

---

## Remarques sur les exercices de C++

---

### Organisation des répertoires

Pour la plupart des exercices proposés dans le cadre de ces Tds, une partie du code est fournie à titre d'exemple ou comme point de départ. Ces informations se trouvent sous le répertoire `/net/public/c/cm/C++EPU2` qui contient

- Un sous-répertoire `Sujets` comportant les éléments nécessaires aux différents Tds
  - une version PDF de chaque énoncé qui se trouve (ou se trouvera...) dans un répertoire `LES_ENONCES`
  - les différents Tds, représentés par un nom (e.g., `Fifo`) et par un lien symbolique portant leur numéro (e.g., `TD_1`)
- Un sous-répertoire `Solutions` qui, au fur et à mesure de l'avancement des TDs contiendra les solutions des exercices.
- Différents éléments utiles, voire indispensables, à la compilation du code fourni ainsi qu'à la compilation de vos solutions: le répertoire `include`, la `Makefile` et le fichier `default.mk`

Afin de pouvoir réaliser ces Tds dans des conditions optimales, vous devez donc récupérer l'ensemble de ces informations en respectant les hiérarchies proposées. Le plus simple est donc de recopier chez vous, lors du premier TD, l'ensemble du contenu du répertoire `C++EPU2` ; par exemple sous **sh**, **bash**, **zsh**...

```
% mkdir mes_tds_c++
% cd mes_tds_c++
% cp -r /net/public/c/cm/C++EPU2/* .
```

### Environnement de développement en C++

#### Environnement par défaut

Les TDs ayant lieu sous Unix (Linux), la solution par défaut est l'utilisation de **g++**, avec Emacs/XEmacs comme éditeur et **gdb** pour la mise au point. Apprenez

- à rédiger des *makefiles* simples (en général, les exemples viennent avec leur `Makefile` qu'il suffit d'éditer pour l'adapter à vos besoins) ;
- à compiler *dans* l'éditeur (Emacs ou XEmacs) : c'est facile, cela évite les erreurs et les pertes de temps ;
- à utiliser un *debugger* comme **gdb** ou, plus agréablement, l'une de ses interfaces graphiques (**xxgdb** ou **ddd**) ; compilez toujours avec l'option **-g** — du moins pour ces exercices.

Certes il s'agit là d'un environnement par bien des côtés rustique comparé aux dorures des environnements intégrés (voir juste après). Mais, bien utilisé, c'est ce que je connais de plus rapide pour le cycle de développement édition-compilation-mise au point.

## Environnements intégrés (IDE)

Bien entendu vous pouvez adopter un autre environnement, à condition d'y être à l'aise. Tant qu'à faire, il vaut mieux privilégier les environnements qui intègrent la compilation et la mise au point (*Integrated Development Environment* ou IDE). Notez cependant qu'ils nécessitent en général un apprentissage non négligeable pour en tirer pleinement partie et sont souvent (beaucoup) plus lents que la solution par défaut évoquée plus haut.

Un grand nombre d'IDE sont disponibles sous Linux : **kdevelop** (VisualC++-like, utilisé par KDE), **eclipse** (personnellement, cet environnement m'est indispensable en Java, mais je trouve son mode C++ un peu limité et assez pénible), **anjuta**, CodeForge (celui-là est payant, mais pas cher !), etc.

Sous Windows, il y a bien sûr l'incontournable Microsoft Visual C++. C'est un environnement très agréable à utiliser ; malheureusement, sous la version 6.0, le compilateur est abominablement mauvais dans son support de la norme C++ ; la version 7.1 (version dite **.net**) est un peu meilleure mais n'est toujours pas exempte de bugs. Sous Windows encore, Cygwin vous permet de développer du C++ « comme sous Linux », avec la compatibilité des outils et des compilateurs et **eclipse** est également disponible. Enfin à noter l'excellent Dev-C++, un environnement intégré « à la Visual », mais simple, léger et rapide, supportant les compilateurs GNU et très bien adapté à des projets de taille pas trop énorme.

## Compatibilité

Toutes les solutions des exercices sont compilables et exécutables à la fois sous Unix/Linux avec les outils GNU et sous Windows avec Cygwin ou Dev-C++. GNU MAKE est indispensable ; La dernière version du compilateur utilisée est **g++-3.4.x** dans les deux environnements (**x** = 1 ou 2).

## Autres ressources

Voir le site Web local de Jean-Paul Rigault pour des compléments sur C++ :  
<http://www-local/~jpr>

Jean-Paul RIGAULT  
[jpr@essi.fr](mailto:jpr@essi.fr)  
Novembre 2004

Mise à jour par Colette Michel  
[cm@essi.fr](mailto:cm@essi.fr)  
Novembre 2005