

Notulen van een vergadering in het Dr NEHER-lab.
van de PTT te LEIDSCHENDAM op donderdag 16 juli '64
van de "kleine" EL X8 - gebruikers,
onderwerp: MICROCOO en ALGOLmachine

Aanwezigen: H.H. Kruseman Aretz , Nederkoorn , Zonneveld (Math. Centr.)
van der Poel (PTT) , van der Sluis , van der Meulen
(ERC Utrecht) , Zwanenburg (Electrologica DEN HAAG)

Tot deze vergadering werd besloten op de vergadering van de Z8-commissie d.d. 9 juli '64 te Amsterdam. Bij die gelegenheid bleken enige verschillen van inzicht te bestaan tussen de vertegenwoordigers van het MC enerzijds en die van de PTT en het ERC-Utrecht anderzijds over het ALGOLsysteem zoals in het MC wordt ontwikkeld.

Een van de resistance was de opvatting van de vertegenwoordigers van het MC, dat de kleine ELX8-installatie met ALGOL solo bedreven zou worden. De vertegenwoordigers van PTT en ERC waren van mening dat dit eenodeloze beperking betekende.

De heer Kruseman Aretz gaf een gedetailleerde uiteenzetting van de ALGOLmachine die het MC wil realiseren. Na lunch en rondgang door het PTT-lab. werd dit ontwerp bekeken in het licht van een bescheiden vorm van multi-bedrijf: er worden geen programma-delen ooit verplaatst, alleen buffers kunnen verhuizen en onder omstandigheden de contrastapel.

Voor kleine ALGOLprogramma's en installaties met meer dan 16K kernen kan het zin hebben te trachten verschillende programma's naast elkaar te draaien, i.h.b. een transput-intensief naast een reken-intensief (niet-noodzakelijk ALGOL-) programma.

De standpunten bleken minder ver uiteen te liggen en het MC-ontwerp bleek meer mogelijkheden te bevatten, dan zich d.d. 9 juli te Amsterdam liet aanzien.

Besloten werd, dat PTT en ERC hun inzichten omtrent de MICROCOO op schrift zouden stellen. Dit is tot op heden niet mogelijk geweest ten gevolge van voortdurende afwezigheid van Prof. van der Poel.

Bijgaand een schets van het ERC, mogelijk een basis voor verdere discussie.

vdM

14. OVER DE BETREKKINGEN
TUSSEN EEN COORDINATORSYSTEEM
EN EEN FAMILIE VAN ALGOLMACHINES
IN EEN KLEINE EL X8 INSTALLATIE

[MICROCOO]

Hoewel wij de opinie onderschrijven, dat op een minimale ELX8-instal-
latie (16K, TP, BL, BP) weinig ruimte is voor multiprogrammering, geloven
wij dat de mogelijkheden reeds bij een geringe uitbreiding (een tweede
BP, BL of TP; een tweede modul van 16K; een regeldrukker) aanzienlijk
groter kunnen worden en dat zelfs op een minimale installatie het naast
elkaar werken van meerdere programma's (met name een transput-intensief-
naast een reken-intensief "achtergrondprogramma") onder omstandigheden
z'n nut kan hebben. Derhalve zouden we het betreuren wanneer een zo vi-
taal project als de ALGOLmachine zich modeloos zou beperken tot solo-
bedrijf.

Onderstaand verhaal bedoelt een schets te zijn van een bedrijfssysteem
voor kleine ELX8-installaties (geen achtergrondgeheugens!) waarin het
principe van de ALGOLmachine zoals geschetst in het memorandum X8A-60 ev.
onaangetast blijft en daarmee vrijwel identiek is zodra één ALGOLmachine
de installatie solo gebruikt.

Uitgangspunt is het volgende:

Een programma (door welke vertaler of assembler dan ook in het geheugen
opgebouwd) is een intern verlopend proces dat z'n transput delegeert
aan routines die essentieel niet tot dat programma behoren. Een hoe-
veelheid in een of andere vorm gepakt bandbeeld wordt door aanroep van
zo'n routine overgedragen (output) resp. ontvangen (input) en na terug-
keer uit die routine is deze hoeveelheid bandbeeld verdwenen resp. aan-
wezig; aan de IOroutine worden twee parameters meegegeven (bijv. A en S):
plaats en hoeveelheid te transputeren bandbeeld. Conversie en deconversie
is een affaire van het programma.

De IOroutines denken we ons gesitueerd in een coördinator die het tran-
sito van bandbeeld verzorgt en de betrekkingen met CHARON onderhoudt. De
coördinator regelt bovendien het verkeer tussen de in het geheugen aan-
wezige programma's: bepaalt welk programma wanneer kan werken op grond van
nader te bepalen prioriteitsregels. De ingreepprogramma's denken we ons
gemakshalve eveneens in de coördinator gesitueerd.

Het totale geheugen denken we ons gesegmenteerd in PAGINA's waarvan de
omvang van later zorg is. We onderscheiden twee soorten pagina's:

- 1) COORDINATOR-PAGINA: door de coördinator in beslag genomen voor een
(voortaan Cpagina) | IOSilo ("buffer" zo u wil),
- 2) PROGRAMMA - PAGINA: door een programma in beslag genomen voor zich-
(voortaan Ppagina) | zelf, een stapel of contrastapel of wat dan ook.

De Ppagina's zijn onaantastbaar voor de coördinator, de Cpagina's
zijn dit niet bij voorbaat voor een programma:

Een programma krijgt een consecutief geheugen-TRAJect ("TRAJ"),
dat voldoende pagina's bevat voor het programma-zelf + eventuele stapel(s).
Het TRAJ van een programma is zijn "maximale speeltuin": zijn geheugen-
behoefte; wanneer de installatie solo wordt bedreven kan het TRAJ het to-
taal nog beschikbare geheugen zijn. Iedere pagina in zijn TRAJ die een
programma (bijv. voor een van zijn stapels) in beslag wil nemen, wordt via
de coördinator aangevraagd en altijd verkregen: wanneer het toevallig een
Cpagina is, heeft de coördinator deze maar te verhuizen. Iedere pagina die
een programma weer verlaat, wordt aan de coördinator teruggegeven, en kan
eventueel weer een Cpagina worden.

De coördinator denken we ons op een vaste plaats zover mogelijk voor-
in het geheugen. De programmaTRAJ's liggen her en der verspreid, ze zijn
niet relocateerbaar.

Omdat de betrekkingen tussen een coördinatorsysteem en eventuele non-ALGOLprogramma's de verrichtingen van een (van de familie van) ALGOLmachine('s) op geen enkele wijze mogen storen, kunnen ze hier verder buiten beschouwing blijven.

We bepalen ons nu tot een schets voor een "bilateraal verdrag" tussen twee partijen:

- 1) Het COördinatorSYsteem (voortaan "COSY"),
- 2) De Familie van Algol-Machines (voortaan "FAM").

FAM:

(Overeenkomstig (afgezien van mogelijke multipliciteit) het memorandum X8A-60 ev. van het MC, denken we ons FAM als volgt gestructureerd (het gaat in het onderstaande om enkele afkortingsdefinities):

- I) Een MONitor + Complex ("MONC"), ~~mm~~ die direct tegen COSY aan voorin het geheugen staat, eveneens dus op een vaste plaats.
- II) Een of meer ALGOL-TRAJ's, als volgt ingedeeld:
 - 1) een Object-Programma ("OP") waaraan we ons gemakshalve maar even de eventuele extra-bibliotheekprogramma's die dit objectprogramma nodig heeft, toegevoegd denken; waaraan aansluitend
 - 2) de Stapel ("S") begint;
 - 3) aan een Plafond ("P") hangt de Contra-Stapel ("CS"), de plaats van P in het TRAJ is geheel ter bepaling van MONC;
 - 4) het gedeelte van P tot Einde-Traject ("PET").

Voor COSY is elk ALGOL-TRAJ pas een "programma", wanneer MONC dit TRAJ als zodanig heeft aangemeld. COSY beschouwt MONC echter als het hoofd van de FAM en meldt dan ook steeds aan MONC welk tot "programma" verklaard TRAJ mag (verder-)werken.

Een bijzonder OP is de AlgolVertaler ("AV"): deze heeft een vaste plaats zover mogelijk achterin het geheugen. De functie van de AV is in 1 of meer slagen een TRAJ te vullen met een OP. Wanneer we deze slagen noemen resp. AV1, AV2, ---, AVn, kunnen we AV1 t/m AVn beschouwen als virtuele transput-machines:

Een via een BL gelezen ALGOLtekst wordt door AV1 getransformeerd tot input voor AV2 etc. Het transitio: input AVk+1 := output AVk kan eventueel via COSY worden gespeeld. AVn deponeert z'n output uiteindelijk in een TRAJ. We nemen hierbij aan, dat de Stapel-ContraStapel ("SCS") van elke AVk in ~~MONC~~ dit TRAJ ligt. (Een en ander vrij naar een suggestie van de heer Kruseman Aretz).

We beschouwen AV voortaan als een tamelijk ~~gewoon~~ "gewoon" TRAJ, met dus twee bijzonderheden waarvan de details COSY bij voorbaat bekend zullen moeten zijn:

- 1) AV is een cascade-transputmachine en heeft een compleet TRAJ voor z'n output.
- 2) AV staat op een vaste plaats, is mogelijk in ruste en mag dan eventueel worden overschreven.

Aldus is $FAM = MONC + TRAJ^1 + TRAJ^2 + \dots + TRAJ^n + AV$

MONC is de instantie (het hoofd van de FAM) die met COSY communiceert:

- A) Bij elk TRAJ drie parameters voor COSY toegankelijk maakt:
 - 1) Sk = eerste pagina boven de stapel,
 - 2) Ck = laatste pagina onder de contra-stapel,
 - 3) Pk = eerste pagina van PET.
- B) Iedere pagina die een stapel of contrastapel wil annexeren, via COSY opeist:
 - Sk ophogen resp. Ck aflagen: indien $Sk \leq Ck$ zal COSY steeds aan de eis voldoen,
 - Pk ophogen: indien $PET \neq 0$ zal COSY steeds aan de eis voldoen.
- C) Iedere pagina die door een stapel of contrastapel wordt verlaten aan COSY teruggeeft.
- D) Al of niet op verzoek van COSY een contra-stapel reorganiseert ("aaneenplemt"), of een plafond verhoogt.
- E) Een uitgediende TRAJ aan COSY teruggeeft.
- F) Voor COSY beschikbaar heeft een volledige lijst van alle TRAJ's onder zijn beheer, met een opgave welke van deze TRAJ's een "programma" zijn en welke nog niet.

We onderscheiden aldus in elk TRAJ de volgende geheugensegmenten:

- | | | |
|--------------|--------------------|--|
| I) segment | OPk + Sk | onaantastbaar voor COSY; |
| II) segment | SkCk | in principe beschikbaar voor MONC, niet onaantastbaar voor COSY; |
| III) segment | CkPk(contrastapel) | onaantastbaar voor COSY; |
| IV) segment | PETk | in principe beschikbaar voor MONC, niet onaantastbaar voor COSY. |

Ten overvloede:

De transput vanuit een TRAJ wordt gespeeld via aanroepen van Ioroutines in COSY. Plaats en hoeveelheid van de transput wordt gespecificeerd in twee parameters. Strikt genomen is nog een derde parameter nodig: het nummer van het TRAJ (k) waarbij deze transput behoort; we kunnen ons voorstellen dat k in de voorste bits van A of S wordt meegegeven. Ieder programma kan dus rustig zijn output in stapel of contrastapel deconverteren en verpakken (na aanroep van de Ioroutine is hij z'n rommel kwijt) en zijn input in stapel of contrastapel ontvangen (na aanroep van de Ioroutine heeft hij zijn materiaal binnen) ter ontpakking en conversie. Niets belet MONC er "eigen buffers" op na te gaan houden, maar dat lijkt ons nu een zinloze doublure (vandaar dat we er bij de parameters sub A geen rekening mee hebben gehouden: dergelijke buffers kunnen misschien in de contrastapel staan).

Samenvattend:

MONC beheert uitsluitend 1 of meer ALGOLmachines: vraagt pagina's aan, geeft pagina's vrij, reorganiseert eventueel een contrastapel al of niet met plafondverhoging.

bij terugkeer uit

COSY bemiddelt tussen FAM en CHARON:

- a) Beheert het VRIJE GEHEUGEN, dat zijn alle pagina's in de installatie die noch P- noch Cpagina zijn.
- b) Output van FAM is Input voor CHARON en vice versa.
COSY verzorgt het transito-verkeer:
De door een Routine of concrete Inputmachine aangeboden transput slaat hij op in buffers, naar rato Cpagina's annexerend; de door een Routine of concrete Outputmachine verlangde transput loost hij, naar rato Cpagina's vrijgevend.
- c) COSY neemt een Cpagina in eerste instantie uit het vrije geheugen; wanneer er geen vrij geheugen is, kan hij beschikken over de segmenten PETk;
wanneer ook hier geen ruimte is, kan hij beschikken over de segmenten SkCk;
wanneer ook hier geen ruimte is, hoort hij een teller op;
wanneer deze teller een grenswaarde overschrijdt, is hij gerechtigd AV tot vrij geheugen te verklaren, mits deze in ruste is;
als alles vol zit, gaat hij wachten of ingrepen uitkomst bieden.
Bij elke pagina-keuze in vrij geheugen of een PETk prevaleert het hoogste paginanummer; voor pagina-keuze in een segment SkCk is misschien een andere strategie beter.
- d) Wanneer MONC een Sk wil ophogen, en ~~MM~~ dit blijkt een Cpagina, zal COSY deze met de hoogste prioriteit verhuizen hierbij handelend als onder c).
Wanneer MONC een Ck wil aflagen, en ~~MM~~ dit blijkt een Cpagina, ^{kan} ~~MM~~ COSY het verzoek retourneren met een verzoek tot reorganisatie van de aangrenzende contrastapel (hij zal dit doen wanneer het aantal vrije pagina's beneden een bepaalde grenswaarde ligt);
wanneer dit verzoek wordt geretourneerd met een herhaling van het verzoek tot aflaging van Ck, handelt COSY als bij aanvraag tot ophoging van een Sk.
Wanneer bij aanvraag tot verhoging van Sk resp. aflaging van Ck blijkt dat $Sk > Ck$, verzoekt COSY aan MONC om Pk te verhogen.
Wanneer $Sk > Ck$ en $PETk = 0$, zal COSY proberen TRAJk uit te breiden; wanneer dit niet mogelijk is (omdat er een ander TRAJ aan grenst) zal COSY de besturing aan een ander programma overdragen;
wanneer (nog steeds in geval een TRAJk moet worden uitgebreid) $Pk =$ fysisch einde van het geheugen, zal COSY dit TRAJ tot vrij geheugen verklaren (eufemisme voor programmamoord).
- e) Wanneer de operateur een ALGOLtekst aankondigt, zal COSY in eerste instantie na~~ga~~an of AV nog aanwezig is en indien dit niet het geval is de AVpagina's te ontruimen overeenkomstig c).
Wanneer de ontruiming is gelukt, wordt AV ingelezen.
Wanneer AV in het geheugen staat, vraagt COSY de PARAMETERS van de aangeboden ALGOLtekst op:
 - 1) zijn GEHEUGENBEHOEFTE (= aantal pagina's van zijn TRAJ),
 - 2) zijn TRANSPUTBEHOEFTE.
 Wanneer aan 1) kan worden voldaan (na eventuele verhuizing van Cpagina's volgens c), kan de tekst worden vertaald: ~~XXXXXXXXXX~~ het ge-eist TRAJ wordt aan MONC overgedragen.

- OPM1: Wanneer $FAM = MONC + TRAJ + AV$ en $TRAJ =$ totaal beschikbaar geheugen, is de hier geschetste organisatie vrijwel identiek met die van X8A-60 ev.
- OPM2: Effectief is het hier geschetste systeem in geval van multi-bedrijf: een verkleining van het per programma beschikbaar geheugen.
- OPM3: We vragen ons af, of $PET = 0$ ($P =$ einde TRAJ) niet een vereenvoudiging zou zijn waarbij niets wordt verloren
- OPM4: We vragen ons af, of het Complex maar niet beter meteen de meest gangbare bibliotheekprogramma's zou kunnen bevatten.