Programmation C

Fiche technique n° 3 : utiliser `Visual Studio Code`

Compiler et déboguer un projet C avec `Visual Studio Code`

`Visual Studio Code` (`VS Code`) est un éditeur : il permet de saisir du texte (par exemple, du code) et de le sauvegarder dans des fichiers. Une fois configuré, on peut compiler et déboguer un projet C directement depuis `VS Code`. Ce n'est pas obligatoire : on peut aussi utiliser le compilateur `gcc` et le débogueur `gdb` dans un terminal. Le terminal peut, au choix, tourner dans `VS Code` ou être un terminal externe. Cette fiche explique comment configurer `VS Code` pour des projets C, et comment l'installer, avec `gcc` et `gdb`, sur un système personnel (Linux, macOS ou Windows).

1 Configuration, à faire avant le TP 1

Chaque dossier de TP que nous vous fournirons contient un sous-dossier `.vscode` déjà configuré. Cependant, la partie de la configuration décrite ici est à faire avant le TP 1.

1.1 Nettoyage de VS Code existant (optionnel mais recommandé)

Le CREMI impose un quota de données sur le compte principal. S'il est dépassé, le compte est limité. Or, `VS Code` produit beaucoup de données cachées. On peut « rediriger » ses répertoires de sauvegarde vers le dossier `~/espaces/travail`, où on dispose d'un quota nettement plus élevé. C'est ce que fait la commande suivante, en réinitialisant au passage les répertoires de configuration de `VS Code`. Notez qu'à cause de cette réinitialisation, il faudra réinstaller les extensions que vous utilisez (ce qui est rapide).

```
$ /net/ens/vscode/reset.sh
```

1.2 Installation d'une configuration pour le langage C

Lancez ensuite la commande suivante :

```
$ cd; bash <(curl -fsSL https://www.labri.fr/perso/zeitoun/teaching/26-27/C/files/vscode.sh)
```

Que fait cette commande ?

- Elle bloque l'utilisation de Copilot, qui peut vous coûter cher s'il se déclenche en TP noté. Ce n'est pas un verrou incontournable : vous pouvez éditer le fichier de configuration pour le ré-autoriser.
- Par défaut, `VS Code` demande de faire confiance aux répertoires visités. Sans réponse, la compilation et le débogage sont bloqués. La commande permet de ne plus jamais voir cette question.
- La commande installe des extensions C : l'extension officielle Microsoft (débogueur, coloration syntaxique), Clangd (analyse du code et les diagnostics en temps réel), et un jeu de thèmes.
- Elle fait que les caractères Unicode (en particulier, les caractères accentués) seront acceptés dans les chaînes et les commentaires sans être surlignés par `VS Code`.
- Elle fait que les sources seront correctement formatés par le logiciel `clang-format` à chaque fois que vous sauvegarderez un fichier `.c` ou `.h`.

Attention ! Lors des TP notés, tout outil IA est interdit et bloqué. Les tentatives de connexions aux sites d'IA sont enregistrées (et bloquées). Si votre `VS Code` tente d'accéder à Copilot, cela est considéré comme une fraude. Il est donc important d'interdire ou de désactiver ces deux extensions.

2 Utilisation pendant les TP

Ouvrir un TP. Téléchargez le TP et lancez `VS Code` dans le répertoire du TP. Pour le TP 1 de C :

```
$ wget -qO- https://www.labri.fr/perso/zeitoun/teaching/26-27/C/files/tp1.tgz | tar -xvz
$ cd tp1
$ make -B compile_commands.json
$ code .
```

Si une fenêtre Welcome to VS Code - Sign in to use GitHub Copilot apparaît, fermez-la (croix en haut à droite).

Éditer un fichier. Lorsque vous éditez un fichier, `VS Code` fournit plusieurs informations :

- Il colorie syntaxiquement le code et donne des informations si on survole un symbole à la souris.
- Il indique les erreurs en les soulignant en rouge. Placer sa souris sur une erreur indique sa nature.
- Il permet de compléter automatiquement certains fragments de texte (structures de contrôle comme les tests ou les boucles, nom de champs de structures, noms de variables, etc.).

Compiler. Le raccourci ctrl + ⇧ + B (⌘ + ⇧ + B sur Mac) lance la compilation par défaut : elle exécute `make`, comme dans un terminal (voir la fiche technique n° 2 sur `make`). Les erreurs de compilation apparaissent directement dans l'onglet Problems, cliquables pour sauter à la ligne concernée. Alternativement, vous pouvez compiler « à la main » en tapant `make` dans un terminal (après s'être positionné dans le répertoire du TP, bien sûr). Ce terminal peut être externe ou interne à `VS Code`.

Déboguer. Cliquez dans la marge à gauche d'une ligne pour poser un point d'arrêt (*breakpoint*), puis tapez F5. Le projet est automatiquement recompilé (`make`) avant le lancement du débogueur. Une fois arrêté sur un point d'arrêt : F10 exécute la ligne courante, F11 entre dans l'appel de fonction, F5 continue jusqu'au prochain point d'arrêt. Le panneau Variables affiche l'état courant des variables locales.

Avertissement

Cette fiche décrit le strict nécessaire pour compiler et déboguer sous `VS Code`. Elle ne remplace pas un cours sur le débogueur. De plus, `VS Code` propose d'autres fonctionnalités, non abordées ici.

3 Help, VS Code ne démarre plus !

Il se peut que votre fichier de configuration `settings.json` soit corrompu, ou que vous ayez oublié de fermer `VS Code` sur une autre machine (ce qui bloque tout nouveau lancement), ou que vous ayez installé une extension qui pose problème. Pour avoir des informations :

- Vous pouvez lancer `code --verbose`.
- Vous pouvez aussi lancer `VS Code` sans les extensions, au cas où la cause du problème serait une extension que vous avez installée, par `code --disable-extensions`.
- Il se peut qu'il y ait une erreur dans un fichier de configuration. Pour tester si votre configuration pose problème, lancez `code --user-data-dir /tmp/vscode-test`. Si cela fonctionne, vous devez avoir un fichier de configuration corrompu. Si c'est votre fichier de configuration global, qui est habituellement `~/.config/Code/User/settings.json`, vous pouvez essayer de :
  - le réparer en l'éditant,
  - le recréer à partir d'une sauvegarde se trouvant dans le sous-répertoire `.ckpt/`,
  - fermer `VS Code`, le supprimer et en recréer un vide en relançant `VS Code`.
- Enfin, `pgrep -a -u "$USER" code` permet de déterminer si `VS Code` tourne sur une machine. Si cette commande affiche un ou plusieurs nombres, il y a des instances de `VS Code` qui tournent. Vous pouvez les supprimer par `pkill -TERM -u "$USER" -x code`. Si vous suspectez qu'une instance de `VS Code` tourne sur une machine, il faut retrouver cette machine pour terminer le processus. Si vous vous souvenez seulement de la salle mais pas de la machine, vous pouvez récupérer la liste des machines sous Nagios et utiliser un script shell lançant la commande `pgrep` sur chacune des machines de la salle concernée, via `ssh`.

4 Installation de VS Code sur machine personnelle

La façon d'installer VS Code et les outils de compilation dépend de votre système d'exploitation.

4.1 Linux (Ubuntu ou Debian)

Installez le compilateur, le débogueur et bear (le générateur de compile_commands.json, la base de compilation utilisée par Clangd) :

```
$ sudo apt update
$ sudo apt install build-essential gdb bear
```

Téléchargez ensuite VS Code et installez-le :

```
$ curl -L "https://update.code.visualstudio.com/latest/linux-deb-x64/stable" -o /tmp/vscode.deb
$ sudo apt install -y /tmp/vscode.deb
```

4.2 macOS

Installez d'abord Homebrew, le gestionnaire de paquets de macOS, depuis Terminal.app :

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Cette commande installe aussi, si nécessaire, les *Xcode Command Line Tools* (compilateur, make, débogueur lldb). Installez ensuite VS Code avec Homebrew, ainsi que bear.

```
$ brew install --cask visual-studio-code
$ brew install bear
```

> Sur macOS, la commande gcc n'est pas le vrai GCC : c'est en réalité Apple clang sous un autre nom. Les deux compilateurs sont proches mais pas identiques (options reconnues, avertissements…). Le Makefile fourni s'adapte déjà automatiquement à celui que vous utilisez ; si vous préférez le vrai GCC (plus proche des machines Linux de la salle de TP), installez-le avec brew install gcc (accessible ensuite sous un nom comme gcc-16).

4.3 Windows

Installez d'abord WSL (*Windows Subsystem for Linux*), qui fournit un véritable environnement Linux (Ubuntu) sous Windows. Ouvrez PowerShell en administrateur et tapez :

```
> wsl --install
```

Redémarrez si demandé, puis lancez « Ubuntu » depuis le menu Démarrer pour terminer sa configuration (nom d'utilisateur, mot de passe). Dans le terminal Ubuntu ainsi ouvert, installez le compilateur, le débogueur et bear, comme sous Linux :

```
$ sudo apt update
$ sudo apt install build-essential gdb bear
```

Installez ensuite VS Code normalement sous Windows depuis le site VS Code. Dans VS Code, installez aussi l'extension WSL (éditeur : Microsoft) : elle permet à VS Code de se connecter à l'environnement Linux, où se trouvent réellement gcc, make et gdb. Ouvrez enfin vos TP depuis le terminal Ubuntu (jamais depuis l'explorateur Windows), en plaçant les fichiers du TP dans le système de fichiers Linux :

```
$ cd ~/chemin/d_acces/a_mon_tp
$ code .
```