

## toolview

Elke ingenieur gebruikt tools en iedereen heeft zijn eigen voorkeur; vaak gebaseerd op wat er toevallig als eerste beschikbaar was. Zo ook de ICT-ingenieur. Ook hij gebruikt tools, software-tools wel te verstaan. Daar zijn er oneindig veel van, waardoor kiezen erg moeilijk is. Veel handige tools worden niet gebruikt, eenvoudigweg omdat ze niet bekend zijn.

Het leren gebruiken van een tool kost tijd. Maar als je de naam en het doel van het tool kent, loont het gebruik al vaak de moeite. Het inleren en inzetten kost vaak minder tijd dan aanmoderen zonder goed tool.

Deze rubriek stelt dergelijke handige tools voor; dit keer eenvoudige interfaces voor de GDB debugger.

## GDB-GUI'S

Langzamerhand begint mijn voorraad tools op te raken! Daarom de oproep om 'handige tools' te mailen. Een naam en een URL is voldoende. Desnoods neem ik contact op voor meer info. En ik zal het schrijfwerk doen. DANK!

Mail me, voor meer info  
ALbert dot Mietus at PTS dot nl

### door & more

*De meeste oude vertrouwde GNU tools zijn ouder dan 'het internet'. Veel oude historie is er dus niet op het WWW; GDB is te oud.*

*Maar er is genoeg nieuws te vinden.*

*De beste plaats om te beginnen is <http://www.gnu.org/software/gdb>*

*De besproken, moderne grafische interfaces zijn te vinden op:*

*<http://sources.redhat.com/insight> en <http://www.gnu.org/software/ddd>*

*Een geheel verouderde screenshot is te vinden op: <http://www.cs.uni.edu/Help/xxgdb.html>*

*En een heel fraai stukje design kun je vinden op <http://sources.redhat.com/gdb/papers/libgdb/2/wp/gdbtk.html> (Klik op het kladje voor nog een voorbeeld van een koffiepoot-ontwerp.)*

*Over eclipse is een boekenkast vol geschreven; bezin voor gij surft naar [www.ecslipse.org](http://www.ecslipse.org)*

*Tot slot, een introductieles over de moderne, grafische en geïntegreerde manier van debuggen met emacs' GDB-UI vind je op: <http://www.linuxjournal.com/node/7876>*

"Als een tester bugs moeten verwijderen" riep een collega ooit, "dan moet de programmeur ze eerst maken". Maar toch, 'debuggen' is werk voor de programmeur. En de debugger is, net als een editor, deel van zijn primaire gereedschap. Maar hoe-wel er heilige oorlogen gevoerd worden over 'vi', 'emacs' of 'MS-studio'; discussies over 'GDB' zijn er nooit. Iedereen lijkt het eens ...

### GDB heeft geen GUI ...

GDB stamt uit de tijd dat programmeurs nog geen klik-en-go interface kenden. Uit de tijd dat 'vi' – met de V van visual (!) – modern was en uit de tijd van K&R C – dus zonder prototypes. Kortom, een GUI was geen optie. Het gebruik van getline() was toen een luxe!

### ... of toch?

Met GDB kan de programmeur een stukje programmacode zien, die regel voor regel doorlopen, in of over functies springen, de waarde van een variabele printen en nog veel meer. Daar was geen X11 voor nodig; een terminal volstond. Of, zoals GNU het bedoelde: emacs. Immers, emacs kan gezien worden als de eerste 'Grafische Geïntegreerde Ontwikkel Omgeving', maar dan niet zo erg grafisch. Het was niet voor niets dat emacs en GDB op dezelfde tape(!) uitgeleverd werden.

Niet iedereen is fan van emacs en een grafische interface heeft zeker voordelen. Dus zijn er al heel lang grafische interfaces voor GDB. Want GDB kan zeker grafisch gebruikt worden; doordat er de optie is tot een (grafisch) front-end. Zo was 10 jaar geleden 'xxgdb' heel populair: de brains van GDB, de looks van X11-Athena. Heel modern, destijds. Ook was er 'gdbtk'. Inderdaad, de gdb-motor met knopjes in Tcl/Tk

Hier en daar zijn nog wel screenshots te vinden van die oudere uitbreidingen; grafisch zijn ze geheel verouderd. De motor, GDB, is niet verouderd; maar steeds verder ontwikkeld. Want wat 5 jaar geleden mooi was, is dat nu niet meer. Maar zonder GUI verouderd het veel minder snel; GDB is nu beter dan ooit!

### Modern Design

Met de komst van Linux kwam Unix op de desktop. En geheel in de traditie van Unix, in meerdere smaken. Dus is er kdbg; gdb met een moderne KDE interface. Maar er zijn ook andere, moderne '3D-' interfaces. Populair is 'DDD' de DataDisplayDebugger. Ook 'Insight' – de opvolger van gdbtk – oogt goed. Beide zijn een soort van "Visual debugger". De kracht van GDB en je kunt direct aan de slag. Het zijn prima, moderne, debuggers.

### Of toch integreren?

De kracht van de 'MS Visual' omgeving is de intergratie van tools; vooral met microsoft tools. GDB zal wel lastig zijn. Maar sinds enkele jaren is er ook 'eclipse'; de open, grafische, geïntegreerde ontwikkelomgeving, voor zowel Unix als Windows; voor Java en webapplicaties. En voor (embedded) C (!) én met ondersteuning voor GDB. Heel modern!

De laatste GUI voor gdb is GDB-UI; vanaf emacs-21.4 beschikbaar. En omdat ook emacs, net als vi(m) veel grafischer is geworden, is debuggen nu echt grafisch. En geheel geïntegreerd in de ontwikkelomgeving en dat in mijn oude, vertrouwde 'editor'. Programmeren doe ik niet zo veel meer; maar als ik een bug moet verwijderen kan ik tenminste zeggen dan mijn debugger beter is!