


Les derniers jours du C++11 ?

Pour ceux qui suivent un peu l'actualité, une nouvelle importante n'a pas dû leur échapper : nous sommes en 2014 (bonne année donc, avec du retard). Pour les développeurs C++, cette année est d'une importance capitale. En effet, cette année, nous allons avoir la réponse à la question existentielle suivante : est-ce que le comité de normalisation du C++ va enfin réussir à faire sortir une norme C++ à la date prévue ?

Bon, accessoirement, cette année sera peut-être aussi la sortie du C++14, mais c'est accessoire par rapport à LA question précédente 

Plus sérieusement, le comité se réunit en février à Issaquah, il est possible que ce soit le dernier meeting avant le C++14. Les différents papiers de travail pour ce meeting ont été publiés, je vais faire un état des lieux de l'actualité sur la normalisation du C++.

Pour ceux qui sont intéressés, vous pouvez aller directement lire les drafts sur le site du comité C++ : C++ Papers for Issaquah. Vous pouvez également aller sur Meeting C++ pour lire la review de Jens Weller sur les drafts pour SG1 Concurrency.

Comme d'habitude, il y a beaucoup d'articles pour le meeting (78 si mes comptes sont bons). Tous ne concernent pas le C++14, on y trouve aussi des propositions pour les TS (Technical Specification), le C++17 ou de simples discussions sur le futur du C++. Pour aider à comprendre les propositions, je vais les classer par groupes de travail. Pour rappel (ce sont des images que j'ai déjà données, mais ça ne fait pas de mal de la remettre) :

Organisation du comité de normalisation C++

Le processus de validation de la nouvelle norme est bien avancé pour le moment. Une proposition de draft final a été présentée lors du dernier meeting en août 2013, pour avoir les commentaires des délégations

nationales. Ces dernières ont rédigé 85 commentaires (plus 35 hors délais, donc non officiels), ce qui est peu par rapport au C++0x et au C++11 (environ 500 commentaires à l'époque). Le comité a donc répondu à ces différents commentaires entre temps et le meeting de février permettra de publier le draft final et voir s'il y a encore des commentaires. Si ce n'est pas le cas, le comité pourra publier un DIS (Draft International Standard), c'est-à-dire le document final qui suivra la fin du processus de validation par l'ISO. Ce processus peut prendre quelques mois, mais si tout se passe bien, on aura bien un C++14 et pas un C++15 :)

À côté de la finalisation du C++14, il y aura également la validation des TS (Technical Specification). Le but de ces TS est de permettre de faire évoluer en souplesse le C++, sans devoir attendre plusieurs années pour la validation d'une norme complète. Cela permettra ainsi aux compilateurs d'implémenter ces fonctionnalités et aux utilisateurs de faire évoluer le code progressivement. Ces TS ne sont pas des normes à proprement parlé, ils ne passent pas par une validation par l'ISO, mais sont validés uniquement par le comité de normalisation du C++. Ils seront intégrés dans la norme suivante par l'ISO (donc dans le C++17 normalement).

Plusieurs TS sont déjà validés par les groupes d'études (ce qui veut dire que certains groupes d'étude vont peut-être se dissoudre) et devrait être validés par le comité. Ce qui explique que certains groupes ne propose pas de draft pour le meeting de février, il faut se référer aux derniers drafts publiés. Les drafts des TS finalisés sont les suivants :

- WG21, Programming Language C++ : N3797 ;
- WG21, Library Fundamentals : N3848 ;
- WG21, Array Extensions : N3820 ;
- SG1, Parallelism \square : N3850 ;
- SG3, File System : N3790 ;
- SG4, URI (Network) : N3827 ;
- SG8, Concepts Lite Specification : N3889.

Pour rappel, voici le planning prévu pour les différentes normes :



Un point important pour terminer : qu'en est-il de la prise en charge par les compilateurs de la prochaine norme ? De ce côté, on peut dire qu'il y a eu une évolution majeure des mentalités. Les compilateurs majeurs du marché ont la volonté de respecter la norme et de l'implémenter rapidement. Au point où un compilateur, Clang 3.4, supporte déjà complètement la prochaine norme (il suffit d'utiliser l'option de compilation `-std=c++1y`). Pour rappel, les binaires de Clang sont maintenant disponibles pour Windows aussi.

GCC 4.9 (Development Version) a commencé également à supporter le C++1y (avec la même option de compilation `-std=c++1y`). Vous pouvez installer cette version via le paquet `gcc-snapshot` sur plusieurs distributions Linux ou sur la page de Equation Solution pour les binaires Windows.

Pour terminer, celui qui a le plus de mérite à mon avis, parce qu'il revient de très très loin en termes de support de la norme : Microsoft Visual C++ 2013. Herb Sutter, qui travaille chez Microsoft et est le secrétaire du comité de normalisation du C++, a annoncé la volonté de se mettre à jour pour le support des anciennes normes (C++11, C++03, etc.) et être moteur dans le support des prochaines normes. Cela va prendre un peu de temps, mais le changement a déjà commencé. Voir le planning des implémentations.

Conclusion

Après cette vue d'ensemble de l'actualité de la normalisation du C++, je

vais détailler dans les prochains articles le détail des différents TS, de la norme C++14 (si vous ne voulez pas attendre, vous pouvez lire les résumés sur isocpp.org ou sur Developpez.com) et des drafts pour le meeting de février.

Si vous souhaitez écrire des articles sur les TS ou la future norme du C++, je me ferais un plaisir de les publier sur ce blog.

[C++11](#)