

Introduction to software development tools

.S

make

.h



.C++

.C

Introduction to software development tools

Michael Opdenacker

Free Electrons

<http://free-electrons.com>

Created with [OpenOffice.org](http://openoffice.org) 2.x

Thanks to Nicolas Rougier (Copyright 2003, <http://webloria.loria.fr/~rougier/>) for the Tux image



Introduction to software development tools
© Copyright 2005-2007, Free Electrons
Creative Commons Attribution-ShareAlike 2.5 license
<http://free-electrons.com>

Jan 14, 2009



Rights to copy



Attribution – ShareAlike 2.5

You are free

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions

Attribution. You must give the original author credit.

BY: **Share Alike.** If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

License text: <http://creativecommons.org/licenses/by-sa/2.5/legalcode>

© Copyright 2005-2007

Free Electrons

feedback@free-electrons.com

Document sources, updates and translations:

<http://free-electrons.com/articles/swdev>

Corrections, suggestions, contributions and translations are welcome!



Best viewed with...

This document is best viewed with a recent PDF reader or with OpenOffice.org itself!

- ▶ Take advantage of internal or external hyperlinks. So, don't hesitate to click on them!
- ▶ Find pages quickly thanks to automatic search
- ▶ Use thumbnails to navigate in the document in a quick way

If you're reading a paper or HTML copy, you should get your copy in PDF or OpenOffice.org format on <http://free-electrons.com/articles/swdev!>



Contents

Source management

- ▶ Program build automation
- ▶ Integrated development environments (IDEs)

Debugging and analysis tools

- ▶ Debuggers
- ▶ Static code checkers
- ▶ Memory checkers
- ▶ System analysis

Development environments

- ▶ Developing on Windows



Embedded Linux development tools

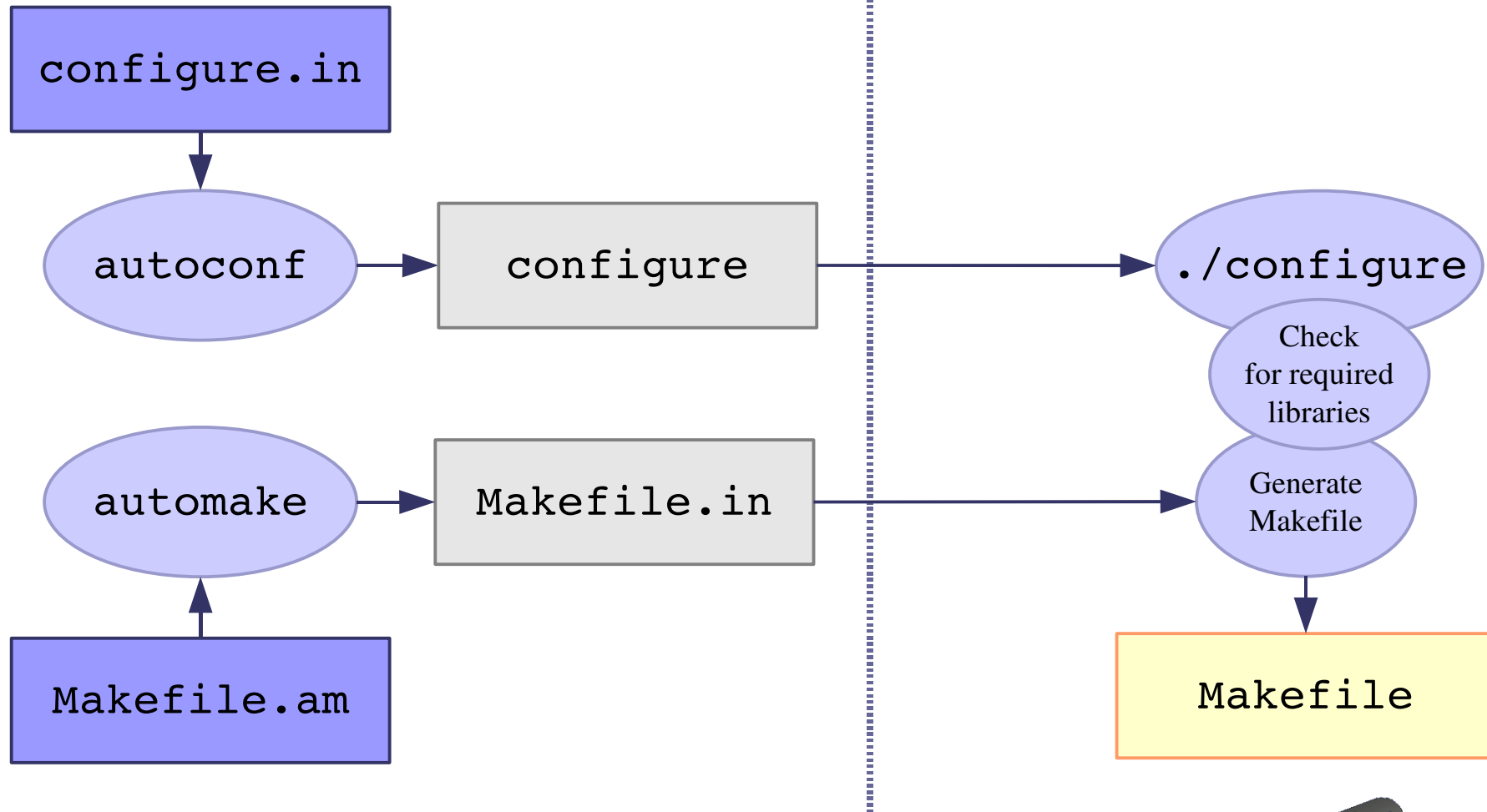
Source management
Program build automation



GNU toolchain flow (simplified)

Developer

User



GNU toolchain strengths and weaknesses

Universal solution for Unix systems

- Still used to build most Free Software C and C++ programs, in particular in GNU
- Very powerful and easy for end users:
`./configure --prefix=/opt`
`make`
`make check`
`make install`
- Causes of check failures on the user side sometimes not easy to figure out.
- No standard configure options
- Output very complex Makefiles

Difficult for developers to master

- Lots of details to learn and understand
- Tedious documentation
- Multiple input and output files
- Causes of check failures on the user side sometimes not very easy to figure out.

See <http://freshmeat.net/articles/view/889/>, <http://freshmeat.net/articles/view/1702/> and <http://freshmeat.net/articles/view/1715/> for discussions about the limitations of `autoconf` / `automake` / `make` and alternatives.



SCons (1)

<http://www.scons.org/>

- ▶ An alternative to **make**, written in Python
A successor of the **Cons** program, written in Perl.
- ▶ Takes care of all the compiling and dependency handling job.
Dependencies handled with MD5 checksums by default.
Running **touch** on source files will have no effect!
- ▶ Also supports **autoconf** like capabilities: checking for existing libraries, headers or programs on the user system.
- ▶ Unlike **make**, **autoconf**, **automake**, no specific format!
Access to all Python syntax and scripting facilities.



SCons (2)

- ▶ Native support for lots of compiler based languages:
C, C++, D, Java, Fortran, Yacc, Lex, Qt and SWIG, TeX and LaTeX
Easily extensible through user-defined builders.
- ▶ Supposed to support building on multiple platforms:
GNU/Linux, Unix, Windows NT, MacOS X, OS/2
- ▶ Built-in support for fetching source files from SCCS, RCS, CVS, and Subversion.
Caution: Monotone, git and Mercurial not supported (Apr. 2007).
- ▶ **SCons** was once considered as the new build system for KDE.
However, KDE developers faced major issues building on non Linux platforms and lacked support from mainstream **SCons** developers.
SCons also has the reputation of being slow and inefficient
(bad for big projects).



SCons (3)

- ▶ Simple SConstruct file:

```
Library('foo', ['f1.c', 'f2.c', 'f3.c'])  
Program('prog.c', LIBS=['foo', 'bar'], LIBPATH='.')
```

- ▶ Nice and friendly user manual full of examples:

<http://www.scons.org/doc/HTML/scons-user/book1.html>



cmake



<http://cmake.org/>

- ▶ Cross-platform, extensible, open-source make system
A strong alternative to **Scons**!
- ▶ Not a build system by itself.
It generates building control files for each target platform:
Makefiles on GNU/Linux and Unix,
Workspaces / projects on **MS Visual C++**.
- ▶ Supports complex, large projects.
Adopted by KDE developers for their version 4 release.
- ▶ Can generate project files for **KDevelop3**, **MSVC 6,7**, **XCode**.

Nice introduction: <http://lwn.net/Articles/188693/>



cmake example code

```
# Project name
PROJECT(HELLO)

# Source code subdirectories
SUBDIRS(Hello Demo)

# Create a library from the hello.cxx file
ADD_LIBRARY(Hello hello.cxx)

# Location of library include files
INCLUDE_DIRECTORIES(${HELLO_SOURCE_DIR}/Hello)

# Library location for the linker
LINK_DIRECTORIES(${HELLO_BINARY_DIR}/Hello)

# helloDemo executable built from demo.cxx and demo_b.cxx
ADD_EXECUTABLE(helloDemo demo.cxx demo_b.cxx)

# Link the executable to the Hello library.
TARGET_LINK_LIBRARIES(helloDemo Hello)
```

Example
CMakeLists.txt file



Embedded Linux development tools

Source management Integrated Development Environments (IDE)



Source browsers

- ▶ **LXR: Linux Cross Reference**

Allows to browse code through a web browser.

- ▶ **cscope**

Console mode source browsing tool.

Integrates with editors like **vi** and **emacs**.

- ▶ **KScope**

A graphical interface to **cscope**.

See our Linux kernel and driver development training materials (<http://free-electrons.com/training/drivers>) for more details.



KDevelop

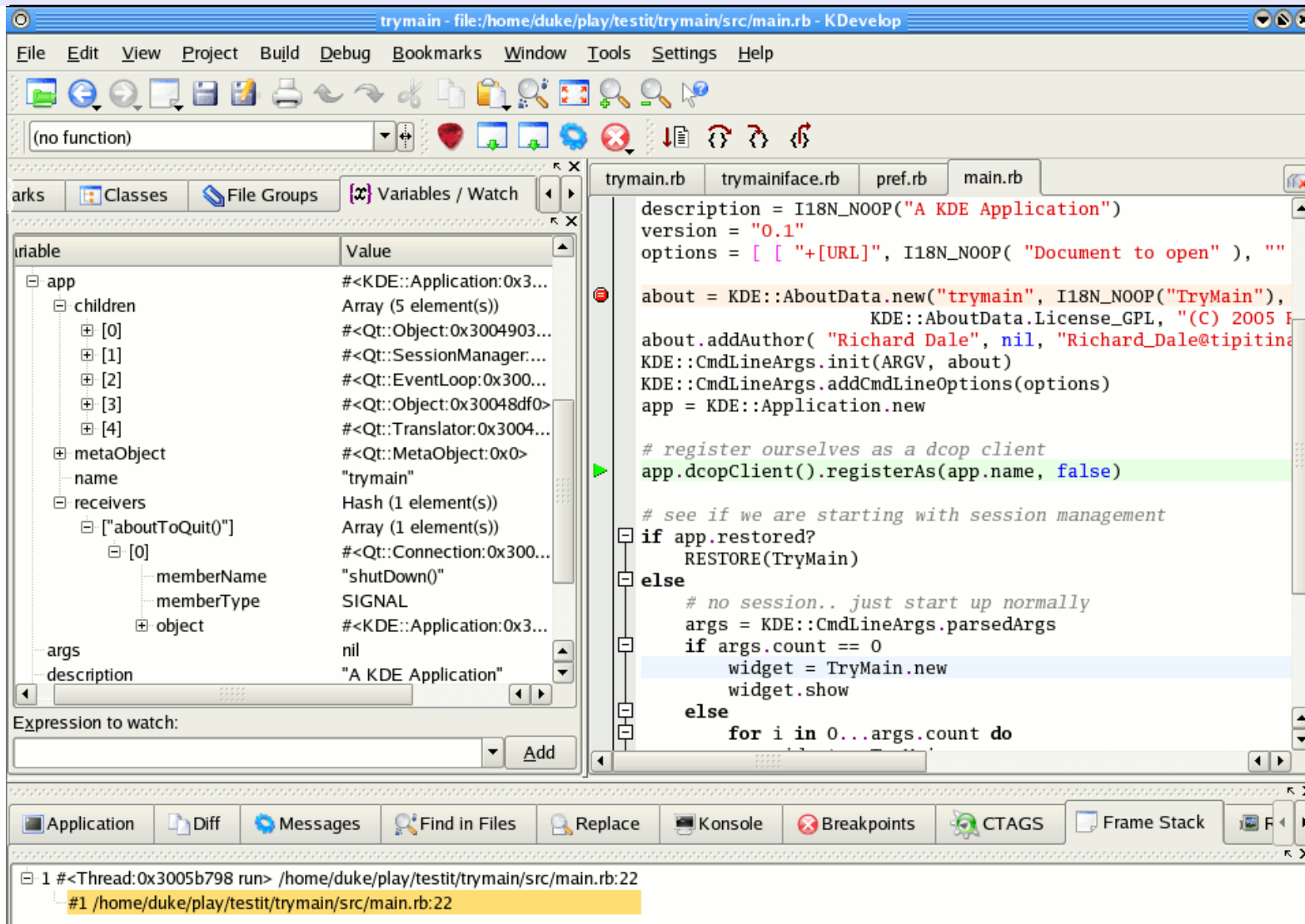
<http://kdevelop.org>

- ▶ A full featured IDE!
- ▶ License: GPL
- ▶ Supports many languages:
Ada, C, C++, Database, Java, Perl, PHP, Python, Ruby, Shell
- ▶ Supports many kinds of projects:
KDE, but also GTK, Gnome, kernel drivers, embedded (Opie)...
- ▶ Many features: editor, syntax highlighting, code completion, compiler interface, debugger interface, file manager, class browser...

Nice overview: <http://en.wikipedia.org/wiki/Kdevelop>



KDevelop screenshot



Ruby
debugger



Anjuta DevStudio

<http://www.anjuta.org/>



Anjuta DevStudio

- ▶ Gnome's IDE
Less features than **KDevelop** so far.
- ▶ License: GPL
- ▶ Supported languages: C and C++
- ▶ Supported project types: command line, **GTK**, **Gnome**, **wxWindows**, **Xlib**.
- ▶ Features: integrated editor, syntax highlighting, code completion, compiler and debugger interface, tag browser.



Anjuta screenshot

The screenshot displays the Anjuta IDE interface. The main window shows an Inheritance Graph with the following structure:

```
graph TD; DocWatcher --> Editor; LexerModule --> ExternalLexerModule; Font --> FontCached; Window --> ListBox; wxApp --> MyApp; Editor --> ScintillaBase; ExternalLexerModule --> ScintillaBase; FontCached --> ListBoxX; ListBox --> ListBoxX; ScintillaBase --> ScintillaGTK;
```

The left sidebar shows a Symbols tree with a list of classes and methods under the 'Classes' category. The right sidebar shows a Project view with a list of files and folders. The bottom window is a Terminal showing the output of the 'ls' command:

```
[naba@localhost naba]$ ls
accounts  belzabar  Documents  evolution  hexagon.png  logo  Movies  rpms  vnc-scripts
anjuta   Desktop  Downloads  GNUstep   Images       mail  Projects  test.txt
[naba@localhost naba]$
```

The terminal status bar at the bottom indicates: Col: 000 Line: 0117 Project: anjuta Mode: INS Zoom: 0

Anjuta's class inheritance graph and terminal



Eclipse (1)

<http://www.eclipse.org/>



- ▶ An extensible, plugin based software development kit, typically used for creating IDEs.
- ▶ Supported by the Eclipse foundation, a non-profit consortium of major software industry vendors (IBM, Intel, Borland, Nokia, WindRiver, Zend, Computer Associates...).
- ▶ Free Software license (Eclipse Public License). Incompatible with the GPL.
- ▶ Supported platforms: GNU/Linux, Unix, Windows

Extremely popular: created a lot of attraction.



Eclipse (2)

- ▶ Eclipse is actually a platform composed of many projects:
<http://www.eclipse.org/projects/>
- ▶ The platform is used by major embedded Linux software vendors for their (proprietary) system development kits:
MontaVista DevRocket, TimeSys TimeStorm, Windriver Workbench, Sysgo ELinOS.

Eclipse is a huge project.

It would require an entire training session!



Embedded Linux development tools

Debugging and analysis tools Debuggers

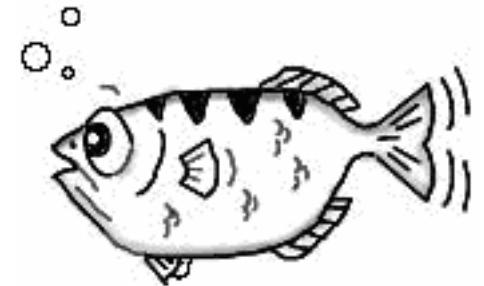


GDB

The GNU Project Debugger

<http://www.gnu.org/software/gdb/>

- ▶ The debugger on GNU/Linux, available for most embedded architectures.
Supported languages: C, C++, Pascal, Objective-C, Fortran, Ada...
- ▶ Console interface (useful for remote debugging).
Graphical front-ends available.
- ▶ Can be used to control the execution of a program, set breakpoints or change internal variables. You can also use it to see what a program was doing when it crashed (by loading its memory image, dumped into a `core` file).



See also <http://en.wikipedia.org/wiki/Gdb>



GDB graphical front-ends

- ▶ **DDD - Data Display Debugger**

<http://www.gnu.org/software/ddd/>

The most popular graphical front-end, with advanced data plotting capabilities.

- ▶ **GDB/Insight**

<http://sources.redhat.com/insight/>

From the **GDB** maintainers (Red Hat).

- ▶ **KDbg**

<http://www.kdbg.org/>

Another front-end, for the **K Display Environment**.



gdbserver

Possibility to debug with `gdb` without needing the full `gdb` (2.4 MB on `i386`) on the target.

- ▶ `gdbserver` is a lightweight program (only 40 KB on `arm`). This makes it easy to include and port on new targets.
- ▶ All the complexity (and possibly a `gdb` front-end) is on the host computer.
- ▶ Host-target connection through a serial or networking link.



gdb server requirements

Target

- ▶ The executables can be stripped!

Host

- ▶ Need a copy of the executable.
- ▶ The executable has to contain debugging information.
- ▶ Graphical front-ends supporting remote execution:
Only **Insight** lets the user specify a connection to a remote target. In other front-ends (**DDD**, **KDBG**), a **gdb** configuration file can be used.



gdbserver connection

Serial line

- ▶ Target:
> `gdbserver /dev/ttyS0 <program> <args>`
- ▶ Host:
`gdb > target remote /dev/ttyS1`

Networking (TCP)

- ▶ Target:
> `gdbserver host:1234 <program> <args>`
Possible to use the host IP address too!
- ▶ Host:
`gdb > target remote target:1234`



Embedded Linux development tools

Debugging and analysis tools Static code checkers



Splint

<http://splint.org/>, from the University of Virginia

- ▶ GPL tool for statically checking C programs for security vulnerabilities and coding mistakes
- ▶ Today's `lint` program for GNU/Linux. The successor of LClint.
- ▶ Very complete manual and documentation
- ▶ Doesn't support C++



Embedded Linux development tools

Debugging and analysis tools Memory checkers



memcheck

<http://hald.dnsalias.net/projects/memcheck/>

- ▶ GNU GPL tool for dynamic memory checking
- ▶ Works by replacing glibc's memory management functions by its own.
- Supports many GNU/Linux and other Unix architectures (unlike GNU Checker): **i386**, **arm**, **alpha**, **ia64**.
Many others not tested.



ElectricFence

`efence`, GNU GPL, from Bruce Perens

<http://directory.fsf.org/devel/debug/ElectricFence.html>

- ▶ Stops your program on the exact instruction that overruns or underruns a `malloc()` memory buffer.
- ▶ GDB will then display the source-code line that causes the bug.
- ▶ Works by using the virtual-memory hardware to create a red-zone at the border of each buffer - touch that, and your program stops.
- Supposed to work on any platform supported by Linux, whatever the cpu (provided virtual memory support is available).



Valgrind (1)

<http://valgrind.org/>



- ▶ GNU GPL Software suite for debugging and profiling programs.
- ▶ Supported platforms: Linux on **x86**, **x86_64**, **ppc32**, **ppc64**
Others: compile your program to these platforms to use **Valgrind**.
- ▶ Can detect many memory management and threading bugs.
- ▶ Profiler: provides information helpful to speed up your program and reduce its memory usage.
- ▶ The most popular tool for this usage.
Even used by projects with hundreds of programmers.



Valgrind (2)

- ▶ Can be used to run any program, without the need to recompile it.
- ▶ Example usage

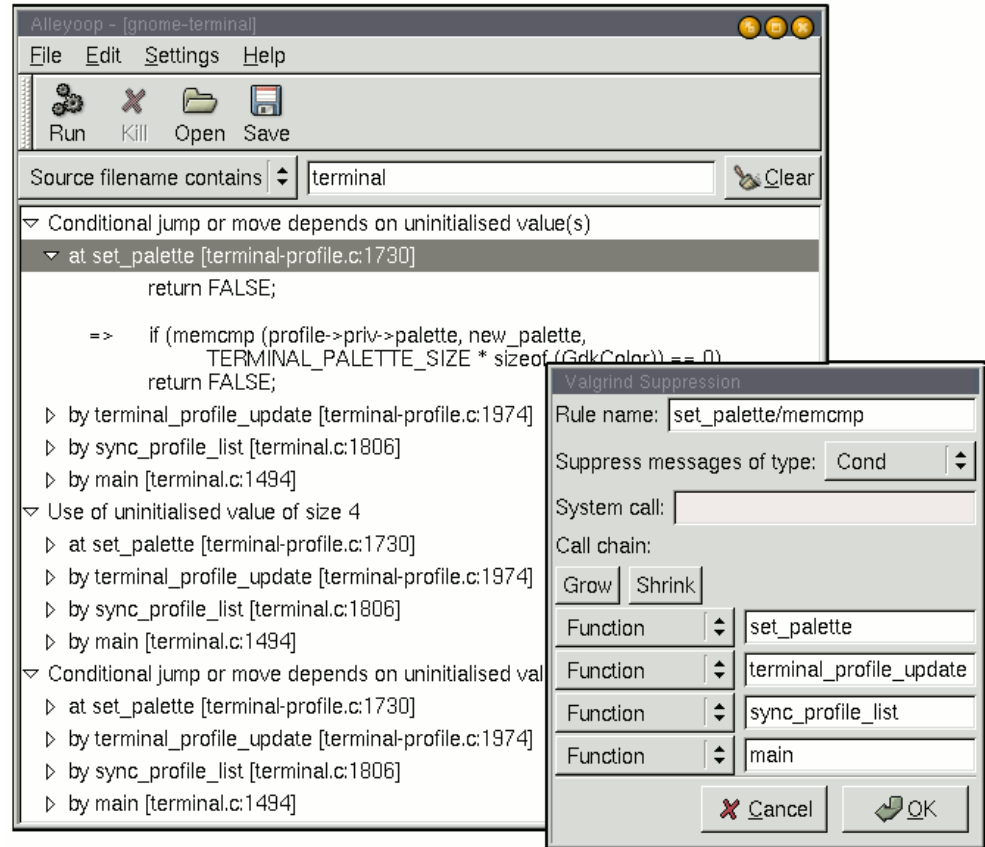
```
valgrind --leak-check=yes ls -la
```
- ▶ Works by adding its own instrumentation to your code and then running in on its own virtual x86 (or ppc) core.
Significantly slows down execution, but still fine for testing!
- ▶ More details on <http://valgrind.org/info/> and http://valgrind.org/docs/manual/coregrind_core.html#howworks



Alleyoop

<http://alleyoop.sourceforge.net/>

Graphical front-end to Valgrind



Embedded Linux development tools

Debugging and analysis tools System analysis



strace

System call tracer

<http://sourceforge.net/projects/strace/>

- ▶ Available on all GNU/Linux systems
- ▶ Allows to see what any of your processes is doing:
accessing files, allocating memory...
Easy substitute to debuggers in simple cases.
- ▶ Usage:
`strace <command>` (starting a new process)
`strace -p<pid>` (tracing an existing process)

See `man strace` for details.

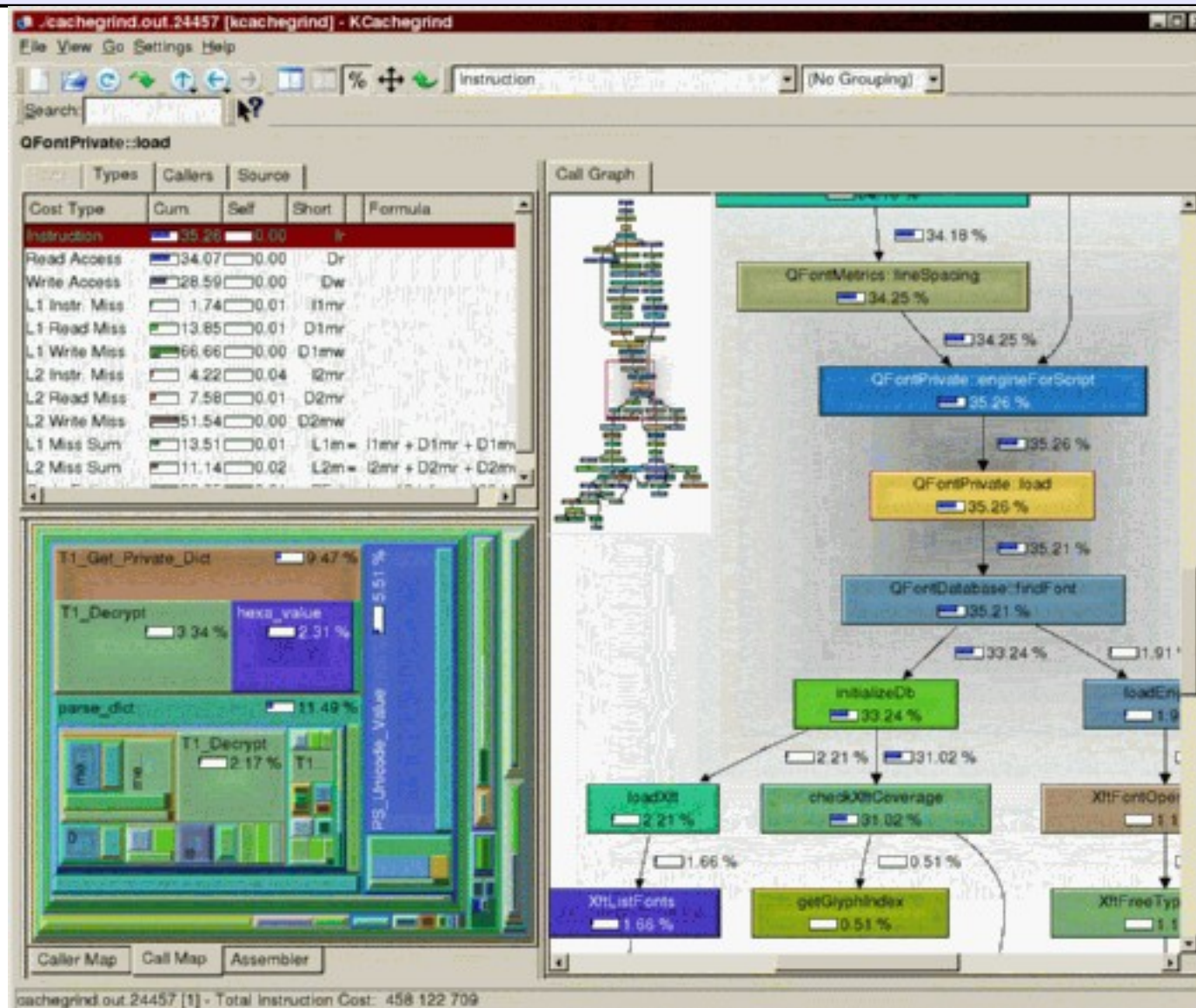


Callgrind / KCachegrind

- ▶ **Cachegrind / Callgrind**: part of the **Valgrind** tool suite
Collects function call statistics and call graphs.
Useful to know in which functions most time is spent.
- ▶ **KCachegrind**: <http://kcachegrind.sourceforge.net/>
An amazing visualizer for **Cachegrind / Callgrind** data.
- ▶ **KCachegrind** can also import data from other profilers (such as **OProfile**), and from profiling output from Python, Perl and PHP.



KCachegrind screenshot



Embedded Linux development tools

Developing on Windows



Developing on Windows !?

Using a GNU/Linux workstation is the easiest way to create software for GNU/Linux or embedded Linux

- ▶ You use the same tools and environment as all community developers do. Much fewer issues you are the only one to face.
- ▶ You get familiar with the system.
Essential for understanding issues.

However, some developers have no choice:

Windows is the only desktop OS allowed in their company.



Cygwin

<http://cygwin.com/>

Linux (POSIX)-like environment for Windows



- ▶ 2 components:
Linux API emulation layer: `cygwin1.dll`
A collection of tools originally found in GNU/Linux
- ▶ Allows to compile and run many GNU/Linux programs on Windows: shells, compiler, http servers, **X Window**, **GTK...**
- ▶ Very easy to install. Can choose which tools to download and install.
- ▶ For embedded Linux system developers:
makes it possible to use GNU toolchains (compiled for Windows) required to build Linux binaries (kernel, libraries or applications).

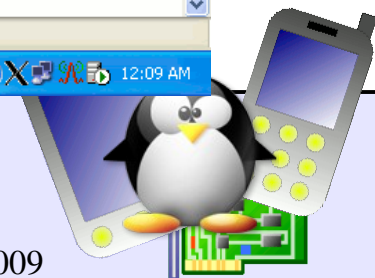


Cygwin screenshot

The screenshot displays a Cygwin environment with three main windows:

- Terminal (X -):** Shows a shell prompt `harold@aerosmith` with the command `$ xeyes` and its output `[3] 4040`. A second prompt shows the command `$` followed by a cursor.
- Debugger Console (DDD: Debugger Console):** Shows a menu bar (File, Edit, View, Program, Commands, Status, Source, Data, Help) and a toolbar with icons for Lookup, Find, Break, Watch, Print, Display, Plot, Show, Rotate, Set, and Undo. The main area is empty.
- Circuit Diagram Editor (Xfig - ex06_fig01.fig):** Shows a circuit diagram with nodes labeled `A`, `X`, `Y`, `B`, `D`, `E`, `F`, `G`, `H`, and `C`. Connections are labeled with delays: `1ms`, `10ms`, and `100ms`. The window includes a toolbar, a menu bar, and a right-hand panel with options like **Depths** (All On, All Off, Toggle, Gray, Blank), **Front** (checked 50), and **Back**.

The taskbar at the bottom shows the Start button, several open windows (Cygwin/X - Screens..., X ~, DDD: Debugger Co..., Xfig - ex06_fig01.fig, xeyes), and the system tray with the time 12:09 AM.



Cygwin limitations

Cygwin is not a complete substitute for a real GNU/Linux system.

- ▶ Almost all developers work on GNU/Linux or on another Unix platform (typically BSD). Don't expect them to test that their tools build on Windows with Cygwin.
- ▶ The number of Cygwin users is quite small.
You may be the first to face or report building issues on this platform for a given compiler or tool version.

So, the best solution is to run Linux inside Windows!



QEMU on Windows

The QEMU logo is displayed in a stylized, multi-colored font (green and yellow) with a slight shadow effect.

<http://www.h7.dion.ne.jp/~qemu-win/>

Fast processor emulator using a portable dynamic translator.

- ▶ License: GNU GPL
- ▶ Allows to run a full virtual PC, emulating processor instructions. Originally runs on **Linux**.
- ▶ Now available on **Windows** too and very stable.
- ▶ Better to use **kqemu** accelerator (now free), otherwise compiling big projects (Linux kernel) can be very long).



VMware

<http://en.wikipedia.org/wiki/VMware>



- ▶ License: proprietary
- ▶ Can run a **GNU/Linux** PC from **Windows**, almost at the host speed.
- ▶ **VMware Player** is now available free of charge.
Many Free Software system images available for download.

The most popular solution in the corporate world.



Related documents

All the technical presentations and training materials created and used by Free Electrons, available under a free documentation license (more than 1500 pages!).

<http://free-electrons.com/training>

- ▶ Introduction to Unix and GNU/Linux
- ▶ Embedded Linux kernel and driver development
- ▶ Free Software tools for embedded Linux systems
- ▶ Audio in embedded Linux systems
- ▶ Multimedia in embedded Linux systems

<http://free-electrons.com/articles>

- ▶ Advantages of Free Software in embedded systems
- ▶ Embedded Linux optimizations
- ▶ Embedded Linux from Scratch... in 40 min!

- ▶ Linux USB drivers
- ▶ Real-time in embedded Linux systems
- ▶ Introduction to uClinux
- ▶ Linux on TI OMAP processors
- ▶ Free Software development tools
- ▶ Java in embedded Linux systems
- ▶ Introduction to GNU/Linux and Free Software
- ▶ Linux and ecology
- ▶ What's new in Linux 2.6?
- ▶ How to port Linux on a new PDA



How to help

If you support this work, you can help ...

- ▶ By sending corrections, suggestions, contributions and translations
- ▶ By asking your organization to order training sessions performed by the author of these documents (see <http://free-electrons.com/training>)
- ▶ By speaking about it to your friends, colleagues and local Free Software community.
- ▶ By adding links to our on-line materials on your website, to increase their visibility in search engine results.



Thanks

- ▶ To the OpenOffice.org project, for their presentation and word processor tools which satisfied all my needs.
- ▶ To the Handhelds.org community, for giving me so much help and so many opportunities to help.
- ▶ To the members of the whole Free Software and Open Source community, for sharing the best of themselves: their work, their knowledge, their friendship.



Free Electrons services

Embedded Linux Training

- Unix and GNU/Linux basics
- Linux kernel and drivers development
- Real-time Linux
- uClinux
- Development and profiling tools
- Lightweight tools for embedded systems
- Root filesystem creation
- Audio and multimedia
- System optimization

Custom Development

- System integration
- Embedded Linux demos and prototypes
- System optimization
- Linux kernel drivers
- Application and interface development

Consulting

- Help in decision making
- System architecture
- Identification of suitable technologies
- Managing licensing requirements
- System design and performance review

Technical Support

- Development tool and application support
- Issue investigation and solution follow-up with mainstream developers
- Help getting started

<http://free-electrons.com>

