

BUIB: Een Fortran programma voor het overbrengen van Fortran programma's van Honeywell-Bull G.E. time sharing naar een IBM 360/65

Mededeling No.

000469

Centraal Laboratorium
TNOFortrom translates from Honeywell
to IBM

Ir. J. VAN DEN EIKHOFF

Centraal Laboratorium TNO

Samenvatting

In dit verslag wordt een computerprogramma besproken, dat dient om programma's welke foutloos draaien op de Honeywell-BULL time-sharing computer, geschikt te maken voor werking op een IBM 360/65. Gebruik van dit programma brengt de hoeveelheid „handwerk”, nodig voor het overbrengen terug tot een aanvaardbaar minimum.

Zonder nieuwe foutenbronnen en met relatief weinig werk kan van het ene systeem op het andere worden overgegaan.

Summary

In this report a computer programme is given which is able to „translate” programmes that work without error on the Honeywell-BULL G.E. time-sharing computer, in such a way that they can be used on an IBM 360/65-computer.

Use of this programme reduces to an acceptable minimum the amount of „handwork”, necessary for switching over. Without introducing new sources of error, and with relatively little work, it is now possible to switch from one system to the other.

Inleiding

Vaak heeft men — zoals ook in het Centraal Laboratorium TNO — de beschikking over twee computer-systemen, waarvan de gebruiksgebieden elkaar nauwelijks overlappen. Deze systemen zijn:

a) Een time-sharing verbinding per terminal

Door het conversationele karakter van deze verbinding — men krijgt vrijwel direkt antwoord indien men zijn programma draait — is dit een uitermate geschikte manier om programma's op redelijk korte termijn te ontwikkelen.

Het voornaamste nadeel is, dat bij toename van de in- en uitvoer de verbindingstijden dermate toenemen, dat het om persoonlijke en economische redenen onaantrekkelijk wordt. Bijkomende nadelen kunnen zijn: beperkte geheugenruimte, programmalengte en nauwkeurigheid.

b) Batchverwerking op een grote computer

Een voorbeeld hiervan is een IBM 360/65. Op deze wijze vervalt het nadeel van de lange verbindingstijden, terwijl ook geheugenruimte, maximale programmalengte en nauwkeurigheid groter zijn.

Nadelig is echter de lange tijd nodig voor het ontwikkelen van een programma. Dit komt door het loketsysteem: Men levert het pro-

gramma op ponskaarten in en kan later de uitvoer ophalen. Kleine vergissingen en ponsfouten zijn de oorzaak, dat het programma vrij vaak moet proefdraaien, terwijl een programma slechts zelden meer dan één maal per dag gedraaid kan worden.

Het verdient daarom aanbeveling om ook programma's, die uiteindelijk in de batch zullen werken, via de terminal te ontwikkelen. Tot voor kort ging echter het grootste deel van de winst weer verloren bij het overbrengen van de ene computer naar de andere. Dit is tijdrovend en arbeidsintensief. Het foutloze programma moest op formuleren overgeschreven worden, rekening houdend met de onvermijdelijke verschillen tussen de computers, waarna een ponsstypiste het geschrevene op ponskaarten zet. Deze werkwijze vormt een geenszins te verwaarlozen foutenbron, o.a. door overschrijf- en overtypfouten. Het programma BUIB vereenvoudigt deze overdrachtsprocedure aanzienlijk.

Ieder die de schoonheid van de Nederlandse taal ter harte gaat, zal bij het lezen van dit verhaal af en toe de wenkbrauwen fronsen. Dit is onvermijdelijk, aangezien het computerjargon een aantal scherp gedefinieerde termen bevat, die niet zonder verlies aan duidelijkheid door mooiere vervangen kunnen worden.

De verschillen in fortran

Het Time-Sharing FORTRAN van Honeywell-BULL G.E. wijkt op vele punten af van dat van de IBM 360/65. Teneinde het snel werken met de terminal te bevorderen zijn enkele faciliteiten ingebouwd, welke het starre formatted FORTRAN van de IBM 360/65 mist. Anderzijds beschikt de IBM 360/65 over faciliteiten, die bij time-sharing niet bestaan, doch daarvan wordt bij omzetting geen hinder ondervonden.

Het opvallendste verschil is de plaatsing van de FORTRAN-opdrachten (statements). Bij het FORTRAN van de IBM 360/65 (verder genoemd: „op de IBM”), begint elke statement op kolom 7 van de ponskaart, terwijl elke volgende statement op een nieuwe kaart begint. Bij het Time-Sharing FORTRAN van Honeywell BULL G.E. (in het vervolg „bij BULL”) is het programma ingedeeld in regels, elk voorzien van een regelnummer. Op elke regel mogen meerdere statements staan gescheiden door „;”.

Elke statement kan men een naam geven. Op de IBM bestaat deze naam uit een statementnummer (label), welke geplaatst wordt in de posities 1 t/m 5 van de ponskaart. Bij BULL worden labelnummers gewoon voor de statement geplaatst. Bovendien kunnen namen van letters gegeven worden, welke dan door „:” worden gevolgd. Wil men met BULL een programma overzetten dan moet men deze tweede faciliteit vermijden.

Een statement die te lang is voor één regel of één kaart kan worden gecontinueerd. De afspraak hiervoor is bij BULL, dat een statement doorloopt indien het eerste niet blanke karakter na het regelnummer van de volgende regel een „+” is, terwijl een continuatie op de IBM gekenmerkt wordt door een teken ongelijk aan een 0 of een spatie in positie 6.

Een conventie, die de leesbaarheid van een programma zeer kan verhogen is de commentaarconventie. Deze opent de mogelijkheid tot het plaatsen van verklarende tekst in het programma. Op de IBM wordt commentaar aangegeven door het plaatsen van de letter C op positie 1, terwijl bij BULL een regel als commentaar wordt opgevat, indien het eerste karakter na het regelnummer geen spatie is.

BULL kent een faciliteit die schrijfwerk bespaart bij de zgn. rekenkundige if-statement. Dit is een driesprong in het programma, waarbij de sprong afhangt van de waarde van een rekenkundige uitdrukking. Op de IBM moet de if-statement worden gevolgd door drie labels. Indien bij BULL de laatste of de laatste twee labels gelijk zijn aan het label van de volgende statement, mogen zij worden weggelaten.

Er is een groot verschil tussen de in- en uitvoerprocedures op de twee computers. Bij BULL zijn er drie methoden:

1. in- en uitvoer via de terminal aangegeven met input resp. print,
2. in- en uitvoer naar vast gereserveerde geheugendeeltes (zgn. files), aangegeven door read resp. write, vergezeld van een file-nummer,
3. in- en uitvoer naar een tijdelijk geheugendeelte, aangegeven door read resp. write zonder file-nummer.

Bovendien is het mogelijk, indien aan in- en uitvoer weinig eisen worden gesteld zgn. zonder format in te lezen, d.w.z. men hoeft de in- en uitvoer niet precies te precisieren.

Op de IBM daarentegen gebeurt alle in- en uitvoer naar zgn. datasets. In aparte datasetdefinities wordt aangegeven van welk medium dit gebeurt. Standaard is hierbij afgesproken, dat aan dataset 5 de ponskaartlezer is aangesloten, terwijl dataset 6 wordt aangesloten aan de regeldrukker. Bovendien is in- en uitvoer zonder format niet zonder meer mogelijk. Het gebruik van expressies in de uitvoerstatement is op de IBM, in tegenstelling met BULL, niet toegestaan. Een volgend verschil ligt in de verwerking van de uitvoer door de regeldrukker van de IBM. Deze gebruikt het eerste teken van elke regel als controlcharacter, voor pagina-indeling en dergelijke. Indien men zorgt, dat van de uitvoer gecreëerd op de BULL terminal elke regel met een spatie begint, dan wordt hier verder geen hinder van ondervonden. Een laatste punt waar bij in- en uitvoer op moet worden gelet is, dat real variabelen op de IBM ook met een real format uitgevoerd moeten worden en integer variabelen met integerformat.

Ook bij de DO-statement is het BULL-FORTRAN iets flexibeler met real's en integers, want op de IBM moeten lusvariabele, begin- en eindwaarde en stapgrootte positieve nonsubscripted integer grootheden zijn, terwijl bij BULL hiervoor ook real's gebruikt mogen worden.

Er wordt op sommige plaatsen van een programma gebruik gemaakt van statements die niets doen. Op de IBM is dit de zgn. CONTINUE-statement, aangegeven met het woord continue beginnend op positie 7. Bij BULL is dit ook mogelijk met een DUMMY statement d.w.z. een lege statement.

Bij het lezen in een lijst gegevens is een test nodig, die kijkt of de lijst uitgeput is. Deze test is voor beide computers zo verschillend, dat de programmeur dit zelf zal moeten aanpassen.

De laatste statement van een DO-statement mag niet elke willekeurige statement zijn. Bij BULL zijn b.v. de GOTO en RETURN-statement verboden, terwijl op de IBM bovendien de IF-, STOP en PAUZE-statement niet toegestaan zijn. Een verschil toont ook de zgn. „COMPUTED GOTO”, welke afhankelijk van de waarde van een variabele een sprong naar verschillende gedeelten van het programma veroorzaakt.

Zie hiertoe het volgende overzicht:

IBM 360/65

Honeywell-BULL

a) de variabele moet zijn:

non-subscripted integer integer of *real*
(*real* wordt afgekapt)

b) ligt de variabele buiten de toegestane grenzen, dan wordt:
verder gegaan met het programma
statement volgend op gestopt
de computed goto

c) variabele *moet* worden variabele *mag* worden
voorafgegaan door een voorafgegaan door een
komma komma

Het laatste verschil slaat op het laatste deel van het programma de „END-Statement”. Op de IBM moet deze aanwezig zijn, terwijl hij niet gelabeld mag zijn. Bij BULL hoeft hij niet aanwezig te zijn en hij mag indien aanwezig wel gelabeld zijn.

In het volgende hoofdstuk bespreken wij het werken met BUIB, terwijl tevens een samenvatting wordt gegeven van de verschillen welke door BUIB worden aangepast, welke verschillen moeten worden vermeden en welke verschillen zelf moeten worden aangepast.

Bespreking van het programma Buib

Teneinde de handarbeid bij het overbrengen van time-sharing naar de IBM 360/65 tot een minimum te beperken, dient men zich enige beperkingen op te leggen. Om te weten welke beperkingen dit zijn moet men weten wat BUIB wel en niet aanpast.

Hoe ziet het programma er uit nadat het door BUIB is verwerkt?

V1: Elke statement begint in kolom 7 van een ponskaart.

V2: Commentaar begint met een „C” in kolom 1.

Tabel 1.

Soort in- of uitvoer,		time-sharing		uitvoer BUIB voor BM 360/65	
met format	terminal	INPUT	2,x	READ	(5,2)x
		PRINT	3,x	WRITE	(6,3)x
	tijdelijk	READ	2,x	READ	(7,2)x
		WRITE	3,x	WRITE	(7,3)x
	file	READ	(1,2)x	READ	(1,2)x
		WRITE	(2,3)x	WRITE	(2,3)x
zonder format	terminal	INPUT,	x	READ	(5,)x
		PRINT,	x	WRITE	(6,)x
	tijdelijk	READ,	x	READ	(7,)x
		WRITE,	x	WRITE	(7,)x
	file	READ	(1)x	READ	(1,)x
		WRITE	(2)x	WRITE	(2,)x

Aanpassing door BUIB van de in- en uitvoeropdrachten

V3: Statementnummers beginnen in kolom 1.

V4: Continuatie van een statement begint met een „+” in kolom 6.

V5: Meerdere statements op één regel zijn op verschillende ponskaarten gezet.

V6: De rekenkundige if-statement is eventueel aangevuld tot drie statement-nummers.

V7: Dummy statements zijn vervangen door continue statements.

V8: Een aantal verschillen in de in- en uitvoer zijn opgeheven, zie Tabel 1.

Aangezien in- en uitvoer zonder format op de IBM niet zonder meer mogelijk is, wordt dit gesignaleerd met de melding:

FORMAT TOEVOEGEN

in het protocol. Een tweede melding:

LABEL(S) VERANDEREN

wordt gegeven als een if-statement aangevuld is tot drie labelnummers, terwijl de volgende statement al gelabeld was.

Waar moet men bij het programmeren op letten, wil men bij het overzetten een minimale hoeveelheid handwerk overhouden?

M1: Gebruik in de uitvoer geen expressies en zorg dat de mode van de variabelen overeenkomt met het bijbehorende format.

M2: Gebruik voor lusvariabele, begin- en eindwaarde en increment van een DO-statement positief gehele non-subscripted integer variabelen.

M3: Neem als laatste statement van een DO-loop geen IF-, STOP of PAUZE-statement.
(GOTO en RETURN mogen toch al niet.)

M4: Gebruik een END-statement, die bovendien niet gelabeld is.

M5: Gebruik alleen labelnummers, geen namen.

M6: Vermijd in de uiteindelijke versie het gebruik van in- en uitvoer zonder format en zorg dat elke regel met een spatie begint.

M7: Gebruik bij de computed GOTO een non-subscripted integer variabele voorafgegaan door een komma.

Aan deze punten kan zonder noemenswaardig verlies aan flexibiliteit worden voldaan bij het schrijven van het programma voor de terminal.

Is het programma door BUIB overgezet, dan moet de programmeur zelf de volgende punten nog aanpassen:

A1: Vervang de endfile-test van BULL: IF(END-FILE) door de overeenkomstige test op de

IBM: END = „statementnummer” in de leesopdracht; b.v.: READ (1, 27, END = 38). Het is mogelijk, dat het programma hierdoor op kleine plaatsen moet worden gewijzigd.

A2: Voeg de nodige dataset definities toe.

A3: Plaats de data uit de tijdelijke file in dataset 7.

A4: Verzorg de meldingen FORMAT TOEVOEGEN en LABEL(S) VERANDEREN, welke in het protocol voorkomen.

A5: Spoor de plaatsen op, waar tegen de punten M1 t/m M7 is gezondigd en corrigeer ze.

Dat de hoeveelheid werk sterk afhangt van de wijze van schrijven zullen we met een eenvoudig voorbeeld toelichten. We kiezen daartoe de functie $f(x) = \sin(x)$ en bepalen het maximum van deze functie ($x = \pi/2$) door stapsgewijs aftasten van de functie (dus niet door differentiëren!).

VO1 is een BULL-FORTRAN programma, dat met het oog op overbrengen naar IBM slecht geschreven is.

Alle in- en uitvoer is zonder format en zal aangepast moeten worden. De grootheden in de DO-statement tot statement 3 zijn real en dienen ook later te worden aangepast, d.w.z. vrijwel het hele programma moet opnieuw worden gemaakt. Dit is vanzelfsprekend een extreem voorbeeld.

VO2, een ander BULL-FORTRAN programma, is wel goed. Het overbrengen met BUIB gaat vrijwel zonder moeite.

VO3, laat het resultaat van de vertaling van VO2 door BUIB zien.

Men ziet, dat in- en uitvoer-opdrachten aangepast zijn. Alleen de input-statement zonder format moet worden aangepast. Bovendien is de IF-statement aangevuld tot drie labels, terwijl de dummy-statement met labelnummer 3 omgezet is in een CONTINUE-statement.

Het is duidelijk, dat de hoeveelheid werk op deze wijze sterk is verminderd.

Belangstellenden voor dit programma kunnen contact opnemen met de auteur, p/a Centraal Laboratorium TNO, Schoemakerstraat 97, Postbus 217, Delft. Of telefonisch 01730-37000, toest. 2661.

```

0  DIMENSION DI(3)
10  PRINT, "X-VALUE SINE RESOLUTION"
20  READ, DI; 1 INPUT, O, B
30  DO 2 I = 1,3; RM = -1; D = DI(I)
40  DO 3 X = O, B, D
50  IF (SIN(X)-RM)3; XO = X; RM = SIN(X)
60  3; 2 PRINT, XO, RM, D; GOTO 1
70  $ DATA
80  .1, .01, .001

```

VO1 Een moeilijk te vertalen BULL-FORTRAN programma

```

0  DIMENSION DI(3)
10  PRINT 4
20  4 FORMAT (" X-VALUE SINE RESOLUTION")
30  READ 5, DI; 5 FORMAT (4X, F3.1, F4.2, F5.3)
40  1 INPUT, O, B
50  DO 2 I = 1,3; RM = -1; D = DI(I);
55  JE = (B-O)/D + 1
60  DO 3 J = 1, JE; X = O + (J-1)*D
70  IF(SIN(X)-RM)3; XO = X; RM = SIN(X)
80  3; 2 PRINT 6, XO, RM, D
90  6 FORMAT (3 F7.3); GOTO 1
100 END
110 $ DATA
120 .1, .01, .001

```

VO2 Dit BULL-FORTRAN programma doet hetzelfde als VO1, maar is gemakkelijk met BUIB te vertalen

```

          DIMENTION DI(3)
          WRITE (6, 4)
4  FORMAT (" X-VALUE SINE RESOLUTION" )
          READ (7, 5) DI
5  FORMAT (4X, F3.1, F4.2, F5.3)
1  READ (5,) O, B

          FORMAT TOEVOEGEN
          DO 2 I = 1,3
          RM = -1
          D = DI (I)
          JE = (B-O)/D + 1
          DO 3 J = 1, JE
          X = O + (J-1) * D
          IF (SIN(X)-RM)3, 9999, 9999
9999 XO = X
          RM = SIN(X)
3  CONTINUE
2  WRITE (6, 6) XO, RM, D
6  FORMAT (3 F7.3)
          GOTO 1
          END

```

VO3 Het resultaat van de vertaling van VO2 met BUIB: Een IBM-FORTRAN programma