Environnement de développement sous xubuntu

Concours Informatique — TP d'Algorithmique

Écoles Normales Supérieures — Session 2016

Utilisateur : `tpalgo` Mot de passe : `tpalgo`

Attention : ne pas mettre les ordinateurs en veille !

Il est conseillé de placer tous ses fichiers dans le dossier personnel, qui s'ouvre par défaut lorsque l'on ouvre un navigateur de fichiers et/ou un terminal.

La clef USB Pensez à faire des sauvegardes régulières sur celle-ci ; en cas de panne de votre machine, une machine de substitution vous sera attribuée, mais seules les données présentes sur la clef USB pourront y être transférées.

Éditeurs de texte et terminal

Deux éditeurs de textes sont fournis par défaut, Gedit et Emacs, ainsi que deux environnements dédiés, Pyzo (Python) et Scilab.
Tous sont accessibles dans `Menu des applications > Développement`.

Gedit C'est un éditeur de texte classique, à préférer si l'on ne connait pas Emacs. Il faudra cependant lancer les programmes via un terminal : aucune fonctionnalité n'est intégrée à l'éditeur.

Emacs Il est doté de plus de fonctionnalités, dont un mode interactif pour Caml Light/OCaml (tuareg-mode), mais utilise des raccourcis claviers non-conventionnels. Les combinaisons de touches sont rappelées dans les menus (C = `Ctrl`, M = `Alt`). Quelques combinaisons usuelles :

- `Ctrl-w` pour couper, `Alt-w` pour copier, `Ctrl-y` pour coller
- `Ctrl-x` puis `Ctrl-s` pour sauvegarder

Le terminal On travaille dans un *dossier courant*, rappelé à gauche entre le « `:` » et le « `$` » (`~` correspond au dossier personnel). Pour utiliser une commande, il suffit de la taper, puis d'appuyer la touche `Entrée`. Quelques commandes utiles :

- `cd` pour se déplacer entre les dossiers (modifie le dossier courant) :
  - `cd dir` pour entrer dans le dossier `dir`
  - `cd ..` pour remonter d'un dossier
  - `cd` tout seul pour revenir au dossier personnel
- `ls` pour afficher les fichiers présent dans le dossier courant
- Touche `Haut` pour ressaisir une commande précédemment exécutée
- `./fichier` pour exécuter `fichier` (celui-ci doit être un exécutable)
- `... > fichier` pour écrire le résultat de `...` directement dans `fichier` (exemple : `python fichier.py > resultat.txt`)
- `Ctrl-C` pour interrompre un programme (utile en cas de boucle infinie)
- `Ctrl-D` pour insérer un caractère EOF (permet de quitter le mode interactif)

Attention : pour copier-coller du contenu dans le terminal, `Ctrl-C` et `Ctrl-V` ne fonctionnent pas : il faut utiliser les combinaisons `Maj-Ctrl-C` et `Maj-Ctrl-V`.

Caml Light — OCaml

Si vous souhaitez utiliser la bibliothèque `num`, il faut rajouter `caml_all` (Caml Light) ou `nums.cma` (OCaml) directement après les commandes `camllight` ou `ocaml`.

Dans le terminal Pour interpréter l'intégralité du fichier `fichier.ml`, les commandes à utiliser sont `camllight < fichier.ml` ou `ocaml fichier.ml`.
Il est possible de lancer Caml en mode interactif [1] à l'aide des commandes `camllight` ou `ocaml` seules. On peut les précéder de `ledit` (`ledit camllight` ou `ledit ocaml`) afin de bénéficier des touches `Gauche`, `Droite` et `Haut`.

Emacs — tuareg-mode Le mode tuareg permet de faire du Caml en mode interactif. Il utilise les raccourcis suivants :

- `Ctrl-c` puis `Ctrl-b` pour interpréter tout le fichier
- `Ctrl-c` puis `Ctrl-r` pour interpréter la partie surlignée
- `Ctrl-c` puis `Ctrl-e` pour interpréter la commande où se situe le curseur.

La première fois, il faut confirmer (touche `Entrée`) le choix de l'interpréteur en bas de l'écran : `camllight` (par défaut) ou `ocaml` (à taper à la place de `camllight`).

Aucune documentation n'est installée pour ces langages.

Python 2 — Python 3

Les bibliothèques `numpy` et `scipy` sont installées.

Pyzo Utilisez `Ctrl-Entrée` pour interpréter votre code. Le langage par défaut est Python 3, mais on peut le remplacer par Python 2 dans `Shell > Edit shell configurations...` en choisissant comme exécutable `/usr/bin/python2.7`.

Dans le terminal Pour interpréter l'intégralité du fichier `fichier.py`, les commandes à utiliser sont `python fichier.py` (Python 2) ou `python3 fichier.py` (Python 3). Il est aussi possible de lancer Python en mode interactif [1] à l'aide des commandes `python` ou `python3`.

Toutes les documentations sont installées.

C — C++ — Pascal — Java

Ces langages sont à compiler dans le terminal, à l'aide des commandes suivantes :

- `gcc fichier.c`
- `gpc fichier.pp`
- `g++ fichier.cpp`
- `javac Class.java`

Les trois premières commandes génèrent un exécutable `a.out`, qui doit être lancé avec la commande `./a.out`. Pour java, il faut utiliser `java Class`. Pour les projets utilisant plusieurs fichiers, la commande `make` est disponible.

À l'exception des pages `man` (C/C++), aucune documentation n'est installée.

Scilab

Pensez à utiliser SciNotes pour sauvegarder votre travail ! Les résultats intermédiaires n'étant pas affichés, utiliser `disp()` pour afficher les variables.

1. Attention : pensez à bien sauvegarder votre code !