

RA

**stichting
mathematisch
centrum**



REKENAFDELING

MR 126/71

SEPTEMBER

J.V.M. VAN DER GRINTEN en C. ZUIDEMA
EEN PROCEDUREBIBLIOTHEEK OP ACHTERGRONDGEHEUGEN
IN HET MC-MILLI-SYSTEEM VOOR DE EL X8

RA

2e boerhaavestraat 49 amsterdam

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

Inhoud

0. Literatuur
1. Inleiding
2. Globale beschrijving
3. Bibliotheekadministratie
4. Vrij-locateerbare code
5. Geheugenindeling
6. Diverse routines
7. Anonieme procedures
8. Codeprocedures
9. Catalogue
10. Gebruiksaanwijzing
11. Foutmeldingen
12. Verschilpunten t.o.v. MILLI
13. ELAN - tekst van de submonitorsectie

O. Literatuur

- [1] P. Naur (ed),
Revised Report on the Algorithmic Language ALGOL 60,
Regnecentralen, Kopenhagen 1962.
- [2] F.E.J. Kruseman Aretz en B.J. Mailloux,
The ELAN sourcetext of the MC ALGOL 60 system for the EL X8,
MR 84, Mathematisch Centrum, Amsterdam 1966.
- [3] F.E.J. Kruseman Aretz,
Het objectprogramma, gegenereerd door de X8 - ALGOL 60 -
vertaler van het M.C.,
MR 121, Mathematisch Centrum, Amsterdam 1971.
- [4] K.K. Koksma,
Een ALGOL 60 bibliotheek systeem voor de EL X8,
MR 117, Mathematisch Centrum, Amsterdam 1970.
- [5] F.E.J. Kruseman Aretz,
Bespreking MILLI-monitor (ongepubliceerde verslagen),
Mathematisch Centrum, Amsterdam 1970.
- [6] MILLILIB, ELAN-sourcetext, Assemblaat dd 01.04.71 (ongepubliceerd),
Mathematisch Centrum, Amsterdam 1971.
- [7] R. Brinkhuijsen,
Systemen op magneetband, NR 13,
Mathematisch Centrum, Amsterdam 1970.

1. Inleiding

In dit rapport wordt een uitbreiding van de MC-ALGOL 60-vertaler besproken die het mogelijk maakt om in ALGOL 60-programma's, zonder voorafgaande declaratie, procedures aan te roepen die zich in vertaalde vorm bevinden in een bibliotheek op achtergrondgeheugen (de trommel). Van de benodigde procedures wordt, voorafgaand aan de executie van het objectprogramma, de objectcode opgehaald en aan het objectprogramma toegevoegd.

Ook het opbouwen van deze trommelbibliotheek en de bijbehorende administratie wordt beschreven.

Het hier beschreven systeem, verder MILLILIB te noemen, is in feite een uitbreiding van het MILLI-systeem van de hand van prof. dr. F.E.J. Kruseman Aretz, dat op 01.10.70, zonder trommelbibliotheek, op het MC reeds in gebruik was genomen. MILLILIB is op 01.04.71 op het MC in gebruik genomen.

Er is bij dit project gebruik gemaakt van het werk van K.K. Koksma, die op het MC het MKL-systeem realiseerde; d.i. het MICRO - systeem, uitgebreid met een trommelbibliotheek. Met name zijn uit MKL overgenomen opbouw en inrichting van de bibliotheekadministratie en de analyse van ALGOL 60-procedures, bestemd om in de trommelbibliotheek opgenomen te worden. Voor een gedetailleerde beschrijving van crosstable, infotable, trafotable en routines als CRF, INSPECTDECL en TRUNCATE wordt dan ook verwezen naar [4].

In de voorliggende versie van MILLILIB is niet gepoogd om in de trommelbibliotheek ruimtewinst te behalen b.v. door de procedures niet in de vorm van instructies maar als een rij macro's op de trommel te zetten.

De tekst van dit rapport is van de hand van J.V.M. van der Grinten. In de beschrijving wordt enige voorkennis verondersteld van het MILLI-systeem, vooral van de daarin gebruikte ALGOL 60-vertaler die nauwelijks afwijkt van de beschrijving in [2]. Een beschrijving van de monitor in het MILLI-systeem vindt men in [5].

2. Globale beschrijving

2.1. Het aanroepen van procedures uit een trommelbibliotheek in ALGOL 60-programma's.

2.1.1. De ALGOL 60-vertaler heeft om het gebruik van procedures uit een trommelbibliotheek, verder trommelprocedures te noemen, aan te kunnen enkele extra taken:

- Tijdens Prescan1 worden alle niet-gedeclareerde namen uit de ALGOL 60-tekst opgezocht (zie [6] p. 306: de routine ASK LIBR). Voor alle namen die daarbij worden gevonden in de catalogue (= bibliotheekcatalogus) wordt een aantekening in de bibliotheekadministratie gemaakt. Zo wordt een lijst opgebouwd van alle direct aangeroepen bibliotheekprocedures. Hieronder vallen zowel de trommelprocedures als ook de procedures die als deel van het systeem permanent in het kerngeheugen staan, de systeemprocedures.

- Aan het begin van de translationscan, voorafgaande aan de opbouw van het eigenlijke objectprogramma, wordt bepaald welke trommelprocedures er nog meer nodig zijn, doordat ze indirect aangeroepen worden (zie 13.18.: de routine CRF). Voor elke direct of indirect benodigde trommelprocedure wordt daarna een dubbelwoord, in het vervolg pseudolvar te noemen, in het objectprogramma gereserveerd en voorlopig gevuld met het rangnummer van de procedure (zie 13.18.: de routine PSEUDOLVAR). De vertaling van de aanroep van een trommelprocedure in de translationscan resulteert dan in een subroutinesprong inhoudelijk via het 1e of 2e woord van de bijbehorende pseudolvar.

2.1.2. Op monitorniveau wordt na gunstige afloop van de vertaling de rest van het werk gedaan (zie 13.15.: de routine SUBMONITOR):

- De benodigde procedures worden nu van de trommel opgehaald (zie 13.16.: de routine READ LIBRARY ROUTINE) en hun beginadressen worden in de bijbehorende pseudolvars ingevuld.
- Bovendien wordt de adressering binnen iedere afzonderlijke procedure in overeenstemming gebracht met de plaats in het kerngeheugen waar hij nu staat (zie 6.2. en 13.16.: de routine ADAPT ADR). Dit gebeurt met behulp van een, samen met de procedure van de trommel gehaalde, vrije-locateerbaarheidsadministratie (zie 3.), die na geraadpleegd te zijn wordt opgeruimd.

2.2. Het vullen van de procedurebibliotheek op de trommel.

2.2.1. De vertaler is geparametriseerd wat betreft de speciale bibliotheekhandelingen. Door de keuze van 1 van 3 lijstjes, base0, base1 en base2, met adressen van routines, en de toekenning van een waarde aan de variabele COMPMODE, wordt op monitorniveau, afhankelijk van de waarde van de variabele NORMAL, bepaald welke versie van de vertaler zal werken. De variabele BASE bevat steeds het adres van een van de genoemde lijstjes, en de variabele COMPMODE heeft de waarde 0, 1 of 2.

Als nu de monitor in de toestand is gebracht waarin hij de trommelbibliotheek wil vullen, doordat de variabele `NORMAL` de waarde `false` heeft gekregen, dan kan men ALGOL 60-procedures eraan toevoegen door ze op te nemen in een ALGOL 60-programma van een speciale vorm nl. bestaande uit de betreffende proceduredeclaraties, elk afgesloten door een puntkomma, en het geheel omsloten door `begin` en `end`. Binnen zo'n bibliotheekprogramma mogen de procedures zichzelf, elkaar, en natuurlijk ook alle reeds eerder opgenomen procedures aanroepen.

2.2.2. Eerst wordt dan (zie 13.1.: de routine `EXTEND LIBRARY`) de vertaler aangeroepen met `COMPMODE = 1` en `BASE = :base1`:

- gecontroleerd wordt of er inderdaad alleen proceduredeclaraties in het programma staan (zie 13.6.: diverse routines en 13.12.: de routine `INSPECT DECL`);
- aan elke procedure wordt een rangnummer toegekend; dit wordt samen met de procedurenaam in de catalogue opgenomen (zie 13.12.: de routine `INSPECT DECL`);
- er wordt geen objectprogramma gegenereerd (zie [6] p. 357 e.v.: de routines `PRODUCE` en `SUBSTT`), maar wel een volledige syntactische analyse uitgevoerd.

2.2.3. Daarna wordt (zie 13.2.: de routine `NEXT PROCEDURE`) voor iedere procedure uit het programma afzonderlijk, omvat door een, door de vertaler zelf toegevoegde, `begin` en `end` (zie 13.9.: de routine `TRUNCATE`), de vertaler aangeroepen met `COMPMODE = 2` en `BASE = :base2`:

- de procedure wordt nogmaals syntactisch geanalyseerd en de opbouw van de objectcode met de bijbehorende vrije-locateerbaarheidsadministratie (zie 4.) vindt plaats;
- objectcode en vrije-locateerbaarheidsadministratie wordt op de trommel gezet (zie 13.16.: de routine `READ LIBRARY ROUTINE`);
- aan de administratie van de bibliotheek als geheel wordt een lijstje met gegevens van de procedure toegevoegd:
 - 1 lengte in het kerngeheugen
 - 2 adres op de trommel
 - 3 verwijzingen naar de trommelprocedures die door de procedure aangeroepen worden (zie 13.3.: de routine `FILL INFOTABLE`).Dit lijstje wordt in het vervolg speciale informatie genoemd.

2.3. Uitbreiding van de monitor.

2.3.1. De monitor in `MILLILIB` is uitgebreid voor de vervulling van twee extra taken na geslaagde vertaling van een ALGOL 60-programma:

- 1 de eventueel benodigde trommelprocedures ophalen; ze achter het objectprogramma plaatsen en hun interne adressering aanpassen;
- 2 daarna de gehele objectcode, die achter de vertaler is opgebouwd, naar beneden schuiven, daarbij de vertaler overschrijvend (zie [6] p. 52: de routine `TRANSPOSE`).

De voor de uitvoering van de eerste taak benodigde routine is tijdens executie van het objectprogramma niet meer nodig, en is daarom bij de vertaler in het geheugen geplaatst onder de naam `SUBMONITOR`.

In de ELAN-tekst van MILLILIB is voor de overzichtelijkheid een submonitor-sectie opgenomen waarin naast SUBMONITOR zoveel mogelijk alle op bibliotheekhandelingen betrekking hebbende vertalererroutines zijn geplaatst. Deze submonitorsectie is aan dit rapport als hoofdstuk 13 toegevoegd.

2.3.2. Naast de in 2.3.1. bedoelde monitor voor de verwerking van normale ALGOL 60-programma's, bevat MILLILIB ook een monitor voor het opbouwen van de trommelbibliotheek. In feite is daartoe de monitor zelf geparametriseerd d.m.v. de variabele NORMAL (zie de 1e instructie van SUBMONITOR, en de 1e instructie van END RUN). Heeft NORMAL de waarde false dan wordt i.p.v. SUBMONITOR (eigenlijk i.p.v. NORMAL PROGRAM) de routine EXTEND LIBRARY uitgevoerd; d.w.z. de monitor tracht de trommelbibliotheek uit te breiden. Een globale beschrijving van EXTEND LIBRARY vindt men in 2.2.

2.3.3. De overgang tussen NORMAL is true en NORMAL is false is niet geautomatiseerd: er is van uitgegaan dat wijziging van de trommelbibliotheek slechts incidenteel nodig is. Het stuk EXT LIB in de submonitorsectie, dat EXTEND LIBRARY en de bijbehorende vertalererroutines bevat (zie 13.1. t/m 13.14.), is onmiddellijk achter de vertaler geplaatst; slechts indien NORMAL door handelingen aan de console (zie 10.) de waarde false gekregen heeft, wordt EXT LIB van de trommel gehaald: de lengte van het traject voor compilertransporten wordt daartoe overeenkomstig vergroot.

2.3.4. Overzicht van de monitorparameter NORMAL en de vertaler parameters COMPMODE en BASE:

NORMAL	COMPMODE	BASE	taak
<u>true</u>	0	:base0	normale verwerking van ALGOL60-programma's
<u>false</u>	1	:base1	analyse van procedure programma's
<u>false</u>	2	:base2	opnemen van trommelprocedures

Wij zullen in het vervolg de waarde van COMPMODE gebruiken om de monitor- en vertalerversie aan te duiden waar we het over willen hebben.

3. Bibliotheekadministratie

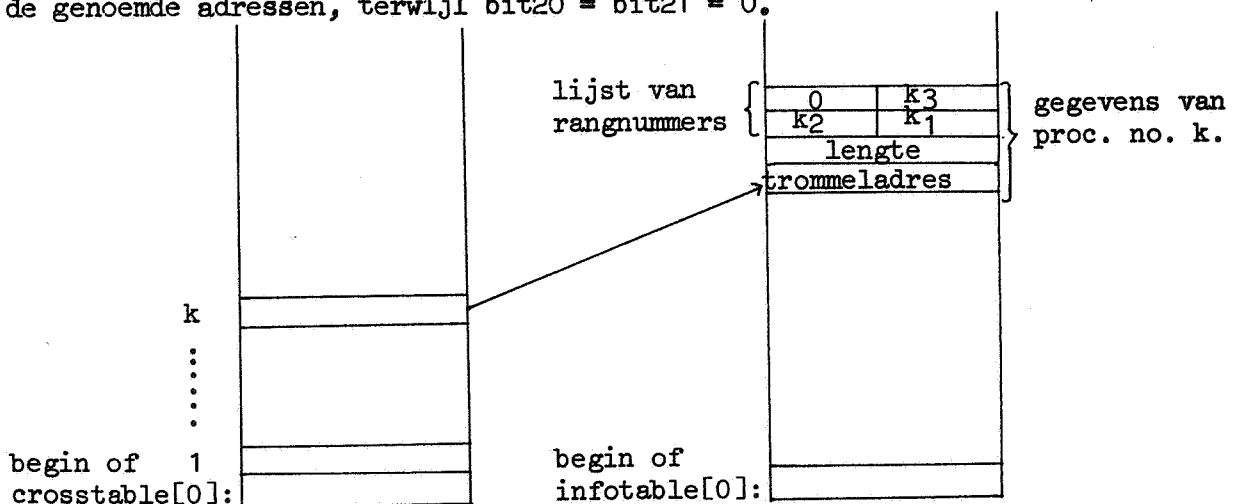
In de catalogue is voor iedere naam van een bibliotheekprocedure behalve een codering van de letters van die naam ook een rangnummer opgenomen dat aan die procedure is toegekend. Als voorbeeld volgt hier de codering in de catalogue van de gegevens van de procedure tab.

```

end of catalogue:    ...
                    ...
                    ...
                    +0          " endmarker
                    ('222 000 000' +1) " nul formele parameters
                    24 x d19 + 32    " rangnummer is 32
                    -'0 00 36 13 14' " naam is tab
                    ...
                    ...
begin of catalogue: ...

```

Als een bibliotheeknaam wordt opgezocht, wordt zijn rangnummer bekend en de speciale informatie over de betreffende procedure, zoals lengte en trommeladres, wordt dan in de infotable gevonden via een arrayelement `crosstable[rangnummer]`, waarin het beginadres van die informatie is geborgen, terwijl bit 20 aangeeft of de betreffende bibliotheekprocedure al (1) dan niet (0) in het programma aangeroepen wordt. Deze indicatie wordt daar aangebracht tijdens `Prescan1`, op het moment dat de naam van de procedure in de catalogue wordt opgezocht door de routine `ASK LIBR` (zie[6] p. 306); bovendien wordt de naamlijst uitgebreid met de gegevens van de procedure, eenvoudig door ze te copieren uit de catalogue, zodat bij elk volgend voorkomen van de procedure in het programma de naam en het rangnummer direct in de naamlijst gevonden worden. De crosstable wordt voor ieder programma samen met de vertaler "vers" opgehaald van de trommel, waar hij bij de vertaler staat, gevuld met de genoemde adressen, terwijl `bit20 = bit21 = 0`.



Onder de speciale informatie bevindt zich ook een lijst van de rangnummers der procedures die worden aangeroepen door de in behandeling zijnde procedure en deze aangeroepen procedures hebben elk ook weer zo'n lijst. Uitgaande van de lijsten van alle direct aangeroepen procedures worden, onder gebruikmaking van de indicatiebits in de crosstable (bit20 en bit21), alle direct of indirect aangeroepen procedures bepaald. Dat gebeurt in de routine CRF (zie 13.18.) en wordt aangegeven in bit21 van de crosstable.

Het opbouwen van de genoemde lijst van rangnummers in de infotable gebeurt in de routine LIBRARY ROUTINE (zie 13.3.). De wijze van aanroep van de routines CRF en LIBRARY ROUTINE vormt een voorbeeld van de parametrisering van de vertaler wat betreft bibliotheekhandelingen, want zij worden op dezelfde plaats aan het begin van de translationscan aangeroepen; CRF voor normale programma's (COMPMODE = 0) en LIBRARY ROUTINE bij de afzonderlijke vertaling van procedures voor opname in de bibliotheek (COMPMODE = 2).

4. Vrij-locateerbare code

- 4.1. In de trommelbibliotheek bevindt zich een reeks procedures in voorvertaalde vorm.

Dat betekent, dat zo'n procedure eens is vertaald met COMPMODE = 2, en de resulterende objectcode in het kerngeheugen is ingezet bv. vanaf kernadres A. Daarna is de objectcode naar de trommelbibliotheek getransporteerd.

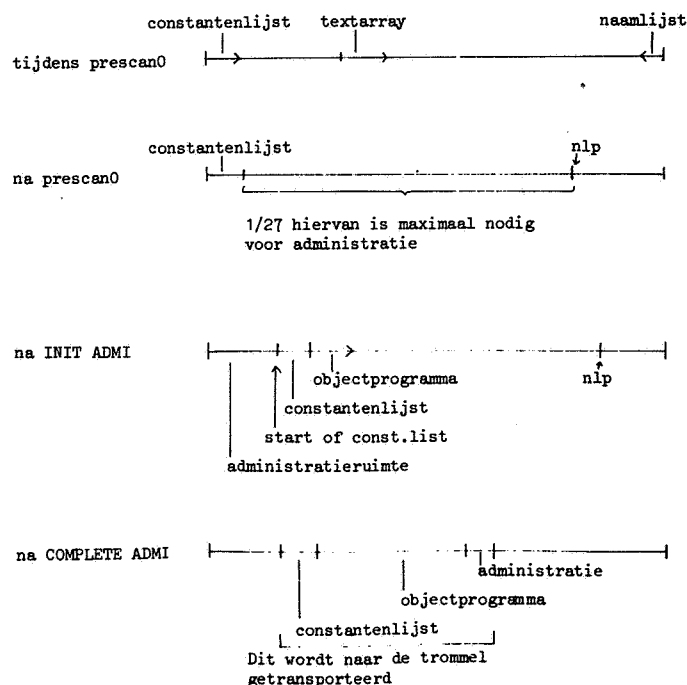
Wordt nu deze procedure aangeroepen in een gebruikersprogramma, en de objectcode daarom opgehaald van de trommel en ingezet bv. vanaf kernadres B, dan moeten, om zinvolle code te verkrijgen, de interne verwijzingen worden aangepast, d.w.z.:

- a alle statische adressen moeten worden gemodificeerd: de correctie $B - A$ moet erbij worden opgeteld;
- b alle rangnummers van procedures moeten door adressen van pseudolvars vervangen worden.

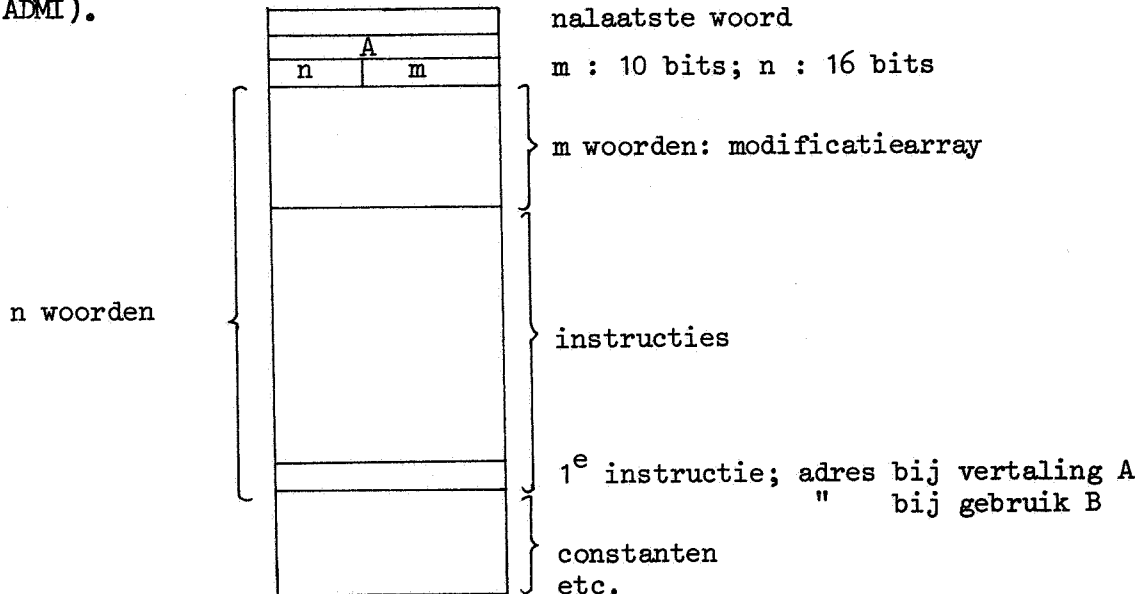
Wat betreft deze rangnummers dient vermeld te worden dat zij bij COMPMODE = 2, in tegenstelling tot COMPMODE = 0 (zie 6.1.), niet tijdens de vertaling door (statische) adressen vervangen kunnen worden, omdat die adressen nog onbekend kunnen zijn (zie ook 9.). Om te kunnen uitzoeken welke geheugenwoorden van de objectcode van een trommelprocedure een statisch adres of een rangnummer bevatten wordt tijdens vertaling met COMPMODE = 2, d.m.v. aanroepen van de routine DO ADMI (zie 13.3.), een vrije-locateerbaarheidsadministratie opgebouwd.

Deze bestaat uit een boolean array, het modificatiearray, met precies een element voor ieder geheugenwoord van de objectcode, waarbij de waarde van het arrayelement true is voor een geheugenwoord met een statisch adres of een rangnummer, en anders false is.

- 4.2. Hoe de reservering van geheugenruimte voor de administratie bij COMPMODE = 2 plaats vindt, is af te lezen uit de volgende schetsen van een gedeelte van het kerngeheugen.



- 4.3. De vrije-locateerbaarheidsadministratie bevat naast het in 4.1. genoemde boolean array nog enkele noodzakelijke verwijzingen, die in de volgende schets zijn aangegeven (zie 13.5.: de routine COMPLETE ADMI).



trommelprocedure met vrije-locateerbaarheidsadministratie

Nadat een procedure van de trommel in het kerngeheugen is geplaatst is het adres van het nalaatste woord bekend, omdat de lengte van het totale transport bekend is. Onder gebruikmaking van n wordt dan het nieuwe adres B bepaald van de 1e instructie; dit is nodig voor de berekening van de correctie $B - A$ (zie 4.1.); m.b.v. m wordt het beginadres van het modificatiearray bepaald.

Hierna is de feitelijke aanpassing der adressen mogelijk (zie 6.2. en 13.16.: de routine ADAPT ADR).

- 4.4. Hoe alle situaties bepaald worden, waarin de vertaler statische adressen in het objectprogramma opneemt, wordt in de volgende secties behandeld.

4.4.1. Analyse van macrowoord en macroparameter.

Het genereren van stukjes objectcode door de vertaler gebeurt via macro's: de vertaler roept de routine MACRO (zie [6] p. 357) aan met in register A een macronummer en eventueel in register S een macroparameter. MACRO roept de routine PRODUCE (zie [6] p. 359) aan, en PRODUCE roept, voorafgaande aan het produceren van objectcode, zonodig de routine PRCS PAR (d.i. process parameter) aan. PRCS PAR leidt uit de macroparameter een bedrag af dat bij een van de door de macro gespecificeerde instructies moet worden opgeteld. Vrijwel alle gevallen waarin een adres in een instructie moet worden ingevuld, worden hierdoor gedekt, en daarom is in de routine PRCS PAR (zie [6] p. 361) een test op statische adressen ingebouwd. De structuur van deze test kan als volgt worden aangeduid:

```
if value like (macro) then inspect macro
else if macro = JU ∨ macro = SUBJ ∨ macro = NIL ∨ macro = LAST then
inspect name
else if 7 dynamic (parameter) then do administration;
```

Toelichting:

- Is een macro "value like" (aangegeven in bit3 van het macrowoord), dan stelt de macroparameter een constante voor, behalve bij de macro's TRC, TIC, COJU, YCOJU, DO en de hiervan afgeleide geoptimaliseerde macro's, waar de parameter een statisch adres voorstelt.
Dit is in MILLILIB in bit2 van het macrowoord, dat nog vrij was, aangegeven. Inspect macro (zie 13.5.: de routine INSP MACRO) test hierop en verricht zonodig administratie in het modificatiearray (zie ook 8.2.).
- Inspect name (zie 13.6.: de routine INSP NAME) verricht administratie behalve in de in 8.2. genoemde situatie.
- In alle overige gevallen verwijst de macroparameter naar de plaats in de naamlijst waar het benodigde adres of rangnummer gevonden kan worden. Daar staat tevens aangegeven of het om een statisch adres of rangnummer, of om een dynamisch adres gaat. De routine dynamic (zie [6] p. 366: de naamlijstprocedure DYNAMIC) zoekt dit uit.

4.4.2. Actuele parameters (zie [3] p. 3).

De vertaling van een procedureaanroep heeft de volgende structuur:

- a voor ingewikkelde parameters, stukjes programma ter berekening van die parameter;
- b daarna de eigenlijke aanroep;
- c daarna voor iedere parameter een codewoord, APD (= actual parameter descriptor) geheten, dat een statisch adres kan bevatten.

Terwijl het onder a genoemde in het objectprogramma wordt opgenomen, worden tegelijk de APD's opgebouwd en voorlopig op de stapel bewaard. Zijn a en b klaar dan worden (zie [6] p. 330: PARLIST13) de APD's een voor een van de stapel genomen en als parameter aan de macro CODE meegegeven. CODE is de enige macro met een parameter waarvoor niet PRCS PAR wordt aangeroepen: hij heeft tot effect dat de meegegeven parameter ongewijzigd in het objectprogramma wordt opgenomen. Daarom wordt in PARLIST13 zelf, waar CODE gegenereerd wordt, een test op statische adressen uitgevoerd (zie 13.5.: de routine INSP APD) en zonodig administratie verricht.

4.4.3. Switch-declaraties (zie [3]: 5.6.).

De vertaling van een switch-declaratie heeft de volgende structuur:

- a voor ingewikkelde designational expressies uit de switchlist, stukjes programma ter berekening van hun waarde;
- b daarna het aantal entries in de switchlist;
- c daarna voor iedere entry een codewoord, SWORD geheten, dat een statisch adres kan bevatten.

Analoog aan het in 4.4.2. vermelde, worden de SWORD's eerst op de stapel bewaard en later m.b.v. de macro CODE aan het objectprogramma toegevoegd. Daarom wordt ook hier ter plaatse getest op statische adressen (zie [6] p. 348: SWDEC6), d.w.z. op het voorkomen van SWORDO. Op het voorkomen van SWORD2 behoeft niet getest te worden, omdat dit SWORD binnen proceduredeclaraties door de vertaler nooit gegenereerd wordt.

4.4.4. For statements (zie[3]: 4.8.).

Het begin van de vertaling van een forlistelement luidt:

```
F = :NEXT          "TSIC(NEXT)
forvar = G
```

waarbij NEXT staat voor de volgende slag in het in behandeling zijnde forlistelement of voor het volgende forlistelement. Omdat hier de value like macro TSIC wordt gebruikt die verder nooit een adres als parameter heeft, is het nodig ter plaatse (zie [6] p. 341: FORLIST1) de administratie te verrichten.

5. Geheugenindeling

- 5.1. Het is van belang om de verschillende systeemdelen zo in het geheugen te plaatsen dat in elke fase van de verwerking van een ALGOL 60-programma de beschikbare werkruimte zo groot mogelijk is.

In het bijzonder heeft dit betrekking op de plaatsing van de compiler: hij werkt slechts beneden 32 K en bouwt een objectprogramma op, dat ook beneden 32 K moet staan, en dat na voltooiing van de vertaling best op de plaats van de compiler mag staan. Wij geven hier drie mogelijkheden aan.

5.1.1. MKL:	monitor	0 - 7 K
	compiler	7 - 13 K
	objectprogramma	tijdens vertaling: op de trommel na vertaling: 7 - 32 K

De vertaling wordt hier in macro-vorm op achtergrondgeheugen opgebouwd, en na voltooiing van de vertaling wordt de uiteindelijke objectcode over de vertaler heen geschreven (zie [4]).

5.1.2. MILLI:	monitor	0 - 7.5 K
	objectprogramma	7.5 K - 26 K
	compiler	26 K - 32 K

Het objectprogramma mag tijdens executie de compilerruimte als werkruimte gebruiken.

5.1.3. MILLILIB:	monitor	0 - 7.5 K
	compiler	7.5 K - 13.5 K
	objectprogramma	tijdens vertaling: 13.5 K - 33 K na vertaling: 7.5 K - 32 K

De vertaler schrijft de instructies van het objectprogramma vanaf 13.5 K, waarbij de adressering in de instructies is aangepast aan uiteindelijke plaatsing vanaf 7.5 K. Na voltooiing van de vertaling wordt het objectprogramma verschoven en over de vertaler heen geschreven.

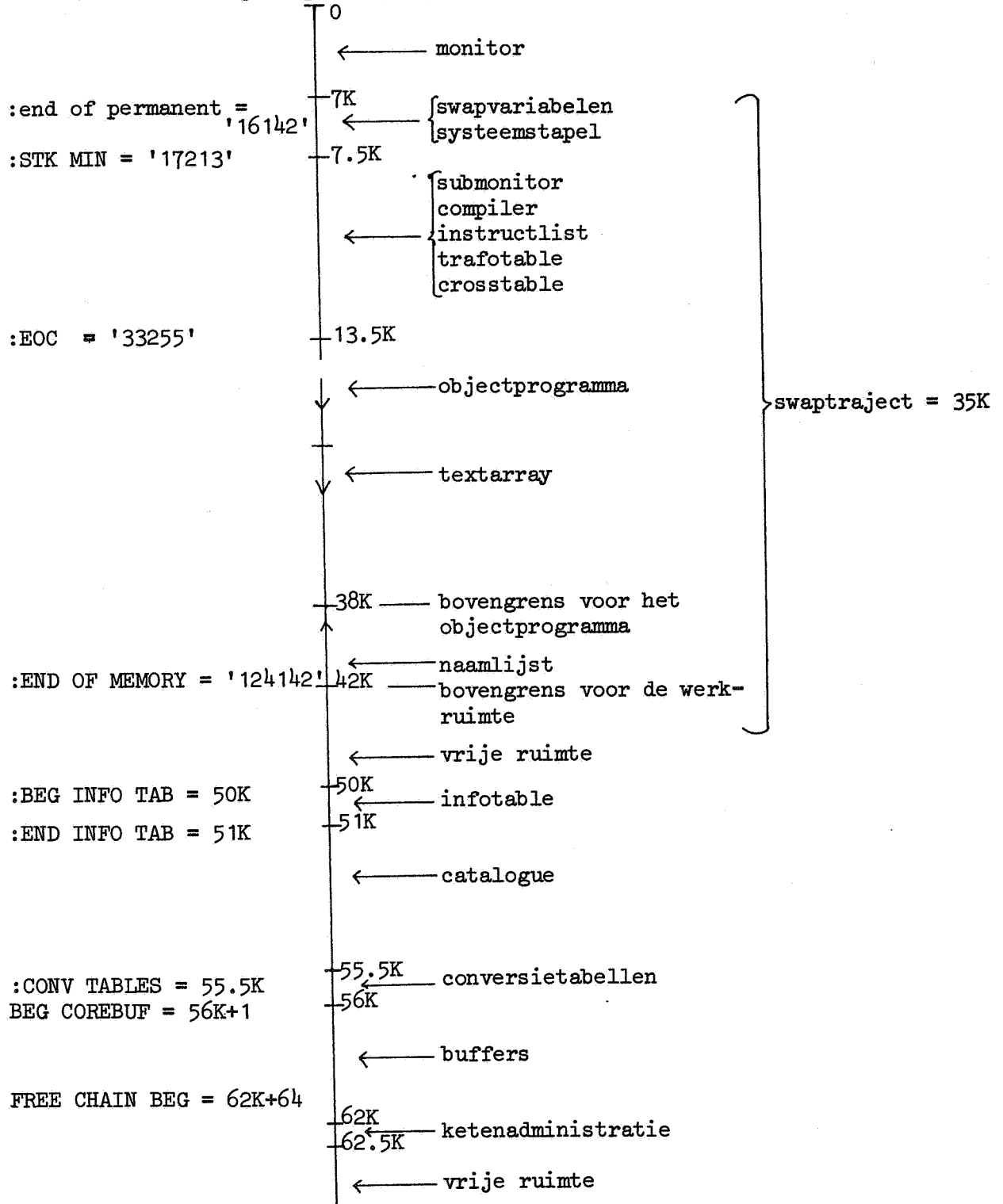
Het verschuiven van het objectprogramma maakt het noodzakelijk om tijdens vertaling voor alle verwijzingen naar het objectprogramma goed onderscheid te maken tussen het actuele adres (d.i. tijdens vertaling) en het virtuele (d.i. tijdens executie). De overgang tussen actuele en virtuele adressen wordt gemaakt m.b.v. variabele transposition en er geldt:

$$\text{actueel adres} = \text{virtueel adres} + \text{transposition.}$$

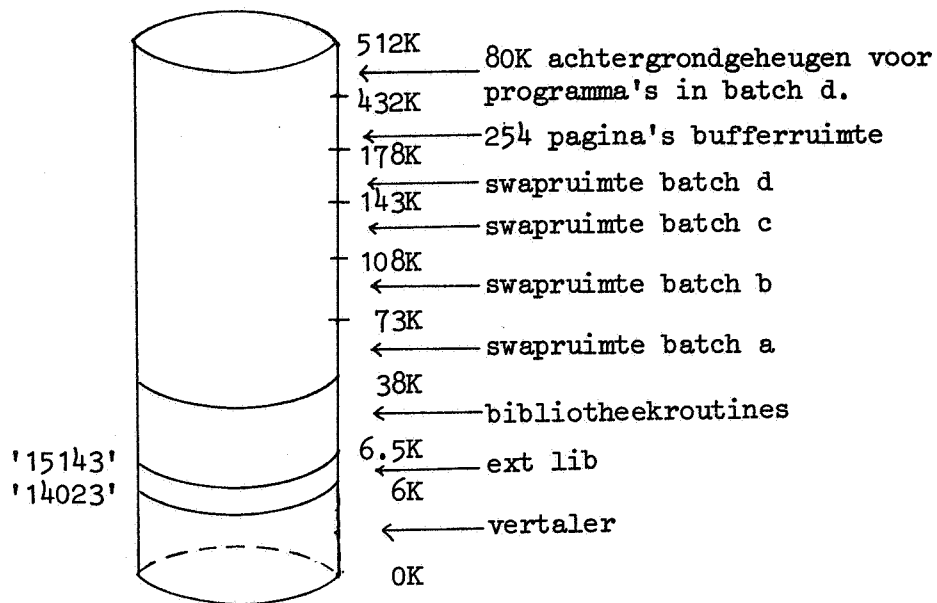
Binnen de compiler is de waarde van de variabele instr cntr altijd een virtueel adres, en ook de adressen die in de naamlijst worden opgenomen zijn virtueel.

Binnen de routine SUBMONITOR echter is de waarde van instr cntr actueel.

5.2. Het kerngeheugen tijdens vertaling.



5.3. Trommelindeling.



6. Diverse routines

6.1. ADDR OF LVAR (zie 13.19.).

Deze routine moet, bij normale programma's tijdens de translationscan, bij de eerste aanroep van een bepaalde bibliotheekprocedure, in de naamlijst het rangnummer vervangen door het adres van de overeenkomstige pseudolvar in het objectprogramma. Dit adres is bij het begin van de translationscan door de routine PSEUDOLVAR, aangeroepen door CRF (zie 13.18.), in een werkarray, trafotable geheten, weggeschreven zodat trafotable [rangnummer] = adres van pseudolvar.

6.2. ADAPT ADR (zie 13.16.).

Deze routine moet, nadat een procedure van de trommel is gehaald, deze gebruiksklaar maken d.w.z. de interne verwijzingen in orde maken aan de hand van de bijgevoegde administratie.

De hoofdcyclus van de routine kan als volgt worden aangeduid:

```
loop:  take next (instruction);
       address:= addresspart (instruction);
       address:= if address < 4096 then convert number
                 else address + correction;
       adapt instruction (address);
       goto loop;
```

Toelichting:

Een adres < 4096, in feite een adres < het aantal bibliotheekprocedures, kan alleen een rangnummer van een bibliotheekprocedure zijn. Analooq aan 6.1. wordt dit door convert number (zie 8.1.) omgezet in het adres van de bijbehorende pseudolvar. Voor de behandeling van LJU1 zie 8.1. Bij adressen > 4095 wordt een vaste correctie, CORRECTION, opgeteld (zie 4.1.).

De macro LAST heeft als bijzonderheid dat het erin voorkomende adres negatief in het geheugen staat. Vandaar dat de correctie wordt afgetrokken.

7. Anonieme procedures

- 7.1. In MILLILIB is de mogelijkheid opgenomen om procedures op te nemen, samen met hulpprocedures die op zichzelf niet aangeroepen kunnen worden vanuit ALGOL60-programma's. Dit wordt bereikt door de namen van de in aanmerking komende, met een vraagteken gemerkte, procedures aan het eind van de verwerking van het bibliotheekprogramma uit de catalogue te verwijderen. Dit anoniem maken is bv. van belang wanneer men als hulpprocedure een codeprocedure heeft die alleen op een zeer bepaalde manier gebruikt kan worden.

```
Vb.      begin   proc ? P1;...;
           proc P2;
           begin   ...;
                        P1
           end P2;
           end
```

Hier wordt P2 wel maar P1 niet aan de gebruiker ter beschikking gesteld.

7.2. Zij COMPMODE = 1.

In Prescan0 en Prescan1 wordt een vraagteken, wanneer dat onmiddellijk voorafgaat aan de procedureidentificer van een proceduredeclaratie met displaylevel = 1, zonder meer geskipt (zie 13.13.: de routine SKIP QM, aangeroepen in [6] p. 280 en 295).

In de translationscan wordt een vraagteken op een dergelijke plaats ontdekt door de routine INSPECT DECL (zie 13.12.). De dan aangeroepen routine HANDLE QM (zie 13.13.) plaatst een indicatie in crosstable[rangnummer], en vervangt het vraagteken in het textarray door twnr.

Twnr is het enige niet-ALGOL 60-symbool dat in het textarray kan voorkomen. Een gevolg van de vervanging is wel dat eventuele erna gegeven foutmeldingen een te hoog regelnummer hebben.

Zij COMPMODE = 2.

Na het verlaten van de loop NEXT PROCEDURE (zie 13.2.) wordt de routine ANONYMIZE (zie 13.14.) aangeroepen. Daar worden m.b.v. de naamlijstprocedure NXT PNR (d.i. next pointer, zie [6] p. 266) achtereenvolgens alle namen van trommelprocedures opgezocht. Bij elke naam wordt gecontroleerd of hiervoor een indicatie is opgenomen in de crosstable. Is dat het geval dan wordt de indicatie verwijderd en de naam in de catalogue vervangen door een bitpatroon dat niet met een naam kan corresponderen, door bit24 op te zetten.

In de huidige versie van MILLILIB wordt de ruimte die de anonieme procedures in de catalogue innemen niet vrijgegeven.

8. Codeprocedures

In de trommelbibliotheek kunnen procedures in code worden opgenomen (zie [6] p. 370, de routine TRL CD). In deze code kunnen o.a. voorkomen macronummers, eventueel met bijbehorende macroparameters. De interpretatie van deze macroparameters in MILLILIB wordt in de volgende secties besproken (zie 4.4.1.).

- 8.1. Bibliotheeknamen als macroparameter worden in de objectcode op de trommel aangegeven door hun rangnummer.
Bij aanroep van de procedure worden de in de bijbehorende objectcode voorkomende rangnummers vervangen door de adressen van de betreffende pseudovar's (zie 6.2.):
bv. SUBC (n) wordt SUBC (pseudolvarn), waarbij pseudolvarn de pseudovar voorstelt behorende bij het rangnummer n.
Hierbij wordt een geval afzonderlijk behandeld nl. de instructie GOTO (pseudolvarn[1]), verkregen uit de macro IJU1 (no 105). Deze instructie is door de macroprocessor als de indirecte sprong GOTO (n+1) in de bibliotheek gezet.
Het voorkomen van deze indirecte sprong wordt nu door de routine ADAPT ADR herkend en de in 6.2. gegeven beschrijving van de werking van ADAPT ADR kan dan ook als volgt aangevuld worden.

```
integer procedure convert number;  
  convert number:=  
    if instructionpart (instruction) = indirect jump  
    then trafotable[address - 1] + 1  
    else trafotable[address];
```

Een gevolg van deze interpretatie is, dat de macro IJU (no 104), die ook een indirecte sprong oplevert, niet met een bibliotheeknaam als parameter gebruikt mag worden, hetgeen ook niet zinvol zou zijn.

- 8.2. Adressen als macroparameter, door een getal aangeduid, worden geïnterpreteerd als vaste systeemadressen, dus onafhankelijk van de plaats waar de procedure wordt ingezet. De resulterende instructies worden dan ook niet gemodificeerd (zie 13.5. e.v., de routines INSP MACRO en INSP NAME).

9. Catalogue

In MILLI vindt men in tegenstelling tot MILLILIB voor elke bibliotheekprocedure via de catalogue het vaste adres van een pseudovar, die verwijst naar de betreffende routine die permanent in het kerngeheugen staat.

Voor de trommelprocedures in MILLILIB zijn pseudovar's zeker nodig, omdat tijdens vertaling (translationscan en de routine ADAPT ADR) de plaatsing van de routines nog onbekend kan zijn, zodat directe verwijzing niet mogelijk is.

Aangezien het niet aantrekkelijk is om voor alle trommelprocedures permanent kernruimte voor hun pseudovar's te reserveren, zijn tijdens het opnemen van procedures in de bibliotheek de eventueel voor de vertaling benodigde adressen van pseudovar's niet bekend. Om deze reden vindt men in MILLILIB in de catalogue voor iedere procedure een rangnummer i.p.v. een adres. Bij aanroep van een trommelprocedure leveren dan de routines ADDR OF LVAR en ADAPT ADR de benodigde adressen van pseudovar's (zie 6.).

Deze rangnummers zijn prettig voor het hanteren van de bibliotheekadministratie. Aangezien er echter voor de systeempcedures naast de catalogue geen verdere bibliotheekadministratie nodig is, zou het goed zijn om twee catalogues te hebben, een voor de permanente bibliotheek met adressen van permanente pseudovar's, die dan zonder bezwaar gecomprimeerd kan worden door een lijst van parameterpatronen afzonderlijk op te nemen, en een tweede voor de trommelbibliotheek met rangnummers.

De Prescan1-routine ASK LIBR is de enige die dan gewijzigd moet worden: slechts voor procedures uit de trommelbibliotheek moet daar een indicatie in crosstable geplaatst worden. De routine ADDR OF LVAR blijft correctwerken.

10. Gebruiksaanwijzing

10.1. In dit hoofdstuk worden de diverse handelingen beschreven, die nodig zijn bij het opbouwen van een bibliotheek en het vastleggen op magneetband van het systeem met een specifieke bibliotheek.

10.2. Protocol voor het opbouwen en vastleggen van een bibliotheek.

- a Als het systeem reeds met een bibliotheek op een magneetband staat, lees dan die magneetband in (zie [7.]), lees een programma met minstens een correcte monitorkop in, en ga verder bij k.
- b Als het systeem zonder bibliotheek op een magneetband staat, lees dan die magneetband in (zie [7.]) en ga verder bij e.
- c SVA↑; LS; IP; lees de ponsband in die het systeem bevat; SVA↓.
- d LS; NB.
- e SVA↑; SVA↓; LS; BGA op '50'.
- f SVA↑; zet in het consolewoord de instructie:
LIBRARY EMPTY = B d.i. '460000513';
HH; SVA↓; BVA;
lees een bibliotheekprogramma in met minstens een correcte monitorkop.
- g Lees bibliotheekprogramma's in indien gewenst.
- h SVA↑; SVA↓; LS; BGA op '51'.
- i Verwerk programma's indien gewenst.
- j Moet de bibliotheek vastgelegd worden, ga dan verder bij m.
- k SVA↑; SVA↓; LS; BGA op '50'.
- l Moet de bibliotheek leeg, ga dan verder bij f, anders bij g.
- m SVA↑; SVA↓; LS; BGA op '52'.
- n Lees een programma in met minstens een correcte monitorkop; wacht tot de machine statisch stopt.
- o Schrijf het systeem naar magneetband met verplaatsing, bv. naar '177000' (zie [7.]).

10.3. Toelichting bij 10.2.

In [6] p. 7 komen de volgende instructies voor:

M['50']:	GOTO (:START EXT LIB)
M['51']:	GOTO (:FINISH EXT LIB)
M['52']:	GOTO (:MILLI TO TAPE)

Bij uitvoering leiden deze instructies naar routines in 13.14.

Een waarde voor NORMAL wordt nu gekozen door het systeem te starten op '50' (NORMAL:= false) of '51' (NORMAL:= true).

De overgang tussen beide toestanden is onbeperkt mogelijk.

Als NORMAL false is kan men de bibliotheek uitbreiden of, indien gewenst, de bibliotheek leeg maken, door LIBRARY EMPTY met de hand true te maken, en een geheel nieuwe bibliotheek opbouwen. Bibliotheekprogramma's mogen in willekeurige batches ingelezen worden, echter niet in verschillende batches tegelijk.

Het systeem is zo gemaakt, dat herhaald starten op '50' resp. '51' hetzelfde effect heeft als eenmaal starten op '50' resp. '51'.

Bovendien heeft drukken op LSNB geen invloed op de waarde van NORMAL.

Waarschuwing: Het is onmogelijk MILLILIB te gebruiken indien niet minstens eenmaal de handelingen beschreven onder 10.2. f uitgevoerd zijn, d.w.z.. eens moet de routine FOUND LIBRARY (zie 13.7.) zijn uitgevoerd ter initialisatie van de bibliotheekvariabelen.

- 10.4. De gang van zaken bij het op magneetband zetten is als volgt (zie 13.14.: de routine MILLI TO TAPE).

Door te starten op '52' komt men in MILLI TO TAPE, waar de lengte van het traject van het eerstvolgende compilertransport passend wordt vergroot, zodat de hele bibliotheek dan mee naar het kerngeheugen wordt getransporteerd, gevolgd door een sprong naar DIRECTION TO DRUM. Na het verwerken van een correcte monitorkop wordt dit transport uitgevoerd, en daarna wordt in DIRECTION TO DRUM de richting van het eerstvolgende (na het schrijven op en weer inlezen vanaf de magneetband) compilertransport ingevuld, nl. "to drum"; tevens wordt een sprong naar MILLI FROM TAPE voorbereid. De machine stopt daarna op de instructie A = MA, omdat A een negatief adres bevat.

Hierna kan men de inhoud van het kerngeheugen op magneetband zetten volgens [7].

Leest men deze magneetband nu in, dan zal de verwerking van de eerste correcte monitorkop gevolgd worden door het hierboven voorbereide compilertransport waarbij vertaler en bibliotheek naar de trommel geschreven worden; daarna wordt dan de lus naar MILLI FROM TAPE uitgevoerd waar de normale situatie hersteld wordt. Merk op, dat indien op het moment van starten op '52' de toestand NORMAL is false heerst, deze toestand ook na het inlezen van de verkregen magneetband weer heerst, iets dat men waarschijnlijk meestal niet zal wensen.

11. Foutmeldingen

- 396 in codebody ontbreekt een comma als scheider tussen
macronummer en bijbehorende parameter.
- 397 in codebody volgt na minus geen getal.
- 398 in codebody begint een parameter niet met letter, cijfer
of minus.
- 399 in codebody als macronummer geen getal.
- 400 codebody niet afgesloten met unquote.
- 401 codebody niet aangevangen met quote.
- 801 bibliotheekprogramma begint niet met begin.
- 802 bibliotheekprogramma is een compound statement.
- 803 het compound tail van een bibliotheekprogramma bestaat niet
alleen uit end.
- 804 bibliotheekprogramma bevat op het niveau van het buitenste
block een declarator, die geen procedure declareert.
- 805 de naam van een aangeboden bibliotheekprocedure komt al voor
in de bibliotheek.
- 806 ruimte voor infotable is uitgeput.
- 807 maximale aantal bibliotheekprocedures overschreden.
- 808 bibliotheekprocedure is na voltooiing van zijn administratie
te lang voor het beschikbare geheugen.
- 809 ruimte voor catalogue is uitgeput.
- 810 systeemfout: de routine TRANSPOSE wordt onjuist gebruikt.
- 811 trommelruimte voor procedures is uitgeput.

12. Verschilpunten t.o.v. MILLI

In dit hoofdstuk worden concrete verschillen t.o.v. MILLI in routines van MILLILIB aangeduid.

12.1. De vertaler.

In de volgende secties worden nummers als verwijzing opgegeven. Bedoeld worden de nummers die in [6], en uitgezonderd 12.1.1. ook in [2], in paginahoofden voorkomen.

12.1.1. General purpose procedures.

<u>Verwijzing</u>	<u>routine</u>	<u>toelichting</u>
0	NXT BSC SBL	COMPMODE = 2: tekst reeds in textarray
7	OUT SBL	idem
13	NAME IN LIBRARY	uit ASK LIBR gelicht
16	INIT	first shift i.v.m. TRUNCATE

12.1.2. Prescan0 procedures.

51 cont	BLOCK(NEXT5)	COMPMODE = 2: aanroep TRUNCATE
52 cont	DECL LST(ELSE1)	COMPMODE = 1: aanroep SKP QM
53/54	ST NUM CS	transposition
mainpr	PRESCAN0	initialisatie textarray i.v.m. TRUNCATE

12.1.3. Prescan1 procedures.

65 cont	PRC DEC	COMPMODE = 1: aanroep SKP QM
66/67	LAB DEC	transposition
70 cont	ASK LIBR	indicatie in crosstable
mainpr	PRESCAN1	TOP OF DISPLAY en INIT ADMI

12.1.4. Translator procedures.

11 cont	PR CALL	COMPMODE = 0: aanroep ADDR OF LVAR
12/13	PAR LIST13	COMPMODE = 2: aanroep INSP APD
	ACT PAR	COMPMODE = 0: aanroep ADDR OF LVAR
17	FOR LST1	COMPMODE = 2: aanroep ADMI4
18 cont	SW DEC6	idem
19/20	BLOCK	COMPMODE = 1: aanroep DEC LST3CM1
	DEC LSTO	COMPMODE = 1: aanroep INSPECT DECL
20 cont	PROGRAM	COMPMODE = 1: aanroep CMPTL3CM1
21	SUBST2	COMPMODE = 1: geen objectcode transposition
21/22	CODE0	COMPMODE = 1: geen objectcode transposition
22/23	PRCS PAR[11]	COMPMODE = 2: aanroep INSP MACRO
	NON VAL[14]	COMPMODE = 2: aanroep INSP NAME
28	TRL CD[7]	COMPMODE = 0: codebody is verboden
	TRL CDO[24]	COMPMODE = 0: aanroep ADDR OF LVAR
28/29	UNS NUM	transposition

mainpr	TRANSL	COMPMODE = 0: CRF en PROGRAM3 COMPMODE = 1: PROGRAM3CM1 COMPMODE = 2: LIBRARY ROUTINE en PROGRAM3CM2
macrolist	MCR LST	bit2 is opgezet bij de nummers: 85(TRC),86(TIC),106(COJU), 107(YCOJU),111(DO)
opt'd macro's	TABEL	bit2 is opgezet bij de nummers: 2,3,7,8,12,13,17,18,22,23,27,28, 32,33,37,38,42,43,47,48,52 en 53.

12.2. Monitor en systeem.

<u>pag. in[6]</u>	<u>routine</u>	<u>toelichting</u>
40	DNTST(LOOP1)	initialisatie van NORMAL en LIBRARY EMPTY
45	NAIGA(RD CMP)	aanroep SUBMONITOR en TRANSPOSE
46	ENDRUN[1]	aanroep ASSURE LIBRARY ter beveiliging van de bibliotheek bij voortijdige beëindiging van bibliotheekprogramma's
52	TRANSPOSE	verschuift objectcode na vertaling
163	TEST IC	expliciete test of objectcode beneden 32 K blijft

13. ELAN-tekst van de submonitorsectie

De in dit hoofdstuk gegeven tekst is een afdruk van een gedeelte van [6], zijnde MILLILIB geassembleerd door de MC-ELAN1-assembler (wordt gepubliceerd).

De drie kolommen links in de tekst geven achtereenvolgens aan: regelnummer, octaal machineadres, en octale codering van de inhoud van dit adres. Paginanummers en sectienummers zijn toegevoegd.

2611 7: - 50%

230

"SUBMONITOR SECTION"

'BEGIN'

```

BASE1, BASE2, EXTEND LIBRARY, NXT PP,
LIBRARY ROUTINE, FILL INFOTABLE, DO ADMI,
LVAR, ASHIFT, APOS, OLD INSTR CNTR, INIT ADMI2,
COMPLETE ADMI, INSP MACRO, INSP APD, INSP NAME, ADMI 4,
ADR OP, CPTL3CM1,
DECL33CM1, PROGRAM3CM1, PROGRAM3CM2, FOUND LIBRARY, UPDATE LIBRARY,
INFORM PROGRAMMER, TRUNCATE, INSPECT DECL,
HANDLE GM, SKIP GM, ANONYMIZE, COMMODE, BASE,
BEG OF ADMI, END OF PROGRAM, RELADDR, RELADDR1, BEGINSAFE,
SHIFTSafe, WORDSAFE, PNTR, WORD1SAFE,
BEGIN OF PROC, FRAME, LOCAL NUMBER, D16M1,
NORMAL PROGRAM, LIBR CELL, READ LIBRARY ROUTINE, CRF,
ADAPT ADR, PREPARE, OBSERVE, BASE0, NXT SBL, INVERT,
DECL LS, DISP LVL, ENTR BLK, NAME IN LIBRARY, NEXT ENTRY, INIT POINTER,
CLEAR CRSTAB, UPDATE CROSTABLE, STOCK1, QUOTE COUNTER, SHIFT, LETTER, LAST SYMBOL,
CHARACTER, WORD COUNT, D20, D21, D24, D25, ADDR OF LVAR,
PR DEC, PROGRAM3, CPTL3, DECL33, ID3,
PAR PART, SAVE NXT SBL, RESTORE NXT SBL, NXT PNR, IN LIBR, CLEAR CRSTAB SPACE

```

EXT L13101:

BASL:

```

:PROGRAM3 CM1
:CMPTL3 CV1
:DECLST3 CM1
A + 0
A + 0
A + 0
A + 0
A + 0
A + 0
SUBC (:SKIP QM)
A + 0

```

JASE2:

```
: PROGRAM 3 CM2
: CRIPTL3
: DECLSTS
SUBC(: INSP MACRO)
SUBC(: ADV 4)
SUBC(: INSP APD)
SUBC(: LIBRARY ROUTINE)
SUBC(: TRUNCATE)
SUBC(: INSP NAME)
A + 0
A + 0
```

EXTEND LIBRARY: 'BEGIN' NEXT PROCEDURE

```

A= LIBRARY EMPTY, P
LIBRARY EMPTY= -B
SUBCD(:FOUND LIBRARY)
SUBC(:CLEAR CRSTAB SPACE)

```

```

COMPMODE= A
G=: BASE 1
BASE= G
SUBC:(PRECAN 0)
S = NBR OF ROUTINES
MAX NBR OF ROUTINES - S, P
" ROUTE

```

260.72 - 502

231

```

- 76 133161 122301447
- 77 133171 1562193561
- 78 133181 1623910201
- 79 133191 1522203344
- 80 133201 122301447
- 81 133211 1400000506
- 82 133221 1400000507
- 83 133231 1223016271
- 84 133241 1600172224
- 85 133251 1260754001
- 86 133261 1260754001
- 87 133271 1260754001
- 88 133281 1260754001
- 89 133291 1260754001
- 90 133301 1260754001
- 91 133311 1260754001
- 92 133321 1260754001
- 93 133331 1260754001
- 94 133341 1260754001
- 95 133351 1260754001
- 96 133361 1260754001
- 97 133371 1260754001
- 98 133381 1260754001
- 99 133391 1260754001
- 100 133401 1260754001
- 101 133411 1260754001
- 102 133421 1260754001
- 103 133431 1260754001
- 104 133441 1260754001
- 105 133451 1260754001
- 106 133461 1260754001
- 107 133471 1260754001
- 108 133481 1260754001
- 109 133491 1260754001
- 110 133501 1260754001
- 111 133511 1260754001
- 112 133521 1260754001
- 113 133531 1260754001
- 114 133541 1260754001
- 115 133551 1260754001
- 116 133561 1260754001
- 117 133571 1260754001
- 118 133581 1260754001
- 119 133591 1260754001
- 120 133601 1260754001
- 121 133611 1260754001
- 122 133621 1260754001
- 123 133631 1260754001
- 124 133641 1260754001
- 125 133651 1260754001
- 126 133661 1260754001
- 127 133671 1260754001
- 128 133681 1260754001
- 129 133691 1260754001
- 130 133701 1260754001
- 131 133711 1260754001
- 132 133721 1260754001
- 133 133731 1260754001
- 134 133741 1260754001
- 135 133751 1260754001
- 136 133761 1260754001
- 137 133771 1260754001
- 138 133781 1260754001
- 139 133791 1260754001
- 140 133801 1260754001
- 141 133811 1260754001
- 142 133821 1260754001
- 143 133831 1260754001
- 144 133841 1260754001
- 145 133851 1260754001
- 146 133861 1260754001
- 147 133871 1260754001
- 148 133881 1260754001
- 149 133891 1260754001
- 150 133901 1260754001
- 151 133911 1260754001
- 152 133921 1260754001
- 153 133931 1260754001
- 154 133941 1260754001
- 155 133951 1260754001
- 156 133961 1260754001
- 157 133971 1260754001
- 158 133981 1260754001
- 159 133991 1260754001
- 160 134001 1260754001

```

NEXT PROCEDURE: SE END OF INFO TABLE

"OLD END OF INFO TABLE

```

N, A = 807
N, SUBC (:ERRORM)
G, ERRORECUS, P
V, GOTO (: ENDRUN)

```

"NUMBER OF PROCEDURES TO INSERT

```

S - LVAR
END OF TRAFOTABLE + S
END OF CROSTABLE + S
A = BEGIN OF TEXTARRAY
BEGINSAFE = A
S = MA
WORDSAFE = S
S = 1
SMIFSAFE = S
F = LVAR
MC = F
WANTED = -E
A = BASE 2
BASE = A
A = 2
COMPMODE = A
G = LIBR CELL
A = D18
MC(?) = A

```

"VARIABLE TO COUNT PROCEDURES
" AND VARIABLE TO POINT INTO CATALOGUE
"NO COMPILING OF LINENUMBERS

```

A = 2
COMPMODE = A
G = LIBR CELL
A = D18
MC(?) = A

```

"INDICATE INSERTING
"ROUTINES AUTHORIZED
"TO DRUM

```

MC = S
SUBC (:CLEAR CRSTAB)
SUBC (:PRESCAN 0)
A = INSTR CNTR
A = -1
A = -START OF CONST LIST
S = MIB-1)
MS(1) = A
SUBC (:COMPLETE ADM1)
A = INSTR CNTR
A = -START OF CONST LIST
S = MC(-1)
A = MS
S = END OF INFO TABLE
NS = A
A = DR LBR END, P
V, A = R11
V, SUBC (:ERRORM)
V, GOTO (: ENDRUN)
END OF DRMLIBRARY = A
A = 1
PLUSA(MIB-2))
A + BEGIN OF CROSTABLE
MA(1) + S
S = A
A = -START OF CONST LIST
A + TRANSPOSITION
SUBC (:PREPARE)
SUBC (: READ LIBRARY ROUTINE)
SUBC (:INFCRM PROGRAMMER)
G = MIB-1)
SUBC (:NXT PP)
MIB-1) = A
N, GOTO (: NEXT PROCEDURE)

```

"LENGTH WITHOUT
"ADMINISTRATION
"INTO INFO TABLE

"LENGTH, INCLUDING ADMINISTRATION
"RESTORE OLD END OF INFO TABLE
"FIRST FREE PLACE ON
"DRUM AT THE END
"OF THE INFO TABLE

"SPACE
"ON DRUM
"EXHAUSTED

"NUMBER OF ROUTINES

"TO DRUM
" POINTER INTO CATALOGUE

260.7. - 502

232

```

10636 '033410' '620316207'
10637 '033411' '562734342'
10638 '033412' '562733746'
10639 '033413' '220300413'
10640 '033414' '466374411'
10641 '033415' '562733756'
10642 '033416' '562733770'
10643 '033417' '522003346'
10644 '033418'
10645 '033419'
10646 '033420'
10647 '033421' '126072400'
10648 '033422' '362077777'
10649 '033423' '222000003'
10650 '033424' '072074377'
10651 '033425' '000000072'
10652 '033426' '311100505'
10653 '033427' '526075377'
10654 '033428'
10655 '033429'
10656 '033430' '122000003'
10657 '033431' '160100510'
10658 '033432' '022000000'
10659 '033433' '066074377'
10660 '033434' '562417434'
10661 '033435' '562417424'
10662 '033436' '522233560'
10663 '033437' '261117710'
10664 '033438' '261317714'
10665 '033439' '562733442'
10666 '033440' '512000006'
10667 '033441'
10668 '033442'
10669 '033443'
10670 '033444'
10671 '033445'
10672 '033446'
10673 '033447'
10674 '033448'
10675 '033449'
10676 '033450'
10677 '033451'
10678 '033452'
10679 '033453'
10680 '033454'
10681 '033455'
10682 '033456'
10683 '033457'
10684 '033458'
10685 '033459'
10686 '033460'
10687 '033461'
10688 '033462'
10689 '033463'
10690 '033464'
10691 '033465'
10692 '033466'
10693 '033467'
10694 '033468'
10695 '033469'

```

G= ERRONEOUS, P
 N, SUBC (:ANONYMIZE)
 N, SUBC (:CLEAR CRSTAB)
 N, S = PROGRAM
 N, MS(9) = B
 N, SUBC (:UPDATE CROSSTABLE)
 N, SUBC (:UPDATE LIBRARY)
 GOTO(: ENDRUN)
 'END' EXTEND LIBRARY

NXT PP:
 S = MG
 S ' * ' 32767
 Muls(3)
 A = - : MS(-1)
 A + G
 U, A- END OF CATALOGUE, Z
 GOTO(MC(-1))

LIBRARY ROUTINE:
 S=3
 PLUS(END OF INFOTABLE)
 A=0
 MS(-1)= A
 SUBC (:INIT POINTER)
 SUBC (: NEXT ENTRY)
 GOTO(: INIT ADM12)
 U, A ' * ' D20, Z
 N, A ' * ' D24, Z
 N, SUBC (: FILL INFOTABLE)
 JUMP(-6)

FILL INFOTABLE: A= END CF INFOTABLE
 U, A= A(-1), Z
 N, A= 1
 JUMP(4)
 PLUS(END OF INFOTABLE)
 LUS(13)
 MAT(-2) + S
 S= 0
 MAT(-1)= S
 A- INFC TAP END, P
 V, A= 806
 GOTO(:ERN)
 V, GOTO(MC(-1))

DC ADI :
 'BEGIN' NEWWORD, LOOP1, LOOP2
 SUBC (:PAR PART)
 S-1
 S + INSTR CNTR
 MC = S
 MC = A
 MC = F
 A= S
 S-OLD INSTR CNTR
 OLD INSTR CNTR= A
 G= APOS
 ASHIFT - S, Z
 N, A = ASHIFT, P
 A = MG

") IF ALREADY A NUMBER IN WORD
 ") THEN INSERT CURRENT NUMBER
 ") AFTER SHIFT AND MAKE
 ") NEXT ENTRY ZERO
 ") ELSE INSERT NUMBER
 " END OF COMPILATION
 " PAR PART IN S
 " POINTS TO WORD TO BE MODIFICATED
 " COPY IN A
 " NUMBER OF WORDS NOT TO BE MODIFICATED
 " POINTS TO ADM1WORD
 ") IS THIS WORD ENOUGH?

260271 - 502

233

10696	'033474'	'100317216'	N, S + ASHIFT	"NUMBER OF SHIFTS IN S
10697	'033475'	'100033520'	S + LUAVAR	"
10698	'033476'	'570400074'	DO(S)	"SHIFT S
10699	'033477'	'002200001'	Y, A + 1	"ADMIWORD
10700	'033500'	'066072400'	Mg = A	"RESTORE REGISTERS
10701	'033501'	'562733506'	N, SUBC(NEWWORD)	
10702	'033502'	'626075376'	P = MC(-2)	
10703				
10704	'033503'	'026075377'	A = MC(-1)	
10705	'033504'	'126075377'	S = MC(-1)	
10706	'033505'	'526475377'	GOTOR(MC(-1))	"EXIT
10707				
10708	'033506'	'122000032'	S = 26	
10709	'033507'	'141017216'	ASHIFT + S, Z	"THIS WORD
10710	'033510'	'020717216'	N, A = ASHIFT, P	"ENOUGH?
10711	'033511'	'602000001'	F + 1	
10712	'033512'	'720117217'	APOS = G	"INCREASE POINTER
10713	'033513'	'022000001'	Y, A = 1	
10714	'033514'	'022300000'	N, A = 0	
10715	'033515'	'066072400'	Mg = A	
10716	'033516'	'526675377'	Y, GOTOR(MC(-1))	"ADMIWORD
10717	'033517'	'522033507'	GOTO(NEWWORD(1))	
10718				
10719	'033520'	'660000040'	LUAVAR:	
10720			LUAVAR:	
10721	'033521'	'130016215'	S = - INSTR CNTR	"ACTUAL
10722	'033522'	'110000436'	S - TRANSPOSITION	"COPY
10723	'033523'	'630160074'	G = - S	"RESERVE PLACE
10724	'033524'	'100016574'	S + NLP	"FOR ADMIWORDS
10725	'033525'	'022000000'	A = 0	
10726	'033526'	'302000033'	DIVAS(27)	
10727	'033527'	'102000001'	S + 1	
10728	'033530'	'140016215'	INSTR CNTR + S	
10729	'033531'	'020000434'	A = BEGIN CF PR AR	
10730	'033532'	'100000073'	S + A	
10731	'033533'	'160000476'	START OF CONST LIST = S	"ACTUAL
10732	'033534'	'000000436'	A + TRANSPOSITION	"
10733	'033535'	'100000436'	S + TRANSPOSITION	"NUMBER OF CONSTANTS
10734	'033536'	'060017215'	BEG OF ADM = A	"AND FIRST WORDS OF-
10735	'033537'	'611100073'	G - A, Z	"OBJECTPROGRAM INTO COUNT
10736	'033540'	'502200010'	Y, JUMP(8)	
10737	'033541'	'720100077'	COUNT = G	
10738	'033542'	'000000072'	A + G	
10739	'033543'	'100000072'	S + G	
10740	'033544'	'012000001'	A - 1	"PLACE
10741	'033545'	'112000001'	S - 1	"CONSTANT LIST
10742	'033546'	'626173400'	G = MA	"BEFORE
10743	'033547'	'726174400'	MS = G	"OBJECTPROGRAM
10744	'033550'	'556433544'	REP(LOOP1)	
10745	'033551'	'022000032'	A = 26	"INITIALIZE ASHIFT
10746	'033552'	'060017216'	ASHIFT = A	"INITIALIZE APOS
10747	'033553'	'020017215'	A = BEG CF ADMI	"FIRST ADMIWORD
10748	'033554'	'060017217'	APOS = A	"EXIT INIT ADMI AND LIBRARY ROUTINE
10749	'033555'	'122000000'	S = 0	
10750	'033556'	'166073400'	MA = S	
10751	'033557'	'526475377'	GOTOR(MC(-1))	
10752				
10753	'033560'	'020016215'	INSTR CNTR	"POINTER TO
10754	'033561'	'012000001'	A - 1	"LAST
10755	'033562'	'060017220'	OLD INSTR CNTR = A	"HANDLED WORD

234

10/56	10/57	10/58	10/59	10/60	10/61	10/62	10/63	10/64	10/65	10/66	10/67	10/68	10/69	10/70	10/71	10/72	10/73	10/74	10/75	10/76	10/77	10/78	10/79	10/80	10/81	10/82	10/83	10/84	10/85	10/86	10/87	10/88	10/89	10/90	10/91	10/92	10/93	10/94	10/95	10/96	10/97	10/98	10/99	10000	10001	10002	10003	10004	10005	10006	10007	10008	10009	10010	10011	10012	10013	10014	10015	10016																																																																																																																																																																																																																															
033563	033564	033565	033566	033567	033568	033569	033570	033571	033572	033573	033574	033575	033576	033577	033578	033579	033580	033581	033582	033583	033584	033585	033586	033587	033588	033589	033590	033591	033592	033593	033594	033595	033596	033597	033598	033599	033600	033601	033602	033603	033604	033605	033606	033607	033608	033609	033610	033611	033612	033613	033614	033615	033616	033617	033618	033619	033620	033621	033622	033623	033624	033625	033626	033627	033628	033629	033630	033631	033632	033633	033634	033635	033636	033637	033638	033639	033640	033641	033642	033643	033644	033645	033646	033647	033648	033649	033650	033651	033652	033653	033654	033655	033656	033657	033658	033659	033660	033661	033662	033663	033664	033665	033666	033667	033668	033669	033670	033671	033672	033673	033674	033675	033676	033677	033678	033679	033680	033681	033682	033683	033684	033685	033686	033687	033688	033689	033690	033691	033692	033693	033694	033695	033696	033697	033698	033699	033700	033701	033702	033703	033704	033705	033706	033707	033708	033709	033710	033711	033712	033713	033714	033715	033716	033717	033718	033719	033720	033721	033722	033723	033724	033725	033726	033727	033728	033729	033730	033731	033732	033733	033734	033735	033736	033737	033738	033739	033740	033741	033742	033743	033744	033745	033746	033747	033748	033749	033750	033751	033752	033753	033754	033755	033756	033757	033758	033759	033760	033761	033762	033763	033764	033765	033766	033767	033768	033769	033770	033771	033772	033773	033774	033775	033776	033777	033778	033779	033780	033781	033782	033783	033784	033785	033786	033787	033788	033789	033790	033791	033792	033793	033794	033795	033796	033797	033798	033799	033800	033801	033802	033803	033804	033805	033806	033807	033808	033809	033810	033811	033812	033813	033814	033815	033816	033817	033818	033819	033820	033821	033822	033823	033824	033825	033826	033827	033828	033829	033830	033831	033832	033833	033834	033835	033836	033837	033838	033839	033840	033841	033842	033843	033844	033845	0

Address	Hex	Assembly	Symbol	Comment
0010	033651	NSP NAME:	S - 0, P	" PARAMETER > 0?
0017	033652		N, A = CODEBODY, P	" ~ NOT IN CODEBODY ?
0018	033653		Y, A = M(B - 2)	" MACRO
0019	033654		Y, GOTO (:DO ADMI)	" ADMINISTRATION
0020	033655		GOTOR (MC(-1))	
0021	033656	ADR OP:	U, A ' 4, Z	
0022	033657		GOTO(:INVERT)	
0023	033657			
0024	033660	CMP TL3CM1:	A = 802	
0025	033661		SUBC(:ERRCRM)	
0026	033662		GOTO(:CMPTL3)	
0027	033662			
0028	033663	DECLST3CM1:	SUBC (:DISP LVL)	" THIS PART TO BE CARRIED OUT
0029	033664		S - 1, Z	" ONLY ONCE FOR THE WHOLE PROGRAM
0030	033665		SUBC(:DEC LST3)	
0031	033666		GOTOR (MC(-1))	
0032	033667		S = LAST SYMBOL	
0033	033671		S - 105, Z	
0034	033672		A=803	" LAST SYMBOL = END?
0035	033673		SUBC(:ERRCRM)	
0036	033673		GOTOR(MC(-1))	
0037	033674			
0038	033675	PROGRAM3CM1:	U, S-104, Z	" LAST SYMBOL = BEGIN?
0039	033676		A = 801	
0040	033677		SUBC(:ERRCRM)	
0041	033678		GOTO(:PROGRAM3)	
0042	033679		SUBC (:NEXT SBL)	
0043	033701	PROGRAM2CM2:	SUBC (: DECL LS)	
0044	033702		SUBC (: ENTR BLK)	
0045	033703		GOTO (:PR DEC)	

26071 - 502

236

FOUND LIBRARY: 'BEGIN' FOUND CROSSTABLE, D23M3

S = PERMANENT CATALOGUE END
END OF CATALOGUE = S

G = BEGIN OF INFOTABLE

S = 0
MG(1) = S

S = DR LIER BEG

END OF DRUMLIBRARY = S

G + 4
END OF INFOTABLE = G
MG = SS = NBR OF PERMANENT ROUTINES
NBR OF ROUTINES = SS + 1
S = BEGIN OF TRAFOTABLE

END OF TRAFOTABLE = S

S + MAX NBR OF ROUTINES
END OF CROSSTABLE = S

G + D23M3

FCUND CROSSTABLE:

S - 1
MS = G

U, S - BEGIN OF CROSSTABLE, Z

N, GOTO(:FCUND CROSSTABLE)

GOTO(:UPDATE LIBRARY)

D23M3:

('040 000 000' - 3)

'END' FOUND LIBRARY

CLEAR CRSTAB SPACE:

A = D24

S = BEGIN OF CROSSTABLE

S + MAX NBR OF ROUTINES

S - 1

U, S - END OF CROSSTABLE, Z

N, MS = A

N, JUMP (-4)

A + END OF INFOTABLE

MS = A

CLEAR CRSTAB: 'BEGIN' LOOP, D21 PLUS D20

SUBC(:INIT POINTER)

SUBC(:NEXT ENTRY)

GOTOR(MC(-1))

A*:-D21 PLUS D20

S + BEGIN OF CROSSTABLE

MS = A

GOTO(:LOOP)

D21 PLUS D20: '014 000 000'

10851	108	'120000474'	
10852	'033704'	'160000505'	
10853	'033705'	'620100503'	
10854	'033706'	'222000000'	
10855	'033707'	'166072403'	
10856	'033710'		
10857	'033711'	'120000467'	
10858	'033712'	'160000511'	
10859	'033713'	'602000004'	
10860	'033714'	'720100510'	
10861	'033715'	'166072400'	
10862	'033716'	'120000471'	
10863	'033717'	'160000504'	
10864	'033720'	'202000001'	
10865	'033721'	'100000500'	
10866	'033722'	'160000506'	
10867	'033723'	'200000472'	
10868	'033724'	'160000507'	
10869	'033725'	'600133733'	
10870	'033726'	'112000001'	
10871	'033727'	'726174400'	
10872	'033730'	'111100502'	
10873	'033731'	'522533726'	
10874	'033732'	'522033770'	
10875	'033733'	'037117775'	
10876			
10877	'033734'		
10878	'033735'		
10879	'033736'		
10880	'033737'		
10881	'033740'		
10882	'033741'		
10883	'033742'		
10884	'033743'		
10885	'033744'		
10886	'033745'		
10887	'033746'		
10888	'033747'		
10889	'033748'		
10890	'033749'		
10891	'033750'		
10892	'033751'		
10893	'033752'		
10894	'033753'		
10895	'033754'		
10896	'033755'		
10897	'033756'		
10898	'033757'		
10899	'033758'		
10900	'033759'		
10901	'033760'		
10902	'033761'		
10903	'033762'		
10904	'033763'		
10905	'033764'		
10906	'033765'		
10907	'033766'		
10908	'033767'		
10909	'033768'		
10910	'033769'		

260.71 - 502

236

```

2071 '034035' : '660000146' : ELOOP1:
2072 '034036' : '263000077' :
2073 '034037' : '062373377' :
2074 '034040' : '166375403' :
2075 '034041' : '120300073' :
2076 '034042' : '502710251' :
2077 '034043' : '126375377' :
2078 '034044' : '56334035' :
2079 '034045' : '526475377' :
2080
2081 110
2082
2083 111
2084
2085 '034046' : '0660/5400' :
2086 '034047' : '562434131' :
2087 '034050' : '562420030' :
2088 '034051' : '113100151' :
2089 '034052' : '522234126' :
2090
2091 '034053' : '1220017614' :
2092 '034054' : '020016572' :
2093 '034055' : '113100002' :
2094 '034056' : '5223354073' :
2095 '034057' : '102000001' :
2096 '034060' : '160017225' :
2097 '034061' : '012000001' :
2098 '034062' : '060017224' :
2099 '034063' : '122000150' :
2100 '034064' : '760000060' :
2101 '034065' : '160017226' :
2102 '034066' : '126073401' :
2103 '034067' : '160017230' :
2104 '034070' : '122000151' :
2105 '034071' : '166073401' :
2106 '034072' : '522034126' :
2107
2108 '034073' : '113100001' : ELSE0:
2109 '034074' : '522334113' :
2110 '034075' : '122000002' :
2111 '034076' : '160017225' :
2112 '034077' : '012000001' :
2113 '034100' : '060017224' :
2114 '034101' : '126073400' :
2115 '034102' : '370017232' :
2116 '034103' : '102006400' :
2117 '034104' : '160017226' :
2118 '034105' : '126073400' :
2119 '034106' : '36007234' :
2120 '034107' : '100034112' :
2121 '034110' : '166073400' :
2122 '034111' : '522034126' :
2123 '034112' : '032200000' :
ENDSBL:

```

END INFORM PROGRAMMER

TRUNCATE: 'BEGIN' ELSE0, ELSE1, END, ENDSBL

MC = A
SUBC(: SAVE NXT SBL)
SUBC(: NXT SBL)
U, S - 105, Z
V, GOTO(:END)

S = SHIFT
A = TEXT ARRAY POINTER
U, S - 2, Z
N, GOTO(:ELSE0)
S + 1

SHIFTSAFE = S
A - 1
BEGINSAFE = A
S = 104

LUS(16)
WORDSAFE = S
S = MAT(1)
WORDSAFE = S
S = 105
MAT(1) = S
GOTO(:END)

" SIMULATE TEXT ARRAY BEGINNING WITH
") BEGIN SYMBOL FOR NEXT PROCEDURE
")
") SAVE SECOND WORD OF TEXT FOR
") NEXT PROCEDURE
" TRUNCATE TEXT BY INSERTING OF END SYMBOL

U, S - 1, Z
N, GOTO(:ELSE1)
S = 2

SHIFTSAFE = S
A - 1
BEGINSAFE = A
S = MA

S '*' - FRAME
S + 26624
WORDSAFE = S
S = MA
S '*' 016, M1
S + ENDSBL
MA = S
GOTO(:END)
+ 6881280

" SHIFT TO POSITION
" CLEAN CHARACTER, DONE?
" CORRECT CODE
" SAVE S
" PRINT CHARACTER
" RESTORE S
" NEXT CHARACTER, IF ANY

" TEXT(BEGINSAFE)
" MAKE PLACE FOR BEGIN SYMBOL
" INSERT BEGIN SYMBOL
" TEXT(BEGINSAFE)

```

260071 - 502
239

11028 '034113': '060017224' ELSE1:
11029 '034114': '122000001'
11030 '034115': '160017225'
11031 '034116': '126073400' S = MA
11032 '034117': '372000377' S 'v' - 295
11033 '034120': '102000050' S + 104
11034 '034121': '160017226' WORDSAFE = S
11035 '034122': '126073400' S = MA
11036 '034123': '370017232' S '+-FRAME
11037 '034124': '102064400' S + 2680
11038
11039 '034125': '166073400' MA = S
11040 '034126': '562434144' SUBC (:RESTORE NXT SBL)
11041 '034127': '026075377' A = MC(-1)
11042 '034130': '526475377' GOTOR(MC(-1))
11043
11044
11045
11046 '034131': '120016212' 'END'TRUNCATE
11047 '034132': '166075400' S = LAST SYMBOL
11048 '034133': '120017610' MC = S
11049 '034134': '166075400' S = STOCK1
11050 '034135': '120017613' MC = S
11051 '034136': '166075400' S = QUOTE COUNTER
11052 '034137': '120016212' MC = S
11053 '034140': '166075400' S = TEXT ARRAY POINTER
11054 '034141': '120017614' MC = S
11055 '034142': '166075400' S = SHIFT
11056 '034143': '524437772' MC = S
11057
11058 '034144': '126075377' GOTOR (MC(-6))
11059 '034145': '164037772' RESTORE NXT SEL:S = MC(-1)
11060 '034146': '126075377' M1 B -6) = S
11061 '034147': '120017614' S = MC(-1)
11062 '034150': '126075377' SHIFT = S
11063 '034151': '160016212' S = MC(-1)
11064 '034152': '126075377' TEXT ARRAY POINTER = S
11065 '034153': '160017613' QUOTE COUNTER = S
11066 '034154': '126075377' S = MC(-1)
11067 '034155': '160017610' STOCK1 = S
11068 '034156': '126075377' S = MC(-1)
11069 '034157': '160016212' LAST SYMBOL = S
11070 '034160': '526475377' GOTOR (MC(-1))

" ) SAVE LINK
")

```

260271 - 502

240

```

11075      112      562423337      : INSPECT DECL:
11076      034161:      113000001      'BEGIN' LOOP, LOOP1, PROC END, END0, END1
11077      034162:      526775377      S = 1, Z
11078      034163:      120017622      N, GOTOR(MC(-1))
11079      034164:      770000063      S = CHARACTER
11080      034165:      112500031      RUS(19)
11081      034166:      113700017      U, S - 25, P
11082      034167:      022201444      N, S - 15, E
11083      034170:      522203560      Y, A = 804
11084      034171:      562434131      GOTO(:ERRCRM)
11085      034172:      562420030      SUBC(:SAVE NXT SBL)
11086      034173:      562434276      SUBC(:HANDLE QM)
11087      034174:      121017616      S = LETTER LAST SYMBOL, Z
11088      034175:      522334274      N, GOTO(:END1)
11089      034176:      562435157      SUBC(:IDF3)
11090      034177:      166075400      MC = S
11091      034200:      562422144      SUBC(:NAME IN LIBRARY)
11092      034201:      022301445      N, A = 805
11093      034202:      562703360      N, SUBC(:ERRCRM)
11094      034203:      124037777      S = MIB-1)
11095      034204:      126074577      S = MS(-1)
11096      034205:      362077777      S = 32767
11097      034206:      100017632      Muls(4)
11098      034207:      100000473      S + WORD COUNT
11099      034210:      110400305      S + CAT END
11100      034211:      022201451      S - END OF CATALOGUE, P
11101      034212:      562603560      Y, A = 809
11102      034213:      522234273      Y, SUBC(:ERRCRM)
11103      034214:      022000001      G, GOTO(:END0)
11104      034215:      040000304      A = 1
11105      034216:      022000001      NBR OF ROUTINES + A
11106      034217:      064040300      A + WORD COUNT
11107      034220:      020016374      MIB) = A
11108      034221:      126073400      A = NLP
11109      034222:      166072400      S = MA
11110      034223:      312000001      MG = S
11111      034224:      632000001      A - 1
11112      034225:      122000001      G - 1
11113      034226:      125040000      S = 1
11114      034227:      522334224      M(B) - S, Z
11115      034228:      024037777      N, GOTO(:LOOP)
11116      034229:      126073400      A = MIB-1)
11117      034230:      372077777      S = MA
11118      034231:      32767      S = 32767
11119      034232:      166072400      S = NBR OF ROUTINES
11120      034233:      126073377      MG = S
11121      034234:      113000001      S = MA(-1)
11122      034235:      362077777      S = 32767
11123      034236:      164040000      M(B) = S
11124      034237:      100017233      S + LOCAL NUMBER
11125      034240:      166072377      MG(-1) = S
11126      034241:      012000007      A - 7
11127      034242:      522234267      Y, GOTO(:PROC END)
11128      034243:      132000001      S = -1
11129      034244:      612000001      G - 1
11130      034245:      612000001      " )222 000 001 + FORMAL COUNT
11131      034246:      612000001      " ) INTO CATALOGUE
11132      034247:      612000001      " )222 000 001 + FORMAL COUNT
11133      034250:      612000001      " ) INTO CATALOGUE
11134      034251:      612000001      " ) SIMULATE LETTERS

```

260471 - 502

241

```

11.55 '034252': '166072377'
11.56 '034253': '126473376'
11.57 '034254': '012000001'
11.58 '034255': '512300003'
11.59 '034256': '370000422'
11.60 '034257': '166072376'
11.61 '034258': '122000000'
11.62 '034259': '166072375'
11.63 '034260': '612000002'
11.64 '034261': '012000001'
11.65 '034262': '122000001'
11.66 '034263': '550400000'
11.67 '034264': '522334250'
11.68 '034265': '122000000'
11.69 '034266': '166072376'
11.70 '034267': '122000000'
11.71 '034268': '166072375'
11.72 '034269': '612000003'
11.73 '034270': '720100505'
11.74 '034271': '412000001'
11.75 '034272': '562444144'
11.76 '034273': '526475377'
11.77 '034274': 'END0:'
11.78 '034275': 'END1:'

```

```

MC[-1] = S
S = MAT[-2], P
A - 1
N, JUMP(-3)
S, *- D18M1
MC[-2] = S
S = 0
MC[-3] = S
G - 2
A - 1
S = 1
M[B] - S, Z
GOTO(:LOOP1)
N,
S = 0
MC[-2] = S
G - 3
END OF CATALOGUE = G
A - 1
SUBC (:RESTORE NXT SBL)
GOTOR(MC[-1])
END, INSPECT DECL

```

```

")
") SKIP LETTERS
") CHARACTER OF FORMAL
") INSERT ZERO AT THE END OF THE PROCEDURE
") THROW AWAY N OF PROC IDENTIFIER

```

```

11.55 '034275': '526475377'
11.56 '034276': 'END0:'
11.57 '034277': 'END1:'
11.58 '034278': 'END2:'
11.59 '034279': 'END3:'

```

```

HANDLE GM:
'BEGIN' INSTR

```

```

11.60 '034280': '120016212'
11.61 '034281': '113100172'
11.62 '034282': '526775377'
11.63 '034283': '120000504'
11.64 '034284': '120000001'
11.65 '034285': '120000502'
11.66 '034286': '020017715'
11.67 '034287': '046074400'
11.68 '034288': '020016572'
11.69 '034289': '020117514'
11.70 '034290': '613000001'
11.71 '034291': '012000001'
11.72 '034292': '126073400'
11.73 '034293': '066075400'
11.74 '034294': '022034326'
11.75 '034295': '600100073'
11.76 '034296': '576472400'
11.77 '034297': '166075400'
11.78 '034298': '122000167'
11.79 '034299': '576472403'
11.80 '034300': '106075577'
11.81 '034301': '026075377'
11.82 '034302': '166073400'
11.83 '034303': '522020030'
11.84 '034304': '360017234'
11.85 '034305': '370017232'
11.86 '034306': '370017232'
11.87 '034307': '760000040'
11.88 '034308': '760000040'
11.89 '034309': '760000050'
11.90 '034310': '113'
11.91 '034311': '562421337'
11.92 '034312': '113000001'
11.93 '034313': '120216212'

```

```

S = LAST SYMBOL
U, S - 122, Z
N, GOTOR (MC[-1])
S + 1
S = NBR OF ROUTINES
S + BEGIN OF CROSSTABLE
A = D25
MS+ A
A = TEXT ARRAY POINTER
G = SHIFT
G - 1, Z
A - 1
S = MA
MC = A
A = : INSTR
G + A
DO (MG)
MC = S
S = 119
DO (MG(3))
S + MC[-1]
A = MC[-1]
MA = S
GOTO (:NXT SBL)
S, *- D16M1
S, *- 255
S, *- FRAME
LUS (16)
LUS (0)
LUS (8)
END, HANDLE GM

```

```

") QUESTIONMARK?
") INDICATION
") INTO
") CROSSTABLE
") TEXT ARRAY POINTER -1
") TEXT

```

```

11.60 '034280': '120016212'
11.61 '034281': '113100172'
11.62 '034282': '526775377'
11.63 '034283': '120000504'
11.64 '034284': '120000001'
11.65 '034285': '120000502'
11.66 '034286': '020017715'
11.67 '034287': '046074400'
11.68 '034288': '020016572'
11.69 '034289': '020117514'
11.70 '034290': '613000001'
11.71 '034291': '012000001'
11.72 '034292': '126073400'
11.73 '034293': '066075400'
11.74 '034294': '022034326'
11.75 '034295': '600100073'
11.76 '034296': '576472400'
11.77 '034297': '166075400'
11.78 '034298': '122000167'
11.79 '034299': '576472403'
11.80 '034300': '106075577'
11.81 '034301': '026075377'
11.82 '034302': '166073400'
11.83 '034303': '522020030'
11.84 '034304': '360017234'
11.85 '034305': '370017232'
11.86 '034306': '370017232'
11.87 '034307': '760000040'
11.88 '034308': '760000040'
11.89 '034309': '760000050'
11.90 '034310': '113'
11.91 '034311': '562421337'
11.92 '034312': '113000001'
11.93 '034313': '120216212'

```

```

SUBC(: DISP LVL)
S - 1, Z
Y, S = LAST SYMBOL

```

```

") REPLACE IT BY NLCR
") INTO TEXTARRAY
") SKIP QUESTIONMARK
") REMOVE QUESTIONMARK

```

```

11.60 '034280': '120016212'
11.61 '034281': '113100172'
11.62 '034282': '526775377'
11.63 '034283': '120000504'
11.64 '034284': '120000001'
11.65 '034285': '120000502'
11.66 '034286': '020017715'
11.67 '034287': '046074400'
11.68 '034288': '020016572'
11.69 '034289': '020117514'
11.70 '034290': '613000001'
11.71 '034291': '012000001'
11.72 '034292': '126073400'
11.73 '034293': '066075400'
11.74 '034294': '022034326'
11.75 '034295': '600100073'
11.76 '034296': '576472400'
11.77 '034297': '166075400'
11.78 '034298': '122000167'
11.79 '034299': '576472403'
11.80 '034300': '106075577'
11.81 '034301': '026075377'
11.82 '034302': '166073400'
11.83 '034303': '522020030'
11.84 '034304': '360017234'
11.85 '034305': '370017232'
11.86 '034306': '370017232'
11.87 '034307': '760000040'
11.88 '034308': '760000040'
11.89 '034309': '760000050'
11.90 '034310': '113'
11.91 '034311': '562421337'
11.92 '034312': '113000001'
11.93 '034313': '120216212'

```

```

SKIP GM:
SUBC(: DISP LVL)
S - 1, Z
Y, S = LAST SYMBOL

```

```

") DISPLAY LEVEL = 1?

```

```

26071 - 502
1195 '034337': '113200172'
1196 '034340': '522220030'
1197 '034341': '526475377'
1198
1199
1200
1201 '034342': '120000505'
1202 '034343': '032000002'
1203 '034344': '066074400'
1204 '034345': '120000474'
1205 '034346': '562421211'
1206 '034347': '013100002'
1207 '034350': '526675377'
1208 '034351': '013100001'
1209 '034352': '112200001'
1210 '034353': '522234344'
1211 '034354': '166075400'
1212 '034355': '112000001'
1213 '034356': '026474400'
1214 '034357': '512500003'
1215 '034360': '166075400'
1216 '034361': '262077777'
1217 '034362': '000000502'
1218 '034363': '126073400'
1219 '034364': '36117715'
1220 '034365': '370317715'
1221 '034366': '166373400'
1222 '034367': '126075377'
1223 '034370': '026074400'
1224 '034371': '240317714'
1225 '034372': '076374400'
1226 '034373': '126075400'
1227 '034374': '522034345'
1228
1229
1230
1231
1232
1233 '012633': '020500512'
1234 '012634': '522330306'
1235 '012635': '020000437'
1236 '012636': '120017711'
1237 '012637': '040000451'
1238 '012640': '040000436'
1239 '012641': '140003056'
1240 '012642': '522003036'
1241
1242
1243 '012643': '020500512'
1244 '012644': '522203036'
1245 '012645': '030000437'
1246 '012646': '130017711'
1247 '012647': '522012637'
1248
1249
1250
1251
1252
1253 '012650': '020000451'
1254 '012651': '060012674'
1255 '012652': '020000470'
1256

```

242

```

V, S - 122, 2
V, GOTO (INXT 88L)
V, GOTO (MC(-1))

ANONYMIZE:
'BEGIN' LOOP

S = END OF CATALOGUE
A = - 2
MS = A
S = PERMANENT CATALOGUE END
SUBC (:NXT PNR)
U, A - 2, Z
V, GATOR (MC(-1))
U, A - 1, Z
V, S - 1
V, GOTO (:LOOP)
MC = S
S - 1
A = MS, P
N, JUMP (-3)
MC = S
A, * 32767
A + BEGIN CF CROSTABLE
S = MA
U, S, * D25, Z
N, S, * -D25
N, MA = S
S = MC(-2)
A = MS
N, A, * D24
N, MS = - A
S = MC
GOTO (:LOOP)
'END' ANONYMIZE

SYSTEM(-37):
START EXT LIB:
U, A = NORMAL, P
N, GOTO (:DNTST)
N, A = LENGTH OF EXT LIB
S = D21
COMPCELL[1]+A
TRANSPPOSITION+A
DNTST[16]+S
GOTO (:DNTST)

FINISH EXT LIB:
U, A = NORMAL, P
V, GOTO (:DNTST)
Y, A = -LENGTH OF EXT LIB
S = D21
GOTO (:START EXT LIB[4])

MILLI TC TAPE: 'BEGIN'
DIRECTION TO PRUH, MILLI FROM TAPE,
INSTR1, INSTR2, INSTR, SAVE
A = COMP CELL[1]
SAVE = A
A = DR LIBR END

```

" AND QUESTIONMARK?
" SKIP IT

" MARK
" END OF
" CATALOGUE

" ENDMARKER?

" FORMAL IDENTIFIER?
" DECREASE PARAMETER FOR NXT PNR

" ADDRESS OF FIRST LETTERS

" N
" NUMBER OF PROCEDURE

" NO INDICATION IN CROSTABLE?
" REMOVE
" INDICATION

" FIRST LETTERS OF PROC. NAME
" NON-EXISTING NAME
" INTO CATALOGUE
" N

" REVERSE INSTRUCTION

260c71 - 50c

243

```

11295 '012653' '06000451' 'COMP CELL(1) = A'
11296 '012654' '020003314' 'A = RD CMP(1)'
11297 '012655' '06001267' 'INSTR = A'
11298 '012656' '020012675' 'A = INSTR1'
11299 '012657' '060003314' 'RD CMP(1) = A'
11300 '012660' '522003036' 'GOTO(:DNTST)'
11301
11302
11303
11304
11305
11306
11307
11308 '012661' '02000421' 'DIRECTION TO DRUM:'
11309 '012662' '06000452' 'A = D16'
11310 '012663' '020012676' 'COMP CELL(2) = A'
11311 '012664' '060003314' 'A = INSTR2'
11312 '012665' '06006100' 'RD CMP(1) = A'
11313 '012666' '02607340' 'DVOFF'
11314 '012667' '02607340' 'A = MA'
11315
11316
11317
11318
11319
11320
11321
11322
11323
11324
11325
11326
11327
11328
11329
11330
11331
11332
11333
11334
11335
11336
11337
11338
11339
11340
11341
11342
11343
11344
11345
11346
11347
11348
11349
11350
11351
11352
11353
11354
11355
11356
11357
11358
11359
11360
11361
11362
11363
11364
11365
11366
11367
11368
11369
11370
11371
11372
11373
11374
11375
11376
11377
11378
11379
11380
11381
11382
11383
11384
11385
11386
11387
11388
11389
11390
11391
11392
11393
11394
11395
11396
11397
11398
11399
11400
11401
11402
11403
11404
11405
11406
11407
11408
11409
11410
11411
11412
11413
11414
11415
11416
11417
11418
11419
11420
11421
11422
11423
11424
11425
11426
11427
11428
11429
11430
11431
11432
11433
11434
11435
11436
11437
11438
11439
11440
11441
11442
11443
11444
11445
11446
11447
11448
11449
11450
11451
11452
11453
11454
11455
11456
11457
11458
11459
11460
11461
11462
11463
11464
11465
11466
11467
11468
11469
11470
11471
11472
11473
11474
11475
11476
11477
11478
11479
11480
11481
11482
11483
11484
11485
11486
11487
11488
11489
11490
11491
11492
11493
11494
11495
11496
11497
11498
11499
11500
11501
11502
11503
11504
11505
11506
11507
11508
11509
11510
11511
11512
11513
11514
11515
11516
11517
11518
11519
11520
11521
11522
11523
11524
11525
11526
11527
11528
11529
11530
11531
11532
11533
11534
11535
11536
11537
11538
11539
11540
11541
11542
11543
11544
11545
11546
11547
11548
11549
11550
11551
11552
11553
11554
11555
11556
11557
11558
11559
11560
11561
11562
11563
11564
11565
11566
11567
11568
11569
11570
11571
11572
11573
11574
11575
11576
11577
11578
11579
11580
11581
11582
11583
11584
11585
11586
11587
11588
11589
11590
11591
11592
11593
11594
11595
11596
11597
11598
11599
11600
11601
11602
11603
11604
11605
11606
11607
11608
11609
11610
11611
11612
11613
11614
11615
11616
11617
11618
11619
11620
11621
11622
11623
11624
11625
11626
11627
11628
11629
11630
11631
11632
11633
11634
11635
11636
11637
11638
11639
11640
11641
11642
11643
11644
11645
11646
11647
11648
11649
11650
11651
11652
11653
11654
11655
11656
11657
11658
11659
11660
11661
11662
11663
11664
11665
11666
11667
11668
11669
11670
11671
11672
11673
11674
11675
11676
11677
11678
11679
11680
11681
11682
11683
11684
11685
11686
11687
11688
11689
11690
11691
11692
11693
11694
11695
11696
11697
11698
11699
11700
11701
11702
11703
11704
11705
11706
11707
11708
11709
11710
11711
11712
11713
11714
11715
11716
11717
11718
11719
11720
11721
11722
11723
11724
11725
11726
11727
11728
11729
11730
11731
11732
11733
11734
11735
11736
11737
11738
11739
11740
11741
11742
11743
11744
11745
11746
11747
11748
11749
11750
11751
11752
11753
11754
11755
11756
11757
11758
11759
11760
11761
11762
11763
11764
11765
11766
11767
11768
11769
11770
11771
11772
11773
11774
11775
11776
11777
11778
11779
11780
11781
11782
11783
11784
11785
11786
11787
11788
11789
11790
11791
11792
11793
11794
11795
11796
11797
11798
11799
11800
11801
11802
11803
11804
11805
11806
11807
11808
11809
11810
11811
11812
11813
11814
11815
11816
11817
11818
11819
11820
11821
11822
11823
11824
11825
11826
11827
11828
11829
11830
11831
11832
11833
11834
11835
11836
11837
11838
11839
11840
11841
11842
11843
11844
11845
11846
11847
11848
11849
11850
11851
11852
11853
11854
11855
11856
11857
11858
11859
11860
11861
11862
11863
11864
11865
11866
11867
11868
11869
11870
11871
11872
11873
11874
11875
11876
11877
11878
11879
11880
11881
11882
11883
11884
11885
11886
11887
11888
11889
11890
11891
11892
11893
11894
11895
11896
11897
11898
11899
11900
11901
11902
11903
11904
11905
11906
11907
11908
11909
11910
11911
11912
11913
11914
11915
11916
11917
11918
11919
11920
119
```

26471 - 506

244

Line	Address	Hex	Assembly	Comment
1315	017244	020416207	A= ERRONEOUS, P	
1316	017245	522203346	GOTO(:END RUN)	
1317	017246	020065966	A= BEGIN OF PROGRAM	
1318	017247	000000436	A + TRANSPORTION	
1319	017250	060017227	PNTN= A	
1320	017251	020000436	A= TRANSPORTION	
1321	017252	012000001	A - 1	
1322	017253	060166215	PLUSA(INSTR CNTR)	
1323	017254	126073400	S=MA	
1324	017255	166075400	MC= S	
1325	017256	1000117223	A + RELADDRI	
1326	017257	562472963	SUBC(:OBSERVE)	
1327	017260	622017317	G=: LIBR CELL	
1328	017261	022000000	A= 0	
1329	017262	066072402	MG(2)= A	
1330	017263	122000002	S= 2	
1331	017264	170117227	MINS(PNTR)	
1332	017265	127074401	S= MS(1), Z	
1333	017266	522217310	GOTO(:END)	
1334	017267	100000502	S + BEGIN OF CROSTABLE	
1335	017270	020016215	A = INSTR CNTR	
1336	017271	562472952	SUBC(:PREPARE)	
1337	017272	562472952	SUBC(:READ LIBRARY ROUTINE)	
1338	017273	562472957	SUBC(:ADAPT ADR)	
1339	017274	122000001	S= 1	
1340	017275	170116215	MINS (INSTR CNTR)	
1341	017276	126074400	S= MS	
1342	017277	114567777	U, S - M(B-1), P	
1343	017300	164237777	Y, M(B-1)= S	
1344	017301	020017231	A= BEGIN OF PROC	
1345	017302	010000436	A - TRANSPORTION	
1346	017303	120017227	S= PNTR	
1347	017304	066074400	MS= A	
1348	017305	002000001	A + 1	
1349	017306	066074401	MS(1)= A	
1350	017307	522017263	GOTO(:LCOP)	
1351	017310	026075377	A= MC(-1)	
1352	017311	122016215	S= INSTR CNTR	
1353	017312	066074400	MS= A	
1354	017313	102000001	S + 1	
1355	017314	110000436	S - TRANSPORTION	
1356	017315	10016215	INSTR CNTR = S	
1357	017316	526475377	GOTOR(MC(-1))	
1358	116		END, SUBMONITOR	
1359				
1360	017317	300017317	LIBR CELL	
1361	017320		SKIP 4	
1362				
1363	017324		READ LIBRARY ROUTINE:	
1364	017324	122017317	S=: LIBR CELL	
1365	017325	562475312	SUBC(:TRANSPORT)	
1366	017326	522000345	GOTO(:ENDTRANS)	
1367				
1368	117		ADAPT ADR:	
1369				
1370	017327	620116215	G= INSTR CNTR	
1371	017330	612000002	F - 2	
1372	017331	720117220	OLD INSTR CNTR= G	
1373	017332	026072400	A= MG	
1374	017333	670000052	RVA(10)	

26071 - 502

245

11375	'017334'	'120000072'	B = G	"BEGIN OF PROGRAM OF PROCEDURE
11376	'017335'	'110000073'	S - A	"
11377	'017336'	'160017231'	BEGIN OF PROC S	"
11378	'017337'	'160016215'	INSTR CNTR = S	"
11379	'017338'	'1100000436'	S - TRANSPOSITION	"
11380	'017339'	'116072401'	S - M(11)	"
11381	'017340'	'160017413'	CORRECTION = S	"
11382	'017341'	'026072400'	A = MG	"
11383	'017342'	'026072400'	A = * 1023	"
11384	'017343'	'026072400'	G - A	"
11385	'017344'	'026072400'	MC = G	"
11386	'017345'	'026072400'	APOS = B	"
11387	'017346'	'026072400'	B = 1	"
11388	'017347'	'026072400'	INSTR CNTR = B	"
11389	'017348'	'026072400'	B = 0	"
11390	'017349'	'026072400'	ASHIFT = B	"
11391	'017350'	'026072400'	A = MG	"
11392	'017351'	'026072400'	NORA = Z	"
11393	'017352'	'026072400'	GOTO(NEXT WORD)	"
11394	'017353'	'026072400'		"
11395	'017354'	'026072400'		"
11396	'017355'	'026072400'		"
11397	'017356'	'026072400'		"
11398	'017357'	'026072400'		"
11399	'017358'	'026072400'		"
11400	'017359'	'026072400'		"
11401	'017360'	'026072400'		"
11402	'017361'	'026072400'		"
11403	'017362'	'026072400'		"
11404	'017363'	'026072400'		"
11405	'017364'	'026072400'		"
11406	'017365'	'026072400'		"
11407	'017366'	'026072400'		"
11408	'017367'	'026072400'		"
11409	'017368'	'026072400'		"
11410	'017369'	'026072400'		"
11411	'017370'	'026072400'		"
11412	'017371'	'026072400'		"
11413	'017372'	'026072400'		"
11414	'017373'	'026072400'		"
11415	'017374'	'026072400'		"
11416	'017375'	'026072400'		"
11417	'017376'	'026072400'		"
11418	'017377'	'026072400'		"
11419	'017378'	'026072400'		"
11420	'017379'	'026072400'		"
11421	'017380'	'026072400'		"
11422	'017381'	'026072400'		"
11423	'017382'	'026072400'		"
11424	'017383'	'026072400'		"
11425	'017384'	'026072400'		"
11426	'017385'	'026072400'		"
11427	'017386'	'026072400'		"
11428	'017387'	'026072400'		"
11429	'017388'	'026072400'		"
11430	'017389'	'026072400'		"
11431	'017390'	'026072400'		"
11432	'017391'	'026072400'		"
11433	'017392'	'026072400'		"
11434	'017393'	'026072400'		"

```

260271 - 502
11435 '017424': '160016215'
11436 '017425': '526475377'
11437
11438
11439 '017426': '122000001'
11440 '017427': '160117222'
11441 '017430': '026074400'
11442 '017431': '111100507'
11443 '017432': '110000502'
11444 '017433': '526075377'
11445
11446 '017434': '120000502'
11447 '017435': '160017222'
11448 '017436': '526475377'
11449
11450
11451
11452 '017437': '562417434'
11453 '017440': '562417426'
11454 '017441': '522417467'
11455 '017442': '261117710'
11456 '017443': '56217445'
11457 '017444': '522017440'
11458
11459
11460
11461
11462
11463
11464 '017445': '100000202'
11465 '017446': '066075400'
11466 '017447': '026074400'
11467
11468 '017450': '261117711'
11469 '017451': '522317465'
11470 '017452': '240017711'
11471 '017453': '066074400'
11472 '017454': '127073402'
11473 '017455': '522217465'
11474 '017456': '362017777'
11475 '017457': '562417445'
11476 '017461': '126073402'
11477 '017462': '771000055'
11478 '017463': '56217445'
11479 '017464': '002300001'
11480 '017465': '522317454'
11481 '017466': '026075377'
11482
11483
11484
11485
11486
11487
11488
11489
11490
11491
11492
11493
11494

PSEUDO LVAR:
S = 1
RELADDR = S
S + TRANSPOSITION
PLUS (INSTR CNTR)
A = 0
MS [-1] = A
SUBC (:INIT POINTER)
SUBC (:NEXT ENTRY)
V, GOTO (:END)

LOOP:
V, GOTO (:END)

INSTR CNTR= S
GOTOR(MC[-1])
'END' ADAPT ADR

S = 1
PLUS(PNTR)
A = MS
U, S - END OF CROSSTABLE, Z
S - BEGIN OF CROSSTABLE
GOTO(MC[-1])

S = BEGIN OF CROSSTABLE
PNTR = S
GOTOR(MC[-1])

'BEGIN' CRF1, CRSS, PSEUDO LVAR, LOOP, END

SUBC (:INIT POINTER)
SUBC (:NEXT ENTRY)
GOTOR(:PSEUDO LVAR)
V, A '* D20, Z
N, SUBC(:CRSS)
GOTO(:CRF1)

'BEGIN' END, CYCLE

CRSS:
S + BEGIN OF CROSSTABLE
MC = A
A = MS
A '* D21, Z
N, GOTO(:END)
A '* D 21
MS = A
S = MA121, Z
V, GOTO(:END)
S '* 8191
SUBC(:CRSS)
S = MA121
RUS(13), Z
N, SUBC(:CRSS)
N, A + 1
N, GOTO(:CYCLE)
A = MC[-1]
GOTOR(MC[-1])

'END' CRSS

" ) POINTER: = POINTER + 1
" )
" ) CONTENTS OF CROSSTABLEWORD
" ) S - END OF CROSSTABLE, Z
" ) NUMBER OF ENTRY IN S
" ) 6 INSTRUCTIONS

" ) CRSS EXPECTS IN S AN ENTRY
" ) OF THE CROSSTABLE AND BESIDES
" ) THAT IF CALLED FROM OUTSIDE IN
" ) A AN ENTRY OF THE INFOTABLE

" ) ALREADY NEEDED?
" ) THEN EXIT CRSS

" ) NOTICE PROCEDURE NEEDED
" ) ANY OTHER PROCEDURE NEEDED?
" ) IF NOT EXIT CRSS

" ) ANY OTHER PROCEDURE NEEDED?
" ) NEXT ENTRY OF INFOTABLE
" ) RESTORE ENTRY OF INFOTABLE

" ) ACTUAL INSTR CNTR
" ) LEADING ZERO OF
" ) PSEUDO LVARs

```

26071 - 502

247

```

1495 '017501' '626173401'
1496 '017502' '26117711'
1497 '017503' '261337714'
1498 '017504' '522217477'
1499 '017505' '020000074'
1500 '017506' '000000502'
1501 '017507' '060000077'
1502 '017510' '036000400'
1503 '017511' '006001400'
1504 '017512' '000037222'
1505 '017513' '010517223'
1506 '017514' '060217223'
1507 '017515' '020000072'
1508 '017516' '04017222'
1509 '017517' '022000002'
1510 '017520' '060116215'
1511 '017521' '166073376'
1512 '017522' '166073377'
1513 '017523' '100000500'
1514 '017524' '010000436'
1515 '017525' '012000002'
1516 '017526' '066074400'
1517 '017527' '522017477'
1518 '017530' '020000436'
1519 '017531' '050016215'
1520 '017532' '526475377'
1521 118
1522
1523 '017533' '026074400'
1524 '017534' '166075400'
1525 '017535' '120000073'
1526 '017536' '272077777'
1527 '017537' '362077777'
1528 '017540' '112507777'
1529 '017541' '126275377'
1530 '017542' '526675377'
1531 '017543' '066075400'
1532 '017544' '100000500'
1533 '017545' '026074400'
1534 '017546' '006075377'
1535 '017547' '126075377'
1536 '017550' '066074400'
1537 '017551' '526475377'
1538
1539 '017552' '622017317'
1540 '017553' '066072403'
1541 '017554' '160000077'
1542 '017555' '136000400'
1543 '017556' '176072404'
1544 '017557' '106001400'
1545 '017560' '166072401'
1546 '017561' '140016215'
1547 '017562' '526475377'
1548
1549 '017563' '010000436'
1550 '017564' '012577777'
1551 '017565' '526775377'
1552 '017566' '620100073'
1553 '017567' '720116215'
1554 '017570' '022000755'

G = MA (1)
U, A, '021, 2
N, A, '024, 2
Y, GOTO (:LOCP)
A = S
A + BEGIN OF CROSSTABLE
D = A
A = - MD
A + M1
A + RELADDR
U, A - RELADDR, P
Y, RELADDR1 = A
A = G
RELADDR + A
A = 2
PLUS A (INSTR CNTR)
MA(-2) = S
MA(-1) = S
S + BEGIN OF TRAFOTABLE
A - TRANSPOSITION
A - 2
MS = A
GOTO (:LCOP)
A = TRANSPOSITION
INSTR CNTR = A
GOTOR (MC(-1))
END, CRF

ADDR OF LVAR:
A = MS
MC = S
S = A
A, ' -32767
S, ' 32767
U, S - 4095, P
Y, S = MC(-1)
Y, GOTOR (MC(-1))
MC = A
S + BEGIN OF TRAFOTABLE
A = MS
A + MC(-1)
MS = MC(-1)
MS = A
GOTOR (MC(-1))

PREPARE:
G = :LIBR CELL
MG(3) = A
D = S
S = - MD
MG(4) = - S
S + M1
MG(1) = S
INSTR CNTR + S
GOTOR (MC(-1))

OBSERVE:
A - TRANSPOSITION
U, A - '77 777', P
N, GOTOR (MC(-1))
G = A
INSTR CNTR = G
A = 493

" LENGTH IN CORE
" LENGTH ON DRUM
" MAX. VALUE IN RELADDR1
" LENGTH IN CORE
" PSEUDO LVAR
" PSEUDO LVAR(1)
" ) VIRTUAL
" ) ADDRESS OF
" ) PSEUDO LVAR
" VIRTUAL AGAIN
" NAMELIST IN]
" NAMELIST IN]
" NUMBER OR ADDRESS OF PROCEDURE
" NOT FIRST CALL?
" NAMELIST [N]
" ADDRESS ON DRUM
" LENGTH INCLUDING ADMINISTRATION
" NEW INSTR CNTR
" VIRTUAL INSTR CNTR
" BEYOND32K?
" VIRTUAL AGAIN

```

```

260871 - 502
11555 *C17571* : 610100434
11556 *C17572* : 720016221
11557
11558 *C17573* : 522012772
11559
11560 *C17574* : 000030140
11561 *C17575* : 000027023
11562 *C17576* : 000050042
11563 *C17577* : 002000000
11564 *C17578* : 002000000
11565 *C17579* : 002000000
11566 *C17601* : 562417437
11567 *C17603* : 002000000
11568 *C17604* : 002000000
11569 *C17605* : 002000000
11570 *C17606* : 562417533

      BASE0:
      :PROGRAM3
      :CMPTL 3
      :DECLST 3
      A + 0
      A + 0
      A + 0
      SUBC(:CRF)
      A + 0
      A + 0
      A + 0
      SUBC (:ADDR OF LVAR)

GOTO(:ERM)

6 - BEGIN OF PR AR
VALUE OF CONSTANT#

") TOTAL LENGTH OF OBJECT-
") PROGRAM INTO ERROR
248

```

