Přechod systému FRED z Autotools na CMake

Miroslav Franc ▪ miroslav.franc@nic.cz ▪ 15. listopadu 2018

Osnova

- Co je to FRED?
- Co je build systém?
- Jak fungují Autotools/CMake?
- Proč nepsat makefily ručně?
- Detekce závislostí na Linuxu
- Co nám přechod na CMake přinese?

Co je fred?

- Free Registry for ENUM and Domains
  - Skupina open source nástrojů pro správu doménového registru
  - Platfoma: C/C++/Python & PostgreSQL & Linux
  - http://fred.nic.cz
  - Backend převážně v C++

Co je build systém?

- spousta zdrojových souborů a hlaviček → knihovny a binárky
- knihovny jsou buď statické (`*.a`) nebo dynamické (`*.so`)
- detekce závislostí
  - knihovny, hlavičky, metadata
- konfigurace
- instalace a odinstalace
- balíčkování

Jak funguje Autotools

- dva soubory: `configure.ac` a `Makefile.am`
- `configure.ac` je napsán v jazyku `m4`
- `Makefile.am` je šablona pro `Makefile`
- `autoconf` transformuje `configure.ac` do `configure` skriptu
- `automake` transformuje `Makefile.am` do `Makefile.in`
- produktem úspěšného doběhnutí `configure` skriptu je `Makefile`

Jak funguje Autotools

```
$ autoreconf -vfi  # really?
$ mkdir build
$ cd build
$ ../configure --with-idldir=/home/mfranc/m/idl
$ make
$ make distcheck
```

Jak funguje CMake

- jeden soubor: `CMakeLists.txt`
- program `cmake` **musí** být přítomen při konfiguraci
- může generovat i něco jiného než `Makefile`
- vlastní jazyk založený na příkazech
  - obsahuje i funkce, makra a proměnné
  - vše je string, dokonce i pole

Jak funguje CMake

```
$ mkdir build
$ cd build
$ cmake ..
$ make
$ make distcheck
```

```
$ mkdir build
$ cd build
$ cmake -G Ninja ..
$ ninja
$ ninja distcheck
```

Proč nepsat makefily ručně?

- nepřenositelnost a nerozšiřitelnost
- Makefile syntax není jednodušší
- detekce závislostí je manuálně složitější
- nemožnost použít předpřipravenou funkcionalitu
- `CMake` a `Autotools` jsou de-facto standard

Detekce závislostí na Linuxu

- `pkg-config --cflags --libs grpc`
  - možnost například vyžadovat minimální verzi
- `boost` nepoužívá `pkg-config`
- apache používá `apxs` tools pro build modulů
  - `apxs -q CFLAGS`
  - `apxs -q LDFLAGS`
  - `apxs -q LIBEXECDIR`
- oddělovač polí mezera (`shell`) vs. středník (`CMake`)

Detekce závislostí na Linuxu

- hlavičky se detekují
  - na přítomnost souboru
  - překladem drobného programu, který je includuje
- funkce se detekují
  - překladem drobného programu, který je používá
- speciální případy
  - detekce verze knihovny podle C/C++ makra

Co nám přechod na CMake přinese?

- podpora kompilační DB: `compile_commands.json`
- další změny jsou jednodušší
- optimalizace procesů vytváření knihoven a binárek
- možnost používat `ninja` místo `make`
- CMake nepodporuje `uninstall`, `dist`, a `distcheck`
- nutná úprava balíčkovaní (`deb` a `rpm`)

Bude to mít vliv na rychlost?

Co nám přechod na CMake přinese?

```
# autoreconf + configure
real    0m34.588s
user    0m27.032s
sys     0m2.800s
```

```
# configure
real    0m27.782s
user    0m22.040s
sys     0m2.528s
```

```
# cmake (make)
real    0m6.326s
user    0m4.828s
sys     0m0.828s
```

```
# cmake (ninja)
real    0m5.110s
user    0m3.952s
sys     0m0.524s
```

Co nám přechod na CMake přinese?

```
-j50 CXXFLAGS='-ggdb -O2 -std=c++14 -Wall -Wextra'

# make (autotools)
real    4m7.755s
user    172m22.092s
sys     11m4.964s


# make (cmake)
real    4m3.132s
user    165m1.940s
sys     10m56.380s
```

```
# ninja (cmake)
real    4m0.559s
user    163m11.496s
sys     10m44.100s
```

Otázky?