

Programmation : LLVM unifie son support des ARM 64 bits

Petit à petit, LLVM et le compilateur Clang (C, C++, ObjectiveC) s'affirment comme une alternative de plus en plus évidente au traditionnel GCC.

FreeBSD utilise ainsi Clang comme compilateur de base depuis début 2014 (voir « [FreeBSD 10.0 joue la carte de la virtualisation, et de Clang](#) ») et Apple s'est appuyé sur LLVM pour son langage de programmation Swift (voir « [Avec Swift, Apple veut révolutionner la programmation iOS et OS X](#) »).

Les responsables du projet viennent récemment de présenter **LLVM 3.5 et Clang 3.5**. Côté Clang, les avancées sont relativement classiques : le compilateur intègre les dernières évolutions du C++, tout en essayant de se rapprocher toujours plus du GCC.

Les changements sont plus profonds au sein de LLVM, en particulier dans sa gestion des puces ARM 64 bits. Un nouveau générateur de code est proposé : il allie celui conçu par Apple pour ses puces A7 et A8 à celui développé pour les autres composants ARM 64 bits.

Uniformisation dans le monde ARM

Cet effort d'unification est à l'image de l'évolution du monde ARM. Pratiquement chaque composant est différent et incompatible avec ses concurrents. Idem pour les cartes-mères, incompatibles entre elles. Seul le code applicatif (et encore...) peut passer librement d'une puce ARM vers une autre.

Toutefois, avec l'arrivée des composants ARM sur les marchés mobile, desktop et serveur, la donne change rapidement. Dans le monde des serveurs, une spécification permettra de créer des produits relativement interchangeables (voir l'article « [Un standard pour les serveurs ARM... et Windows Server en prime ?](#) »).

Linux s'y est également mis. Le célèbre noyau open source abandonne en effet progressivement le modèle 'un kernel par composant ARM' pour proposer des offres plus génériques, compatibles avec plusieurs SoC différents. Une évolution aussi logique que bienvenue.

Crédit photo : © AMD