

Programmer avec Emacs

Laurent Réveillère

Enseirb-Matmeca / IPB
Département Télécommunications

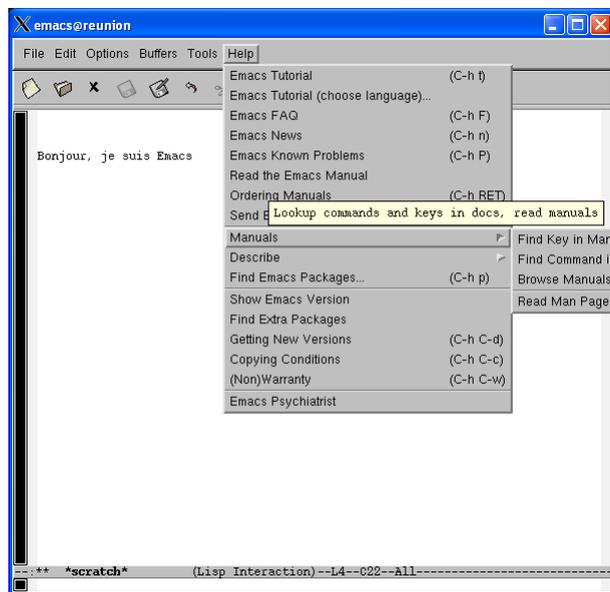
`Laurent.Reveillere@ipb.fr`
`http://reveille.vvv.enseirb-matmeca.fr/`

Techniques de programmation avancée

Qui es-tu Emacs?

- ◆ Éditeur de texte polyvalent
 - Programmation : C, C++, Java, shell, ...
 - Lettres, rapports : LaTeX
 - Mail, News
 - Bloc note
 - Tout est possible ☺
- ◆ Éditeur extensible
 - Programmation en LISP
 - Système de macros
 - Tout est reconfigurable ...
- ◆ Éditeur multiplateformes
 - GNU/LINUX, BSD, SunOS, MacOS, Windows ...

Emacs par l'exemple



Conventions

- ◆ RET frappe de la touche Enter
- ◆ DEL frappe de la touche Delete
- ◆ SPC frappe de la touche Espace
- ◆ ESC frappe de la touche Esc
- ◆ C-x frappe de la touche Ctrl et de la touche x simultanément (C-X produit le même effet)
- ◆ M-x frappe différée de la touche Esc puis de la touche x
- ◆ M-x toto frappe différée de la touche Esc puis de la touche x puis de la séquence de caractères toto
- ◆ S-right frappe simultanée de la touche shift et du bouton droit de la souris
- ◆ S-C-right frappe simultanée de la touche shift, de la touche Ctrl, et du bouton droit de la souris
- ◆ C-x y frappe de la touche Ctrl et de la touche x simultanément, puis frappe différée de la touche y

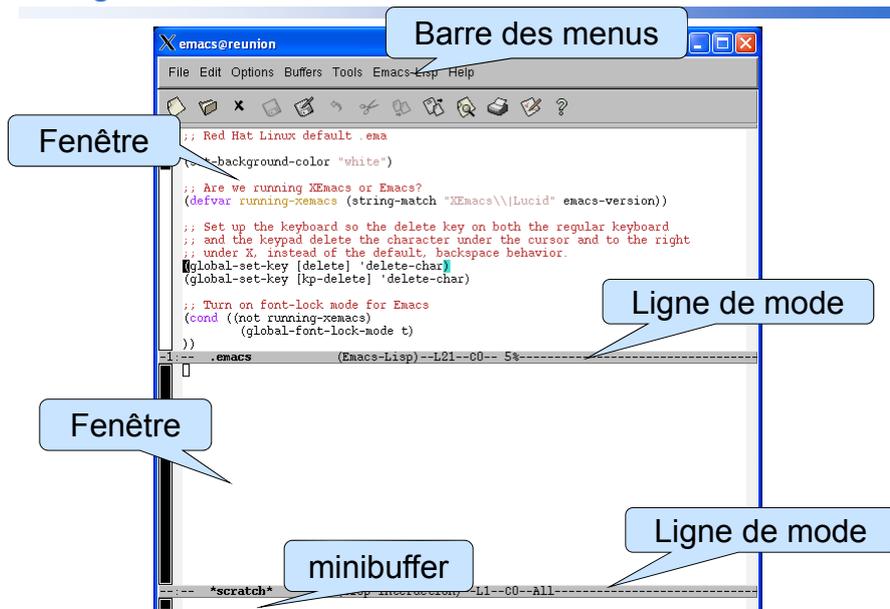
Buffer, Fichier, Fenêtre

- ◆ Emacs n'édite pas de *fichiers*
 - Emacs édite des *buffers* (*tampons*)
- ◆ Notion de fichier
 - Suite d'octets sur un support quelconque
- ◆ Notion de buffer
 - Copie de travail d'un fichier
 - Toutes les modifications apportées à un texte sont en fait apportées à un buffer
 - Certains buffers ne sont pas associés à des fichiers sur le disque (buffers spéciaux)
- ◆ Notion de fenêtre
 - Espace sur l'écran qui affiche un buffer
 - Contient une ligne de mode qui donne des indications sur le buffer en cours d'édition
- ◆ Donc...
 - Lorsque l'utilisateur saisit du texte, c'est dans la fenêtre qu'il le saisit, mais c'est dans le buffer que sont faites ses modifications.
 - Lorsque l'utilisateur change de buffer la fenêtre reste la même mais le buffer affiché change.

Techniques de programmation avancée

5

Organisation de l'écran



Techniques de programmation avancée

6

Lancer EMACS

- ◆ \$ emacs

Ouverture de Emacs avec un buffer par défaut : `*scratch*`

- ◆ \$ emacs nomfichier

- Le fichier existe

- » Ouverture de Emacs avec chargement du fichier dans un buffer et ouverture d'une fenêtre pour afficher le buffer

- Le fichier n'existe pas

- » Ouverture de Emacs avec création d'un buffer « `nomfichier` » et ouverture d'une fenêtre pour afficher le buffer
- » Il faut sauvegarder le buffer pour créer le fichier
- » Il faut avoir les droits suffisant en écriture

Zone d'écho (echo area / minibuffer)

- ◆ Ligne située en bas de la fenêtre Emacs

- ◆ Deux rôles différents

- Afficher des messages : erreur ou informations
- Entrer des commandes

- ◆ Tous les messages affichés dans la ligne d'échos sont copiés dans le buffer `*Messages*`

- ◆ Exercice

- Lancer emacs
- Ouvrir un fichier : File → Open File
- Utiliser la touche TAB pour la complétion sur les noms de fichier comme en shell

Ligne de mode

- ◆ Chaque fenêtre possède sa propre ligne de mode
- ◆ Contient des informations sur le buffer édité
 - nom du buffer
 - les modes en cours
 - si le buffer contient des modifications non sauvegardées
 - position de la fenêtre dans le buffer
 - ...

Manipulation de fichiers

- ◆ $C-x C-f nom$
 - Ouverture du fichier *nom* dans la fenêtre courante
 - Le buffer édité est mis en arrière plan, et reste dans la liste de buffers
 - Un nouveau buffer est créé

Manipulation de fichiers (suite)

- ◆ C-x C-s
 - Enregistre le contenu du buffer courant dans le fichier sur lequel il est ouvert
- ◆ C-x s
 - Enregistre le contenu de tous les buffers ouverts
- ◆ C-x C-w *nom*
 - Enregistre le contenu du buffer édité sous le nom *nom*

Manipulation de buffers

- ◆ C-x b *buffer*
 - Sélectionne ou crée (s'il n'existe pas) le buffer *buffer* dans la fenêtre courante
- ◆ C-x C-b
 - Ouvre une nouvelle fenêtre contenant la liste de tous les buffers ouverts
- ◆ C-x k
 - Détruit le buffer courant

Manipulation de fenêtres

- ◆ C-x 2
 - Découpe verticalement la fenêtre courante en deux fenêtres
- ◆ C-x 3
 - Découpe horizontalement la fenêtre courante en deux fenêtres
- ◆ C-x o
 - Sélectionne la fenêtre suivante
- ◆ C-x 0
 - Détruit la fenêtre sélectionnée
- ◆ C-x 1
 - Détruit toutes les fenêtres sauf la fenêtre courante
- ◆ C-x +
 - Réorganise toutes les fenêtres, et leur donne une taille égale

Exemple à saisir

```
/* hello.c */
#include <stdio.h>
#include <stdlib.h>

void Hello(void)
{
    printf("Hello World\n");
}
```

Exécution de commande

- ◆ M-X nomcommande
 - Exécute la commande nomcommande
- ◆ Si une commande requiert des arguments
 - Saisis dans le minibuffer
- ◆ De nombreuses commandes sont accessibles par des raccourcis clavier
 - Exemple: `find-file` et `C-x C-f`
- ◆ Astuce
 - `C-g` annule la commande en cours d'exécution
 - Très utile 😊

Interaction avec Emacs

- ◆ Point
 - Endroit particulier d'un buffer
 - » Où les commandes d'édition prennent effet
 - » Où la saisie de texte est insérée
 - Correspond au curseur du terminal
 - Curseur repéré par un caractère en inverse vidéo
 - Le point se trouve juste avant ce caractère
- ◆ Saisie
 - Mode insertion
 - Mode overwrite

Déplacements (principaux raccourcis)

- ◆ C-a
 - Déplace le point au début de la ligne courante
- ◆ C-e
 - Déplace le point à la fin de la ligne courante.
 - Le curseur se trouve donc après les dernier caractère de la ligne
- ◆ M-a
 - Déplace le point au début du paragraphe courant
- ◆ M-e
 - Déplace le point à la fin du paragraphe courant
- ◆ M-b
 - Déplace le point au début du mot courant
- ◆ M-f
 - Déplace le point à la fin du mot courant
- ◆ M-<
 - Déplace le point au début du buffer
- ◆ M->
 - Déplace le point à la fin du buffer

Déplacements (suite)

- ◆ M-g N
 - Aller à la ligne N
- ◆ M-x goto-line N
 - Idem

Marque et région

- ◆ Marque
 - Position particulière dans un buffer (comme le point)
- ◆ Région
 - Zone de texte dans un buffer situé entre le point et la marque
 - Permet d'appliquer des commandes sur une zone
 - Sélection par la souris en mode fenêtré
- ◆ Déplacement de la marque
 - C-SPC
 - » Fixe la marque au niveau du point
 - C-x C-x
 - » Échange le point et la marque

Copie de texte

- ◆ Kill-ring
 - Espace commun à tous les buffers ou Emacs range les textes détruits précédemment
 - » Sorte de pile circulaire
- ◆ Commandes de destruction de texte
 - C-k
 - » Détruit la ligne courante à partir du *point*
 - C-w
 - » Détruit la région (du point jusqu'à la *marque*)
 - M-DEL
 - » Détruit le mot précédent (ou le mot courant si le curseur est au milieu d'un mot)
 - M-d
 - » Détruit le mot suivant (ou le mot courant si le curseur est au milieu d'un mot)

Copie de texte (suite)

- ◆ Commandes de "collage" de texte et de manipulation de *kill-ring*
 - ☐ C-y
 - » Insère le bloc de texte situé à la fin du *kill-ring* au niveau du point
 - ☐ M-y
 - » Juste après un C-y, remplace le texte inséré par le bloc de texte précédent dans le *kill-ring*
 - ☐ M-w
 - » Enregistre la région dans le *kill-ring* sans la détruire

Recherche

- ◆ Recherche incrémentale de texte dans un buffer
 - ☐ S'affine au fur et à mesure des lettres saisies
 - ☐ Commence au niveau du point
- ◆ Commandes
 - ☐ C-s *chaîne*
 - » Recherche incrémentale en avant
 - ☐ C-r *chaîne*
 - » Recherche incrémentale en arrière
- ◆ Remarque
 - ☐ S'arrête dès qu'une occurrence du mot recherché est trouvé
 - ☐ Reprise de la recherche sur les occurrences suivantes par C-s ou C-r
- ◆ Recherches d'expressions régulières
 - ☐ C-M-s ou C-M-r
 - ☐ Exemple
 - » C-M-s [-a-zA-Z_\.\.]+\@[-a-zA-Z0-9_\.\.]+\.(fr|com|net|org\)

Remplacement

- ◆ Deux types de remplacement
 - Remplacement inconditionnel
 - Remplacement conditionnel
- ◆ Pour chaque type, deux variantes
 - Remplacement de chaînes
 - Remplacement d'expressions régulières
- ◆ Remplacement inconditionnel
 - `M-x replace-string chaîne remplacement`
 - » Remplace sans confirmation toutes les occurrences de la chaîne *chaîne* par la chaîne *remplacement*.
 - `M-x replace-regexp regexp remplacement`
 - » Remplace sans confirmation l'expression régulière *regexp* par la chaîne *remplacement*

Remplacement (suite)

- ◆ Remplacement conditionnel
 - `M-% chaîne remplacement`
 - » Remplace toutes les occurrences de la chaîne *chaîne* par la chaîne *remplacement* en demandant confirmation à l'utilisateur.
 - `M-x query-replace chaîne remplacement`
 - » Effectue la même tâche que la précédente
 - `M-x query-replace-regexp chaîne remplacement`
 - » Remplace toutes les occurrences de l'expression régulière *regexp* par la chaîne *remplacement* en demandant confirmation à l'utilisateur.

Modes

◆ Modes *majeurs*

- Modifient le comportement d'Emacs en fonction du type de texte édité
- Modes pour la programmation C, l'édition LaTeX, ...
- Chaque buffer possède un mode majeur et un seul

◆ Modes *mineurs*

- Options supplémentaires ajoutés aux buffers
- Aucun ou plusieurs modes mineurs possible simultanément sur un même buffer
- Exemple
 - » auto-fill passage à la ligne automatique
 - » auto-save sauvegarde automatique
 - » iso-accents production de lettres européennes
 - » scroll-bar barre de défilement à côté de chaque buffer
- Chargement de mode par `M-x nomdumode-mode`

Aide contextuelle

- ◆ C-h c *séquence*
 - Donne le nom de la commande rattachée à la séquence de touches *séquence*
- ◆ C-h f *fonction*
 - Affiche la documentation pour la fonction *fonction*
- ◆ C-h k *séquence*
 - Donne le nom de la commande rattachée à la séquence de touches *séquence* ainsi que sa documentation
- ◆ C-h m
 - Affiche de la documentation sur le mode majeur utilisé
- ◆ C-h t
 - Entre dans le tutorial d'Emacs
- ◆ C-h v *variable*
 - Affiche la documentation sur la variable *variable*
- ◆ C-h w *commande*
 - Affiche la séquence de touches permettant d'exécuter la commande *commande*

Programmation C sous Emacs

- ◆ Mode d'édition spécifique
 - M-x C-mode
 - Chargement automatique du mode lorsque le fichier possède l'extension .c
- ◆ Couleurs spécifiques pour les mots clés du langage, chaînes, etc.
 - Suivant configuration de Emacs
- ◆ Indentation du code
 - Indentation d'une ligne par la touche `TAB`
 - Indentation d'une région par la commande `indent-region`

Trucs et astuces

- ◆ Complétion
 - Complète le début du mot situé sur le point en fonction des mots présent dans le buffer
 - Commande : M- /
 - Utilisation
 - » Plus d'excuses pour ne pas avoir des noms de variables explicites
 - » Fonctionne quelque soit le mode chargé

Macros

- ◆ Permet d'automatiser une tâche répétitive et complexe
 - Comme les scripts en shell
- ◆ Définition d'une macro
 1. C-x (Début de l'édition de la macro
 2. Taper la suite de commande à enregistrer
 3. C-x) Fin de macro
- ◆ Commande utiles
 - C-x e
 - » Exécuter la dernière macro définie
 - M x name-last-kbd-macro
 - » Nommer la dernière macro définie
 - M x macro
 - » Exécuter la macro macro

Commandes utiles

- ◆ Lancer un shell dans une fenêtre Emacs
 - Utile pour tester un programme sans sortir d'emacs
 - M-x shell
- ◆ M-x compile
 - Lance l'utilitaire de compilation (make par défaut)
- ◆ M-x gdb
 - Lance le debugger gdb

Configuration de Emacs

- ◆ Le fichier .emacs dans votre home contient votre configuration de Emacs
- ◆ Permet de :
 - Définir des variables
 - Charger des modes
 - Définir des macros
 - Modifier des raccourcis claviers
 - Etc ...
- ◆ Écrit en Emacs LISP
 - Aller voir sous Google

Exemple de .emacs

```
(set-background-color "white")

;; Quelques raccourcis claviers
(global-set-key [M-left]      'backward-kill-word)
(global-set-key [A-left]     'backward-kill-word)
(global-set-key [f2]         'save-buffer)
(global-set-key [f3]         'find-file)

;; Turn on font-lock mode for Emacs
(global-font-lock-mode t)
(turn-on-font-lock)

(transient-mark-mode 1)           ; Affiche la selection
(setq inhibit-startup-message t)  ; Il est gentil mais pas de pub...
(setq column-number-mode t)       ; Affiche le numéro de colonne
```

Version complète en ligne