

## Introduction

---

Outre des compétences théoriques qui sont abordées dans d'autres parties de ce document, la maîtrise d'un système informatique implique également une excellente connaissance des outils informatiques qui sont inclus dans ce système. Cette connaissance se construit bien entendu en utilisant activement ces logiciels. Cette section comprend les informations de bases sur quelques logiciels importants qui sont utilisés dans le cadre du cours. Elle est volontairement réduite car de nombreux logiciels pourraient faire l'objet de livres complets. Les étudiants sont invités à proposer des améliorations à cette section sous la forme de pull-requests via <https://github.com/obonaventure/SyllabusC>.

## Éditeurs

---

De nombreux éditeurs sont utilisables pour manipuler efficacement du code source en langage C. Chaque étudiant doit choisir l'éditeur qui lui convient le mieux. En voici quelques uns :

**vi(1)** est un des premiers éditeurs à avoir été écrit pour Unix. Il a fortement évolué, et reste un outil de choix pour de nombreux administrateurs systèmes. De nombreux tutoriels permettent d'apprendre rapidement **vi(1)**, dont <https://developer.ibm.com/tutorials/l-vi/>

**emacs** est un autre éditeur fréquemment utilisé sous Unix. Il existe de très nombreuses extensions à emacs qui lui permettent de faire tout ou presque, y compris de jouer à des jeux comme Tetris. Son extensibilité peut rebuter certains utilisateurs. De nombreux tutoriels sont disponibles sur Internet, dont <https://www.gnu.org/software/emacs/tour/>

**gedit** est l'éditeur de base dans l'environnement graphique **GNOME** utilisé dans les distributions Linux.

**Sublime Text** est un éditeur de texte léger et facile d'utilisation. La version complète est payante, mais il est possible d'utiliser la version gratuite à volonté. Cet éditeur de texte est un bon choix pour les débutants qui n'auraient jamais touché à la programmation.

**Atom** est un éditeur de texte facile d'utilisation et proposant de nombreuses extensions. Il a été développé par l'équipe de GitHub, et profite donc d'une interface simple pour le version control des projets en utilisant Git (voir la section consacrée à Git).

**Visual Studio Code** est un éditeur de texte proposant également de nombreuses extensions, et régulièrement utilisé dans l'entreprise. Il permet, entre autres, de profiter d'un terminal de commandes sur la même fenêtre que l'éditeur de texte.

**eclipse** est un environnement complet de développement écrit en Java pour supporter initialement ce langage. De nombreuses extensions à **eclipse** existent, dont **CDT** qui permet la manipulation efficace de code source en langages C et C++.

Pour les débutants, les éditeurs conseillés sont Sublime Text pour sa simplicité, Atom pour son intégration de Git (qui sera largement utilisé lors du projet), ou Visual Studio Code pour sa popularité et son accès direct à un terminal.

## Gestionnaires de code partagé

---

Dans de nombreux projets informatiques, il est nécessaire d'utiliser des outils qui permettent d'organiser efficacement le partage du code source entre plusieurs développeurs. On parle de **version control**. Les plus anciens gestionnaires de code sont **cvs** ou **rcs**. Ces logiciels ont été créés lorsque Unix était utilisé sur des mini-ordinateurs qui servaient tout un département. Aujourd'hui, les logiciels de gestion de code source s'utilisent en combinaison avec des serveurs Internet pour permettre un partage large du code source. Les plus connus sont :

**Git** qui est décrit dans ce syllabus. Il s'agit de l'outil de version control le plus utilisé, et donc également celui que vous utiliserez. Il est aussi décrit dans plein d'autres ressources comme **git-scm** contenant le livre Pro Git, une **vidéo** faite par le créateur de ce [sites](#), ou encore [la documentation de GitHub](#). **site**

**subversion** qui est décrit dans ce syllabus également.

**mercurial**

**bazaar**

## Compilateurs

---

Le compilateur C utilisé dans de nombreuses distributions Linux est **gcc(1)**. C'est un compilateur open-source développé activement dans le cadre du projet GNU par la **Free Software Foundation**. Nous utiliserons principalement **gcc(1)** dans le cadre de ce cours.

Il existe des alternatives à **gcc(1)** comme **llvm** que nous utiliserons lorsque nous analyserons le code assembleur généré par un compilateur C. Les variantes commerciales de Unix utilisent généralement des compilateurs propriétaires, dont par exemple **Oracle Studio** ou la **suite de compilateurs** développée par **intel**.