

Installation et premiers pas avec l'IDE Visual Studio Code C++ pour linux

Ce document explique comment installer *Visual Studio Code IDE for C/C++* sous linux. Cet IDE est disponible sur Moodle (<https://moodle.univ-ubs.fr/course/view.php?id=9497>).

Il explique également comment créer un premier projet C++ très simple qu'il faudra compiler et exécuter.

A. INSTALLATIONS PACKAGES LINUX

Vérifiez les installations suivantes, sinon installez les packages nécessaires (si échecs, passez quand-même à l'étape suivante) :

- `sudo apt install x11-apps`
- `sudo apt install xterm`
- `run apt-get update`

B. INSTALLATION VSCODE BASIC POUR LINUX/UBUNTU

1. Récupérer l'archive d'installation pour linux *VisualStudioCodeAmd64Linux.deb* sur Moodle
2. Installer cette version sous linux/Ubuntu : `sudo apt install ./VisualStudioCodeAmd64Linux.deb`
3. VSCode se lance simplement en tapant la commande "code". Un "prompt" vous demande si vous désirez vraiment continuer avec la version linux (répondre yes). Pour éviter que ce "prompt" apparaisse à chaque lancement, définir la variable d'environnement dans votre `.bashrc` et la positionner à 1 : `export DONT_PROMPT_WSL_INSTALL=1`. Tenir compte de cette variable en exécutant votre `.bashrc` : `source ~/.bashrc`. Relancer la commande "code" (le « prompt » n'apparaît plus).

C. INSTALLATION DE L'EXTENSION C++ POUR VSCODE

1. Installer le compilateur C++ sous linux : `sudo apt install g++`
2. Installer le Debugger C++ sous linux : `sudo apt install gdb`
3. Lancer VSCode (commande "code")
4. L'extension C++/linux pour VSCode consiste simplement à installer "C++ extension for VS Code" (voir <https://code.visualstudio.com/docs/cpp/config-linux> et n'exécuter QUE le point 2 du paragraphe « Prerequisites »)

D. INSTALLATION DE L'EXTENSION MAKEFILE

1. Installer l'outil *Makefile* sous Linux : `sudo apt install make`
2. Lancer *VSCode* et dans les extensions proposées par *VSCode* (menu View/Extensions), installer simplement "Makefile Tools"
3. Sous *VSCode*, donner le chemin qui permet d'accéder à l'emplacement de votre fichier *Makefile* qui se trouvera systématiquement en */ws*. Pour ce faire sélectionner *File -> Preferences -> Settings -> Extensions -> Makefile Tools*, puis dans "Makefile Path" rentrer le chemin relatif *./ws/*. Fermer ensuite la fenêtre "Settings".

E. UN PREMIER PROJET VSCode C++ AVEC MAKEFILE

Ce premier projet correspond exactement à ce qui vous est demandé dans le **TP1/Exercice1**.

1. Créer le projet sous linux

D'abord fermer *VSCode*.

Suivre les étapes suivantes :

- créer le répertoire *.../VisualStudioCode*
- créer le répertoire de projet *.../VisualStudioCode/ProjetTP1*
- se mettre dans */ProjetTP1* et lancer *VSCode* avec la commande "`code .`" (**attention à bien mettre .**) : à ce moment là, *VSCode* considère que votre répertoire de travail (workspace) est *VisualStudioCode/ProjetTP1* et tous les chemins relatifs s'expriment par rapport à ce répertoire */ProjetTP1*

2. Créer les répertoires */src*, */bin*, */ws* et les fichiers sources *.h* et *.cpp*

Sous *VSCode*, créer en dessous de */ProjetTP1* les 3 répertoires */src*, */bin*, */ws* au même niveau (cliquer sur "Explorer" menu barre de gauche).

Sous *VSCode*, créer les fichiers *.h* et *.cpp* et pour ce faire :

- clic droit sur */src/New File/* et rentrer *Exo1.h* (+ écrire le code dans ce fichier suivant les indications du TP1/Exo1)
- clic droit sur */src/New File/* et rentrer *Exo1.cpp* (+ écrire le code dans ce fichier suivant les indications du TP1/Exo1)
- clic droit sur */ws/New File/* et rentrer *Makefile* (+ écrire la cible de compilation, suivant les indications du TP1/Exo1, qui s'appellera *lanceurExo1*)

3. Compiler et exécuter l'application sous VSCode

- cliquer sur "Makefile" dans le menu de la barre de gauche : pour l'option "Build target" sélectionner la cible "lanceurExo1" et pour l'option "Launch target" sélectionner "Exo1.bin"
- pour compiler : cliquer sur "Makefile" dans le menu de la barre de gauche puis cliquer sur l'icône "Build the current target" (lire et corriger les éventuelles erreurs de compilation)
- pour exécuter : cliquer sur "Makefile" dans le menu de la barre de gauche puis cliquer sur l'icône "Run the selected binary target in the terminal"

4. Quitter VSCode

File -> Exit

A chaque fois que l'on relancera *VSCode* par la suite il n'est plus nécessaire de se placer dans */ProjetTP1* avec la commande `"code ./"` : par défaut maintenant, après avoir lancé *VSCode* avec la commande `"code"`, il se positionnera automatiquement dans le répertoire */ProjetTP1*.