toolview

Albert Mietus'

toolview

Elke ingenieur gebruikt tools en iedereen heeft zijn voorkeur. Vaak gebaseerd op dat wat er toevallig als eerste beschikbaar was. Zo ook de ICT-ingenieur. Ook die gebruikt tools, software-tools wel te verstaan. En daarvan zijn er oneindig veel, waardoor ook gelijk een probleem ontstaat. Vaak wordt een *handige tool* niet gebruikt, gewoon omdat niet bekend is dat er een tool voor is. Of omdat je soms eindeloos kan zoeken naar dat ene onbekende tool. Daarom deze rubriek; een plek om allerlei tools eens voor te stellen. Zodat **jij** ze eens kunt gaan gebruiken, omdat je ze kent. Zodra je ze nodig het, kun je je er verder in verdiepen. Ken jij ook zo'n handig tool, dat veel te weinig gebruikt wordt? Schrijf dan eens een **ToolView**.

Mail me voor meer info.

ALbert.Mietus@PTS.nl

door & more

SCO

Momenteel is SCO veel in het nieuws door hun claim tegen Linux; ze lijken anti-opensource. Toch is SCO niet altijd de **bad guy** *geweest. Zij hebben cscope ontwikkeld en het vrijgegeven als opensource; met een BSD-achtige licentie. Zodat iedereen het tool mag gebruiken en mag aanpassen, zodat het voor veel systemen beschikbaar is. Kijk op http://cscope.sourceforge.net voor meer info.*

cscope

Het doorgronden van grote hoeveelheden C-code is vaak lastig. Het zoeken naar functie-aanroepen, zeker als dat in tientallen files gebeurd, kan dan veel tijd kosten. Unix-programmeurs kunnen een heel eind komen met grep en dergelijke. Maar ook dat is te langzaam bij honderden directories of duizenden files. Voortdurend find'en en grep'en kost dan te veel tijd. Een tool als cscope kan die taak automatiseren. Het maakt zoeken naar C-symbolen, evenals het vinden van functie definities en aanroepen, gemakkelijk.

klein maar fijn

Het tool **cscope** is klein (op mijn FreeBSD systeem maar zo'n 122Kbyte) en *ouderwets*; er is *geen* grafische user interface. Maar juist daardoor is het erg prettig werken met **cscope**. Als je **cscope** voor het eerst aanroept, zal het razendsnel alle dot-c en dot-h files in de huidige directory doorzoeken.
Het resultaat wordt opgeslagen in de file csope.out. Je kun ook zelf opgeven welke files doorzocht moeten worden; wat handig is als er meerdere directories gebruikt worden. Ook kan **cscope** lex- en yacc-files doorzoeken; en ook C++ wordt (grotendeels) ondersteund. Zodra er de *cscope.out* database eenmaal gemaakt is, zal **cscope** die up-to-date houden, zodat veranderingen in de code 'direct' bekend zijn. Bij elke aanroep van **cscope** wordt alleen die code, die veranderd is, opnieuw gescand, waardoor het gebruik flitsend snel is.

commandline met vi

Je kunt **cscope** vanaf de commandline aanroepen. Je ziet dan een eenvoudig menu, met een 9-tal (zoek)opties. De muis is niet nodig. Met de pijltjes loop je naar je zoekoptie van je keuze; je tikt een functienaam in en de entertoets geeft een overzicht. Nogmaals de pijltjes en enter en je bent in vi, bij de gewenste functie! Eenvoudiger kan het niet. Als je vi verlaat, kom je weer terug in het menu en kun je naar een andere plek springen, of opnieuw zoeken. Behalve naar functies, kun je zoeken naar globale variabelen of willekeurige strings, of zelfs zoeken naar reguliere expressies.

emacs' xcscope.el

Ook vanuit (X)Emacs kun je **cscope** gebruiken. Met de standaard uitbreiding *xcscope.el* is er een volledige interface naar **cscope**. Je kunt **cscope** gebruiken met *'M-x'* commando's, maar ook via keybindings en zelfs met een grafisch menu. Dat is het mooie van een tool als **cscope**. Het concentreert zich op het doorzoeken van C-code en dat doet het goed. Het dwingt geen manier van werken af en daardoor is het altijd te gebruiken.

tenslotte

Als je ooit een stuk C-code in beheer moet nemen en het inleren kost je meer dan 3 à 4 uur werk, probeer dan **cscope** eens. Het kost misschien een halve dag om te installeren en te begrijpen, maar daarna gaat het navigeren veel sneller. Die tijd win je zeker terug. En als je cscope eenmaal kent, dan is het zelfs voor heel kleine projecten nuttig.