1157 66 FR2921587032 FR1095001791 FR3522992903 1
572 3754 587 66 FR2921611337 FR5694028588 FR2928495361 1
573 4861 496 66 FR2921632649 FRNR83938054 FR2297020831 1
574 4861 1092 66 FR2921632653 FRNR83938054 FR2998035163 1
575 5090 496 66 FR2921632670 FRUS02289548 FR2297020831 1
576 6118 547 66 FR2921632748 FRUS17058140 FR2912263258 1
577 4774 1083 66 FR2922049683 FRNR11111639 FR2998012590 1
578 4774 1055 66 FR2923926546 FRNR11111639 FR2996042170 1
579 5071 1057 66 FR2924287713 FRUS02205082 FR2996051509 2
580 406 1030 66 FR2924287739 FR2292002618 FR2995050716 2
581 5088 1057 66 FR2924287777 FRUS02266008 FR2996051509 2
582 3865 580 66 FR2924287838 FR5994022699 FR2924287739 1
583 5071 453 66 FR2924642106 FRUS02205082 FR2295001272 2
584 5090 583 66 FR2924642178 FRUS02289548 FR2924642106 1
585 5090 1094 66 FR2926655450 FRUS02289548 FR2998048141 1
586 6134 1211 66 FR2926769223 FRUS17226843 FR3556861871 1
587 406 1019 66 FR2928495361 FR2292002618 FR2995001576 2
588 5091 1068 66 FR2928822260 FRUS02294436 FR2997015432 1
589 3754 1085 66 FR2928822272 FR5694028588 FR2998012685 1
590 4861 1042 66 FR2929504175 FRNR83938054 FR2996001261 2
591 6118 590 66 FR2929504314 FRUS17058140 FR2929504175 1
592 4846 4787 66 FR2929868582 FRNR79292043 FRNR16877809 2
593 5059 6133 66 FR2929868593 FRUS02183007 FRUS17221958 2
594 5090 4871 66 FR2929868627 FRUS02289548 FRU120461325 1
595 4774 4897 66 FR2929868629 FRNR11111639 FRU127030645 1
596 5091 4895 66 FR2929868636 FRUS02294436 FRU127026082 1
597 5070 945 66 FR2929952017 FRUS02203438 FR2992000616 2
598 5091 597 66 FR2929952154 FRUS02294436 FR2929952017 1
599 3754 1065 66 FR2929952157 FR5694028588 FR2997001645 1
--:-- ped complet 9% (599,47) (Fundamental Isearch)--- jeu fév 7 9:01
I-search: fr5694028

```

FIGURE 2.1 – Mise en surbrillance d’une recherche

Chaque nouvel appel de M-p rappellera l’expression précédente dans l’historique des recherches. Inversement, à l’aide de M-n on fera défiler dans l’ordre inverse les dernières commandes.

Se déplacer facilement dans un fichier est intéressant dans un premier temps, mais l’étape suivante est souvent de faire des modifications. Nous allons ici voir comment utiliser emacs au mieux pour traiter un fichier texte à modifier.

## 2.2 Sélectionner

Après avoir décrit les moyens de se déplacer dans du texte, nous allons voir comment le modifier. Sinon, le terme “Éditeur” de texte serait un peu usurpé...non ?

Nous allons ici définir un ensemble de termes spécifiques à emacs.

Une marque est un point du texte qui est enregistré par emacs. On définit une marque à l’aide de la combinaison C-espce (le message suivant “mark set” s’affiche alors dans le mini-buffer. La marque nous servira à délimiter une zone de sélection qui s’étendra de la marque à la position du curseur, cette zone est appelée “Region”. Ainsi, pour sélectionner un mot, il faut :

1. Se positionner au début de ce mot.
2. Poser une marque (C-espce)
3. Se déplacer à la fin du mot (C-→)

La région est alors mise en surbrillance<sup>4</sup>. Une fois une région sélectionnée on va pouvoir

4. Si ce n’était pas le cas, il faut cliquer sur options, et cocher “Active Region hightlighting”.



Commande	Action
C-w	Couper
M-w	Copier
C-y	Coller
C-k	Couper ce qui est à droite du curseur

TABLE 2.3 – Raccourcis clavier pour copier coller

effectuer dessus des modifications.

Avec Emacs, on ne coupe pas une zone de sélection, ... on la tue “Kill”, ou plus sagement on l’écrit “Write” dans un “kill ring”. De même, on ne colle pas une région mais on l’arrache “Yank”. Vous ne trouverez donc pas de C-C C-V sous emacs !

Les raccourcis correspondants sont donc 2.3 :

La dernière commande sera surtout utile dans le mini-buffer pour indiquer notamment des chemins d’accès (cf Chapitre 2).

Nous avons parler du “Kill ring”. Il s’agit d’un espace mémoire qui va stocker l’ensemble des régions coupées/copiées. On pourra ainsi choisir quelle région parmi les dernières coupées on veut coller. La procédure est la suivante :

- On se positionne à l’endroit où l’on veut coller notre région
- On colle (C-y)
- Puis on remplace successivement la région collée par celles précédemment coupées à l’aide des touches M-y

### 2.3 Remplacer

Si l’on sait comment rechercher, on peut dans la foulée remplacer un mot par un autre.

On procèdera ainsi

1. Appeler la fonction de remplacement M-%
2. Renseigner (dans le mini-buffer) le mot à rechercher et valider par entrée
3. Renseigner le mot de substitution et valider par entrée
4. Répondre Y pour effectuer le remplacement ou N sinon
5. (Si vous préférez effectuer toutes les modifications d’un seul coup appuyer sur !)

La recherche se fait à partir de la position du curseur et jusqu’à la fin du fichier. Une astuce si vous souhaitez faire des changements sur une zone particulière du fichier est de sélectionner d’abord une région, puis d’effectuer un remplacement, ce dernier ne s’effectuera que dans la région sélectionnée.

Commande	Action
C-x r t <texte>	Insère un rectangle contenant le texte “texte”
C-x r k	Couper
C-x r y	Coller

TABLE 2.4 – Raccourcis utiles pour les rectangles

Expression	interprétation
.	N’importe quel caractère
*	la répétition $n$ ( $n \geq 0$ ) fois du caractère présent avant *
+	la répétition $n$ ( $n \geq 0$ ) fois du caractère présent avant +
[acd]	Un des caractères entre crochet
[a-n]	Un caractère compris entre a et n

TABLE 2.5 – Principaux symboles pour les expressions régulières

## 2.4 Les rectangles

Nous allons présenter ici une fonction intéressante d’emacs, la création de rectangles. Emacs permet en effet de créer des sélections rectangulaires de texte

Pour faire une sélection rectangulaire le principe est simple. La région que l’on désire sélectionner sera définie à l’aide d’une marque posée au niveau d’un coin du rectangle et l’on positionnera le curseur au niveau du coin opposé du rectangle.

Une fois le rectangle créé, plusieurs actions sont possibles : la création d’un rectangle de texte, le coupage d’un rectangle ou son stockage.

## 2.5 Recherches sophistiquées : Les expressions régulières

Les recherches peuvent parfois s’avérer plus complexes que la simple recherche d’un simple mot. Pour gérer ces cas plus complexes il existe les “expressions régulières” qui sont un ensemble de conventions qui permettent d’apporter plus de flexibilité dans la recherche.

Il existe ainsi de nombreuses autres combinaisons<sup>5</sup> mais leur emploi reste moins fréquent.

5. <http://www.linux-france.org/article/appli/emacs/manuel/html/regexps.html>

## 2.6 Exemples d'utilisation

Après avoir exposé les fonctionnalités d'Emacs nous allons les mettre en oeuvre au travers d'exemples. Mais de commencer voici la procédure à mettre en oeuvre pour obtenir les exercices d'exemple.

1. Copiez dans le repertoire que vous souhaitez l'archive des exemples.
2. Extrayez l'archive
3. Entrez dans le repertoire
4. Lancez le script d'initialisation emacs
5. Entrez dans le dossier EXEMPLE/Ex1
6. Lancez emacs

Ce qui revient à taper dans la console Xterm les commandes suivantes.

```
cp \ugen\ugenfgu\commun\EMACS.tgz .
tar -xvf EMACS.tgz
cd EMACS
init
cd EXEMPLE\Ex1
emacs pedigree &
```

Ce premier exercice a pour but d'utiliser les déplacements.

Prenons la situation suivante, vous souhaitez lire un fichier de pedigree (avec un programme SAS par exemple) et vous obtenez des erreurs à l'exécution dues à des problèmes de format de lecture.

1. Le code race n'est pas renseigné
2. Les numéros nationaux sont codés en 2+12 soit sur 14 positions

Nous allons donc essayer de corriger tout cela. Premièrement nous allons ajouter une colonne de 46 en position 20, correspondant au code race des bovins laitiers de race Montbéliarde.

Attention, Emacs commence la numérotation des colonnes à 0, SAS la commence en colonne 1 ! Il faut bien tenir compte de ceci dans vos déplacements et lorsque vous vérifiez la position de votre curseur indiquée dans la ligne de base.

Une fois ceci fait, nous allons maintenant effacer les deux premiers caractères des numéros nationaux "FR".

Une fois ces modifications réalisées votre programme devrait lire correctement le fichier de pedigree.

Une première vérification de la qualité des données nous indique que la valeur maximale pour la variable sexe est 3...ce qui est assez gênant. L'animal problématique est l'individu n°100. Nous allons donc nous rendre à la ligne 100.

La question est maintenant de connaître le sexe réel de l'individu 100. On pourra par exemple poser un marque page à la position de la ligne 100, puis lancer une recherche sur le nombre 100 pour savoir si cet individu apparaît en tant que père ou mère. On retourne alors à l'individu 100 pour corriger l'erreur de sexe.

Enfin, après vérification nous avons fait une erreur de race, le code race à indiquer n'est en effet pas le code 46 mais le code 66. Essayons de "remplacer"<sup>6</sup> le code 46 par le code 66.

---

6. Essayez pour vous amusez de faire ce remplacement à la fois avec et sans expression régulière par exemple

## Emacs un gestionnaire de fichiers

La gestion de fichiers est une tâche très courante sous unix. Emacs permet de réaliser assez facilement un ensemble de tâches courantes qui vous permettront de gagner en efficacité.

### 3.1 Ouverture de dossiers

Un travail bien structuré nécessite souvent de disséminer ses fichiers dans un ensemble cohérent de dossiers, sous-dossiers. La contre-partie de cette structuration est de multiplier les commandes pour passer d'un dossier à un autre. Nous allons présenter ici comment se déplacer dans un dossier à l'aide d'Emacs.

Le mode Dir, est un mode servant à la gestion/navigation de dossiers. On l'appelle à l'aide de la commande C-x d (ou en cliquant sur l'icône répertoire). Le mini-buffer est alors activé, et un emplacement à regarder vous est demandé par défaut à l'ouverture il s'agit de votre home directory (/ugen/ugenxxx). Il vous suffit alors de renseigner le dossier que vous souhaitez afficher et de valider par entrée. Vous obtiendrez alors un affichage ressemblant à celui de la figure 3.1.

Les informations contenus par le buffer sont : Le nom des fichiers, leur propriétaire, leur droits d'accès.

Par convention, les dossiers sont marqués d'un d dans la ligne des droits, de plus emacs colore les répertoires d'une certaine couleur (ici en bleu). On notera de plus que les fichiers cachés sont présentés (fichiers commençant par un point). Les fichiers de sauvegardes (c'est à dire ceux finissant par ~ générés automatiquement) ont également une couleur différente (ici grise). Enfin, les fichiers de liens symboliques ont également leur couleur atitrée (ici violet). On peut ainsi avoir en un clin d'oeil une bonne information de ce que contient un dossier.

Une fois le mode Dir activé, on peut déplacer le curseur sur les différents fichiers du dossier,

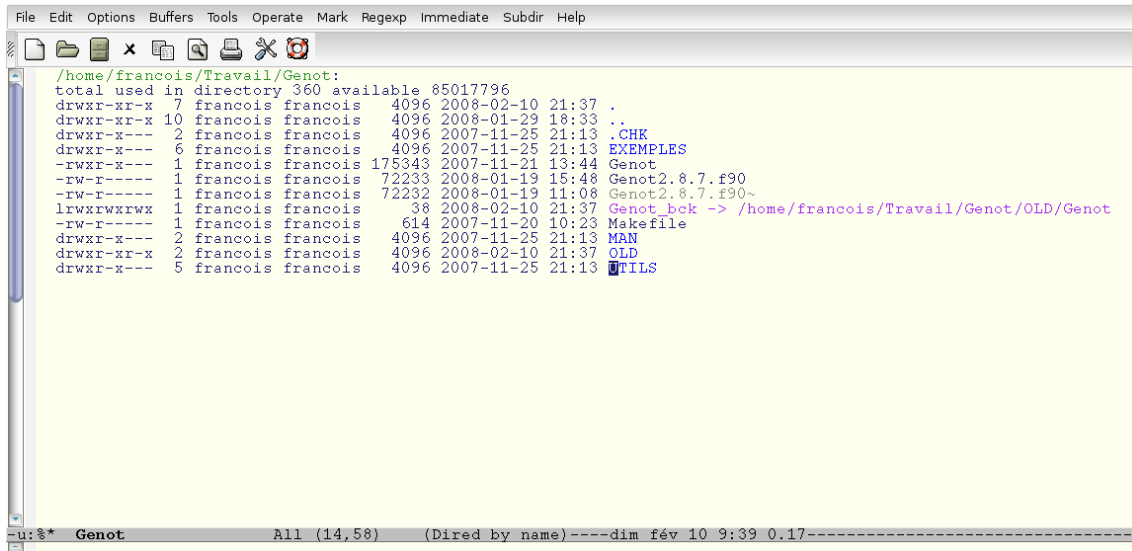


FIGURE 3.1 – Mode répertoire

et ouvrir ces derniers en validant sur entrée. On pourra également sélectionner le fichier à l'aide de la souris et l'ouvrir à l'aide d'un clique du milieu sur le fichier. Le fichier s'ouvre en général dans un nouveau buffer. Lorsque l'on clique sur le nom d'un dossier, c'est ce dossier qui s'ouvre en mode répertoire dans un autre buffer.

## 3.2 Gestion des Buffers

Comme précisé précédemment, Emacs gère facilement l'ouverture de plusieurs fichiers simultanément si bien qu'il n'est pas nécessaire d'ouvrir emacs plus d'une fois. En général, Emacs n'affiche que deux buffers simultanément. Pour choisir les buffers que l'on souhaite afficher, il suffit de positionner le curseur dans un des buffers, puis cliquer sur le bouton droit de la souris tout en appuyant sur la touche control. Un menu contextuel s'affiche comme ce qui est présenté dans la figure 3.2, on déplace la souris sur le buffer que l'on souhaite afficher et on relâche la touche control.

Une autre manière de procéder est de taper `C-x C->` et faire défiler dans l'ordre les buffers. On peut revenir au buffer précédemment affiché à l'aide de la commande `C-x C-<`. Enfin, si vous connaissez le nom des buffers, la commande `C-x b <nom du buffer>` est possible. Tout comme nous l'avons vu précédemment si vous appelez souvent les mêmes buffers, les touches `M-p` et `M-n` vous permettront de faire défiler dans le mini-buffer les derniers buffers appelés.

Lorsque deux buffers sont affichés, on peut passer de l'un à l'autre par la commande `C-x o`. On peut également cliquer à l'aide de la souris.

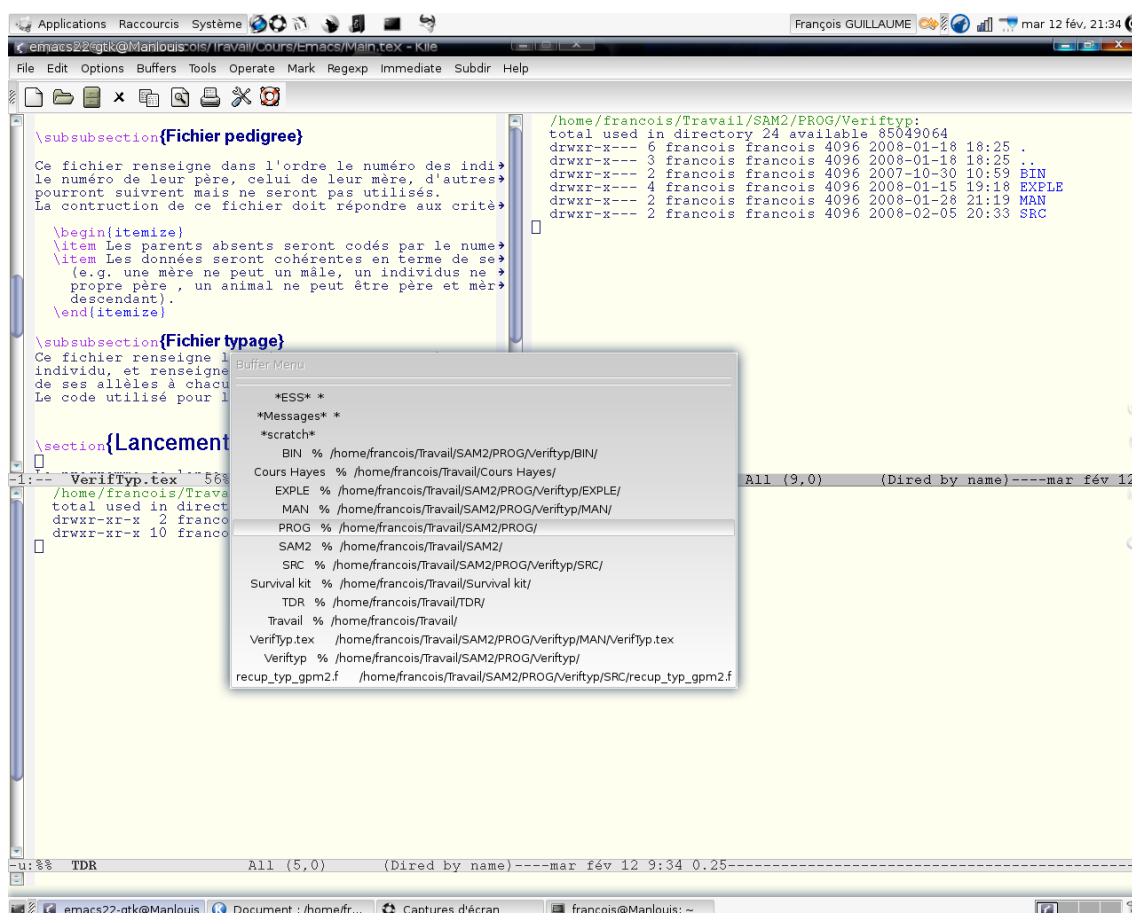


FIGURE 3.2 – Menu contextuel

### 3.3 Déplacement au travers des dossiers

Nous avons vu une méthode simple d’ouverture d’un répertoire, nous allons voir une méthode un peu plus efficace.

Lorsque vous tapez le chemin d’accès au dossier, vous pouvez recourir à la complétion en tapant sur la touche Tab. Pour rappel, la complétion est un procédé permettant de compléter le début d’un mot.

Par exemple, vous voulez écrire le chemin d’accès suivant `/MonDossierBin`, il vous suffira de taper par exemple `/MonDos`, puis TAB et Emacs complétera le chemin d’accès par `sierBin`. Si la complétion est équivoque et que deux complétions sont possibles, Emacs créera un nouveau buffer qui affichera les complétions possibles. Il vous restera à désambigüer la situation.

Lorsque vous renseignez l’emplacement à atteindre, le mini-buffer se comporte comme n’importe quel buffer de texte. Vous pouvez ainsi vous déplacer mot par mot à l’aide de la commande `C←` et bien souvent effacer ce qui se trouve à droite du curseur `C-k`. Vous pou-

Action sur le buffer	Raccourci
Division en 2 buffers sens horizontal	C-x 2
Division en 2 buffers sens vertical	C-x 3
Fusion des buffers en un seul buffer	C-x 0

TABLE 3.1 – Raccourcis pour l’organisation des buffers

vez également coller un chemin que vous auriez coupé précédemment C-y. Enfin, puisque l’on est régulièrement amené à taper les mêmes chemins, le mini-buffer conserve en mémoire l’ensemble des chemins déjà tapés. On fera défiler ces derniers à l’aide de M-p ou M-n ou encore avec les flèches haut et bas.

### 3.4 Disposition des buffers

Par défaut emacs affiche deux buffers. Vous pouvez néanmoins adapter cette affichage en divisant en deux chacun des buffers. Le tableau 3.4 indique les raccourcis pour modifier l’affichage des buffers.

On pourra ajuster la taille des buffers en positionnant la souris sur la ligne de base des buffers (la flèche de la souris se modifie alors) et en faisant glisser cette ligne tout en maintenant le bouton droit de la souris enfoncé.

Enfin, lorsque l’écran Emacs a été divisé plusieurs fois on peut être amené à changer la police de caractère afin d’avoir une meilleure vision d’ensembles des buffers. Pour se faire, il suffit de maintenir la touche Shift enfoncée et de faire un clic gauche de souris. Un menu contextuel apparait et vous permet de choisir le type de police pour l’affichage des buffers.

### 3.5 Recherche d’un fichier

Lorsqu’un dossier contient un nombre important de fichiers, il peut devenir difficile de trouver le fichier qui nous intéresse. Plusieurs petites astuces peuvent permettre de mieux s’y retrouver.

Par défaut, les fichiers sont affichés par ordre alphabétique. On peut néanmoins les classer selon leur date de création par un appui sur la touche s.<sup>1</sup>

Une autre possibilité est d’avoir recours à la fonction recherche. Celle-ci s’appelle par C-s, (comme en mode texte) et on fait défiler les occurrences de l’expression à rechercher par appuis successifs sur ces touches.

1. Note, il se peut que le buffer ne se mette pas à jour (dans ce cas un fichier créé récemment n’apparait pas dans le buffer, il suffit de le mettre à jour en tapant sur la touche g)



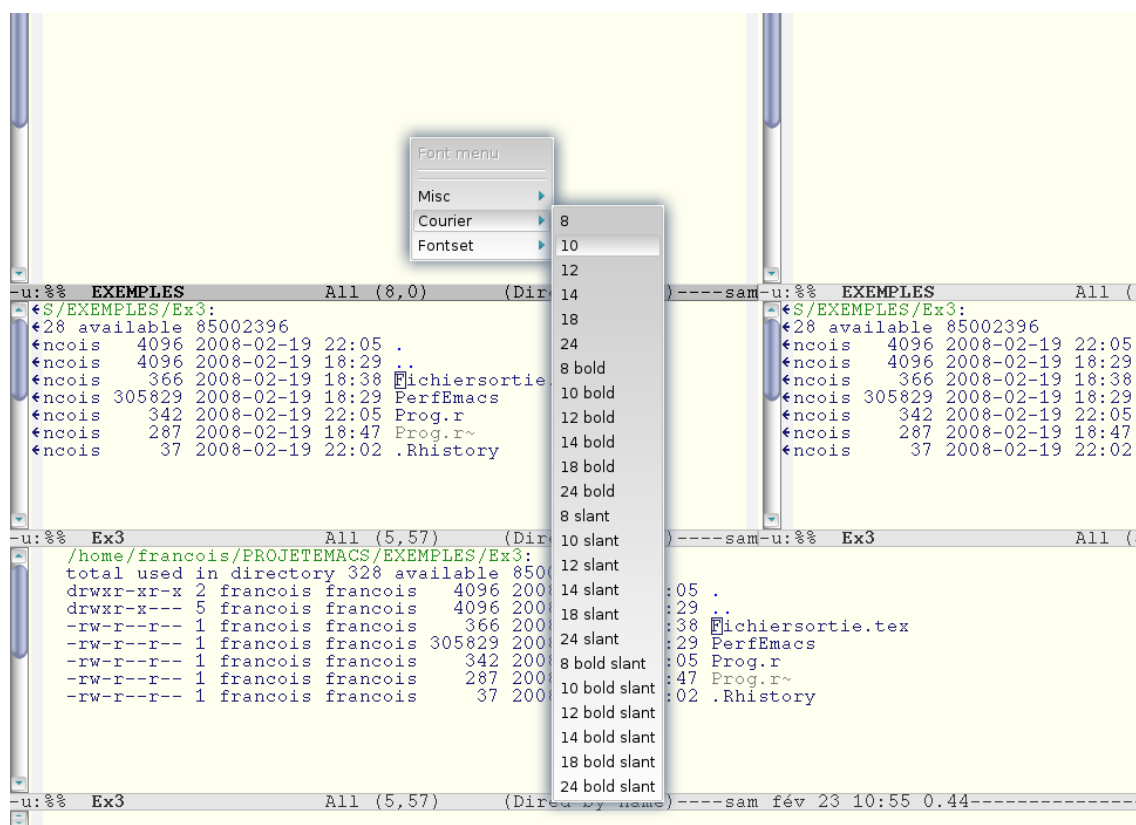


FIGURE 3.3 – Menu contextuel pour le choix des polices

## Expressions régulières

On peut utiliser des expressions régulières lorsque l’on indique le chemins d’accès vers un dossier. Supposons que vous avez un dossier “MonDossierSAS” contenant des fichiers liés à SAS (.sas,.log et.lst) et que vous ne souhaitez voir que les fichiers .log. On peut alors procéder comme précédemment pour ouvrir le dossier, mais lorsque vous indiquerez le chemin vers le dossier, vous complèterez le chemin par une expression régulière. Le chemin deviendra : “MonDossierSAS/\*.\*log” après validation, on obtient un buffer n’affichant que les fichiers .log.

## 3.6 Opérations sur les fichiers

L’interface d’emacs propose un ensemble de fonctions sur les fichiers. Nous allons présenter un petit échantillon de celles-ci.

Pour créer un nouveau dossier il suffit de taper sur + puis de renseigner dans le minibuffer le nom de ce dossier.

Pour déplacer, modifier le nom d’un fichier, il suffit de positionner le curseur sur le fichier à

modifier, taper sur R et renseigner le nouveau nom/emplacement du fichier.

Pour copier un fichier, il faut positionner le curseur sur le fichier, taper C et renseigner le nom du nouveau fichier

Pour effacer un fichier, il faut sélectionner le fichier,taper D et valider l'effacement.

Pour les commandes précédentes, la complétion est possible et le rappel des derniers noms de fichiers renseignés se fait par la combinaison M-p ou M-n.

## 3.7 Autres fonctions utiles

### 3.7.1 Droits d'un fichier

Pour changer les droits d'un fichier, il faut positionner le curseur sur ce fichier,taper sur Shift-M et entrer le code de droits. Pour rappel, différentes syntaxes sont possibles (ma préférée étant le code sur trois bits, dont les règles sont résumées dans le tableau 3.7.1). Le droits accordés sont la somme de droits

droit	Coeff du droit	Valeur à ajouter
Lecture	$2^2$	4 ou 0
Ecriture	$2^1$	2 ou 0
Exécution	$2^0$	1 ou 0
Code	Somme des valeurs pour chaque droits	

Le code le plus large 7, se décompose ainsi en : 4 (droit en lecture) + 2 (droit en écriture) + 1 (droit en exécution). Pour permettre à n'importe qui de copier un fichier (le fichier .emacs par exemple) le code sera 4 (lecture seule)

On donne un droit pour chaque type d'utilisateur, tout d'abord pour le propriétaire, ensuite pour le groupe (i.e. toute personne appartenant au groupe de l'utilisateur) et enfin pour tout utilisateur. La commande suivante : `chmod 754 <fichier>` signifie ainsi

**7** Droits de lecture, écriture et exécution pour le propriétaire

**5** Droits de lecture et exécution pour le groupe

**4** Droit de lecture seul pour tout utilisateur

### 3.7.2 Lien symbolique

Emacs permet de créer facilement des liens symboliques. Ces derniers peuvent être utiles notamment dans des situations où vous utilisez un programme qui nécessite un nom spécifique

pour un fichier<sup>2</sup>, que ce dernier est volumineux et que vous préférez ne pas le recopier inutilement. Une autre utilisation peut être pour intégrer un programme dans votre bin et ainsi ne plus avoir à recopier le chemin complet d'accès à ce programme.

Pour créer un lien symbolique, on sélectionne le fichier à lier, S et on indique le chemin d'accès vers le lien.

### 3.7.3 Compression

Pour décompresser un fichier compressé, il vous suffit de placer le curseur sur ce fichier et taper Z. Cette même commande permet de compresser un fichier.

## 3.8 Travailler sur un groupe de fichiers

Les tâches répétitives sont le lot des utilisateurs d'Unix. Emacs propose pour simplifier ces tâches le travail par lot. Par exemple, votre dossier contient trois fichiers contenant la même erreur (un numéro national éroné par exemple). Vous pourriez ouvrir un à un ces fichiers pour corriger cette erreur en effectuant un remplacement. Une autre solution est de traiter ces trois fichiers simultanément. On procèdera ainsi :

1. Marquer les fichiers à modifier en :  
positionnant le curseur sur chaque fichier  
appuyant sur M
2. Appuyer sur Q
3. Indiquer l'expression à rechercher et valider
4. indiquer l'expression de remplacement et valider
5. l'ensemble des remplacements possibles vous seront proposer, la validation d'un remplacement se faisant exactement comme en mode texte.

L'ensemble des commandes présentées précédemment peuvent être réalisée par groupe. Attention, cependant à bien retirer la marque des fichiers une fois la modification appliquée (sous peine d'effacer un jour des fichiers par erreur par exemple). Pour désélectionner un seul fichier précédemment sélectionné il faut sélectionner ce fichier puis taper U. Pour désélectionner tous les fichiers précédemment sélectionnés, on pourra utiliser la séquence suivante : \* !

Lorsque vous éditez un texte avec emacs, une copie de sauvegarde est automatiquement effectuée. Si vous êtes sûr de ne pas en avoir besoin, ces sauvegardes peuvent être effacées. Pour cela, il faut :

---

2. Nous verrons un exemple plus loin

- Marquer les fichiers de sauvegarde en appuyant sur AltGr-2 (en fait on veut accéder à la touche tilde )
- Effacer en appuyant sur x
- Confirmer l’effacement des fichiers

Note : les fichiers marqués deviennent rouges et la lettre D apparait en début de ligne. Si vous n’observez pas de changements appuyez sur g pour mettre à jour le buffer.

On peut procéder de même avec d’autres types de fichiers spécifiques. Ces types sont renseignés dans le menu contextuel Mark d’emacs.

### 3.9 Exercices

Nous allons démarrer cette fois-ci emacs sans préciser de fichier à ouvrir. À l’aide du mode directory, rendez-vous dans le fichier d’exemple 2 “Ex2”.

Un fichier nommé surprise s’y trouve. Nous allons placer le curseur sur ce fichier

Si vous avez des difficultés, essayez de voir ce qui ne va pas et remédiez à ce problème.

Nous allons ici démarrer emacs sans l’associer à un fichier. La commande à taper est donc emacs & (l’esperluette signifie que le programme est lancé en arrière plan. et qu’on pourra donc continuer à entrer des commandes dans le shell. Sans cette esperluette, vous ne pouvez pas entrer de nouvelle commande avant d’avoir fermé emacs.

La première chose à faire est donc d’ouvrir le répertoire / votre home directory.

Pour info, le home directory est votre répertoire atitré, celui-ci est sauvegardé régulièrement, mais ne doit normalement pas contenir de fichiers de données. À chaque ouverture d’un shell ce répertoire est lu pour charger des paramètres personnel par exemple. De même, si vous possédez un répertoire appelé /bin l’ensemble des programmes/fichiers exécutables qui s’y trouveront pourront être appelé par leur simple nom (i.e. sans leur chemin complet d’accès) depuis n’importe quel répertoire !

Une fois que vous avez ouvert votre répertoire entrez dans le fichier EMACS/Exemple/Ex2

Nous allons maintenant effectuer une commande qui ne sert pas tous les jours sur le fichier appelé surprise. Positionnez, le curseur sur ce fichier. puis appuyez sur la touche !, le minibuffer est activé, validez par entrée.

Bon maintenant, vous devez regretter d’avoir fait ce qui vous a été demandé ! Ceci nous amène à deux constats !

1. Unix permet de faire en très peu de commandes beaucoup de choses.
2. Il est donc très important de vérifier ce que l’on fait !

En fait, vous ne pouviez pas réellement savoir ce qui allait se passer puisque par défaut le

## CHAPITRE 3. EMACS UN GESTIONNAIRE DE FICHIERS

---

fichier était protégé en écriture et pas en exécution. Modifiez ceci afin de ne plus réitérez cette erreur et de pouvoir voir et comprendre ce qui s'est passé.

Entrez maintenant dans le fichier surprise, pour voir sa structure. Il s'agit en fait d'un script, pour le moment, notons juste que ce script est une succession de commande shell courantes, imbriquées dans une boucle. Nous verrons plus tard comment créer de tels scripts.

Réparons maintenant nos erreurs. Essayez de supprimer le fichier ficvar.

Pour aller plus vite essayez de trouver un moyen de supprimer tous les fichiers dont le nom commence par fichier de la façon la plus efficace possible. Il existe bien entendu plusieurs façons de procéder.

Un fichier ne s'est pas effacer, en effet fichier 17 n'est pas un fichier, mais un dossier ! On ne pourra pas l'effacer tant qu'il contiendra des données.

Entrons dans ce répertoire. Trois fichiers s'y trouvent. Ces fichiers portent l'extension .Z, il sont donc compressés. Décompressez ces trois fichiers. (Si possible en même temps).

Nous allons maintenant copier le fichier ped dans le répertoire ex2.

Ceci n'étant pas très efficace, nous allons prendre le risque de tout perdre et faire plutôt déplacer ces trois fichiers dans ex2.

Une fois ceci fait, il ne doit plus rester de fichiers dans le répertoire fichier17, on peut donc procéder à son effacement.

Puisque vous avez eu cette fois-ci la curiosité de regarder les fichiers que nous venons de copier, vous avez vu qu'il s'agit d'un fichier de pedigree, un fichier de performances et d'un fichier de paramètres.

Il y a deux performances enregistrés dans le fichier de performances, nous allons donc pouvoir réaliser deux analyses de composantes de la variance. Pour ce faire, copiez le fichier de paramètres.

Par mégarde le fichier de paramètre n'est pas valide, en fait le fichier de pedigree qui s'appelle geneal dans les fichier de paramètres, s'appelle en réalité pedigree dans notre exemple.

Remplacez dans les deux fichiers de paramètres le mot geneal par pedigree.

Nous avons donc les fichiers nécessaires à une analyse des composantes de la variance de deux caractères. Il nous manque un logiciel pour réaliser les analyses. Nous allons utiliser le logiciel de I Misztal. remlf90. Une version de ce dernier est présent dans le répertoire MISC de EMACS. Pour l'exercice, créez dans le répertoire ex2 un lien symbolique vers remlf90.

Pour info, cette pratique présente plusieurs avantages :

1. Elle évite de copier X versions d'un même logiciel (on a donc toujours la dernière version (celle qui dans l'idéal présente le moins de bug).
2. Elle simplifie l'écriture des commandes de lancement (on a pas à taper tout le chemin

d'accès vers un programme)

# Shell, scripts et autres outils

## 4.1 Le mode shell

Il s'agit d'un mode qui pourrait paraître inutile (après tout pour lancer emacs on passe déjà par un shell) L'atout du shell d'emacs est de pouvoir continuer à employer les raccourcis que vous avez déjà appris. En effet, le shell emacs se comporte comme un fichier texte, vous pouvez donc utiliser les commandes de déplacement, copie, collage etc. Ceci présente un intérêt notamment pour pouvoir récupérer des résultats du shell et les sauvegarder dans un autre fichier.

Ce shell permet également la complétion. Enfin, le rappel des dernières commandes est également possible à l'aide des commandes `M-p` et `M-n`.

Le shell a par contre quelques défauts, notamment pour l'affichage sur plusieurs pages des résultats de certaines commandes. Par exemple, lors de l'appel de la fonction `man`<sup>1</sup>.

Le mode shell est appelé par la commande `M-x shell`. Un buffer s'ouvre alors, par défaut il se trouve dans le dossier où vous étiez lorsque vous l'avez appelé.

Comme il arrive que certains processus ne s'arrêtent pas ou qu'on ait rencontré un problème, on peut tuer un processus par la commande `C-c C-c`. D'autres commandes sont disponibles dans le menu "signal".

## 4.2 Quelques éléments du langage shell

Si vous êtes aguerri au langage shell, vous pouvez passer cette partie. Nous verrons ici quelques bases utiles lors de l'utilisation du shell.

---

1. Rassurez-vous pour afficher les pages des manuels il existe un mode spécifique appelé par `M-x man`

Commande	Fonction
cd	Change de répertoire
ls	Liste un répertoire
rm	Efface un fichier
mv	Déplace (ou renomme un fichier)
cp	copie un fichier
echo <expression>	Ecrit l'expression <expression> dans la console
date	Ecrit la date dans la console

Le shell est une coquille qui est remplie de petits logiciels très utiles. Ceux qui sont les plus utilisés sont référencés dans le tableau

Pour obtenir une aide sur ces commandes, une autre commande est possible `man`. Il vous suffit de taper `man ls` pour avoir une description de la commande `ls`. Dans Emacs, il est recommandé d'utiliser le mode Man (appelé par la commande `M-x man`).

Les options les commandes que vous utiliserez peuvent être suivies d'options. Leur rôle est décrit dans les pages de manuel. La syntaxe habituelle est donc `commande -options`. Par exemple : `ls -al` listera (l) tous les fichiers (a) i.e. même ceux cachés.

Un mécanisme très intéressant sous unix est la redirection de fichier symbolisée par les signes `>` et `<`. Il s'agit ici de diriger le contenu d'un fichier vers un programme ou le contraire. Par exemple, `Programme <FichierEntrée >FichierSortie` Va lire `FichierEntrée` et envoyer son contenu comme argument de `Programme`. `Programme` va produire un nombre de messages de résultats qui seront stockés dans `FichierSortie`. On évite par ce procédé de devoir répondre aux questions du programme et on peut aussi s'abstenir de rester face à son écran pour voir les résultats en temps réel du programme, tout en conservant une trace écrite des messages que le programme aura envoyer.

Le pipe symbolisé par le signe `"|"`, ici c'est une autre fonctionnalité qui permet de renvoyer le résultat d'un programme directement en entrée d'un autre programme. Un exemple la commande `ls -al | wc -l` va lister l'ensemble des fichiers du répertoire courant puis renvoyer ce listing vers la commande `wc -l` qui compter le nombre de ligne d'un fichier. Nous saurons donc ainsi combien de fichiers se trouvent dans notre dossier.

Les chemins. Sous Unix tous les fichiers sont désignés par un chemin à partir de la racine `"/"`. Pour accéder à n'importe quel fichier, on pourra écrire l'intégralité du chemin pour y accéder. Une autre façon de procéder est de se servir de chemin relatifs. Il s'agit ici de désigner un fichier par référence au répertoire dans lequel on se trouve. Si le fichier que l'on veut atteindre se trouve dans le répertoire courant, on l'appellera par son nom. Si l'on doit remonter d'un niveau pour atteindre le fichier, on utilisera le symbole `"../"` qui signifie remonter d'un niveau. On pourra combiner ces symboles. ainsi : `"../../.."` nous fait remonter de 3 niveaux.

Les meta caractères. Il s'agit ici de caractères qui vont se substituer à un ensemble d'autre caractères. Par exemple, le signe `"*"` va se substituer à n'importe qu'elle chaine de caractère,



donc, si vous exécutez la commande : `ls Fic*`, vous allez lister l'ensemble des fichiers commençant par `Fic` et suivi de n'importe quelle chaîne de caractères.

### 4.3 Les scripts

Pour automatiser encore plus le travail sous unix, on peut mettre dans un fichier un ensemble de commandes unix. Celle-ci seront exécutées les unes après les autres. Il faut ensuite penser à leur donner un droit d'exécution. On pourra alors exécuter ce script en tapant `./NomDuFichier` par exemple.

Ceci correspond aux bases du langage de script. En fait, on peut réaliser bien plus qu'une succession de commandes en réalisant des structures plus complexes (boucles, recours à des variables). La maîtrise des scripts demanderait un manuel à lui tout seul, nous allons donc plutôt revenir à emacs et aux fonctionnalités qu'il propose.

Un mode shell-script existe dans emacs. On l'appelle par la commande `M-x sh-mode`.

Parmi les fonctionnalités que propose ce mode, on a une coloration syntaxique (qui permet de bien identifier les éléments du script en fonction de leur type (variable, commentaires, opérateurs de langage). Une autre fonction bien pratique est l'aide à la construction de certaines structures. Ainsi, pour faire une boucle `for` le minibuffer va s'activer et vous demander le nom de la variable d'index puis après validation, vous obtiendrez le squelette d'une boucle `for`.

### 4.4 Autres outils

Il existe quelques utilitaires très pratiques que l'on pourra utiliser dans nos scripts.

`head` et `tail`.

Ces utilitaires vont afficher le début et la fin d'un fichier, l'option `-n10` permet de préciser le nombre de lignes à extraire (10 ici).

`wc`

Cet outil permet de compter le nombre de lignes, mots et caractères d'un fichier. Il est très utile pour connaître les dimensions d'un fichier (pour l'allocation de matrices par exemple). En conjonction avec `head`, il permet facilement de connaître le nombre de colonnes d'un fichier (`head -n1 Fichier | wc -w`).

`grep`

Cet outil permet d'extraire d'un fichier les lignes correspondant à un motif particulier. ex : `grep 5619216053 pedigree`, permettra de sélectionner les lignes du fichier `pedigree` dont une ligne contient le motif `5619216053`.

Awk

Il s'agit en fait plus d'un langage de programmation que d'un simple outil. Dans les scripts, il permettra par exemple d'extraire facilement une colonne de donnée, grâce à une syntaxe assez simple. Par exemple pour extraire la colonne 1 du fichier pedigree, on exécutera : `awk 'print $1' pedigree`.

### 4.5 Exercices

Nous allons reprendre l'analyse du jeu de données du répertoire Ex2 là où nous l'avions laissée. Ouvrez emacs et rendez-vous dans le répertoire Ex2.

Lancez un buffer de shell

Nous allons récupérer des informations sur la taille des fichiers.

Tout d'abord, récupérez le nombre de lignes du fichier de pedigree.

## Programmation

Emacs est avant tout écrit par des développeurs pour des développeurs. Il offre ainsi un grand nombre de commodités pour la programmation que nous allons présenter.

Tout d’abord, Emacs s’adapte au type de fichier que vous voulez éditer. Ainsi, si vous ouvrez un fichier portant une extension `.f90`, Emacs se mettra en mode `fortran90` (indication `F90-mode` dans la ligne de base du buffer, au niveau de la barre des menus, une rubrique spécifique s’affichera).

Les langages disposant d’un mode sur la version d’emacs installée sur DGA2 sont récapitulés dans le tableau 5

Language	Suffixe du fichier
Fortran 90	<code>.f90</code>
Fotran 77	<code>.f77</code>
C	<code>.c</code>
C++	<code>.cpp</code>
Perl	<code>.pl</code>
Shell	<code>.sh</code>
Awk	<code>.awk</code>

TABLE 5.1 – Langages courants ayant un mode emacs

### 5.1 Fonctionnalités avec Fortran 90

Le mode du langage permet de réaliser une coloration syntaxique du code source. Il permet d’effectuer également une indentation du code, en sélectionnant la zone à indenter et en tapant `C-M-q`.

Lors de l’écriture de code source emacs permet la complétion par exemple en fortran 90 si

une boucle “do” est étiquetée, en écrivant le mot clef de fin “End” puis en tapant sur tabulation, l’étiquette de fin vient s’ajouter toute seule.

On peut également naviguer facilement vers les sous-routines en utilisant le i-menu. Pour l’utiliser, il faut d’abord cliquer sur F90, puis add i-menu. On a alors une nouvelle rubrique dans la barre des raccourcis. en cliquant sur i-menu, on aura désormais un sous-menu avec une entrée pour chaque sous routine du programme.

L’appel du compilateur se fait par la commande `M-x compile`. Par défaut, emacs vous propose d’utiliser la commande `make -k` (nous détaillerons son utilisation plus tard) . Il est néanmoins possible d’effacer cette commande par défaut et de la remplacer par la commande classique de compilation. Quelque chose comme `f90 -o prog prog.f90` ou `gfortran -o prog prog.f90`. Dans ce cas, la complétion ne fonctionne pas à l’intérieur du mini-buffer, par contre le rappel des dernières commandes ainsi que les fonctions textes (couper coller) fonctionnent.

Une fois la commande de compilation entrée, un buffer est ouvert affichant le résultat de la compilation. En cas de problème de compilation, les messages d’erreurs sont affichés et un simple clique sur l’un de ces messages positionnera le curseur dans le fichier de code source sur le point ayant posé problème lors de la compilation. Ceci permet de corriger assez facilement ses erreurs.

Une fois vos corrections apportées, vous pouvez recompiler soit comme précédemment (`M-x compile`) à noter que du coup la dernière commande de compilation vous sera proposée par défaut. Vous pouvez également utiliser la commande `M-x recompile` qui est plus directe car elle ne vous demande pas confirmation de la commande de compilation.

Une interface avec le déboggeur gdb est possible (mais nous ne verrons pas cet aspect ici, car gdb n’est pas installé sous dga2).

Comme nous l’avons vu, on peut facilement agencer les buffers d’emacs. Une configuration envisageable est donc d’avoir 3 buffers d’ouverts, l’un comportant le programme à compiler, le second affichant le message de compilation et enfin le dernier comportant un buffer shell pour tester le bon fonctionnement du shell.

Ces quelques astuces apportent un certain confort pour la programmation.

## 5.2 programmation avancée

Nous présenterons ici quelques concepts avancés qui permettent de résoudre des problèmes rencontrés notamment sur des gros programmes.

### 5.2.1 makefile

Le makefile est un fichier regroupant des commandes de compilation. Son nom est invariable “Makefile”. Dans sa forme la plus simple il peut ressembler à cela :

```
Test: test.o
f90 -o test test.o
```

Qui correspond à la structure

```
Cible : dépendance(s)
commande
```

La cible est toujours suivie de “:” la seconde ligne commence toujours par une tabulation (mais emacs se chargera de vérifier que ceci est le cas) On peut rentrer des commandes de compilations, mais aussi des commandes shell plus simples comme cp mv etc. Cette présentation reste très succincte on pourra se référer à des documentations plus étoffés afin de construire un Makefile digne de ce nom <sup>1</sup>.

Enfin, le makefile est invoqué par la commande make. La première règle rencontrée est évaluée. L’évaluation d’une règle se fait en plusieurs étapes :

- Les dépendances sont analysées, si une dépendance est la cible d’une autre règle du Makefile, cette règle est à son tour évaluée.
- Lorsque l’ensemble des dépendances est analysé et si la cible ne correspond pas à un fichier existant ou si un fichier dépendance est plus récent que la règle, les différentes commandes sont exécutées.

L’avantage dans cette façon de procéder est de ne pas tout recompiler si seulement un programme à été modifié.

Si vous souhaitez tester, ou recompiler une partie du projet, vous pouvez faire appel à une cible particulière. Ainsi, make dépendance va juste compiler la cible “dépendance”

Pour créer un fichier makefile avec emacs, le plus simple est de créer dans le répertoire où se trouvent vos sources à compiler un fichier appelé makefile (C-x C-f , nom du fichier “makefile”).

Le mode Makefile permet de naviguer d’une cible à l’autre via les commande M-p et M-n.

Une fois votre fichier makefile écrit, vous pourrez désormais modifier un fichier de source et le recompiler par la commande Make (proposée par défaut lorsque l’on tape M-x compile).

---

1. Le manuel présent ici <http://gl.developpez.com/tutoriel/outil/makefile/> est déjà un mieux

### 5.2.2 Tags

Lorsqu'un projet fait appel à différentes sous-routines présentes dans des fichiers différents, le débogage peut devenir plus difficile. Heureusement pour simplifier ce travail l'utilitaire etags est là.

Lorsque vous avez 3 fichiers fortran e.g. fic1.f90, fic2.f90 et fic3.f90. Admettons que fic1.f90 fasse appel à des fonctions définies dans fic2.f90 et fic3.f90.

On peut alors faire appel au programme etag en ouvrant un buffer de shell dans le répertoire contenant les fichiers sources puis en tapant la commande :

```
etags *.f90
```

Cette commande va inspecter tous les fichiers .f90 présents dans le dossier courant et créer un fichier "TAGS" qui contient des infos sur quel fichier contient quelle fonction.<sup>2</sup>

À l'ouverture de fic1.f90 dans emacs, si dans le code source un appel à une fonction Fonc1 est fait, on pourra se placer sur le F de Fonc1 puis taper M-. Le minibuffer s'active et affiche "Find tag : (default fonc1)". Si vous désirez atteindre la définition de la fonction fonc1 validez (sinon, rentrez le nom de la fonction que vous souhaitez atteindre. Après validation, un buffer s'ouvre et positionne le curseur au niveau de la fonction fonc1.

### 5.2.3 CVS

Emacs permet de travailler sur un fichier sous le contrôle d'un serveur de version concurrente de type CVS.

Ces serveurs sont très utilisés dans le monde des projets open-source. Leur but est de gérer un code source, enregistrer toutes les modifications que celui peut connaître et vérifier que des modifications faites par deux personnes dans des parties différentes du code ne vont pas créer de problèmes. Le principe est illustré dans la figure 5.2, on a donc un depot present sur un serveur CVS qui gère un projet, puis chaque client peut demander à créer une copie locale du projet. Le client pourra effectuer sur sa copie des modifications et les soumettre au serveur qui se mettra à jour et proposera cette nouvelle version à l'ensemble des clients.

Ces serveurs ont le grand avantage de pouvoir revenir rapidement à n'importe quelle ancienne version d'un programme. Leur usage est néanmoins un peu compliqué et contraignant pour de petits programmes.

Lorsque l'on ouvre une copie locale d'un fichier source sous contrôle CVS, on voit apparaître dans la ligne de base la version de ce fichier. La première question à se poser est de savoir si nous travaillons sur la dernière version existante du fichier. Pour ce faire on va taper la com-

2. On pourra éventuellement créer une cible spécifique "tags" dans le makefile qui permettra de réaliser l'indexation des sous routines et fonctions automatiquement.

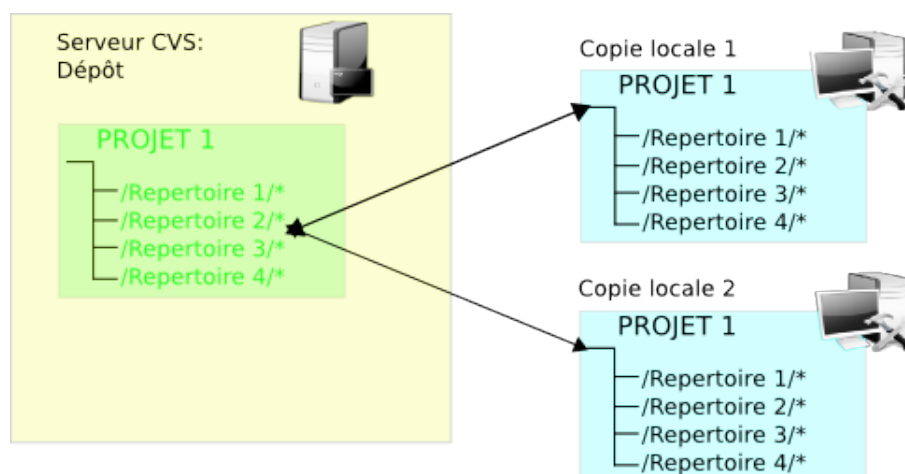


FIGURE 5.1 – Principe de la gestion sous CVS

mande “M-x cvs-examine“, un buffer s’ouvre listant l’ensemble des fichiers sous cvs présent dans le dossier ainsi que leur statut (U up-to date, M modifié) On se placera alors sur l’un des fichiers et on pourra soit le mettre à jour depuis le serveur, soit soumettre la version locale au serveur.

### 5.3 Un exemple

Nous allons à titre d’illustration réaliser la compilation d’un programme Fortran90.

Exemple le programme simul.f90.

Ouvrez le fichier Simul.f90 dans le répertoire ex4 Tout d’abord, afin de rendre la lecture du programme plus aisée, sélectionnez l’ensemble du programme et indentez le code. (Menu F90, indent subprogram) ensuite procédez à sa compilation par la commande M-x compile (un fichier makefile ayant été écrit pour l’occasion), puis validez .

Emacs ouvre un buffer et commence à nous insulter. Il s’agit des problèmes qu’il a rencontré lors de la compilation. Ces messages sont interactifs, ainsi en cliquant sur l’un d’entre eux, le curseur va aller se positionner sur le point où le compilateur a buté.

L’erreur rencontrée est due à une boucle non fermée, en ligne 58, ajoutez la balise end dans le texte, sauvegardez et recompiliez. On peut ré-invoquer la commande compile, ou taper la commande M-x recompile. La compilation devrait avoir lieu sans problèmes. Si tel n’est pas le cas, ou si vous désirez voir les modifications réalisées depuis la dernière sauvegarde du fichier, vous pouvez avoir recours au programme ediff. Celui-ci peut être invoqué en cliquant sur Tools, compare (puis en sélectionnant le type de comparaisons) La commande emacs est quant à elle M-x ediff. Trois nouveaux buffers s’ouvrent et indiquent les différences observées entre les deux fichiers à comparer.

```

MultiPrec=MultiPrec*formule1(position)
endif
end do
!Puisqu'on veut NumHap allant de 1 à Cum on ajoute 1
Numhap=NumHap+1
end subroutine Convert

subroutine ReverseConvert(formule1,Haplo1,NumHap)
!====
!Conversion d'un numéro d'haplotype en haplotype
!====
!Cette sous routine va convertir un entier (NumHap) en un vecteur contenant
!un haplotype (Haplo1) à l'aide d'une formule
!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!formule1 :Formule du nombre d'allèles à chaque marqueur (<=> base)
!Haplo1 :Haplotype de marqueurs correspondant au numéro d'haplotype
!NumHap :Numéro d'haplotype à convertir
!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
integer::NumHap,InMarq,Cumul
integer,dimension(1)::Haplo1(:,) ,formule1(:,)

!patites vérifications de cohérence
Cumul=0 !MR2
do InMarq=1,size(formule1,1)
Cumul=Cumul+formule1(InMarq)
end do
DL.f90 87% (394/0) CVS-1.5 (F90)---dim oct 19 12:09 0.40 Mail---
Repository : /home/francois/.cvs
Module : DL/DL/SRC
Working dir: ~/Travail/DL/DL/SRC/

In directory .:
Removed DL
committed 09-10-04 1.5 DL.f90
Modified Makefile
Modified utilselect.mod

----- End -----
-- last cmd: cvs -f commit -m <log message> DL.f90 --

*cv* All [9,38] (CVS: exit)---dim oct 19 12:09 0.40 Mail---
CVS process has completed in 'cvs*'

```

FIGURE 5.2 – Fenetres emacs CVS

La compilation effectuée, il vous faut désormais vérifier que le programme fonctionne correctement, premièrement ouvrez un buffer en mode shell. Dans un second temps placez vous dans le répertoire EX3, lancez le programme de simulation, celui-ci effectuera des simulations en se basant sur les fichiers pedigree et perf que vous avez précédemment créés.

Le programme vous demande un certains nombre de paramètres, imaginez un script shell qui pourra automatiquement vous donner ces paramètres. puis lancer le programme de simulation. Imaginez ensuite un script qui effectuera une centaine de simulation, pour vérifier qi votre programme fonctionne bien.



## Emacs et ESS

Nous allons ici présenter ESS aka. Emacs Speack Statistics. Il s'agit d'un mode permettant d'interagir avec SAS, R, Splus .Ce mode contrairement aux autres n'est pas inclus avec emacs. Il nous faudra donc l'installer<sup>1</sup>.

### 6.1 Mode SAS

Par défaut lorsque vous entrez dans un fichier SAS et que ESS est lancé, le fichier sera coloré en fonction de mots clef du langage. L'intérêt d'Emacs dans l'édition de programme SAS est d'avoir un réel éditeur de texte<sup>2</sup>

Si vous souhaitez soumettre votre programme à SAS, il faut entrer dans le mode SAS. par la commande `M-x SAS`. Emacs va alors lancer SAS, Il vous est ensuite demandé de renseigner le répertoire de travail. Une fois ce dernier renseigné Emacs est divisé en 4 buffers. Le premier contient le programme, le second correspond à un shell dans lequel les instructions soumises sont retranscrites. Le troisième buffer contient la log de SAS. Enfin,le quatrième buffer contient le listing des résultats.

La soumission du programme se fait par sélection du code à soumettre puis `C-c M-j`.

### 6.2 R et Emacs

Par défaut R est simplement un langage qui interprete les commandes dans un shell. L'inconvénient de ceci est de devoir taper toutes des instructions au fur et à mesure d'une analyse.

---

1. Si vous utiliser le fichier .emacs créé dans au chapitre 1, vous n'avez pas besoin de faire une telle installation  
2. Comprenez par là un éditeur qui par exemple se place en fin de ligne lorsque l'on appuie sur la touche FIN au lieu d'exécuter tout le programme SAS.

ESS propose de scinder votre écran en deux avec d'une part un fichier contenant les instructions à exécuter et d'autres part le shell d'interprétation. Donnant un écran ressemblant à la figure 6.1

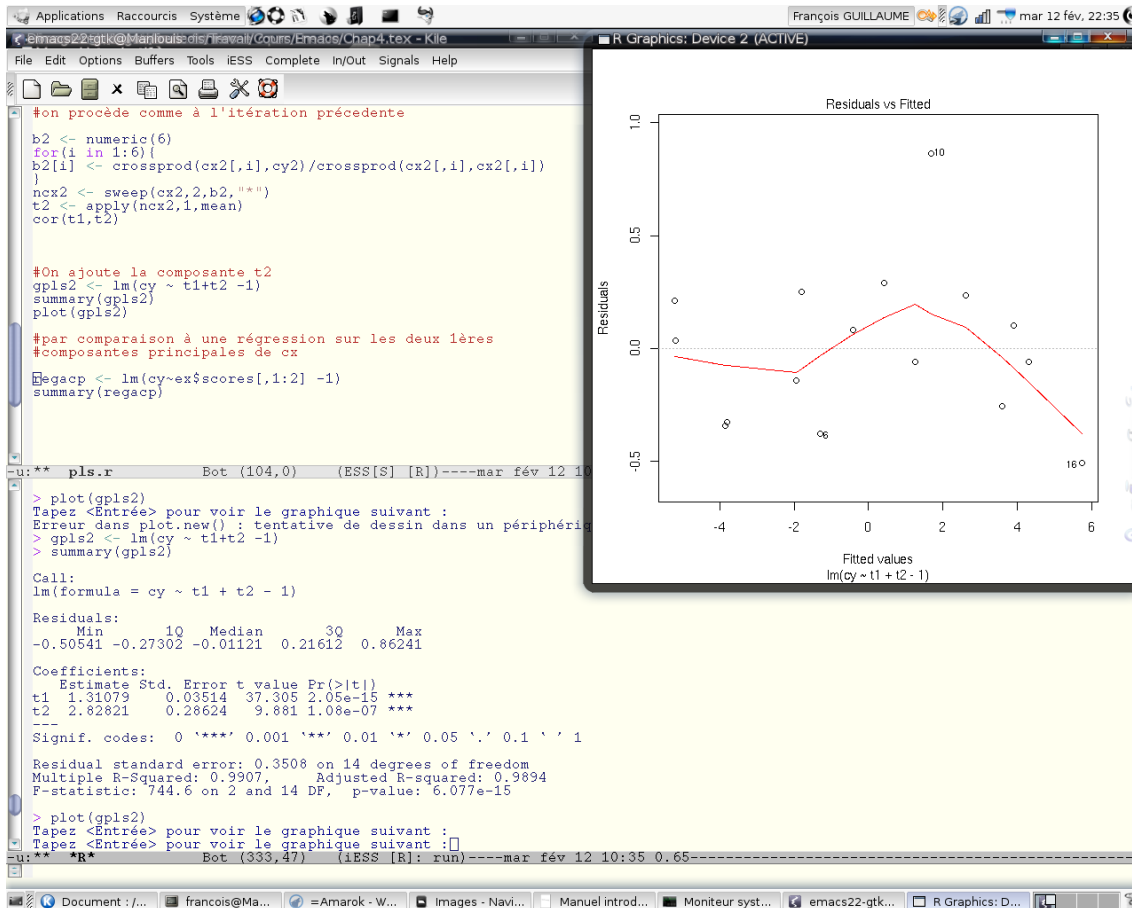


FIGURE 6.1 – Mode ESS. On notera que les graphiques sont présents dans un fenêtre X indépendante

Lorsque l'on ouvre ou crée un fichier source R (finissant par l'extension .r) le mode ESS s'active. Si vous désirez ne pas créer de fichier finissant par .r, le mode ESS est toujours activable via la commande M-x r-mode.

Il faut ensuite invoquer R en cliquant sur l'icone R par exemple. Le Minibuffer s'active et demande un répertoire de travail. Après validation, le buffer de R s'affiche et attends vos instructions. Pour soumettre un programme, il faut sélectionner le code à soumettre dans le buffer de source et le soumettre par le raccourci C-x M-j. (on peut bien sûr également utiliser les boutons prévus à cet effet).

Une fois les instructions soumises, elles sont retranscrites dans le buffer R. Vous obtenez alors les résultats de vos analyses. Vous pourrez alors par exemple sélectionner ces résultats pour les recopier dans un fichier de résultats par exemple.

## 6.3 Exercice

Nous allons ouvrir le fichier `Analyse.r` présent dans le répertoire `Ex4`. Il suffit dans un premier temps d'exécuter les commandes du programme.

Dans un second temps, après avoir lu la documentation de la fonction `hist`, commencez à améliorer votre graphique.



# L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X est langage de programmation orienté dans la mise en page. C'est d'ailleurs avec ce langage que ce manuel est écrit.

Les avantages de L<sup>A</sup>T<sub>E</sub>X sont entre autre une mise en page de grande qualité et un très bon rendu des formules mathématiques. De plus on a plus le soucis de la mise en page (celle-ci étant faite à l'aide d'algorithmes) Le pendant négatif de ceci est d'avoir un langage de plus à apprendre, de ne pas voir instantanément le résultats de ce que l'on écrit<sup>1</sup> car il faut passer par une étape de compilation avant de pouvoir lire le document final.

Il existe plusieurs mode L<sup>A</sup>T<sub>E</sub>X sous emacs. Le meilleur étant AucTeX. Sous dga2 il n'existe en fait que le mode classique L<sup>A</sup>T<sub>E</sub>X. Nous allons donc nous contenter de présenter ce dernier.

## 7.1 Rudiments de L<sup>A</sup>T<sub>E</sub>X

Nous allons présenter quelques bases de L<sup>A</sup>T<sub>E</sub>X afin de pouvoir créer vos premiers documents.

Un document L<sup>A</sup>T<sub>E</sub>X sera tout d'abord un fichier dont l'extension sera .tex. Il comporte deux parties :

- Une partie déclarative
- Une partie corps du texte

### Partie déclarative

Cette partie doit au minimum indiquer le type du document, son format. Ensuite, on fait appel à des packages spécifiques qui vont ajouter des fonctionnalités à L<sup>A</sup>T<sub>E</sub>X, grâce à des décl-

---

1. mais on s'en passe vite...n'ayez pas peur

ration

`usepackage[option]Titre` Package Deux packages essentiels si vous rédigez en français seront donc “babel” et “Fontenc“ . Le prmeier appliquera les règles de typographie française, tandis que l’autre accèptera les touches accentuées.

Lorsque vous avez un fichier tex qui fonctionne bien, il peut être intéressant de le réutiliser comme base pour de nouveaux documents.

## Le corps du texte

Contre toute attente, il contient ... du texte. En fait, la logique de L<sup>A</sup>T<sub>E</sub>X est d’avoir recours à des environnements, dans lesquels une mise en forme spéciale est appliquée. Ainsi, le corps du texte est un environnement commençant par le mot clef `begindocument` et finissant par `enddocument`. Entre ces deux balises vous pouvez entrer le texte que vous désirez. Vous pouvez également insérer d’autres environnements, les plus fréquents sont : `itemize` (insertion d’une liste), `tabular` (insertion d’un tableau), `figure` (insertion d’une image) et bien sûr `displaymath` (insertion d’éléments mathématique).

Le code L<sup>A</sup>T<sub>E</sub>X peut être commenté à l’aide du signe `%` . De nombreuses macros, peuvent également être programmé, mais ceci dépasse largement le nut de ce manuel. Nous nous arrêterons ici pour la description de L<sup>A</sup>T<sub>E</sub>X. si vous souhaitez en savoir plus de très bonnes références existe sur internet.

## 7.2 Mode L<sup>A</sup>T<sub>E</sub>X

Ce mode se charge automatiquement à l’ouverture d’un fichier `.tex`. La saisie du texte est aidée par des fonctions de complétion automatique. Une fois que vous avez tapé la commande `beginquelquechose` la commande `C-c C-e` va automatiquement compléter cet environnement par `endquelquechose`

L<sup>A</sup>T<sub>E</sub>X étant un langage à compiler, la commande de compilation pourra être appelé par la commande `C-c C-f`. Un buffer s’ouvre alors avec les informations de compilation. Pour voir le résultat de la compilation, la commande à exécuter est `C-c C-v`. Enfin, pour imprimer votre résultats la commande est `C-c C-p..`

## 7.3 Mode Postscript

Postscript est un langage utilisé par les imprimantes. Les fichier de types postscripts peuvent donc être directement dirigés vers une imprimante pour être imprimé. On peut néanmoins ne

visualiser ces documents sans les imprimer. Lors de l'ouverture d'un fichier .ps ou .eps par emacs le mode postscript est activé. Vous pouvez alors démarrer un interpréteur postscript C-c C-s. Puis envoyer le contenu du fichier vers cet interpréteur C-c C-b.

### 7.4 Exemple

Nous allons ici réaliser un document simple avec L<sup>A</sup>T<sub>E</sub>X. Dans un premier temps, nous allons regarder le fichier Bilan.tex présent dans le répertoire Ex5. Lisez les commentaires pour comprendre ce qui se trouve dans notre fichier et compilez une première fois le document.

Dans un second temps, vous pouvez ajouter un texte introductif pour expliquer ce que vous avez réalisé comme simulation.

En vous basant, sur la sortie automatique de R (Summary.tex) voyez comment on écrit un tableau en L<sup>A</sup>T<sub>E</sub>X et essayez de voir comment à l'aide d'Emacs, vous pourriez faire un tableau qui listerait les fichiers présents dans le répertoire Ex4. (à l'aide d'un buffer de shell, de la commande ls -al et des sélections rectangulaire).





## Gadgets d'Emacs

D'après la légende Emacs sait tout faire sauf le café ! Nous allons donc passer en revue quelques éléments de ce que emacs sait faire et qui ne font pas partie des utilisations les plus courantes.

### 8.1 ftp et telnet

Pour accéder à un site distant via ftp ou telnet, on peut utiliser les mode ftp et telnet appelé respectivement par : M-x ftp et M-x telnet.

Le mode ftp ne permet pas la complétion. Par contre on pourra tirer parti des fonctions d'édition de texte d'Emacs. Ainsi, on peut faire un ls pour lister les fichiers d'un dossier puis copier les fichiers que l'on veut rapatrier via la commande get.

Le mode telnet ne permet pas non plus de complétion.

Attention, ces protocoles (Ftp et telnet) sont utilisés en mode texte...n'essayez donc pas de lancer une application avec interface graphique.

Il existe également un mode navigateur web qui n'est pas installé sur dga2, mais qui permet néanmoins une bonne navigation.

### 8.2 L'agenda et les calculatrices

Tout d'abord un agenda. On y accède en cliquant sur tools puis agenda. Parmi les fonctions utiles, on a le renseignement des vacances, d'anniversaires et l'exportation au format  $\text{\LaTeX}$  de calendrier. Parmi, les fonctions moins utiles mais amusantes, on a la conversion des dates dans

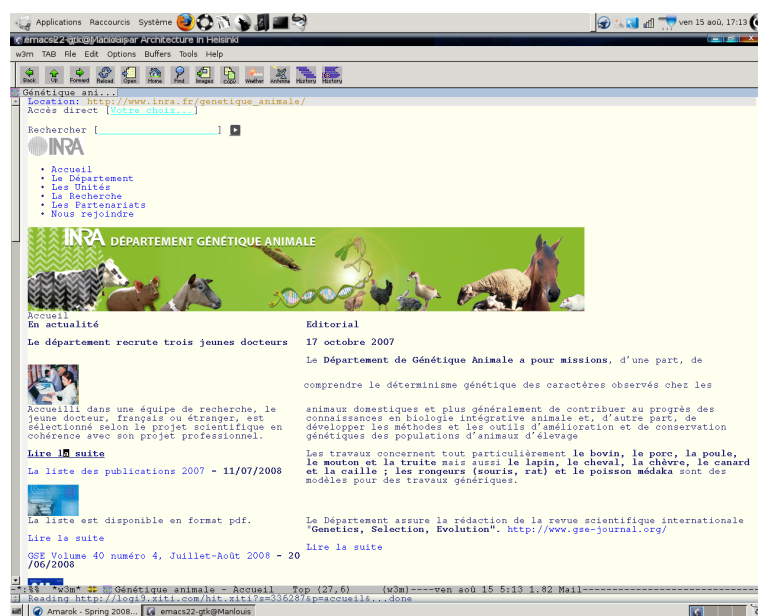


FIGURE 8.1 – Mode web : la page du département

différents calendrier (chinois, Hébreux, maya<sup>1</sup> !), les différentes phases de la lune et l'ensemble des jours fériés dans le monde (ça pousse à être très cosmopolite !

La calculatrice simple. On peut la démarrer via la commande `M-x calculator`, ou en cliquant sur `Tools/calculator`. Les raccourcis clavier pour des fonctions un peu compliquées sont indiquée dans le sous-menu `calculator`. Depuis la version 22 emacs possède également une version calculatrice programmable.

### 8.3 Lecteur de mail

Lorsque vous recevez un mail celui-ci est notifié dans la ligne de base. Un clique sur le mot `Mail`, ouvrira un buffer avec ce mail. Le mode `Mail` permet de recevoir et d'envoyer des mails., la limite est l'impossibilité de voir des fichiers joints.

### 8.4 Jeux

Bien sûr, un éditeur de texte qui propose la conversion des dates selon le calendrier hébreux ne pouvait pas ne pas comporter de jeux ! Il y a donc 11 jeux disponibles sous emacs. Ceux-ci peuvent être lancé de diverses façons, il ne tient qu'à vous de trouver comment.

1. Si vous arrivez à comprendre ce calendrier faite moi signe

## 8.5 Dessin

Certes il s'agit ici également d'un outil futile dont les résultats ne sont pas à proprement parler des chefs d'oeuvres, mais c'est une fonctionnalité qui peut permettre de faire quelques schémas au format texte. M-x

## 8.6 Le psychanalyste

Ici, on arrive à la plus futile des fonctionnalités d'Emacs, le psychanalyste. Il s'agit en fait d'un programme qui vous pose des questions et essaye de tenir une conversation cohérente avec vous. C'est assez amusant à faire ...une fois. On y accède en cliquant sur help/Emacs psychotherapist ou M-x doctor.