

Serie: DOSBox unter Windows kompilieren 2/10

Diese Artikelserie zeigt auf, wie man unter Windows eine Entwicklungsumgebung mit Open Source basierten Programmen erstellt und darin Programme und deren Abhängigkeiten kompiliert. Die daraus entstehenden Dateien können unter Windows wie normale Programme verwendet werden.

Im zweiten Teil dieser Artikelserie beschreibe ich die Installation einer Entwicklungsumgebung basierend auf dem MinGW Projekt. Eine Entwicklungsumgebung besteht aus einer Zusammenstellung von Programmen und Werkzeugen, welche zur Erstellung beliebiger Software verwendet werden kann. In solch einer Zusammenstellung können z.B. ein Texteditor zum Bearbeiten der Quelltexte, ein Dateiverwaltungssystem, ein Compiler und ein Linker vorhanden sein.

Wikipedia schreibt zu MinGW

MinGW oder Mingw32 (Minimalist GNU for Windows) ist eine Softwareportierung der GNU-Entwicklerwerkzeuge (GCC, GDB) auf die Windows-Plattform, mit der man Programme für Windows entwickeln kann. MinGW entstand aus dem Cygwin Projekt heraus. Es wird, anders als bei Cygwin, keine Kompatibilitätsschicht in Form einer DLL benötigt. Auch kann hiermit entwickelte Software unter Lizenzen veröffentlicht werden, die nicht mit der GNU GPL kompatibel sind.

Prinzipiell kann man einfach den Installer von der MinGW Webseite herunterladen, anklicken und abwarten. Es hat sich jedoch bei mir gezeigt, dass diese Installationsart im späteren Verlauf zu Problemen führt. So wird z.B. MinGW wie auch MSYS in ein und das selbe Verzeichnis installiert. Das sollte bei neueren Versionen eigentlich kein Problem sein. Sollte... Meine Erfahrungen haben gezeigt, dass es besser ist

alles von Hand zu installieren.

Vorinformationen

- Da eine Entwicklungsumgebung naturgemäss relativ viel Platz belegt, sollte man sich für die Installation eine Disk aussuchen, welche mindestens noch 4-5GB übrig hat.
- Für alle Installationen in dieser Artikelserie wird von folgendem Basis-Installationspfad ausgegangen:
[D:\Entwicklung\MinGW-MSYS](#)
- Wer sich bereits ein wenig mit der Thematik des kompilierens auseinander gesetzt hat, kann getrost die neusten Versionen aller in dieser Artikelserie beschriebenen Pakete herunterladen. Man muss aber damit rechnen, dass unerwartete Fehler auftreten können. Wer auf Nummer sicher gehen will, sollte genau die in diesem Artikel angegebenen Versionen herunterladen.

Installation und Konfiguration

1. In dem oben angegebenen Basis-Installationspfad werden als erstes alle Pakete aus der **unten stehenden Tabelle** heruntergeladen.
2. Im Basis-Installationspfad wird vom gzip Archiv **bin/gzip** entpackt, vom libarchiv die Dateien **bin/{bsdtar,libarchive2.dll}** und vom Dependencie Paket die beide Bibliotheken **bin/{bzip2,zlib1}.dll**.
3. **bsdtar.exe** in **tar.exe** umbenennen.
4. Innerhalb des Basis-Installationspfades muss ein **MinGW** Verzeichnis angelegt werden in welchem alle heruntergeladenen Pakete entpackt werden. Hierzu öffnet man eine Eingabeaufforderung (Start/Ausführen/cmd [Enter]) und gibt darin die folgenden Befehle ein.

```
cd /D D:\Entwicklung\MinGW-MSYS\  
mkdir MinGW
```

```

tar -xvzf binutils-* -C ./MinGW/
tar -xvzf gcc-core-* -C ./MinGW/
tar -xvzf gcc-g++-* -C ./MinGW/
tar -xvzf gcc-ada-* -C ./MinGW/
tar -xvzf gcc-g77-* -C ./MinGW/
tar -xvzf gcc-java-* -C ./MinGW/
tar -xvzf gcc-objc-* -C ./MinGW/
tar -xvzf mingwrt-*-mingw32-dev* -C ./MinGW/
tar -xvzf mingwrt-*-mingw32-dll* -C ./MinGW/
tar -xvzf w32api-* -C ./MinGW/

```

Paket	Beschreibung	Link
binutils	Enthält Programme wie den Assembler, den Linker und viele andere.	binutils-2.19.1
gcc-core	Grundlegende Kompiliertools sowie der C-Compiler aus der Gnu Compiler Collection.	gcc-core-3.4.5
gcc-g++	Der C++ Compiler aus der Gnu Compiler Collection.	gcc-g++-3.4.5
gcc-ada	Der ada Compiler aus der Gnu Compiler Collection.	gcc-ada-3.4.5
gcc-g77	Der g77 Compiler aus der Gnu Compiler Collection.	gcc-g77-3.4.5
gcc-java	Der java Compiler aus der Gnu Compiler Collection.	gcc-java-3.4.5
gcc-objc	Der Objective C Compiler aus der Gnu Compiler Collection.	gcc-objc-3.4.5
mingw32-dev	Grundlegende Dateien der MinGW Runtime Umgebung.	mingwrt-3.15.2-dev
mingw32-dll	Basisbibliotheken der MinGW Runtime Umgebung.	mingwrt-3.15.2-dll

mingw32-API	Essenzielle Header Dateien sowie Bibliotheken der MinGW Runtime Umgebung.	w32api-3.13
gzip	gzip ent-/packer um die .gz Dateien entpacken zu können	gzip.zip
tar	tar ent-/packer um die .tar Dateien entpacken zu können	tar.zip
dependencies	Bibliotheksabhängigkeiten zu gzip und tar.	deps.zip

Im dritten Teil dieser Artikelserie wird die MSYS Umgebung installiert, welche erst den sinnvollen Umgang mit den Entwicklungswerkzeugen ermöglicht.