**INTRODUCTION A CMAKE**

CMake est une famille d’outils open-sources pour construire, tester et intégrer des logiciels.

Il permet de générer facilement des fichiers Makefile ainsi que les projets pour Codeblocks ( ou autre environnement)

Une image contenant texte

Description générée automatiquement

Pour créer un CmakeLists.txt :

Cmake\_minimum\_required(VERSION 2.8) (obligatoire)

Project(hello)

Add\_executable (Hello prog.c hello.c)

Il est d’usage de créer un répertoire build pour contenir les fichiers de compilation :

Mkdir build : cd build

Cmake .. (cmake list est a l’ext du repertoire build)

Make (compile tout et créer l’executable Hello)

./Hello (compile le Hello)

QQES OPTIONS

Cmake -G CodeBloks -Unix Makefiles (genere un Hello.cbp = nous evite de versionner nos fichiers de base pour coder sur codeBloks)

BONNE PRATIQUES

* Faire le cmake dans un repertoire build
* Mettre ts les fichiers source (.c, .cpp) dans un repertoire src
* Mettre ts les fichiers d’entetes (.h, .hpp) dans include
* Mettre ts les fichiers d’application (contenant le main) à part (apps, bin, exe, ..)
* Mettre les fichiers de test à part (dans tests)
* Commenter son code

mkdir src include apps

mv prog.c apps/

mv hello.h include/

mv hello.c src/

donc :

CMakeLists.txt devient :

cmake\_minimum\_required(VERSION 2.8)

project(hello)

include\_directories( include)

add\_executable (Hello apps/prog.c src/hello.c)

Quand il y a bcp de fichiers à compiler ,, il est courant de les rassembler en librairie :

cmake\_minimum\_required(VERSION 2.8)

project(hello)

include\_directories( include)

add\_library(toto SHARED src/hello.c) peu etre rajouté pour dire que cest une librairie dynamique et no n statique

add\_executable (Hello apps/prog.c)

target\_link\_libraries(Hello toto)

on peut utilizer des variables, ou rassembler les fichiers ou faire afficher des messages :

Une image contenant texte

Description générée automatiquement