

Trolltech DevDays 2006 : Multi-Threading sous Qt4

par

Date de publication : 01/11/2006

Dernière mise à jour :

Multithreading avec Qt, résumé de la conférence de Bradley Hughes lors des Trolltech DevDays 2006.

- I - Qui est Bradley Hughes ?
- II - Résumé de la présentation
- III - Conclusion

I - Qui est Bradley Hughes ?

Bradley Hughes est le responsable du développement cross-plateforme de la base de Qt.

II - Résumé de la présentation

Tout d'abord, il faut considérer le support des threads dans Qt4 comme l'évolution de Qt3, lui-même évolution de Qt2.

QThread, la classe de base de support des threads, dispose maintenant de 3 signaux - **finished**, **started**, **terminated** - et d'un slot supplémentaire - **quit** -. Ces fonctions permettent d'indiquer la fin d'un thread, par exemple. C'est possible car les signaux et slots sont enfin thread-safe.

Naturellement, il existe une classe de mutex, **QMutex**, ainsi que **QReadWriteLock** permettant de créer un mutex sensible à un accès en lecture ou en écriture. Pour simplifier l'écriture, il existe des objets bloqueurs, comme **QMutexLocker** ou **QReadLocker** et **QWriteLocker**. De plus, il est possible d'endormir un thread à l'aide de **QWaitCondition** et de le réveiller à l'aide de l'appel à **wakeOne** ou **wakeAll**. Une nouvelle classe de sémaphore est proposée, **QSemaphore**.

QThreadStorage permet aussi de stocker pour un thread et de manière transparente pour n'importe quel thread, un espace de stockage unique par thread.

Depuis Qt4.0, on peut utiliser les signaux et les slots entre threads. En fait, il est possible de poster des événements entre les threads, et c'est ainsi que les signaux/slots fonctionnent. Par défaut, la connexion se fait par messages entre les threads, et directement dans un même thread. Ensuite, il est possible de forcer la connexion directe - comme dans Qt3 - ou par message. Pour des messages avec des paramètres particuliers, il faut enregistrer les types à l'aide de **QMetaType**.

III - Conclusion

Le support du multithreading a été largement amélioré depuis la version 3 de Qt, et est mieux conçu que d'autres bibliothèques - par exemple Boost -. Il reste encore des changements à effectuer, comme le support de boîtes de dialogues dans les threads non GUI. Pour plus de détails, veuillez consulter [cette page](#).

