	Cycle préparatoire 1^{ère} année Prise en main d'un éditeur de code <i>Florent DEVIN, Peio LOUBIÈRE</i>	
	<i>Matière :</i> Informatique	<i>Date :</i> Octobre 2018
		<i>Durée :</i>
		<i>Nombre de pages :</i> 10

1 Introduction

Le but de ce Tp n'est pas de fournir une explication complète d'emacs, mais de vous fournir les bases nécessaires à l'utilisation d'emacs pour la programmation. Avant de commencer à parler d'édition de code, il est important d'expliquer la notion d'éditeur de code.

Un éditeur de code permet d'éditer du code dans un certain formatage. Il sert généralement à créer des programmes. Un éditeur de code n'est pas un éditeur de texte ¹ encore moins un traitement de texte.

Un traitement de texte offre à l'utilisateur des outils de formatage de texte, que ne propose pas un éditeur. Un traitement de texte sert par exemple à taper une lettre, un cours, ...

Vous découvrirez par la suite aussi la notion de formateur de texte. Un formateur de texte est a mi-chemin entre un éditeur de texte, et un traitement de texte. Le formateur permet via un langage de "programmation" de créer du texte mis en forme (c'est à dire avec une mise en page correct).

Pour ne citer que les plus connus, KWord est un traitement de texte; emacs, vi, des éditeurs de code; notepad, gedit des éditeurs de texte, et LaTeX un formateur de texte. Comme tout outil, un éditeur de code nécessite un investissement temporel, parfois non négligeable, afin de se sentir à l'aise.

1.1 Comprendre la notion de fichier et de buffer

En général, les éditeurs n'éditent jamais directement un fichier. Ils copient le contenu du fichier dans un buffer, et édite ce dernier. Le fichier ne change que lors de la sauvegarde.

Un buffer est une sorte de fichier temporaire. À ce titre, les buffers possèdent un nom. En général, le nom du buffer est le même que celui du fichier que vous souhaitez éditer. Ce qui permet de modifier un fichier, et éventuellement de le sauvegarder, si vous appréciez les modifications, ou non.

Gardez toujours à l'esprit qu'il vous faut sauvegarder pour que les modifications apportées au buffer soit prises en compte.

1.2 Modes

Les modes sont un des avantages d'emacs. En effet, lorsqu'on utilise un éditeur, il n'est pas rare de vouloir qu'il réagisse par rapport à ce que l'on veut faire. Par exemple, si l'on programme en Pascal, on voudrait que l'éditeur mette en forme automatiquement le programme. De même, si l'on tape du texte, on préfère que l'éditeur passe à la ligne automatiquement, ...

Les modes permettent à emacs de se comporter comme on l'attend. Sous emacs, il existe deux types de mode : les *major-modes*, et les *minor-modes*.

Les major-modes fixent les règles générales d'édition. Normalement, emacs essaye de "deviner" le mode approprié. En fait, il s'appuie sur les extensions des fichiers que l'on édite, ou essaye d'analyser le fichier pour connaître son major-mode. S'il ne trouve aucun mode particulier, il se met dans le mode *Fundamental*. C'est le mode le plus général, qui ne procure aucune action particulière.

Les minor-modes sont des fonctionnalités supplémentaires que l'on peut ajouter comme par exemple, le correcteur orthographique (utile lorsque vous tapez du \LaTeX), le retour à ligne automatique, ...

1. Même si certain éditeur de code, comme emacs, peuvent très bien éditer du texte

1.3 Démarrer emacs

Pour démarrer emacs, il suffit de taper la commande emacs suivi du nom du fichier que vous désirez éditer. Si le fichier n'existe pas, emacs le crée. En tapant la commande emacs test vous devriez arriver sur une fenêtre du style de la figure 1. Bien sûr il est possible d'éditer un fichier existant, dans

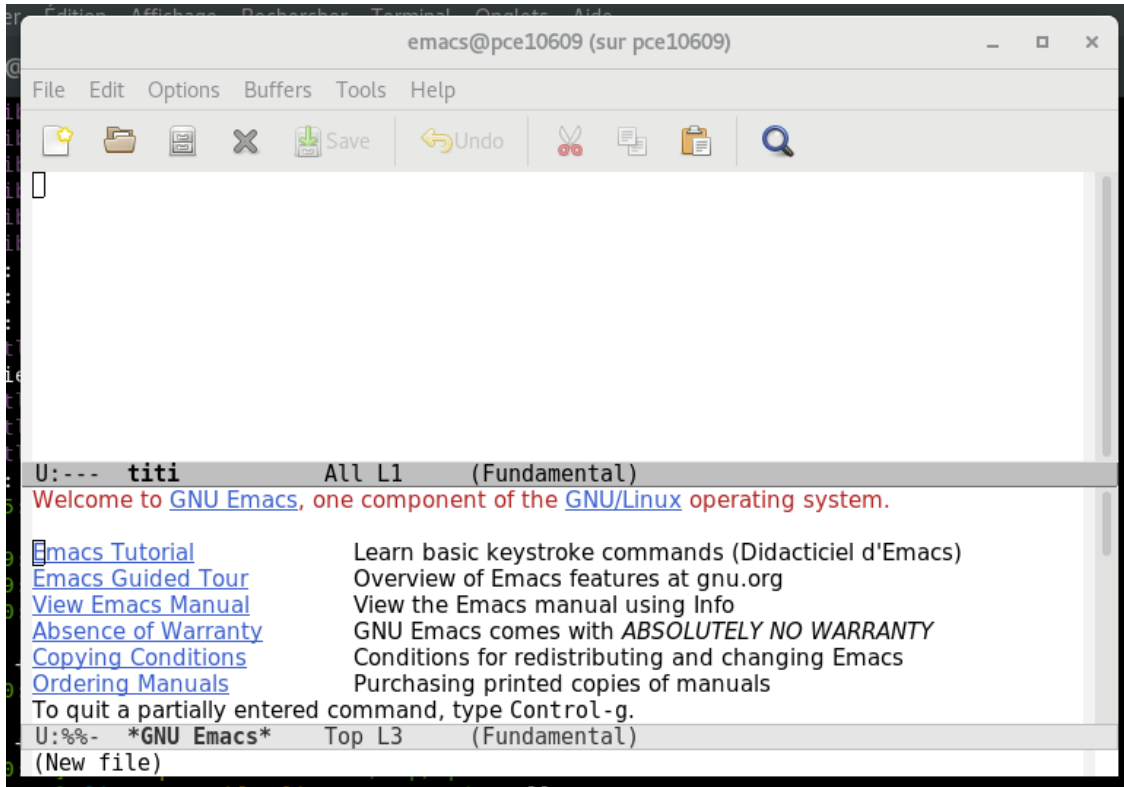


FIGURE 1 – emacs édite le fichier titi qui n'existe pas encore

ce cas emacs lit le fichier et l'affiche, voir la figure 2. Si vous omettez le nom de fichier, alors emacs démarre sans charger de fichier comme sur la figure 3.

Emacs est séparé en plusieurs zones. La première, tout en haut, c'est la barre de menu, comportant les menus File, Edit, Options, Buffer, Tools, Help. En dessous, la barre d'outils qui comporte de base les bouton Open, Dired, Save, Print, Cut, Copy, Paste, Undo, Spell, Replace, Mail, Info, Compile, Debug, News. Vient ensuite un zone réservée pour les onglets qui indique quels sont les buffer du même major-mode en cours d'édition, ensuite la zone de texte proprement dite. Après cette zone de texte se trouve ce qui s'appelle le mode-line. Sur la figure 1 on voit dans l'ordre U, nous nous occuperons pas de cette information, puis ---, qui indique l'état du fichier. Si vous voyez 2 astérisque (**) c'est que le fichier a été modifié, si vous voyez %, le fichier que vous édité est en lecture seule, enfin %* signifie que vous avez modifié un fichier en lecture seule. Suit un texte entre parenthèses. Il vous indique en rouge les major-modes actifs, ainsi qu'en vert les différents minor-modes activés. Après cette zone vous trouverez une ligne (sur la figure 1 elle comporte le texte (New file)) que l'on appelle le mini-buffer. C'est dans cette zone qu'emacs "communique" avec vous.

1.4 Commandes emacs

Maintenant que vous avez eu un aperçu visuel d'emacs, vous êtes prêt pour apprendre les commandes. Cependant, avant d'apprendre ces commandes, il est nécessaire de prendre quelques lignes pour présenter les habitudes...

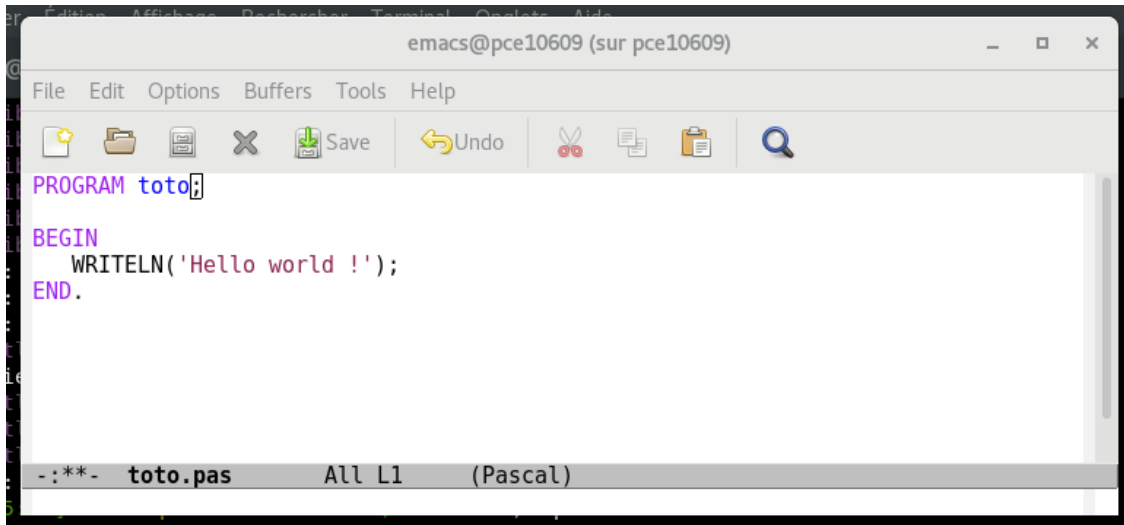


FIGURE 2 – emacs édite le fichier toto.pas

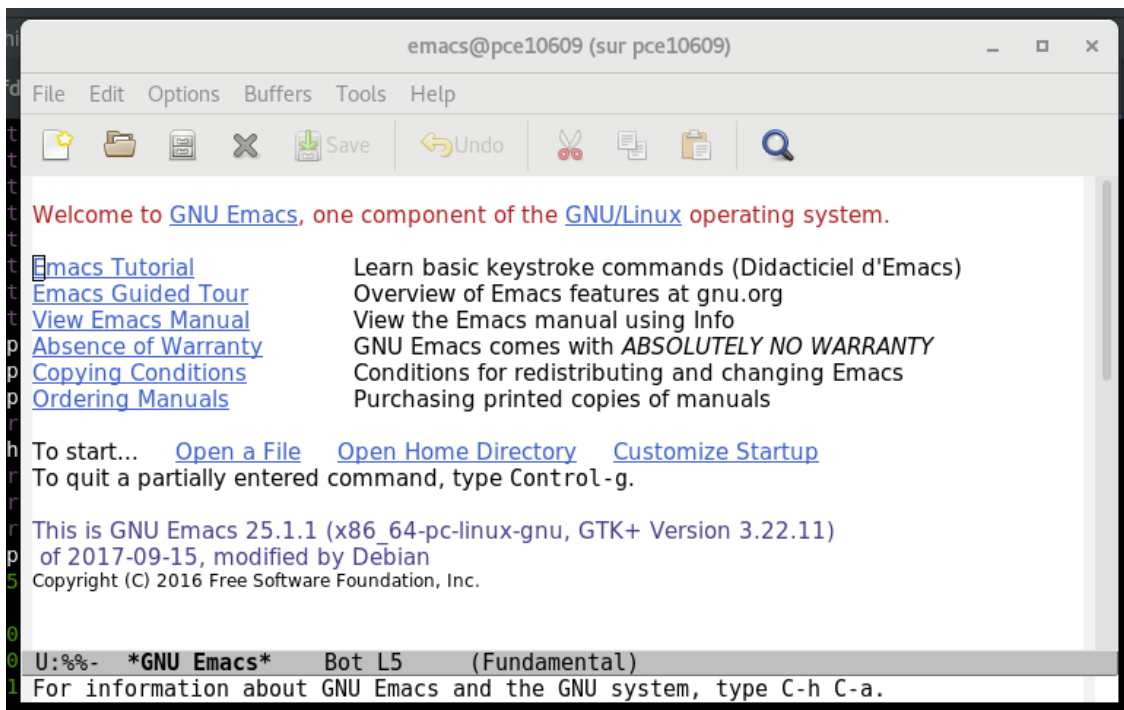


FIGURE 3 – emacs démarre

Il existe sous emacs de centaines voire des milliers de commandes possibles ! Il n'est pas nécessaire de les connaître toutes pour utiliser emacs, comme il n'est pas nécessaire de connaître tous les mots d'un dictionnaire pour parler une langue. . . Toutes les commandes possèdent un nom par lequel on va appeler la commande. Cependant les noms de commandes sont relativement long. Pour éviter d'avoir à taper les commandes pour exécuter différentes actions, emacs réalise ce qu'on appelle un "binding" : une commande est associée une séquence de touches (appelé aussi raccourcis-clavier). Dans l'énoncé des bindings vous verrez apparaître les conventions suivantes :

- C-x où x représente n'importe quel caractère. Ce qui signifie qu'il faut appuyer sur la touche Control et la touche x
- M-x où x représente n'importe quel caractère. Ce qui signifie qu'il faut appuyer sur la touche Meta et la touche x. En général, la touche Meta est la touche Echap, ou la touche Alt
- C-x qqch où qqch représente une touche ou une séquence de touche. Il faut donc appuyer sur C-x puis sur la séquence indiquée
- Enfin, si aucune touche n'est disponible, ou que l'on ne connaît pas le binding associé à la commande, il reste la possibilité de taper le nom de la commande après avoir tapé la séquence M-x

De nombreuses commandes sont accessibles par l'intermédiaire des menus. Prenez quelques instants pour parcourir les différents menus, et repérer celles qui pourraient vous intéresser. Vous trouverez, dans les menus à droite de la commande, son raccourcis clavier si ce dernier existe.

1.5 Édition de fichiers

Pour ouvrir un fichier il suffit de taper la séquence suivante C-x C-f ou M-x find-file. Il n'est pas nécessaire de relâcher la touche Control lors de cette séquence. Vous pouvez appuyer sur la touche Control, la maintenir enfoncée (pas trop quand même) pendant que vous appuyez sur les touches x et f. Dès que vous appuyez sur cette séquence de touches, emacs vous demande via le mini-buffer quel fichier vous souhaitez éditer : *Find file: ~/*. Il vous faut donc entrer le nom d'un fichier. La complétion marche aussi dans le mini-buffer. Pour entrer le nom du fichier que vous désirez éditer, vous avez le choix entre trois méthodes :

- entrer le nom, et le chemin d'accès exact au fichier;
- entrer le début du nom, puis appuyer une fois sur Tab, pour qu'emacs fasse la complétion. Si il existe plusieurs possibilités de complétion, emacs, vous indique dans une seconde fenêtre la liste des fichiers correspondants;
- appuyer deux fois sur la touche Tab, et sélectionner via la souris le fichier qui vous intéresse.

1.6 Sauvegarde de fichiers

Pour sauvegarder un fichier en cours d'édition, la séquence C-x C-s². Emacs vous indique alors (toujours dans le mini-buffer) : *Wrote nom du fichier*, s'il y avait des modifications à sauvegarder. Sinon il vous indique : *(No changes need to be saved)*. Contrairement à Word³, ou à d'autres éditeurs du même style, emacs sauvegarde réellement le contenu affiché. Word sauvegarde aussi les attributs du texte et autres information qui n'ont rien à voir avec le texte, ce qui en fait un éditeur de texte inutilisable pour la programmation. De plus Word, plutôt que de sauvegarder le texte, sauvegarde le texte plus les modifications qui ont été faites depuis un certain temps. Ce qui augmente considérablement la taille du fichier de sauvegarde.

Parfois, vous allez éditer un fichier, pour le modifier et le sauvegarder sous un autre nom. Il vous faut alors utiliser la séquence C-x C-w⁴. Dans ces conditions emacs vous demande sous quel nom vous désirez sauvegarder le buffer en cours d'édition.

Il est plutôt conseillé de copier le fichier sous le nom que vous désirez, puis de modifier cette copie, que de faire l'inverse. En effet, avec l'habitude, vous utiliserez rapidement les séquences de touches. Vous vous apercevrez qu'il n'est pas rare que la séquence C-x C-s "tombe" facilement sous les doigts.

2. Cette séquence correspond en fait à l'appel de la fonction Save File du menu File

3. Un traitement de texte sous Windows

4. ou Save File As



Lors de l'édition d'un fichier, veillez à sauvegarder fréquemment. En effet, si vous ne prenez pas soin de sauvegarder votre travail à intervalle régulier, vous vous exposez à perdre tout ce que vous aviez tapé, à cause d'une coupure de courant, ou autre événement externe

Emacs travaille sur des buffers et des fichiers. Lors de la sauvegarde d'un fichier, il génère automatiquement un fichier nom du fichier~. Ce fichier est un fichier comportant l'ancienne version du fichier que vous avez édité. De plus, emacs sauvegarde automatiquement le buffer (à intervalle régulier) dans le fichier #nom du fichier#. Nous verrons plus tard l'utilisation de ces différents fichiers.

1.7 Quitter emacs

Pour les plus impatients C-x C-c. Si vous quittez emacs alors que des buffers ont été modifié mais non sauvegardé emacs vous demande si vous désirez les sauvegarder. Une fois tous les buffers sauvegardés, emacs se termine. Si vous ne souhaitez pas sauvegarder un buffer, mais que vous désirez quand même quitter, emacs vous demandera : *Modified buffers exist; exit anyway? (yes or no)*. Assez explicite, non ?

1.8 Résumé

Voici un résumé des commandes que vous devriez connaître.

Séquence	Nom de commande	Action
C-x C-f	find-file	Ouvre un fichier
C-x C-s	save-buffer	Sauvegarde un fichier
C-x C-w	write-file	Enregistrer le buffer sous un nom
C-x C-c	save-buffers-kill-emacs	Quitte emacs
C-h ? ?	help-command	Affiche de l'aide

2 Un peu de manipulation

1. Lancer emacs.
2. Éditer un nouveau fichier appelé fichier_test, notez le major-mode qui est associé à ce fichier.
3. Changer le major-mode en mode texte⁵.
4. Taper le texte suivant : Un premier petit exercice.
5. Dans un shell regardez où se trouve le fichier fichier_test.
6. Quitter emacs, sans sauvegarder.
7. Relancer emacs en éditant le fichier fichier_test.
8. Que vous dit emacs?
9. Toujours dans un shell afficher le fichier #fichier_test#, grâce à une commande shell. Que constatez vous?
10. Récupérez le fichier de sauvegarde automatique d'emacs grâce à la commande recover-file, puis en appuyant sur entrée à la question *Recover file :~/*. Que constatez vous?
11. Ajouter le texte La suite de l'exercice. Sauvegardez. Regardez les fichiers présents dans votre répertoire de travail. Lesquels, générés par emacs, sont présents?
12. Ajouter maintenant le texte J'en sais rien. Sauvegardez. Regardez les fichiers présents dans votre répertoire de travail. Lesquels, générés par emacs, sont présents?
13. Expliquez les différences des réponses entre les 2 questions précédentes.
14. Sauvegardez le fichier en cours sous le nom fichier_test2. Modifiez ce fichier. Regardez les fichiers présents dans votre répertoire de travail. Lesquels, générés par emacs, sont présents?
15. Ouvrez le fichier fichier_test modifier le. Utilisez la séquence C-x s. À quoi sert cette commande?

5. La commande qui permet de changer le major-mode en mode texte est text-mode

3 Quelques pas dans l'édition

3.1 Introduction

Normalement, vous devriez en cours d'année utiliser plusieurs fois emacs. De nombreuses fois pour programmer, d'autres fois pour taper du texte. À la longue vous vous apercevrez qu'emacs est un éditeur très pratique, notamment grâce à ses modes, et fonctions de configuration. Nous détaillerons par la suite comment adapter emacs pour ses propres besoins.

3.2 Mode texte, et Fill mode

Avant de taper quoi que ce soit, regardez le mode-line. Si vous apercevez le mot *Fill*, c'est que vous vous trouvez dans le minor-mode auto-fill-mode. Dans ce mode si vous tapez du texte au "kilomètre"⁶, emacs retourne automatiquement à la ligne. C'est ce que beaucoup d'éditeurs de texte appellent *Word wrap*. Cette fonctionnalité est pratique lorsque vous tapez du texte.

Cependant il se peut que cette fonctionnalité soit active lorsque vous entrez un programme. Dans ce cas, cette fonctionnalité devient gênante. Vous pouvez la désactiver, ou l'activer si vous n'êtes pas dans ce minor-mode, par la commande `auto-fill-mode`, ou en faisant un clic droit sur le minor-mode que vous souhaitez désactiver.

3.3 Déplacement dans le texte

Pour se déplacer dans le texte que l'on tape, il existe plusieurs solutions. La première, la plus simple, c'est de cliquer avec la souris à l'endroit où vous désirez vous positionner. Les flèches du clavier sont aussi actives, c'est à dire que vous pouvez vous déplacer, grâce à celles-ci. Il existe cependant d'autres façons de se déplacer.

La plus ancienne, qui existait avant que la souris ne soit intégrée aux environnements graphiques, sont les séquence suivantes : C-f (forward-char), C-b (backward-char), C-p (previous-line), C-n (next-line).

Si vous souhaitez vous déplacer plus rapidement, vous pouvez en appuyant sur la touche Control, et les flèches, vous déplacer d'un mot en avant ou en arrière, ou d'un groupe de lignes au dessus, ou en dessous.

Ces 2 dernières actions n'ont pas le même déplacement que lors de l'appui sur les touches PageUp, PageDown.

3.4 Copier - couper / coller

Pour couper, ou copier du texte, il faut d'abord le sélectionner. Pour le sélectionner, vous pouvez le faire à la souris, en appuyant sur le bouton gauche de la souris à l'endroit où vous désirez commencer la sélection, de bouger la souris là où vous souhaitez terminer la sélection, et de relâcher la souris.

Vous pouvez encore un fois procéder d'une autre façon. Vous vous placez là où vous voulez commencer la sélection. Appuyez sur la séquence C-SPACE, emacs vous dit *Mark set*. Puis déplacer vous.

Une zone en gris apparaît. C'est la zone sélectionnée. Vous pouvez aussi sélectionner du texte en maintenant la touche Shift enfoncé, puis en utilisant les flèches de déplacement.

Un appui sur la séquence C-g arrête la sélection (que vous l'ayez réalisée à la souris, ou au clavier)⁷. Une fois que vous avez sélectionné le texte voulu, vous pouvez le couper⁸, ou le copier⁹ grâce aux séquence C-w, M-w respectivement. L'appui sur C-y permet de coller¹⁰ le texte là où vous vous trouvez. Le presse-papier n'est pas effacé par cette action, vous pouvez donc coller le texte autant de fois que vous désirez.

6. C'est à dire sans taper de retour chariot

7. Cette séquence est en fait une séquence qui permet de sortir de beaucoup de situation. C-g signifie *quit* l'action en cours

8. Couper signifie que vous effacer le texte pour le mettre dans une zone particulière, souvent appelée presse-papier

9. Copie le texte dans le presse-papier

10. Afficher le contenu du presse-papier

Un petit résumé des différentes utilisation des copier/coller, et autres.

- Pour déterminer une région et l'effacer :
 - Positionner le curseur au début de la zone que vous désirez effacer.
 - Appuyer sur C-SPACE. Emacs affiche *Mark set*.
 - Déplacer le curseur à la fin de la zone.
 - Appuyer sur C-w.
 - Emacs efface la sélection
- Pour déplacer une partie d'un texte :
 - Effacer le texte par la méthode précédente.
 - Positionner le curseur là où vous le désirez.
 - Appuyer sur C-y.
- Copier un texte :
 - Réaliser la méthode pour effacer du texte, mais taper M-w, à la place de C-w.
 - Positionner le curseur là où vous le désirez.
 - Appuyer sur C-y.

Enfin, il arrive fréquemment, que l'on désire annuler une action, pour cela appuyer sur C-_.

4 De la pratique toujours de la pratique

1. Si vous avez quitté emacs, démarrez le de nouveau.
2. Créez un fichier `facto`.
3. Dans ce fichier, tapez le texte suivant :

```
(*
-----
-- Fichier      : facto.pas
-- Auteur       : Florent Devin <fd@eisti.eu>
-- Date de creation : Fri Oct 19 07:07:54 2018
--
-- But          : But
-- Remarques    : Aucune
-- Compilation  : fpc
-- Edition des liens : fpc
-- Execution    : shell
-----
*)
PROGRAM Facto;

(*
-----
-- Fonction     : fact (n : Integer) : Integer
-- Auteur       : Florent Devin <fd@eisti.eu>
-- Date de creation : Fri Oct 19 07:08:16 2018
--
-- But          : calculer la factorielle
-- Remarques    : Aucune
-- Pré conditions : n >= 0
-- Post conditions : calculer la factorielle
-----*)
FUNCTION fact (n : Integer) : Integer;
VAR
  res : Integer;
BEGIN
```

```

IF (n <= 1) THEN
BEGIN
res := 1;
END
ELSE
BEGIN
res := n * facto(n-1);
END;
fact := res;
END;

(*Début du programme principal*)
VAR
n : Integer;
BEGIN
WRITE('Entrez un nombre positif : ');
READLN(n);
IF (n >= 0) THEN
BEGIN
WRITELN('La factorielle de ', n, ' vaut :', fact(n));
END
ELSE
BEGIN
WRITELN('Vous n''avez pas entré un nombre positif');
END;
END.

```

4. Sauvegardez.
5. Exécutez la commande suivante : font-lock-mode. Que constatez-vous?
6. Exécutez la commande suivante : c-mode. Que constatez-vous?
7. Exécutez la commande suivante : font-lock-mode. Que constatez-vous?
8. Quittez emacs.
9. Rechargez emacs, et le fichier programme.
10. Quittez emacs.
11. Dans un shell renommer le fichier programme en programme.c.
12. Lancer à nouveau emacs, et ouvrez le fichier programme.c. Que constatez-vous?
13. Vérifier si le minor-mode *Font* est actif. Sinon démarrez-le via la commande font-lock-mode
14. Sélectionnez tout le texte, et tapez la commande indent-region. Que constatez-vous?

5 Juste pour rire

1. Mettez vous au tout début du fichier. Appuyez sur la séquence C-SPACE. Vous venez de marquer le début d'un bloc.
2. Placer vous sur le début de la dernière ligne du programme, *grâce au clavier* afin de sélectionner tout le texte. Puis appuyer sur la séquence C-x r t. Emacs vous dit *String rectangle* :. À cette question répondez le texte suivant : test. Que constatez-vous?
3. Répétez la première étape.
4. Placer sur au début de la dernière ligne du programme, juste après le texte test. Appuyer sur la séquence C-x r k. Que constatez-vous?

6 Emacs et Pascal

Cette année, nous vous fournissons une configuration “minimale” afin qu’emacs soit correctement configuré pour une utilisation dans les cours d’informatique de classe préparatoire première année. Pour cela, il vous faut suivre ces étapes :

1. Quittez emacs.
2. Décompressez l’archive `emacsConf.tgz` dans votre répertoire principal (`~/`)
3. Lancez emacs (si vous ne constatez pas de changement, c’est que vous n’avez pas réalisé correctement l’étape précédente).
4. Renseignez votre `<login>` qui vous est demandé dans le mini-buffer;
5. puis votre mot de passe.
6. Patientez un certain temps, le temps qu’emacs télécharge tous les paquetages nécessaires.
7. À la fin, vous devriez arriver à l’image 4
8. Quittez emacs
9. Lancer la commande `emacs ~/.emacs.el`
10. Modifier les lignes 2 et 3.
11. Éventuellement modifier la ligne 469, pour changer de navigateur.

Votre emacs est configuré pour pouvoir éditer du code pascal. Certains snippets ont été défini pour vous. Pour voir ces snippets, entrez un mot clef, et appuyer sur la touche tabulation. Les mots clefs définis sont: `proc`, `case`, `while`, `for`, `fn`, `record`, `repeat`, `fic`, `com`, `if`, `ife`.

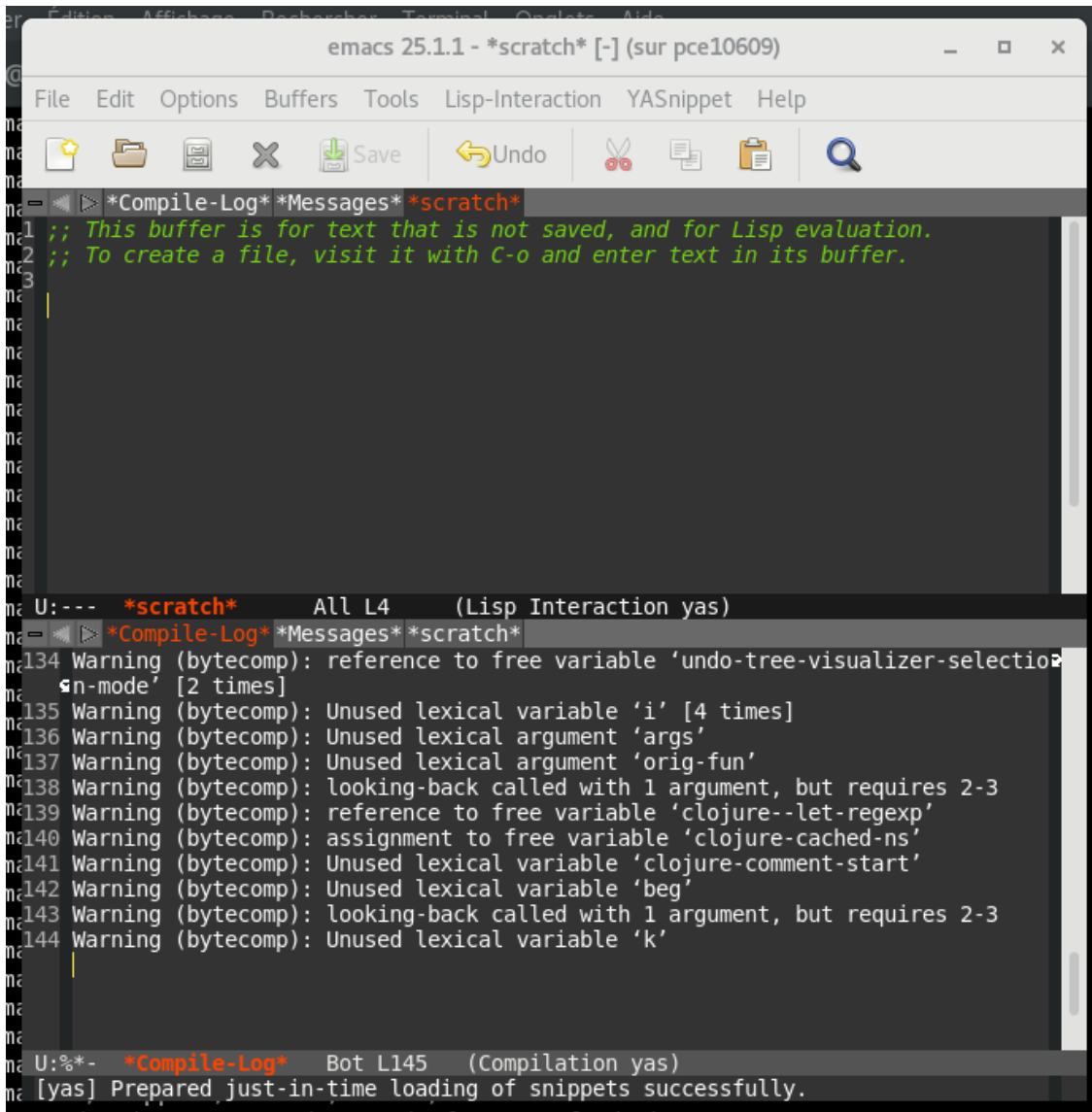


FIGURE 4 – Emacs est configuré