EWD309

Het idee is zo simpel. Je hebt kleine programma's en grotere programma's. Plaats daarom een configuratie van wat kleine machines en een grotere, dan kunnen de kleinere programma's op de kleine en de groter programma's op de grotere machine. Simple comme Bonjour!

Bij nadere beschouwing blijkt het echter een drogreden, want het is bij rekenmachines *niet* waar, dat vele kleintjes een grote maken. Het zou het leven voor gebruikers een stuk ingewikkelder maken, zelfs als de machines compatibel zouden zijn (wat in het onderhavige voorstel niet eens het geval is, maar daarover later).

Een gewetensvol programmeur houdt rekening met de beperkingen van de machine, die zijn programma moet uitvoeren. (De gemaakte opmerkinge, dat lang niet alle programmeurs in dit opzicht altijd zo gewetensvol zijn is irrelevant: je intensiefste gebruikers worden het heus wel en we hebben het over hen, niet over de beginnertjes die ook eens een broddellapje mogen breien.) Het voorstel zou om te beginnen impliceren, dat de gebruiker de macroscopische karakteristieken van twee verschillende machines moet kennen. Hinderlijk. Nog hinderlijker is, dat de gebruiker moet kiezen, welke machine het programa moet uitvoeren. Vaak moet die keuze gemaakt worden voordat het programma gemaakt is omdat de keuze de structuur van het te maken programma zal beinvloeden. Dit betekent dat een programmeur moet uitmaken, wanneer een programma "klein" en wanneer "groot" is. Het werd voorgesteld of rekentijd hiervoor de maatstaf was. Dat is al niet aantrekkelijk, want afhankelijk van invoergegevens kan hetzelfde programma dan soms klein en soms groot zijn. Een ander aspect is benodigde werkruimte, maar ook dit kan voor het zelfde programma van run tot run verschillen. Tenslotte is er nog zoiets als lengte van de programmatekst. Gebruikers zouden zich bv. misschien wel moeten realiseren hoe lang de teksten van bibliotheekprocedures zijn! Maar het allerhinderlijkste is, dat deze verschillende maatstaven best tegen elkaar in kunnen wijzen, wat de gebruiker zou confronteren met het soort conflicten, die wij hem nu juist zo graag wilden besparen.

Totzover opmerkingen, die (enigszins) betrekking hebben op het programmabestand zoals zich dat tot op heden heeft ontwikkeld. Bij de keuze van een volgende machine hebben wij echter meer verantwoordelijkheden: de machine moet de wijze van werken, waar wij naar streven, bevorderen. Willen meer geavanceerde of in enigerlei opzicht meer gestroomlijnde machinetoepassingen van de grond komen, dan kan dit niet door ieder programma iedere keer weer van begin tot eind opnieuw uit te drukken op hetzelfde standaardniveau van een vaste programmeertaal. Middelen hiertoe zijn de opbouw van grote procedurebibliotheken en (special purpose) language processors. Het gebruik hiervan heeft ongetwijfeld tot gevolg dat meer programma's —in elk geval qua tekstlengte— groot worden. Dit is een gewenste ontwikkeling; door de voorgestelde configuratie van "vele kleintjes die niet een grote maken" zou hij echter tegengewerkt worden. (Tot overmaat van ramp kunnen we hierbij nog aantekenen, dat alle machines uit de configuratie als voorgesteld, met linkage editing werken, waardoor programmalengte een extra penalty veroorzaakt,)

Tenslotte zijn de machines niet compatibel, zo incompatibel, dat zelfs de ALGOL-systemen *moeten* verschillen. De hoop, dat deze verschillen door software realistisch gemaskeerd zouden kunnen worden, moet als illusoir gekenschetst worden. Zonder dit incompatibiliteitsdefect is de voorgestelde multi-machine-configuratie al hoogst onaantrekkelijk, met dit defect volstrekt onacceptabel.

15 april 1971 prof.dr.Edsger W.Dijkstra

---

transcribed by Hans Terlouw
revised Sat, 9 Jan 2010