


Crosskompilace s MinGW

Dneska se v rychlosti podíváme na to, jak na GNU/Linuxu narychlo vyprodukovat windowsí spustitelnou binárku pomocí MinGW. Tento článek je zaměřen spíše na začátečníky, pokročilí předpokládám pochopí vše z následujícího screenshotu (trochu okurková sezóna no...):

```
0 ;) harvie@harvie-ntb Temp $ cat hello.c
#include <stdio.h>
#include <windows.h>
void main(void) {
    char text[] = "This is MinGW hello world!";
    puts(text);
    MessageBox(0, text, "some caption", MB_OK | MB_ICONINFORMATION);
    //For info about MessageBox: http://msdn.microsoft.com/en-us/library/ms645505(VS.85).asp
}
0 ;) harvie@harvie-ntb Temp $ i486-mingw32-gcc hello.c -o hello.exe
0 ;) harvie@harvie-ntb Temp $ WINEDEBUG=fixme-all wine hello.exe
This is MinGW hello world!
```



Instalace MinGW

V rychlosti si ověříme, jestli naše distribuce (v mém případě ArchLinux) má balíčky s vývojovým prostředím MinGW.

```
0 ;) harvie@harvie-ntb ~ $ pacman -Ss mingw
community/mingw32-binutils 2.19.1-1
    A set of programs to assemble and manipulate binary and object files
    (mingw)
community/mingw32-gcc 4.4.0-1
    A C and C++ cross-compilers for building Windows executables on Linux
community/mingw32-gcc-base 4.4.0-1
    A C cross-compiler for building Windows executables on Linux
community/mingw32-runtime 3.15.2-1
    mingw32 run-time library
community/mingw32-w32api 3.13-1
    Windows header files fro mingw32 C cross-compiler
```

Vidím, že balíčky jsou v repozitáři community, neměl by tedy být žádný problém je stáhnout a nainstalovat. Já jsem zvolil instalaci balíčku **mingw32-gcc**, přičemž ostatní balíčky se nainstalovaly jako závislosti.

Kompilátor

Po instalaci se podíváme, co se nám vlastně nainstalovalo:

```
0 ;) harvie@harvie-ntb ~ $ ls -l /usr/bin/*mingw*
-rwxr-xr-x 1 root root 20776 19. úno 2009 /usr/bin/i486-mingw32-addr2line
-rwxr-xr-x 2 root root 44464 19. úno 2009 /usr/bin/i486-mingw32-ar
-rwxr-xr-x 2 root root 253608 19. úno 2009 /usr/bin/i486-mingw32-as
-rwxr-xr-x 2 root root 209972 25. srp 15.52 /usr/bin/i486-mingw32-c++
-rwxr-xr-x 1 root root 19552 19. úno 2009 /usr/bin/i486-mingw32-c++filt
-rwxr-xr-x 1 root root 209044 25. srp 15.52 /usr/bin/i486-mingw32-cpp
-rwxr-xr-x 2 root root 68412 19. úno 2009 /usr/bin/i486-mingw32-dlltool
-rwxr-xr-x 1 root root 33920 19. úno 2009 /usr/bin/i486-mingw32-dllwrap
-rwxr-xr-x 2 root root 209972 25. srp 15.52 /usr/bin/i486-mingw32-g++
-rwxr-xr-x 1 root root 207316 25. srp 15.52 /usr/bin/i486-mingw32-gcc
-rwxr-xr-x 1 root root 81176 19. úno 2009 /usr/bin/i486-mingw32-gprof
-rwxr-xr-x 2 root root 265696 19. úno 2009 /usr/bin/i486-mingw32-ld
-rwxr-xr-x 2 root root 31428 19. úno 2009 /usr/bin/i486-mingw32-nm
-rwxr-xr-x 2 root root 176864 19. úno 2009 /usr/bin/i486-mingw32-objcopy
-rwxr-xr-x 2 root root 226028 19. úno 2009 /usr/bin/i486-mingw32-objdump
-rwxr-xr-x 2 root root 44468 19. úno 2009 /usr/bin/i486-mingw32-ranlib
-rwxr-xr-x 1 root root 270432 19. úno 2009 /usr/bin/i486-mingw32-readelf
-rwxr-xr-x 1 root root 22856 19. úno 2009 /usr/bin/i486-mingw32-size
-rwxr-xr-x 1 root root 22840 19. úno 2009 /usr/bin/i486-mingw32-strings
-rwxr-xr-x 2 root root 176868 19. úno 2009 /usr/bin/i486-mingw32-strip
-rwxr-xr-x 1 root root 47592 19. úno 2009 /usr/bin/i486-mingw32-windmc
-rwxr-xr-x 1 root root 129940 19. úno 2009 /usr/bin/i486-mingw32-windres
```

Jak vidíte, na disku je hromada zajímavých utilit, které názvy odpovídají utilitám pro sestavování linuxových binárek. Z těchto názvů také můžeme odhadnout, že jde hlavně o assembler a kompilátory jazyků C a C++ (všechny optimalizované pro architekturu i486). Použití je naprosto analogické ke kompilaci céčkových programů ve Windows a na Linuxu.

Hello, world!

Vytvoříme si soubor hello.c s následujícím obsahem:

hello.c

```
#include <stdio.h>
#include <windows.h>
void main(void) {
    char text[] = "This is MinGW hello world!";
    puts(text);
    MessageBox(0, text, "some caption", MB_OK | MB_ICONINFORMATION);
    //For info about MessageBox:
    http://msdn.microsoft.com/en-us/library/ms645505(VS.85).as
}
```

Tento program by po spuštění měl vypsat do konzole text a stejný text zobrazit jako popup.

Kompilace

Ve chvíli, kdy máme uložený soubor `hello.c`, můžeme provést samotnou kompilaci:

```
i486-mingw32-gcc hello.c -o hello.exe
```

tím vznikne soubor `hello.exe`, který už můžeme spustit pomocí WINE...

Pozn.: Protože MinGW gcc defaultně kompiluje windowsí binárky se subsystémem, který zobrazí příkazový řádek a to platí i pro grafické aplikace, u kterých nechceme, aby naše aplikace otvírala nějaké okno terminálu navíc, musíme někdy gcc přidat parametr **-mwindows**, který zajistí že okno bude schované. Takto lze mimo jiné také přeměnit běžný program pro příkazový řádek napsaný v C v daemona, který po spuštění poběží na pozadí a vidět bude jen ve výpisu procesů.

Mimochodem toto je IMHO jeden z četných nedostatků MS Windows. Na Windows musím překompilovat (nebo alespoň přelinkovat) celý program, abych povolil nebo zakázal zobrazení konzole, což vadí hlavně u grafických programů, kde se používá na ladění a třeba u interpreterů různých jazyků, kde se předem neví, jestli program bude textový nebo grafický. A tak výrobci softwaru musí většinou distribuovat dvě verze jinak naprosto stejného programu. Oproti tomu GNU/Linux a podobné se s problémem vypořádaly o dost lépe. Tam totiž platí, že standardní textový (samozřejmě i chybový) výstup má naprosto každý program a záleží jen na rodičovském procesu, který tento program spouští, jak s ním naloží. Například pokud tedy spouštím Chromium (nebo jiný grafický webový prohlížeč) z příkazového řádku, jeho výstup se zobrazí, protože shell jeho výstup čte a zobrazuje, pokud ho ale spouštím pomocí ikonky z grafického prostředí, tak je tento výstup ignorován a vidím jen samotné okno prohlížeče. Samozřejmě je zde i možnost nastavit spouštěcí ikonku tak, aby spustila místo programu terminál, který teprve tento program spustí a výstup zobrazí. Což se opět využívá především u textových aplikací, nebo pro ladění programů.

Spuštění ve WINE

```
0 ;) harvie@harvie-ntb Temp $ WINEDEBUG=fixme-all wine hello.exe  
This is MinGW hello world!
```

Všimněte si snad jen toho, že jsem na začátek řádku, kde spouštím WINE přidal výraz `WINEDEBUG=fixme-all`, který potlačí ladící výpisy z WINE. Ale i tak je je WINE trošku línější a zavedení programu nějakou tu chvíli trvá. Naštěstí výslednou binárku lze použít nejen ve WINE, ale samozřejmě také přímo na Microsoft Windows.

a to je prozatím vše přátelé



From:

<https://wiki.spoje.net/> - **SPOJE.NET**

Permanent link:

<https://wiki.spoje.net/doku.php/howto/programming/mingw-crosscompile>

Last update: **2016/10/29 23:05**

