

# **GCCSDK**

**<http://gccsdk.riscos.info/>**

**29 mei 2010**

**John Tytgat**

# Wat ?

Alles wat je nodig hebt qua tools om te programmeren  
in C/C++ voor RISC OS

- ARM assembler, disassembler
- C/C++ compiler (gcc, clang, lcc)
- CMunge (nodig voor RISC OS modules te maken)
- mkresfs (om RISC OS Resource:\$ datablobs aan te maken)
- Make (tool om sources te bouwen)

# Wat ?

- Gratis tegenhanger van Acorn C/C++ (en... meer & beter ? ☺ )
- Sources van alle tools zijn beschikbaar (Open Source)
- Tools draaien op RISC OS zelf maar ook op niet RISC OS platformen

# Geschiedenis

- 1990 ish : ports van Perl, Ada en GCC 2.4.5
- April 1997 : port van GCC 2.7.2.1, Nick Burrett
- Mid 1999 “The GCC for RISC OS initiative”  
<http://www.arcsite.de/hp/gcc/> (Steffen Huber)  
Huber)
- 24 Sep 1999 : GCC 2.95.1 Pre-release 1
- 16 Nov 1999 : GCC 2.95.2 Pre-release 2

# Geschiedenis

- 5 Jan 2003 : GCC 2.95.4 Release 3
- 18 Apr 2004 : GCC 3.3.3 Pre-release 2
- 5 Oct 2004 : GCC 3.4.1 Release 1
- 25 Dec 2004 : GCC 3.4.4 Release 1
- 6 May 2005 : GCC 3.4.4 Release 2
- 15 Feb 2006 : GCC 3.4.5 Release 1
- 16 Jun 2006 : GCC 3.4.6 Release 1

# Geschiedenis

- 17 Sep 2006 : GCC 3.4.6 Release 2
- 17 Jul 2007 : GCC 3.4.6 Release 3
- 17 Nov 2007 : GCC 4.1.1 Pre-release 1
- 23 Dec 2008 : GCC 4.1.1 Release 1
- 27 Dec 2009 : GCC 4.1.1 Release 2

# Geheim wapen : UnixLib

- Een “runtime library” zoals SharedCLibrary maar ‘meer’ dan dat alleen
- Voorziet standaard C library functies (C90, C99), zo goed mogelijk POSIX compliant
- Voorziet threading mogelijkheid
- Kan automatisch vertalen tussen Unix en RISC RISC OS file paden
- Kan subprocessen opstarten

# Geheim wapen : UnixLib

Kortom: ideale basis library om te gebruiken bij  
bij RISC OS ports van Unix (gebaseerde)  
programma's !

Geschiedenis:

- Ontwikkeling gestart in 1989 voor NetHack  
(gereleased op 30 Oct 1990)
- UnixLib 2.0 : Eerste officiële release was 18  
Jan 1991.



# Nieuw in GCC 4.1.1

- GCC compiler : significante verbetering
- Intermediaire object code en finale binary is ELF, en niet meer AOF / AIF
- Elf2aif
- Shared libraries

# Hoe bouw je een compiler ?

- Compiler = code transformator, b.v. van C naar ARM code
- Een compiler die loopt op processor XYZ hoeft niet noodzakelijk code te produceren voor die processor XYZ  
Indien wel: “native compiler”  
Indien niet: “cross compiler”
- Probleem: we hebben een platform waar geen compiler voor bestaat. Wat kan je doen ?

# Hoe bouw je een compiler ?

- Je hebt nodig : op platform *XYZ* een werkende C compiler die code produceert voor processor *XYZ*  
“C code” → “C compiler @ platform *XYZ* met output output *XYZ*” → “*XYZ* code”
- Op platform *XYZ*: “C sources van C compiler + beschrijving van ARM processor” → “C compiler @ platform *XYZ* met output *XYZ*” → “C compiler @ platform *XYZ* die ARM output genereert”
- We hebben nu een cross compiler gemaakt

# Hoe bouw je een compiler ?

- Op platform XYZ: “C sources van C compiler compiler + beschrijving van ARM processor” processor”
  - “C compiler @ platform XYZ met output ARM” (= cross compiler)
  - “C compiler @ platform ARM die ARM output genereert”
- We hebben nu een native compiler gemaakt

# GCCSDK : praktisch

- Ofwel download je RISC OS native compiler en gebruik je die op RISC OS
- Ofwel bouw je de cross compiler op een niet RISC OS platform naar keuze

[http://www.riscos.info/index.php/Using\\_GCCSDK](http://www.riscos.info/index.php/Using_GCCSDK)

- Kies: Linux (Ubuntu), Cygwin (Windows), etc
- Installeer Subversion en download GCCSDK
- “./build-world”

# GCCSDK : praktisch

Hoe gebruik je de cross-compiler ?

- Vanuit RISC OS inloggen via Nettle <http://nettle.sourceforge.net>, sources accesseren via Sunfish (NFS)

- Installeer syslogd op RISC OS

```
$ export THROWBACK_HOST=riscos_machine_naam
```

```
$ arm-unknown-riscos-gcc -mthrowback -o file.o -c file.c
```

# Autobuilder

- Build systeem om ports te automatiseren en met minimale moeite te onderhouden
- Gestart als "Unix Porting Project" door Peter Naulls Naulls
- Maakte in 2005 mogelijk om de eerste Firefox port te te doen (met verschillende updates nadien)
- Meer dan 350 projecten/ports gedefinieerd maar een klein deel is recentelijk gebuild en gepubliceerd.
- Enorm potentieel maar... onderbenut
- Geïnteresseerde vrijwillers om te helpen ?

# GCCSDK successen

- Autobuilder : Firefox & andere ports (cvs, Subversion, games, ...)
- RISC OS GCC compiler zelf (met alle tools hierrond)
- PostScript 3 printer driver

Nabije toekomst ?

- Gebruik van de cross compiler voor ROOL



# Toekomst

- C++ & netwerk support voor SharedCLibrary programma's (incl modules)
- Binutils 2.20 : full support voor NEON & VFP ARM ARM instructies (Cortex A8)
- GCC 4.5/4.6 port : support voor NEON & VFP
- clang
  - Open Source compiler gebaseerd op LLVM
  - Vooral door Apple ontwikkeld (en gebruikt in Apple XCode, iPhone, iPad SDKs)
  - Moderne architectuur
  - Sneller, vrij goede code generatie, extreem gebruiksvriendelijke fout meldingen

# Toekomst

- Switch van APCS-32 naar EABI ?
- Verdere ontwikkeling van remote debugger
- Verdere ontwikkeling van asasm : ObjAsm syntax ARM assembler met AOF/ELF output, output, essentieel voor ROOL builds
- Autobuilder gebruiken !

# Konklusie

Als je C/C++ programmeur bent:

- Gebruik deze tools.
- Laat je ervaring (goed & minder goed) weten op de mailing list en help/participeer mee in de ontwikkeling hiervan.
- Overweeg om je programma's te ontwikkelen via cross compilatie (kan je veel tijd besparen).

<http://gccsdk.riscos.info/>