

Prezentacja Cmake'a

Agenda

1. Wprowadzenie
2. Podstawowe pojęcia
 - target
 - library
 - executable
3. Workflow
4. Definiowanie zależności
5. Cross-kompilacja
6. Porównanie z make i qmake

Wprowadzenie

Narzędzie do zarządzania build systemem w projekcie. Powstało w roku 1999 - rozwijany od ponad 15 lat! Możliwość ustawienia (niezależność od):

- kompilatora (zazwyczaj C, C++):
 - gcc
 - clang
- narzędzia do budowania (z możliwością przystosowania do środowiska i platformy):
 - makefile
 - ninja
- Definicję struktury projektu tworzy się w plikach CMakeLists.txt. CMake ma strukturę hierarchiczną:
- pliki CMakeLists.txt są zlokalizowane w katalogach projektu
- każdy CMakeLists.txt dziedziczy zmienne po swoim rodzicu
- struktura katalogów w projekcie powinna odwzorowywać logiczny podział projektu

Workflow

1. Ściągamy projekt.

2. Uruchamiamy CMake'a -> generowanie makefile lub ninja
3. Kompilacja wybranym narzędziem.

Podstawowe pojęcia

Target - część projektu, która może zostać zbudowana. Zazwyczaj library lub executable. Może zostać zbudowany po wygenerowaniu make'a lub ninja.

Library - kontener, dzięki któremu można podzielić projekt na mniejsze części. Podział na libki pozwala na osiągnięcie modułowości i przydaje się przy konstruowaniu testów. Typy libek w CMake'u:

- statyczne, dynamiczne - standardowe typy bibliotek (*.a i *.so)
- logiczne - biblioteki, które mogą być definiowane tylko jako zależności do innych komponentów, nie mogą być budowane niezależnie (nie mogą być targetami)

Executable - target, który po skompilowaniu może być uruchomiony (binarka).

Definiowanie zależności

Definiowanie zależności określa kolejność budowania i linkowania targetów. Zależności do definiowania:

- libka -> libka
- libka -> executable

Cross-kompilacja

CMake umożliwia przeprowadzenie procesu kompilacji na inną architekturę. Nie jest to możliwe na poziomie narzędzia do budowania - należy wygenerować nowe pliki make/ninja. Do cross-kompilacji CMake potrzebuje pliku z konfiguracją zawierającego definicje:

- sysroota sdk
- cross-kompilatora
- systemu operacyjnego, na którym wykonywana jest kompilacja

Porównanie z make i qmake

make

+

- jeden tool do zarządzania i budowania projektu
- stabilność, kompatybilność, przenośność
- dobre docsy i pomoc online

-

- integracja z Qt

qmake

+

- nie skaluje się do dużych projektów
- nieprzystosowany do dzielenia projektu

-

- słaba dokumentacja
- zależność od Qt