

Informations, Annales & Bibliographie



Contents

Examen, Partiels & TMEs	1
Soumission des Comptes Rendus de TME	2
Support de Cours	2
Annales	2
Bibliographie	2
Livres	2
Ressources en Ligne	3
Organisation du Code & Compilation	3
Compilation avec <code>cmake</code> & <code>make</code>	3

Examen, Partiels & TMEs

- Le barème de la note finale de l'UE est le suivant:

Note	Poids
Examen Final	60%
Contrôle Continu	40%

- L'examen final dure deux heures, tous documents autorisés. Il s'agit d'un examen écrit classique.
- La note de rattrapage remplace uniquement la note d'examen, la note de contrôle continu est conservée.
- L'ensemble des TME sont notés et constituent la note de contrôle continu. En conséquence, **il n'y a pas de partiel.**

Soumission des Comptes Rendus de TME

Le compte rendu d'un TME doit être rendu **avant** le début du TME suivant (i.e. vous disposez d'une semaine pour le finir).

La lisibilité du code est notée.

Le compte rendu devra uniquement contenir:

- Le répertoire des sources `<top>/src/` (cf. Organisation du Code et Compilation).
- Une explication concise sous forme d'un fichier texte `CR.txt` ou PDF `CR.pdf`.

Tous ces fichiers devront être fournis sous forme d'une archive compressée de type `.tar.gz`:

```
etudiant@pc:dir> cd <top>
etudiant@pc:top> tar zcvf TME1.tar.gz src
src/
src/cBox.h
src/cBox.c
src/cMain.c
src/CMakeLists.txt
src/CR.txt
etudiant@pc:top>
```

Cette archive doit être jointe en attachement d'un e-mail envoyé à l'encadrant de votre groupe de TME, Jean-Paul.Chaput@lip6.fr ou Marie-Minerve.Louerat@lip6.fr. Le message de l'e-mail doit faire apparaître *clairement* les noms et prénoms de toutes les personnes ayant participé au dit TME.



Note

Nous n'accepterons pas de liens vers des sites tiers (stockage de type *cloud* du style de Dropbox ou Google Drive).

Support de Cours

Annales

- [Examen 2014/2015](#).
- [Rattrapage 2015/2016](#).

Bibliographie

Livres

- H. GARETTA, **Le langage et la bibliothèque C++**, Ellipse 2000.
 - [Le Langage C++](#)
 - [Version abrégée](#)
- C. DELANNOY, **Programmer en langage C++** (5ième édition), Eyrolles 2000.
- M. NEBRA, M. SCHALLER, **Programmer avec le langage C++**, Le Site du Zéro & Simple IT 2011.
- B. STROUSTRUP, **Programmation - Principes et pratique avec C++**, Pearson Education 2010.
- B. STROUSTRUP, **Le langage C++** (3ième édition), Compupress 1999.
- S. MEYERS, **Effective C++** (3ième édition), Addison-Wesley 2005.
- S. MEYERS, **Effective STL**, Addison-Wesley 2001.
- E. GAMMA, R. HELM, R. JOHNSON, J. VLISSIDES, **Design Patterns** Addison-Wesley 1995.

Ressources en Ligne

- Wiki & Référence pour C et C++: <http://en.cppreference.com/>
- Reference C++: <http://www.cplusplus.com/reference/>.
- Documentation STL par SGI: <http://www.sgi.com/tech/stl/>.
- Site libre de Questions & Réponses: <http://stackoverflow.com/>
- Articles de S. Meyers: <http://www.aristeia.com/publications.html>
- Blog de S. Meyers: <http://scottmeyers.blogspot.fr>
- Boost C++ Libraries: <http://www.boost.org>
- Clang, compilateur alternatif à GCC: <http://clang.llvm.org> et <http://clang-analyzer.llvm.org>

Organisation du Code & Compilation

Pour tous les TMEs, il vous est demandé de travailler en respectant l'arborescence suivante:

Répertoire	Contient
<top>/src/	Les fichiers sources (.h, .cpp, cmake)
<top>/build/	La compilation générée par cmake.
<top>/install/include/	Les fichiers .h installés
<top>/install/lib/	Les éventuelles bibliothèques
<top>/install/bin/	Les binaires.
<top>/work/	Le répertoire de travail dans lequel vous lancerez votre programme

Ce choix d'arborescence des répertoires utilise la stratégie de compilation *out of source*. C'est à dire que les modules compilés (.o et exécutable) sont créés dans des répertoires séparés de ceux du code source. Le code source reste ainsi parfaitement propre. Après compilation le binaire, les fichiers d'en tête (.h) et les éventuelles bibliothèques (libraries, en anglais) sont aussi installés dans une arborescence séparées (<>/install/). Enfin, pour exécuter le programme on se placera dans le répertoire de travail <>/work.

Compilation avec cmake & make

Sous les systèmes d'exploitation UNIX en général et GNU/Linux en particulier, le programme de référence pour gérer la compilation d'un projet C/C++ est `make`. Toutefois, l'écriture de son fichier de configuration est fastidieuse. On utilisera donc `cmake`, qui générera les fichiers de `make` à partir d'un fichier de configuration plus simple: `CMakeLists.txt`. Il détermine comment compiler les différents fichiers sources et construire le binaire. Accessoirement, il détecte les caractéristiques du système, vérifie la présence des composants nécessaires et effectue l'installation. Son fichier de configuration se trouve dans le répertoire source et vous sera fourni pour chaque TME.

La compilation s'effectue en deux temps:

1. Exécution de `cmake`, dans le répertoire <>/build/. Cette étape n'est à effectuer qu'une seule fois. Elle va générer les fichiers nécessaires à `make`.
2. Compilation proprement dite, et installation en lançant la commande `make`, toujours dans <>/build/.

Ce qui donnera la séquence de commandes suivantes:

```
etudiant@pc:dir> mkdir build # Une seule fois.
etudiant@pc:build> cd build # Une seule fois.
etudiant@pc:build> cmake ../src # Une seule fois.
etudiant@pc:build> make install
etudiant@pc:build> # Modification du code...
etudiant@pc:build> make install
etudiant@pc:build> # Modification du code...
etudiant@pc:build> make install
etudiant@pc:build> # Modification du code...
etudiant@pc:build> make install
```

Note

Recompiler en restant dans <>/work: il est fastidieux de sans arrêt changer de répertoire entre <>/build ou l'on compile et <>/work ou l'on exécute le programme. On peut éviter cela et rester en permanence dans <>/work en utilisant la commande suivante:



```
etudiant@pc:dir> cd work
etudiant@pc:work> (cd ../build; make install)
etudiant@pc:work> # Modification du code...
etudiant@pc:work> (cd ../build; make install)
etudiant@pc:work> # Modification du code...
etudiant@pc:work> (cd ../build; make install)
```