stichting
mathematisch
centrum

MC

REKENAFDELING MR 126/71 SEPTEMBER

J.V.M. VAN DER GRINTEN en C. ZUIDEMA

EEN PROCEDUREBIBLIOTHEEK OP ACHTERGRONDGEHEUGEN TN HET MC-MILLI-SYSTEEM VOOR DE EL X8

2e boerhaavestraat 49 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] 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 vertalerroutines 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 vertalerroutines 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
true	0	:base0	normale verwerking van ALGOL60-programma's
false	1	:base1	analyse van procedure programma's
false	2	:base2	opnemen van trommelprocedures

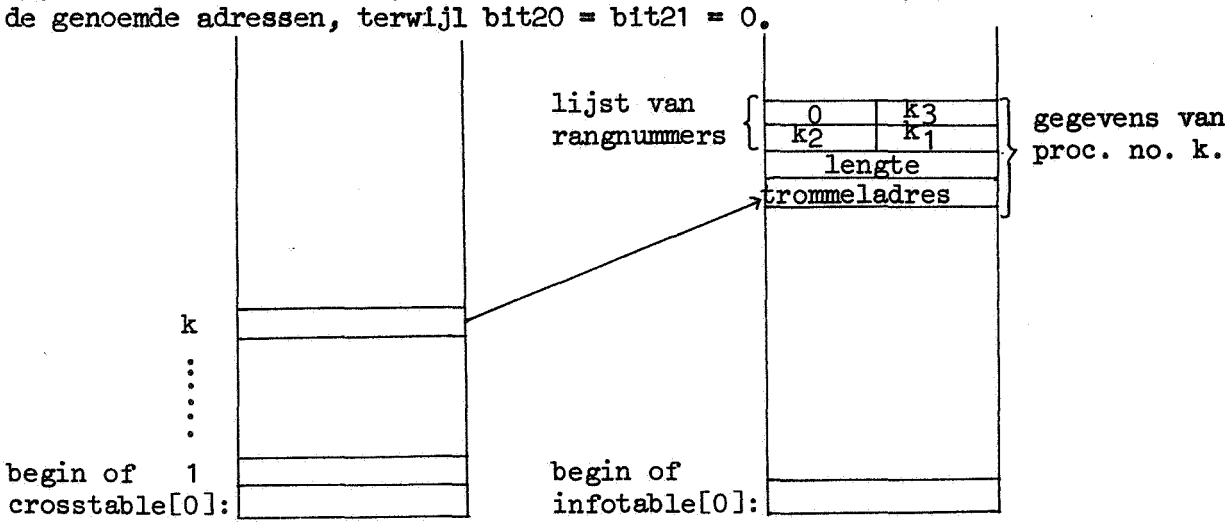
Wij zullen in het vervolg de waarde van COMPMODE gebruiken om de monitor- en vertalerversie aan te duiden waar we het over willen hebben.

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 × 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).

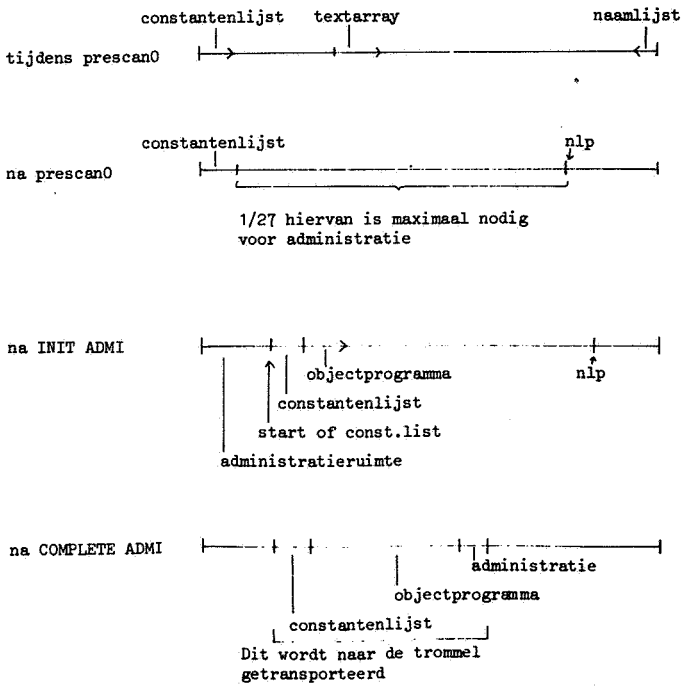
4. Vrij-locateerbare code

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.:

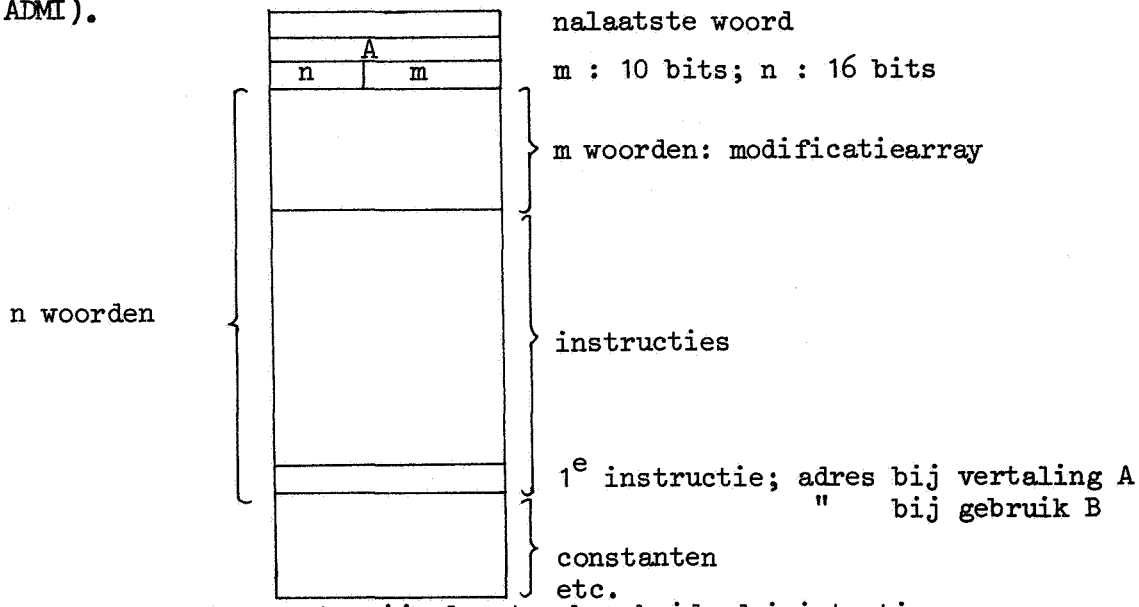
a alle statische adressen moeten worden gemodificeerd: de correctie B - A moet erbij worden opgeteld;
b alle rangnummers van procedures moeten door adressen van pseudolvars vervangen worden.

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 ¬ 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 SWORD0. 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.

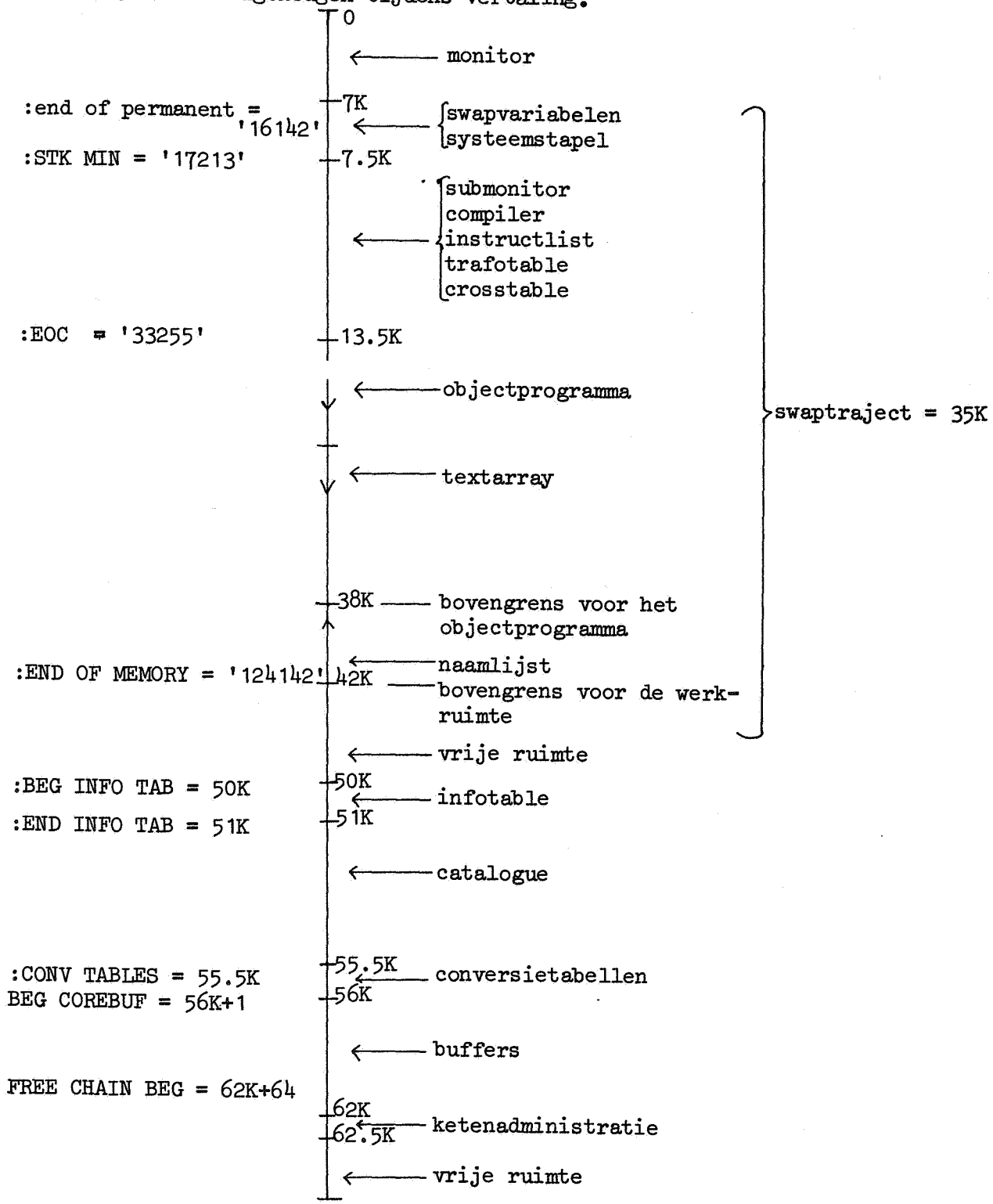
5.1.3. MILLILIB:	monitor	0 - 7.5 K
	compiler	7.5 K - 13.5 K
	objectprogramma	tijdens vertaling: 13.5 K - 38 K na vertaling: 7.5 K - 32 K

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:

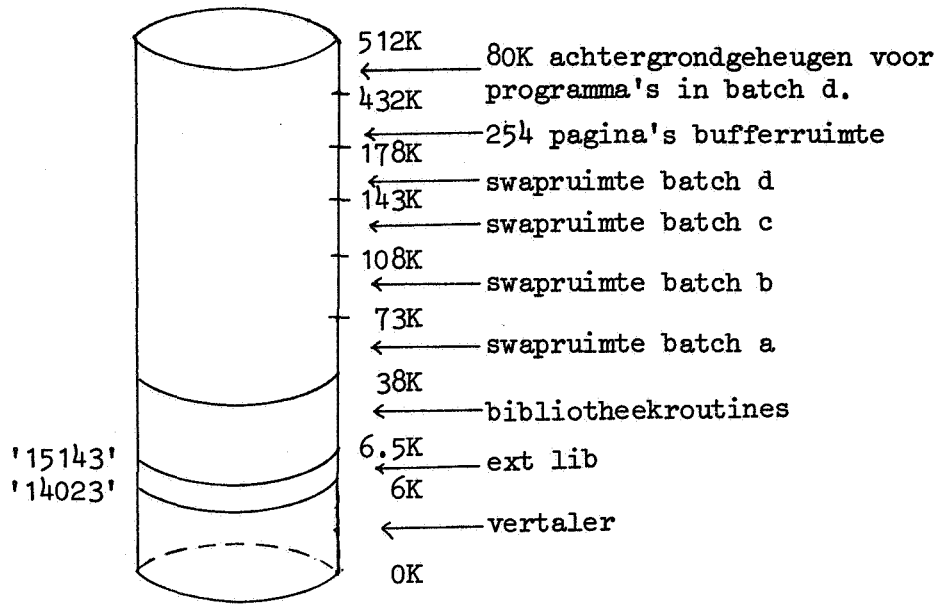
actueel adres = virtueel adres + transposition.

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. Analoog aan 6.1. wordt dit door convert number (zie 8.1.) omgezet in het adres van de bijbehorende pseudolvar. Voor de behandeling van IJU1 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 procedureidentifier 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 pseudolvar's (zie 6.2.):
bv. SUBC (n) wordt SUBC (pseudolvarn), waarbij pseudolvarn de pseudolvar 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 geinterpreteerd 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 pseudolvar, die verwijst naar de betreffende routine die permanent in het kerngeheugen staat.
Voor de trommelprocedures in MILLILIB zijn pseudolvar'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 pseudolvar's te reserveren, zijn tijdens het opnemen van procedures in de bibliotheek de eventueel voor de vertaling benodigde adressen van pseudolvar'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 pseudolvar's (zie 6.).
Deze rangnummers zijn prettig voor het hanteren van de bibliotheekadministratie. Aangezien er echter voor de systeemprocedures 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 pseudolvar'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.

Verwijzing	routine	toelichting
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 LST0	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.

pag. in[6]	routine	toelichting
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 beeindiging 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.

```
10b16                          "SUBMONITOR SECTION"
10b17
10b18     105                  'BEGIN'          BASE1, BASE2, EXTEND LIBRARY, NXT PP,
10b19                                           LIBRARY ROUTINE,  FILL INFOTABLE, DO ADMI,
10b20                                           LUAVAR, ASHIFT, APOS, OLD INSTR CNTR, INIT ADMI,INIT ADMI2,
10b21                                           COMPLETE ADMI,. INSP MACRO, INSP APD, INSP NAME, ADMI 4 ,
10b22                                           ADR OP, CMPTL3CM1,
10b23                                           DECLST3CM1,  PROGRAM3CM1 , PROGRAM3CM2,  FOUND LIBRARY, UPDATE LIBRARY,
10b24                                           INFORM PROGRAMMER, TRUNCATE, INSPECT DECL,
10b25                                           HANDLE QM, SKIP QM, ANONYMIZE, COMPMODE, BASE,
10b26                                            BEG OF ADMI, END OF PROGRAM, RELADDR , RELADDR1, BEGINSAFE,
10b27                                           SHIFTSAFE, WORDSAFE, PNTR, WORD1SAFE,
10b28                                           BEGIN OF PROC,  FRAME, LOCAL NUMBER, D16M1,
10b29                                           NORMAL PROGRAM, LIBR CELL, READ LIBRARY ROUTINE, CRF,
10b30
10b31                                           ADAPT ADR, PREPARE, OBSERVE, BASE0, NXT SBL, INVERT,
10b32                                           DECL LS, DISP LVL, ENTR BLK, NAME IN LIBRARY, NEXT ENTRY, INIT POINTER,
10b33                                           CLEAR CRSTAB, UPDATE CROSSTABLE,  STOCK1, QUOTE  COUNTER, SHIFT, LETTER LAST SYMBOL,
10b34                                            CHARACTER, WORD COUNT, D20, D21, D24, D25,ADDR OF LVAR,
10b35                                           PR DEC, PROGRAM3, CMPTL3, DECLST3, IDF3,
10b36                                           PAR PART,  SAVE NXT SBL, RESTORE NXT SBL, NXT PNR, IN LIBR  , CLEAR CRSTAB SPACE
10b37
10b38                          EXT LIB[0]:
10b39
10b40 '033255':  '000033674'   BASE1:          :PROGRAM3 CM1
10b41 '033256':  '000033660'                   :CMPTL3 CM1
10b42 '033257':  '000033663'                   :DECLST3 CM1
10b43 '033260':  '002000000'                   A + 0
10b44 '033261':  '002000000'                   A + 0
10b45 '033262':  '002000000'                   A + 0
10b46 '033263':  '002000000'                   A + 0
10b47 '033264':  '002000000'                   A + 0
10b48 '033265':  '002000000'                   A + 0
10b49 '033266':  '562434334'                   SUBC (:SKIP QM)
10b50 '033267':  '002000000'                   A + 0
10b51
10b52 '033270':  '000033700'   BASE2:          :PROGRAM3 CM2
10b53 '033271':  '000027023'                   :CMPTL3
10b54 '033272':  '000030042'                   :DECLST3
10b55 '033273':  '562433632'                   SUBC(:INSP MACRO)
10b56 '033274':  '562433646'                   SUBC(: ADMI 4)
10b57 '033275':  '562433636'                   SUBC(: INSP APD)
10b58 '033276':  '562433427'                   SUBC(: LIBRARY ROUTINE)
10b59 '033277':  '562434046'                   SUBC (: TRUNCATE)
10b60 '033300':  '562433651'                   SUBC (:INSP NAME)
10b61 '033301':  '002000000'                   A + 0
10b62 '033302':  '002000000'                   A + 0
10b63
10b64     106                  EXTEND LIBRARY: 'BEGIN' NEXT PROCEDURE
10b65 '033303':  '020400513'                       A= LIBRARY EMPTY, P
10b66 '033304':  '470200513'                   Y, LIBRARY EMPTY= -B
10b67 '033305':  '563633704'                   Y, SUBCD(:FOUND LIBRARY)
10b68 '033306':  '562433734'                       SUBC( :CLEAR CRSTAB SPACE)
10b69 '033307':  '022000001'                       A= 1
10b70 '033310':  '060017213'                       COMPMODE= A
10b71 '033311':  '622033255'                       G=: BASE 1
10b72 '033312':  '720117214'                       BASE= G
10b73 '033313':  '562422426'                       SUBC(:PRESCAN 0)
10b74 '033314':  '120000504'                       S = NBR OF ROUTINES
10b75 '033315':  '150500472'                   U, MAX NBR OF ROUTINES - S, P       "  ROOM?
```

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

                       S = LVAR                          "NUMBER OF PROCEDURES TO INSERT
                       END OF TRAFOTABLE + S
                       END OF CROSSTABLE + S
                       A= BEGIN OF TEXTARRAY
                       BEGINSAFE= A
                       S= MA
                       WORDSAFE= S
                       S= 1
                       SHIFTSAFE= S
                       F= LVAR                           "VARIABLE TO COUNT PROCEDURES
                       MC= F                             " AND VARIABLE TO POINT INTO CATALOGUE
                       WANTED= -8                        "NO COMPILING OF LINENUMBERS
                       A=: BASE 2
                       BASE= A
                       A= 2                              ")INDICATE INSERTING
                       COMPMODE= A                       ")ROUTINES AUTHORIZED
                       G=: LIBR CELL
                       A= D18                            "TO DRUM
                       MG[2]= A

NEXT PROCEDURE:        S= END OF INFO TABLE              "OLD END OF INFOTABLE
                       MC= S
                       SUBC(:CLEAR CRSTAB)
                       SUBC(:PRESCAN 0)
                       A= INSTR CNTR
                       A-1
                       A-START OF CONST LIST             ")LENGTH WITHOUT
                       S= M[B-1]                         ")ADMINISTRATION
                       MS[1]= A                          ")INTO INFOTABLE
                       SUBC(:COMPLETE ADMI)
                       A= INSTR CNTR
                       A-START OF CONST LIST             "LENGTH, INCLUDING ADMIWORDS
                       S= MC[-1]                         "RESTORE OLD END OF INFOTABLE
                       A + MS                            ")FIRST FREE PLACE ON
                       S= END OF INFOTABLE               ")DRUM AT THE END
                       MS= A                             ")OF THE INFOTABLE
                    U, A = DR LIBR END, P
                    V, A = 611                           ") SPACE
                    V, SUBC (:ERRORM)                    ") ON DRUM
                    V, GOTO (:ENDRUN)                    ") EXHAUSTED
                       END OF DRUMLIBRARY= A
                       A= 1
                       PLUSA(M[B-2])                     "NUMBER OF ROUTINES
                       A + BEGIN OF CROSSTABLE
                       MA[1] + S
                       S= A
                       A = START OF CONST LIST
                       A + TRANSPOSITION
                       SUBC(:PREPARE)
                       SUBC(: READ LIBRARY ROUTINE)      " TO DRUM
                       SUBC(:INFORM PROGRAMMER)
                       G= M[B-1]                         " POINTER INTO CATALOGUE
                       SUBC(:NXT PP)
                    N, M[B-1]= A
                    N, GOTO(: NEXT PROCEDURE)
```

```
10636
10637 '033410':  '620516207'                G= ERRONECUS, P
10638 '033411':  '562734342'              N, SUBC (:ANONYMIZE)
10639 '033412':  '562733746'              N, SUBC (:CLEAR CRSTAB)
10640 '033413':  '120300413'              N, S = PROGRAM                          ") SIMULATE
10641 '033414':  '466374411'              N, MS[9] = B                            ") ENDRUN PHASE
10642 '033415':  '562733756'              N, SUBC (:UPDATE CROSSTABLE)
10643 '033416':  '562733770'              N, SUBC (:UPDATE LIBRARY)
10644 '033417':  '522003346'                 GOTO(: ENDRUN)
10645     106                              'END' EXTEND LIBRARY
10646
10647 '033420':  '126072400'  NXT PP:        S= MG
10648 '033421':  '362077777'                S ' * ' 32767                         ")NEXT PROCEDURE POINTER
10649 '033422':  '222000003'                MULS(3)                               "):= ENTRY IN CATALOGUE
10650 '033423':  '072074377'                A= -: MS[-1]                          ")-(3 * FORMAL COUNT + 1)
10651 '033424':  '000000072'                A + G
10652 '033425':  '311100505'              U, A= END OF CATALOGUE, Z               ")LAST PROCEDURE?
10653 '033426':  '526075377'                GOTO(MC[-1])
10654
10655 '033427':               LIBRARY ROUTINE:
10656 '033427':  '122000003'                S=3
10657 '033430':  '160100510'                PLUSS(END OF INFOTABLE)
10658 '033431':  '022000000'                A=0
10659 '033432':  '066074377'                MS[-1]= A
10660 '033433':  '562417434'                SUBC (:INIT POINTER)
10661 '033434':  '562417426'                SUBC(: NEXT ENTRY)
10662 '033435':  '522233560'              Y, GOTO(: INIT ADMI2)                   "INIT ADMI AND EXIT
10663 '033436':  '261117710'              U, A '*' D20, Z                         "PROCEDURE NOT USED?
10664 '033437':  '261317714'              N, A ' *' D24, Z                        "NOT ON DRUM?
10665 '033440':  '562733442'              N, SUBC(: FILL INFOTABLE)
10666 '033441':  '512000006'                JUMP(-6)
10667
10668 '033442':  '020000510'  FILL INFOTABLE: A= END OF INFOTABLE
10669 '033443':  '027173377'              U, A= MA[-1], Z
10670 '033444':  '022300001'              N, A= 1                                 ") IF ALREADY A NUMBER IN WORD
10671 '033445':  '502200004'              Y, JUMP(4)
10672 '033446':  '060100510'                PLUSA(END OF INFOTABLE)             ")THEN INSERT CURRENT NUMBER
10673 '033447':  '760000055'                LUS(13)                             ")AFTER SHIFT AND MAKE
10674 '033450':  '146073376'                MA[-2] + S                          ")NEXT ENTRY ZERO
10675 '033451':  '122000000'                S= 0                                ")
10676 '033452':  '166073377'                MA[-1]= S                           ")ELSE INSERT NUMBER
10677 '033453':  '010400501'                A=INFO TAB END, P
10678 '033454':  '022201446'              Y, A= 806
10679 '033455':  '522212772'              Y, GOTO(:ERM)                           "END OF COMPILATION
10680 '033456':  '526475377'                GOTO(MC[-1])
10681
10682     107                  DO ADMI:       'BEGIN' NEWWORD, LOOP1, LOOP2
10683 '033457':  '562430561'                SUBC(:PAR PART)                       "PAR PART IN S
10684 '033460':  '112000001'                S-1
10685 '033461':  '100016215'                S + INSTR CNTR                        "POINTS TO WORD TO BE MODIFICATED
10686 '033462':  '166075400'                MC = S
10687 '033463':  '066075400'                MC = A
10688 '033464':  '726075400'                MC = F
10689 '033465':  '020000074'                A= S                                  "COPY IN A
10690 '033466':  '110017220'                S-OLD INSTR CNTR                      "NUMBER OF WORDS NOT TO BE MODIFICATED
10691 '033467':  '060017220'                OLD INSTR CNTR= A
10692 '033470':  '620117217'                G= APOS                               "POINTS TO ADMIWORD
10693 '033471':  '151017216'                ASHIFT - S, Z                         ")
10694 '033472':  '020717216'              N, A = ASHIFT, P                        ") IS THIS WORD ENOUGH?
10695 '033473':  '026072400'                A = MG
```

```
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
10700 '033500':  '066072400'                 MG= A                       "ADMIWORD
10701 '033501':  '562733506'              N, SUBC(:NEWWORD)
10702 '033502':  '626075376'                 F= MC[-2]                   "RESTORE REGISTERS
10703
10704 '033503':  '026075377'                 A= MC[-1]
10705 '033504':  '126075377'                 S = MC[-1]
10706 '033505':  '526475377'                 GOTOR(MC[-1])               "EXIT
10707
10708 '033506':  '122000032'  NEWWORD:       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':  '022200001'              Y, A= 1
10714 '033514':  '022300000'              N, A= 0
10715 '033515':  '066072400'                 MG= A                       "ADMIWORD
10716 '033516':  '526675377'              Y, GOTOR(MC[-1])
10717 '033517':  '522033507'                 GOTO(:NEWWORD[1])
10718
10719 '033520':  '660000040'  LUAVAR:        LUA(0)
10720
10721 '033521':  '130016215'  INIT ADMI:     S = - INSTR CNTR
10722 '033522':  '110000436'                 S - TRANSPOSITION           "ACTUAL
10723 '033523':  '630100074'                 G= - S                      "COPY
10724 '033524':  '100016574'                 S + NLP                     ") RESERVE PLACE
10725 '033525':  '022000000'                 A= 0                        ")FOR ADMIWORDS
10726 '033526':  '302000033'                 DIVAS(27)
10727 '033527':  '102000001'                 S + 1
10728 '033530':  '140016215'                 INSTR CNTR + S
10729 '033531':  '020000434'                 A= BEGIN OF PR AR
10730 '033532':  '100000073'                 S + A
10731 '033533':  '160000476'                 START OF CONST LIST= S
10732 '033534':  '000000436'                 A + TRANSPOSITION           ")ACTUAL
10733 '033535':  '100000436'                 S + TRANSPOSITION           ")
10734 '033536':  '060017215'                 BEG OF ADMI= A
10735 '033537':  '611100073'                 G - A, Z                    ")NUMBER OF CONSTANTS
10736 '033540':  '502200010'              Y, JUMP(8)                     ")AND FIRST WORDS OF
10737 '033541':  '720100007'                 COUNT= G                    ")OBJECTPROGRAM INTO COUNT
10738 '033542':  '000000072'                 A + G
10739 '033543':  '100000072'                 S + G
10740 '033544':  '012000001'  LOOP1:        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'                 REPP(:LOOP1)
10745 '033551':  '022000032'                 A = 26
10746 '033552':  '060017216'                 ASHIFT= A                   "INITIALIZE ASHIFT
10747 '033553':  '020017215'                 A= BEG OF ADMI
10748 '033554':  '060017217'                 APOS= A                     "INITIALIZE APOS
10749 '033555':  '122000000'                 S= 0                        ") INITIALIZE
10750 '033556':  '166073400'                 MA= S                       ") FIRST ADMIWORD
10751 '033557':  '526475377'                 GOTOR(MC[-1])               "EXIT INIT ADMI AND LIBRARY ROUTINE
10752
10753 '033560':  '020016215'  INIT ADMI2:    A = INSTR CNTR              ") POINTER TO
10754 '033561':  '012000001'                 A- 1                        ") LAST
10755 '033562':  '060017220'                 OLD INSTR CNTR = A          ") HANDLED WORD
```

```
10756 '033563':  '526475377'                  GOTOR (MC[-1])
10757
10758 '033564':  '620117217'  COMPLETE ADMI:  G= APOS
10759 '033565':  '026072400'                  A= MG                          "ADMIWORD
10760 '033566':  '120017216'                  S= ASHIFT
10761 '033567':  '100033520'                  S + LUAVAR                     ")
10762 '033570':  '570400074'                  DO(S)                          ")SHIFT S
10763 '033571':  '066072400'                  MG= A                          "COMPLETE LAST ADMIWORD
10764 '033572':  '122000001'                  S= 1
10765 '033573':  '100017217'                  S + APOS
10766 '033574':  '110017215'                  S - BEG OF ADMI                "NUMBER OF ADMIWORDS
10767 '033575':  '160000007'                  COUNT= S                       ")PREPARE LOOP IN ORDER
10768 '033576':  '120017215'                  S= BEG OF ADMI                 ")TO PLACE THE ADMIWORDS
10769 '033577':  '620116215'                  G= INSTR CNTR                  ")AFTER THE OBJECTPROGRAM
10770 '033600':  '600100436'                  G + TRANSPOSITION              "ACTUAL
10771 '033601':  '020000072'                  A= G
10772 '033602':  '000000007'                  A + COUNT
10773 '033603':  '002000002'                  A + 2
10774 '033604':  '010500477'                U, A - BEGIN OF NLI, P
10775 '033605':  '022201450'                Y, A = 808
10776 '033606':  '522212772'                Y, GOTO (:ERM)
10777 '033607':  '026074400'  LOOP2:          A= MS
10778 '033610':  '066072400'                  MG= A
10779 '033611':  '102000001'                  S + 1                          "STEP UP
10780 '033612':  '602000001'                  F + 1
10781 '033613':  '556433607'                  REPP(:LOOP2 )
10782 '033614':  '130016560'                  S= -BEGIN OF PROGRAM
10783 '033615':  '176072401'                  MG[1]= -S                      "ABSOLUTE ADDRESS
10784 '033616':  '110000436'                  S - TRANSPOSITION              "ACTUAL NOW
10785 '033617':  '100000072'                  S + G                          ")RELATIVE POINTER TO
10786 '033620':  '760000052'                  LUS(10)                        ")BEGIN OF PROGRAM
10787
10788 '033621':  '166072400'                  MG= S                          ")
10789 '033622':  '130016215'                  S= - INSTR CNTR                ")AND NUMBER OF
10790 '033623':  '110000436'                  S - TRANSPOSITION              ")
10791 '033624':  '100000072'                  S + G                          ")
10792 '033625':  '146072400'                  MG   + S                       ")ADMIWORDS
10793 '033626':  '602000002'                  F + 2
10794 '033627':  '610100436'                  G - TRANSPOSITION              "VIRTUAL
10795 '033630':  '720116215'                  INSTR CNTR= G                  "FIRST FREE PLACE
10796 '033631':  '526475377'                  GOTOR(MC[-1])
10797     107                                 'END' DO ADMI
10798
10799 '033632':  '562433656'  INSP MACRO:      SUBC (:ADR OP)                 " ADDRESS OPERAND?
10800 '033633':  '130617705'                Y, S = -CODEBODY, P              " AND NOT IN CODEBODY?
10801 '033634':  '526775377'                N, GOTOR(MC[-1])
10802 '033635':  '522033457'                   GOTO(: DO ADMI)
10803
10804 '033636':  '020000074'  INSP APD:       A= S                           "COPY APD
10805 '033637':  '261100421'                U, A '*' D18, Z
10806 '033640':  '526775377'                N, GOTOR (MC[-1])
10807 '033641':  '670000064'                   RUA (20)
10808 '033642':  '013000006'                   A-6,Z                          ")EXCLUDE 6 * D20
10809 '033643':  '013300001'                N, A-1, Z                         ")        7 * D20
10810 '033644':  '013300006'                N, A-6,Z                          ")   AND 13 * D20
10811 '033645':  '526675377'                Y, GOTOR(MC[-1])                  ")
10812 '033646':  '166075400'  ADMI 4 :        MC = S
10813 '033647':  '120016215'                   S = INSTR CNTR
10814 '033650':  '522033463'                   GOTO(: DO ADMI[4])
10815
```

```
10616 '033651':  '112400000'  INSP NAME:      S - 0, P                  " PARAMETER > 0?
10617 '033652':  '030717705'                N, A =- CODEBODY, P       " - NOT IN CODEBODY ?
10618 '033653':  '024237776'                Y, A = M[ B - 2]          " MACRO
10619 '033654':  '522233457'                Y, GOTO (:DO ADMI )       " ADMINISTRATION
10620 '033655':  '526475377'                   GOTOR (MC[-1])
10621
10622 '033656':  '263100004'  ADR OP:     U, A ' + ' 4,Z
10623 '033657':  '522020616'                   GOTO(:INVERT)
10624
10625 '033660':  '022001442'  CMP TL3CM1:      A= 802
10626 '033661':  '562403560'                   SUBC(:ERRCRM)
10627 '033662':  '522027023'                   GOTO(:CMPTL3)
10628
10629 '033663':  '562421337'  DECLST3CM1:      SUBC (:DISP LVL)          ")THIS PART TO BE CARRIED OUT
10630 '033664':  '113000001'                   S - 1, Z                  ")ONLY ONCE FOR THE WHOLE PROGRAM
10631 '033665':  '562430042'                   SUBC(:DEC LST3)
10632 '033666':  '526775377'                N, GOTOR (MC[-1])
10633 '033667':  '120016212'                   S= LAST SYMBOL
10634 '033670':  '113100151'                U, S - 105, Z             "LAST SYMBOL= END?
10635 '033671':  '022001443'                N, A=803
10636 '033672':  '562703560'                N, SUBC(:ERRCRM)
10637 '033673':  '526475377'                   GOTOR(MC[-1])
10638
10639 '033674':  '113100150'  PROGRAM3CM1: U, S-104, Z                  "LAST SYMBOL= BEGIN?
10640 '033675':  '022001441'                N, A= 801
10641 '033676':  '562703560'                N, SUBC(:ERROR M)
10642 '033677':  '522030140'                   GOTO(:PROGRAM3)
10643 '033700':  '562420030'  PROGRAM3CM2:     SUBC (:NXT SBL)
10644 '033701':  '562420627'                   SUBC (: DECL LS)
10645 '033702':  '562421364'                   SUBC (: ENTR BLK)
10646 '033703':  '522027646'                   GOTO (:PR DEC)
```

```
10851     108                FOUND LIBRARY:'BEGIN' FOUND CROSSTABLE, D23M3
10852
10853 '033704':  '120000474'                 S = PERMANENT CATALOGUE END
10854 '033705':  '160000505'                 END OF CATALOGUE = S
10855
10856 '033706':  '620100503'                 G = BEGIN OF INFOTABLE
10857 '033707':  '122000000'                 S = 0
10858 '033710':  '166072403'                 MG[3] = S                          ") CROSSREFERENCE
10859                                                                           ") IRRELEVANT FOR
10860                                                                           ") PERMANENT PART
10861 '033711':  '120000467'                 S = DR LIBR BEG
10862 '033712':  '160000511'                 END OF DRUMLIBRARY = S             ") NO PROCEDURE ON
10863                                                                           ") DRUM YET
10864 '033713':  '602000004'                 G + 4
10865 '033714':  '726100510'                 END OF INFOTABLE = G               " 3 ENTRIES
10866 '033715':  '166072400'                 MG = S                             " FIRST FREE PLACE ON DRUM
10867
10868 '033716':  '120000471'                 S = NBR OF PERMANENT ROUTINES
10869 '033717':  '160000504'                 NBR OF ROUTINES = S
10870 '033720':  '102000001'                 S + 1
10871 '033721':  '100000500'                 S + BEGIN OF TRAFOTABLE
10872 '033722':  '160000506'                 END OF TRAFOTABLE = S
10873 '033723':  '100000472'                 S + MAX NBR OF ROUTINES
10874 '033724':  '160000507'                 END OF CROSSTABLE = S
10875 '033725':  '600133733'                 G + D23M3                          " ((('040 000 000' - 3) +:PSEUDO ENTRY)
10876
10877 '033726':               FOUND CROSSTABLE:
10878 '033726':  '112000001'                    S - 1                           " ) ALL PERMANENT ROUTINES
10879 '033727':  '726174400'                    MS = G                          ") USE PSEUDO ENTRY
10880 '033730':  '111100502'              U, S - BEGIN OF CROSSTABLE, Z
10881 '033731':  '522033726'              N, GOTO(:FOUND CROSSTABLE)
10882 '033732':  '522033770'                 GOTO(:UPDATE LIBRARY)
10883 '033733':  '037777775'  D23M3:         ( '040 000 000' - 3)
10884
10885     108                                'END' FOUND LIBRARY
10886
10887 '033734':               CLEAR CRSTAE SPACE:
10888 '033734':  '020017714'                 A =D24
10889 '033735':  '120000502'                 S = BEGIN OF CROSSTABLE            ") INITIALIZE
10890 '033736':  '100000472'                 S + MAX NBR OF ROUTINES            ") NOT
10891 '033737':  '112000001'                 S - 1                              ") USED
10892 '033740':  '111100507'              U, S - END OF CROSSTABLE,  Z           ") PART
10893 '033741':  '066074400'              N, MS = A                              ") OF
10894 '033742':  '512000004'              N, JUMP (-4)                           ") CROSSTABLE
10895 '033743':  '000000510'                 A + END OF INFOTABLE                ") M[END OF CROSSTABLE] :=
10896 '033744':  '066074400'                 MS = A                              ") END OF INFOTABLE
10897 '033745':  '526475377'                 GOTOR ( MC [-1])
10898
10899
10900     109                CLEAR CRSTAB:   'BEGIN' LOOP, D21 PLUS D20
10901
10902 '033746':  '562417434'                 SUBC(:INIT POINTER)
10903 '033747':  '562417426'  LOOP:          SUBC(:NEXT ENTRY)
10904 '033750':  '526675377'              Y, GOTOR(MC[-1])
10905 '033751':  '270033755'                 A'*'-D21 PLUS D20
10906 '033752':  '100000502'                 S + BEGIN OF CROSSTABLE
10907 '033753':  '066074400'                 MS = A
10908 '033754':  '522033747'                 GOTO(:LOOP)
10909 '033755':  '014000000'  D21 PLUS D20 :  '014 000 000'
10910
```

13.7.

-31-

```
10911     109                          'END' CLEAR CRSTAB
10912
10913 '033756':              UPDATE CROSSTABLE:
10914 '033756':  '622017317'              G = :LIBR CELL
10915 '033757':  '020000472'              A = MAX NBR OF ROUTINES
10916 '033760':  '066072401'              MG[1] = A                          " LENGTH OF TRANSPORT
10917 '033761':  '020000421'              A = D18
10918 '033762':  '066072402'              MG[2] = A                          " DIRECTION TO DRUM
10919 '033763':  '020000502'              A = BEGIN OF CROSSTABLE
10920 '033764':  '066072403'              MG[3] = A                          " CORE ADDRESS
10921 '033765':  '012017232'              A = :BEGIN OF SUBMONITOR
10922 '033766':  '066072404'              MG[4] = A                          " DRUM ADDRESS
10923 '033767':  '522017324'              GOTO (:READ LIBRARY ROUTINE)       " CROSSTABLE TO DRUM
10924
10925
10926 '033770':              UPDATE LIBRARY:
10927 '033770':  '620000504'              F = NBR OF ROUTINES                ")
10928 '033771':  '720000514'              LVAR = F                           ") SEE ASSURE LIBRARY
10929 '033772':  '620000506'              F = END OF TRAFOTABLE              ")
10930 '033773':  '720000516'              LVAR[2] = F                        ")
10931 '033774':  '620000510'              F = END OF INFOTABLE               ")
10932 '033775':  '720000520'              LVAR[4] = F                        ")
10933 '033776':  '526475377'              GOTOR(MC[-1])
10934
10935 '033777':              ASSURE LIBRARY:
10936 '033777':  '620000514'              F = LVAR
10937 '034000':  '720000504'              NBR OF ROUTINES = F
10938 '034001':  '620000516'              F = LVAR [2]
10939 '034002':  '720000506'              END OF TRAFOTABLE = F
10940 '034003':  '620000520'              F = LVAR [4]
10941 '034004':  '720000510'              END OF INFOTABLE = F
10942 '034005':  '526475377'              GOTOR (MC[-1])
10943
10944 '034006':              INFORM PROGRAMMER:
10945     110                          'BEGIN' PR IDF, PRINT, PRINT WORD, ELOOP1
10946
10947 '034006':  '622000002'              F = 2                              ") DOUBLE SPACE
10948 '034007':  '562410562'              SUBC(:CARRIAGE)                    ") VERTICALLY
10949 '034010':  '022000001'  PR  DF:     A = 1
10950 '034011':  '074137776'              MINA(M[B-2])
10951 '034012':  '126473401'              S = MA[1], P                       " NO LONGER IDENTIFIER?
10952 '034013':  '562734031'          N,  SUBC(:PRINT WORD)
10953 '034014':  '522634010'          N,  GOTO(:PR IDF)
10954 '034015':  '062074400'              A=:MS                              " NUMBER OF PROCEDURE
10955 '034016':  '562434023'              SUBC(:PRINT)
10956 '034017':  '026000401'              A = M0[1]
10957 '034020':  '562434023'              SUBC(:PRINT)                       " LENGTH IN CORE
10958 '034021':  '136000400'              A =-M0
10959 '034022':  '000000511'              A + END OF DRUMLIBRARY             " LENGTH ON DRUM
10960 '034023':  '166075400'  PRINT:      MC = S
10961 '034024':  '122000006'              S= 6
10962 '034025':  '562410233'              SUBC (:LP FIXT)
10963 '034026':  '126075377'              S = MC[-1]
10964 '034027':  '622000006'              F = 6
10965 '034030':  '522010314'              GOTO(:SPACE)
10966
10967 '034031':  '130000074'  PRINT WORD:  S = - S
10968 '034032':  '022000004'              A = 4
10969 '034033':  '060000007'              COUNT = A
10970 '034034':  '760000002'              LCS(2)                             " DISCARD UNUSED BITS
```

```
10971 '034035':  '660000146'  ELOOP1:        LUAS(6)                          " SHIFT TO POSITION
10972 '034036':  '263000077'                  A '*' 63, Z                      " CLEAN CHARACTER, DONE?
10973 '034037':  '062373377'                N, A =:MA[-1]                      " CORRECT CODE
10974 '034040':  '166375400'                N, MC=S                            " SAVE S
10975 '034041':  '120300073'                N, S = A
10976 '034042':  '502710251'                N, SUBC(:PRNTIS)                   " PRINT CHARACTER
10977 '034043':  '126375377'                N, S = MC[-1]                      " RESTORE S
10978 '034044':  '556434035'                   REPP(:ELOCP1)                   " NEXT CHARACTER, IF ANY
10979 '034045':  '526475377'                   GOTOR(MC[-1])
10980
10981     110                        'END' INFORM PROGRAMMER
10982
10983     111                TRUNCATE:       'BEGIN' ELSE0, ELSE1, END, ENDSBL
10984
10985 '034046':  '066075400'                   MC = A
10986 '034047':  '562434131'                   SUBC(: SAVE NXT SBL)
10987 '034050':  '562426030'                   SUBC(: NXT SBL)
10988 '034051':  '113100151'                U, S - 105,Z
10989 '034052':  '522334126'                Y, GOTO(:END)
10990
10991 '034053':  '120017614'                   S = SHIFT
10992 '034054':  '020016572'                   A =TEXT ARRAY POINTER
10993 '034055':  '113100002'                U, S - 2,Z
10994 '034056':  '522334073'                N, GOTO(:ELSE0)
10995 '034057':  '102000001'                   S + 1
10996 '034060':  '160017225'                   SHIFTSAFE = S
10997 '034061':  '012000001'                   A - 1
10998 '034062':  '060017224'                   BEGINSAFE = A
10999 '034063':  '122000150'                   S = 104                          ")SIMULATE TEXT ARRAY BEGINNING WITH
11000 '034064':  '760000060'                   LUS(16)                          ")BEGIN SYMBOL FOR NEXT PROCEDURE
11001 '034065':  '160017226'                   WORDSAFE = S                     ")
11002 '034066':  '126073401'                   S = MA[1]                        ") SAVE SECOND WORD OF TEXT FOR
11003 '034067':  '160017230'                   WORD1SAFE = S                    ") NEXT PROCEDURE
11004 '034070':  '122000151'                   S = 105
11005 '034071':  '166073401'                   MA[1] = S                        " TRUNCATE TEXT BY INSERTING OF END SYMBOL
11006 '034072':  '522034126'                   GOTO(:END)
11007
11008 '034073':  '113100001'  ELSE0:        U, S - 1, Z
11009 '034074':  '522334113'                N, GOTO(:ELSE1)
11010 '034075':  '122000002'                   S = 2
11011 '034076':  '160017225'                   SHIFTSAFE = S
11012 '034077':  '012000001'                   A - 1
11013 '034100':  '060017224'                   BEGINSAFE = A
11014 '034101':  '126073400'                   S = MA                           " TEXT[BEGINSAFE]
11015 '034102':  '370017232'                   S '*' -FRAME                     " MAKE PLACE FOR BEGIN SYMBOL
11016 '034103':  '102064000'                   S + 26624                        " INSERT BEGIN SYMBOL
11017 '034104':  '160017226'                   WORDSAFE = S
11018 '034105':  '126073400'                   S = MA                           " TEXT[BEGINSAFE]
11019 '034106':  '360017234'                   S '*' D16 M1
11020 '034107':  '100034112'                   S + ENDSBL
11021 '034110':  '166073400'                   MA = S
11022 '034111':  '522034126'                   GOTO(:END)
11023 '034112':  '032200000'  ENDSBL:          + 6881280
```

13.9.

-33-

```
11028 '034113':  '060017224'  ELSE1:          BEGINSAFE = A
11029 '034114':  '122000001'                  $ =1
11030 '034115':  '160017225'                  $HIFTSAFE = S
11031 '034116':  '126073400'                  S = MA                          " TEXT[BEGINSAFE]
11032 '034117':  '372000377'                  S '*' - 255                     " MAKE PLACE FOR BEGIN SYMBOL
11033 '034120':  '102000150'                  S + 104                         " INSERT BEGIN SYMBOL
11034 '034121':  '160017226'                  WORDSAFE = S
11035 '034122':  '126073400'                  S = MA                          " TEXT[BEGINSAFE]
11036 '034123':  '370017232'                  S '*'-FRAME                     " MAKE PLACE FOR END_SYMBOL
11037 '034124':  '102064400'                  S + 26880                       " INSERT END SYMBOL
11038
11039 '034125':  '166073400'                  MA = S                          " TRUNCATE TEXT
11040 '034126':  '562434144'  END:            SUBC (!RESTORE NXT SBL)
11041 '034127':  '026075377'                  A = MC[-1]
11042 '034130':  '526475377'                  GOTOR(MC[-1])
11043
11044     111                          'END'TRUNCATE
11045
11046 '034131':  '120016212'  SAVE NXT SBL:   S = LAST SYMBOL
11047 '034132':  '166075400'                  MC = S
11048 '034133':  '120017610'                  S = STOCK1
11049 '034134':  '166075400'                  MC = S
11050 '034135':  '120017613'                  S = QUOTE COUNTER
11051 '034136':  '166075400'                  MC= S
11052 '034137':  '120016572'                  S = TEXT ARRAY POINTER
11053 '034140':  '166075400'                  MC = S
11054 '034141':  '120017614'                  S = SHIFT
11055 '034142':  '166075400'                  MC = S
11056 '034143':  '524437772'                  GOTOR (M[B -6])
11057
11058 '034144':  '126075377'  RESTORE NXT SBL:S = MC[-1]                       " ) SAVE LINK
11059 '034145':  '164037772'                  M[ B -6] = S                    ")
11060 '034146':  '126075377'                  S = MC[-1]
11061 '034147':  '160017614'                  SHIFT = S
11062 '034150':  '126075377'                  S = MC[-1]
11063 '034151':  '160016572'                  TEXT ARRAY POINTER = S
11064 '034152':  '126075377'                  S = MC[-1]
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])
```

```
11075     112                 :NSPECT DECL:   'BEGIN' LOOP, LOOP1, PROC END, END0, END1
11076
11077 '034161':  '562421337'                  SUBC(:DISP LVL)                  " IF DISPLAY LEVEL>1 THEN
11078 '034162':  '113000001'                  S = 1, Z                         " INSPECT DECL NOT
11079 '034163':  '526775377'              N, GOTOR(MC[-1])                     " APPROPRIATE
11080 '034164':  '120017622'                  S = CHARACTER
11081 '034165':  '770000063'                  RUS(19)
11082 '034166':  '112500031'              U, S = 25, P                         " PROCEDURE DECLARATION?
11083 '034167':  '113700017'              N, S = 15, E
11084 '034170':  '022201444'              Y, A = 804                           ") IF NO PROCEDURE REPORT THIS FACT
11085 '034171':  '522203560'              Y, GOTO(:ERRCRM)                     ") AND LEAVE INSPECT DECL
11086 '034172':  '562434131'                  SUBC (:SAVE NXT SBL)
11087 '034173':  '562420030'                  SUBC (:NXT SBL)
11088 '034174':  '562434276'                  SUBC (:HANDLE QM)
11089 '034175':  '121017616'                  S = LETTER LAST SYMBOL, Z
11090 '034176':  '522334274'              N, GOTO(:END1)
11091 '034177':  '562431157'                  SUBC(:IDF3)
11092 '034200':  '166075400'                  MC = S                           " N
11093 '034201':  '562421244'                  SUBC(:NAME IN LIBRARY)
11094 '034202':  '022301445'              N, A = 805                           " REPORT NAME ALREADY USED
11095 '034203':  '562703560'              N, SUBC(:ERRCRM)                     " IN LIBRARY
11096 '034204':  '124037777'                  S = M[B-1]                       " N
11097
11098 '034205':  '126074377'                  S = MS[-1]
11099 '034206':  '362077777'                  S '*' 32767                      " FORMAL COUNT + 1
11100 '034207':  '222000004'                  MULS(4)                          " 4* FORMAL COUNT + 4 + WORD COUNT
11101 '034210':  '100017632'                  S + WORD COUNT
11102 '034211':  '100000473'                  S + CAT END
11103 '034212':  '110400505'                  S - END OF CATALOGUE, P          " SPACE ENOUGH?
11104 '034213':  '022201451'              Y, A = 809
11105 '034214':  '562603560'              Y, SUBC(:ERRCRM)
11106 '034215':  '522234274'              Y, GOTO(:END0)
11107 '034216':  '620100505'                  G = END OF CATALOGUE
11108 '034217':  '022000001'                  A = 1
11109 '034220':  '040000504'                  NBR OF ROUTINES + A
11110 '034221':  '000017632'                  A + WORD COUNT
11111 '034222':  '064040000'                  M[B] = A
11112 '034223':  '020016574'                  A = NLP
11113 '034224':  '126073400'  LOOP:           S = MA                           ")
11114 '034225':  '166072400'                  MG = S                           ") JUST READ
11115 '034226':  '012000001'                  A = 1                            ") PROCEDURE IDENTIFIER
11116 '034227':  '612000001'                  G = 1                            ") INTO
11117 '034230':  '122000001'                  S = 1                            ") CATALOGUE
11118 '034231':  '155040000'                  M[B] = S, Z                      ") BY COPYING FROM
11119 '034232':  '522334224'              N, GOTO(:LOOP)                       ") BOTTOM OF NAMELIST
11120 '034233':  '024037777'                  A = M[B-1]                       " N OF PROC IDENTIFER
11121 '034234':  '126073400'                  S = MA                           ")
11122 '034235':  '372077777'                  S '*' -32767                     ") CHAR*D19 OUT OF NAMELIST
11123 '034236':  '340000504'                  S '+' NBR OF ROUTINES            ") + NUMBER OF ROUTINES
11124 '034237':  '166072400'                  MG = S                           ") INTO CATALOGUE
11125 '034240':  '126073377'                  S = MA[-1]
11126 '034241':  '362077777'                  S '*' 32767
11127 '034242':  '113000001'                  S = 1, Z
11128 '034243':  '164040000'                  M[B] = S
11129 '034244':  '100017233'                  S + LOCAL NUMBER                 ")'222 000 001' + FORMAL COUNT
11130 '034245':  '166072377'                  MG[-1] = S                       ") INTO CATALOGUE
11131 '034246':  '012000007'                  A = 7
11132 '034247':  '522234267'              Y, GOTO(:PROC END)
11133 '034250':  '132000001'  LOOP1:          S = -1
11134 '034251':  '612000001'                  G = 1                            ") SIMULATE LETTERS
```

```
11135 '034252':  '166072377'                MG[-1] = S                          ")
11136 '034253':  '126473376'                S = MA[-2], P
11137 '034254':  '012000001'                A - 1                               ") SKIP LETTERS
11138 '034255':  '512300003'             N, JUMP(-3)                            ")
11139 '034256':  '370000422'                S '*'- D18M1
11140 '034257':  '166072376'                MG[-2] = S                          " CHARACTER OF FORMAL
11141 '034260':  '122000000'                S = 0
11142 '034261':  '166072375'                MG[-3] = S
11143 '034262':  '612000002'                G - 2
11144 '034263':  '012000001'                A - 1
11145 '034264':  '122000001'                S = 1
11146 '034265':  '155040000'                M[B] - S, Z
11147 '034266':  '522334250'             N, GOTO(:LOOP1)
11148 '034267':  '122000000'  PROC END:     S = 0                               " INSERT ZERO AT THE END OF THE PROCEDURE
11149 '034270':  '166072376'                MG[-2] = S
11150 '034271':  '612000003'                G - 3
11151 '034272':  '720100005'                END OF CATALOGUE = G
11152 '034273':  '412000001'  END0:         B - 1                               " THROW AWAY N OF PROC IDENTIFIER
11153 '034274':  '562434144'  END1:         SUBC (:RESTORE NXT SBL)
11154 '034275':  '526475377'                GOTOR(MC[-1])
11155
11156     112                               'END' INSPECT DECL
11157
11158     113               HANDLE QM:      'BEGIN'          INSTR
11159
11160 '034276':  '120016212'                S = LAST SYMBOL
11161 '034277':  '113100172'             U, S - 122, Z                          " QUESTIONMARK?
11162 '034300':  '526775377'             N, GOTOR (MC[-1])
11163 '034301':  '120000004'                S = NBR OF ROUTINES
11164 '034302':  '102000001'                S + 1
11165 '034303':  '100000502'                S + BEGIN OF CROSSTABLE             ") INDICATION
11166 '034304':  '020017715'                A = D25                             ") INTO
11167 '034305':  '046074400'                MS+ A                               ") CROSSTABLE
11168 '034306':  '020016572'                A = TEXT ARRAY POINTER
11169 '034307':  '620117614'                G = SHIFT
11170 '034310':  '613000001'                G -1 ,Z
11171 '034311':  '012200001'             Y, A - 1                               " TEXT ARRAY POINTER -1
11172 '034312':  '126073400'                S = MA                              " TEXT
11173 '034313':  '066075400'                MC = A
11174 '034314':  '022034326'                A = :INSTR
11175 '034315':  '600100073'                G + A
11176 '034316':  '576472400'                DO (MG)                             " REMOVE QUESTIONMARK
11177 '034317':  '166075400'                MC = S
11178 '034320':  '122000167'                   S =119
11179 '034321':  '576472403'                   DO (MG[3])
11180 '034322':  '106075377'                S + MC[-1]                          " REPLACE IT BY NLCR
11181 '034323':  '026075377'                A = MC[-1]
11182 '034324':  '166073400'                MA = S                              " INTO TEXTARRAY
11183 '034325':  '522020030'                GOTO (:NXT SBL)                     " SKIP QUESTIONMARK
11184 '034326':  '360017234'  INSTR:        S '*' D16M1
11185 '034327':  '372000377'                S '*' - 255
11186 '034330':  '370017232'                S '*' - FRAME
11187 '034331':  '760000060'                LUS (16)
11188 '034332':  '760000040'                LUS (0)
11189 '034333':  '760000050'                LUS (8)
11190     113                               'END'   HANDLE QM
11191
11192 '034334':  '562421337'  SKIP QM:      SUBC(: DISP LVL)                    " DISPLAY LEVEL = 1?
11193 '034335':  '113000001'                S - 1, Z
11194 '034336':  '120216212'             Y, S = LAST SYMBOL
```

```
11195 '034337': '113200172'              Y, S - 122, Z                    " AND QUESTIONMARK?
11196 '034340': '522220030'              Y, GOTO (:NXT SBL)               " SKIP IT
11197 '034341': '526475377'                 GOTOR (MC[-1])
11198
11199     114                ANONYMIZE:     'BEGIN'          LOOP
11200
11201 '034342': '120000505'                 S = END OF CATALOGUE          ") MARK
11202 '034343': '032000002'                 A= - 2                        ") END OF
11203 '034344': '066074400'                 MS = A                        ") CATALOGUE
11204 '034345': '120000474'                 S = PERMANENT CATALOGUE END
11205 '034346': '562421211' LOOP:           SUBC (:NXT PNR)
11206 '034347': '013100002'              U, A- 2 , Z                      " ENDMARKER?
11207 '034350': '526675377'              Y, GOTOR (MC[-1])
11208 '034351': '013100001'              U, A - 1, Z                      " FORMAL IDENTIFIER?
11209 '034352': '112200001'              Y, S - 1                         " DECREASE PARAMETER FOR NXT PNR
11210 '034353': '522234346'              Y, GOTO (:LOOP)
11211 '034354': '166075400'                 MC = S                        " ADDRESS OF FIRST LETTERS
11212 '034355': '112000001'                 S - 1
11213 '034356': '026474400'                 A = MS, P
11214 '034357': '512600003'              N, JUMP (-3)
11215 '034360': '166075400'                 MC = S                        " N
11216 '034361': '262077777'                 A '*' 32767                   " NUMBER OF PROCEDURE
11217 '034362': '000000502'                 A+ BEGIN OF CROSSTABLE
11218 '034363': '126073400'                 S = MA
11219 '034364': '361117715'              U, S '*' D25 ,Z                  " NO INDICATION IN CROSSTABLE?
11220 '034365': '370317715'              N, S '*' -D25                    ") REMOVE
11221 '034366': '166373400'              N, MA = S                        ") INDICATION
11222 '034367': '126075376'                 S = MC[-2]
11223 '034370': '026074400'                 A = MS                        " FIRST LETTERS OF PROC. NAME
11224 '034371': '240317714'              N, A '+' D24                     ") NON-EXISTING NAME
11225 '034372': '076374400'              N, MS = - A                      ") INTO CATALOGUE
11226 '034373': '126075400'                 S = MC                        " N
11227 '034374': '522034346'                 GOTO (:LOOP)
11228     114                               'END'   ANONYMIZE
11229
11230                          SYSTEM[-37]:
11231
11232 '012633':               START EXT LIB:
11233 '012633': '020500512'              U, A= NORMAL, P
11234 '012634': '522303036'              N, GOTO(:DNTST)
11235 '012635': '020000437'                 A= LENGTH OF EXT LIB
11236 '012636': '120017711'                 S= D21
11237 '012637': '040000451'                 COMPCELL[1]+A
11238 '012640': '040000436'                 TRANSPOSITION+A
11239 '012641': '140003056'                 DNTST[16]+S                   " REVERSE INSTRUCTION
11240 '012642': '522003036'                 GOTO(:DNTST)
11241
11242 '012643':               FINISH EXT LIB:
11243 '012643': '020500512'              U, A= NORMAL, P
11244 '012644': '522203036'              Y, GOTO(:DNTST)
11245 '012645': '030000437'                 A= -LENGTH OF EXT LIB
11246 '012646': '130017711'                 S=- D21
11247 '012647': '522012637'                 GOTO(:START EXT LIB[4])
11248
11249     115                MILLI TO TAPE: 'BEGIN'          DIRECTION TO DRUM, MILLI FROM TAPE,
11250                                        INSTR1, INSTR2, INSTR, SAVE
11251
11252 '012650': '020000451'                 A = COMP CELL[1]
11253 '012651': '060012674'                 SAVE = A
11254 '012652': '020000470'                 A = DR LIBR END
```

```
11255 '012653':  '060000451'                COMP CELL[1] = A
11256 '012654':  '020003314'                A = RD CMP[1]
11257 '012655':  '060012677'                INSTR = A
11258 '012656':  '020012675'                A = INSTR1
11259 '012657':  '060003314'                RD CMP[1] = A
11260 '012660':  '522003036'                GOTO(:DNTST)
11261
11262 '012661':            DIRECTION TO DRUM:
11263 '012661':  '020000421'                A = D18
11264 '012662':  '060000452'                COMP CELL[2] = A
11265 '012663':  '020012676'                A = INSTR2
11266 '012664':  '060003314'                RD CMP[1] = A
11267 '012665':  '660061000'                DVOFF
11268 '012666':  '026073400'                A = MA          " FULL STOP
11269
11270 '012667':            MILLI FROM TAPE:
11271 '012667':  '020012674'                A = SAVE
11272 '012670':  '060000451'                COMP CELL[1] = A
11273 '012671':  '020012677'                A = INSTR
11274 '012672':  '060003314'                RD CMP[1] = A
11275 '012673':  '522003314'                GOTO(:RD CMP[1])
11276
11277 '012674':            SAVE:           'SKIP' 1
11278 '012675':  '522012661'  INSTR1:       GOTO(:DIRECTION TO DRUM)
11279 '012676':  '522012667'  INSTR2:       GOTO(:MILLI FROM TAPE)
11280 '012677':            INSTR:          ' SKIP ' 1
11281     115              'END'
11282
11283                       "SUBMONITOR VARIABLES
11284
11285 '017213':            SUBMOVAR[0]:    COMPMODE:                   'SKIP' 1
11286 '017214':                            BASE    :                   'SKIP' 1
11287 '017215':                            BEG OF ADMI :               'SKIP' 1
11288 '017216':                            ASHIFT:                     'SKIP' 1
11289 '017217':                            APOS:                       'SKIP' 1
11290 '017220':                            OLD INSTR CNTR:             'SKIP' 1
11291 '017221':                            END OF PROGRAM:             'SKIP' 1
11292 '017222':                            RELADDR:                    'SKIP' 1
11293 '017223':                            RELADDR1:                   'SKIP' 1
11294 '017224':                            BEGINSAFE:                  'SKIP' 1
11295 '017225':                            SHIFTSAFE:                  'SKIP' 1
11296 '017226':                            WORDSAFE:                   'SKIP' 1
11297 '017227':                            PNTR:                       'SKIP' 1
11298 '017230':                            WORD1SAFE:                  'SKIP' 1
11299 '017231':                            BEGIN OF PROC:              'SKIP' 1
11300
11301                       BEGIN OF SUBMONITOR[0]:
11302
11303 '017232':  '000177400'                FRAME :                     '000 177 400 '
11304 '017233':  '222000001'                LOCAL NUMBER :              '222 000 001'
11305 '017234':  '000177777'                D16M1 :                     '000 177 777'
11306
11307     116              SUBMONITOR:     'BEGIN'                     LOOP, END
11308 '017235':  '020400512'                A= NORMAL, P
11309 '017236':  '522333303'                    N, GOTO(:EXTEND LIBRARY)
11310 '017237':  '022017574'  NORMAL PROGRAM: A=: BASE 0
11311 '017240':  '060017214'                BASE= A
11312 '017241':  '022000000'                A= 0
11313 '017242':  '060017213'                COMPMODE= A
11314 '017243':  '562422426'                SUBC(:PRESCAN 0)
```

```
11615 '017244':  '020416207'              A= ERRONECUS, P
11616 '017245':  '522203346'           Y, GOTO(:END RUN)
11617 '017246':  '020016966'              A= BEGIN OF PROGRAM
11618 '017247':  '000000436'              A + TRANSPOSITION                "ACTUAL
11619 '017250':  '060017227'              PNTR= A
11620 '017251':  '020000436'              A= TRANSPOSITION                 "INSTR CNTR + TRANSPOSITION
11621 '017252':  '012000001'              A - 1                            "REMOVE SUM OF MAXIMA
11622 '017253':  '060116215'              PLUSA(INSTR CNTR)
11623 '017254':  '126073400'              S =MA                            " SUM OF MAXIMA
11624 '017255':  '166075400'              MC= S
11625 '017256':  '000017223'              A + RELADDR1
11626 '017257':  '562417563'              SUBC(:OBSERVE)                   "SPACE FOR LIBRARY ROUTINES?
11627 '017260':  '622017317'              G=: LIBR CELL
11628 '017261':  '022000000'              A= 0                             "FROM DRUM
11629 '017262':  '066072402'              MG[2]= A
11630 '017263':  '122000002'  LOOP:       S= 2
11631 '017264':  '170117227'              MINS(PNTR)
11632 '017265':  '127074401'              S= MS[1], Z                      "LEADING ZERO OF PSEUDO  LVARS?
11633 '017266':  '522217310'           Y, GOTO(:END)
11634 '017267':  '100000002'              S + BEGIN OF CROSSTABLE          " NUMBER OF PROC. IN S
11635 '017270':  '020016215'              A = INSTR CNTR
11636 '017271':  '562417552'              SUBC(:PREPARE)
11637 '017272':  '562417524'              SUBC(:READ LIBRARY ROUTINE)      "FROM DRUM
11638 '017273':  '562417527'              SUBC(:ADAPT ADR)
11639 '017274':  '122000001'              S = 1
11640 '017275':  '170116215'              MINS (INSTR CNTR)                " FIRST FREE PLACE
11641 '017276':  '126074400'              S = MS                           " SUM OF MAXIMA OF PROCEDURE
11642 '017277':  '114567777'           U, S - M[B-1], P                    "SUM OF MAXIMA OF PROGRAM
11643 '017300':  '164267777'           Y, M[B-1]= S
11644 '017301':  '020017231'              A= BEGIN OF PROC                 "BEGIN OF PROGRAM OF PROCEDURE
11645 '017302':  '010000436'              A - TRANSPOSITION                " VIRTUAL
11646 '017303':  '120017227'              S= PNTR
11647 '017304':  '066074400'              MS= A
11648 '017305':  '002000001'              A + 1
11649 '017306':  '066074401'              MS[1]= A
11650 '017307':  '522017263'              GOTO(:LOOP)
11651 '017310':  '026075377'  END:        A= MC[-1]
11652 '017311':  '120016215'              S = INSTR CNTR                   " ACTUAL
11653 '017312':  '066074400'              MS = A                           ") FINAL SUM OF MAXIMA
11654 '017313':  '102000001'              S + 1                            ") INTO OBJECTPROGRAM
11655 '017314':  '110000436'              S - TRANSPOSITION                " VIRTUAL AGAIN
11656 '017315':  '160016215'              INSTR CNTR = S
11657 '017316':  '526075377'              GOTOR(MC[-1])                    "EXIT SUBMONITOR
11658     116                             'END' SUBMONITOR
11659
11660 '017317':  '000017317'  LIBR CELL:  :LIBR CELL
11661 '017320':                           'SKIP' 4
11662
11663 '017324':               READ LIBRARY ROUTINE:
11664 '017324':  '122017317'              S=: LIBR CELL
11665 '017325':  '562406312'              SUBC(:TRANSPORT)                 "GET PROCEDURE FROM DRUM
11666 '017326':  '522003345'              GOTO(:ENDTRANS)                  "WAIT FOR COMPLETION
11667
11668     117             ADAPT ADR:    'BEGIN' LOOP, LOOP3, NEXT WORD,CORRECTION, SAVE, IJU1, LAST
11669
11670 '017327':  '620116215'              G= INSTR CNTR
11671 '017330':  '612000002'              F - 2
11672 '017331':  '720117220'              OLD INSTR CNTR= G                "POINT OVER END OF ADMIWORDS
11673 '017332':  '026072400'              A= MG
11674 '017333':  '670000052'              RUA(10)                          "RELATIVE ADDRESS BEGIN OF PROGRAM
```

```
11375 '017334':  '120000372'                S= G
11376 '017335':  '110000373'                S - A                          "BEGIN OF PROGRAM OF PROCEDURE
11377 '017336':  '160017231'                BEGIN OF PROC= S
11378 '017337':  '160016215'                INSTR CNTR= S                  ") VIRTUAL INSTR CNTR
11379 '017340':  '110000436'                S - TRANSPOSITION              ")MINUS
11380 '017341':  '116072401'                S - MG[1]                      ") ABS. ADDRESS BEGIN OF PROCEDURE
11381 '017342':  '160017413'                CORRECTION= S                  ") IS CORRECTION
11382 '017343':  '026072400'                A= MG
11383 '017344':  '262001777'                A '*' 1023                     " NUMBER OF ADMIWORDS
11384 '017345':  '610100073'                G - A                          "POINTS TO FIRST ADMIWORD
11385 '017346':  '726175400'                MC = G                         "STACK IT
11386 '017347':  '460017217'                APOS= B                        "SAVE B
11387 '017350':  '422000001'                B = 1
11388 '017351':  '450016215'                INSTR CNTR-B                   "POINTS TO LAST HANDLED WORD
11389 '017352':  '422000000'  LOOP:         B= 0
11390 '017353':  '460017216'                ASHIFT= B                      "INITIALIZE ASHIFT
11391 '017354':  '026072400'                A= MG                          "ADMIWORD
11392 '017355':  '661000240'  LOOP3:        NORA, Z                        "WORD EMPTY?
11393 '017356':  '522217414'              Y, GOTO(:NEXT WORD)
11394
11395 '017357':  '660000041'                LUA(1)                         "SHIFT ONE BIT MORE
11396 '017360':  '402000001'                B + 1
11397 '017361':  '440017216'                ASHIFT + B
11398 '017362':  '460116215'                PLUS B(INSTR CNTR)
11399
11400 '017363':  '124040000'                S = M[B]
11401 '017364':  '362077777'                S '*' 32767                    " ADDRESS PART
11402 '017365':  '112507777'              U, S - 4095, P                   " ADDRESS PART > 4095?
11403 '017366':  '502000007'              N, JUMP(7)                       " NO, THEN NUMBER OF PROC.
11404 '017367':  '124040000'                S= M[B]
11405 '017370':  '372077777'                S '*' -32767                   " INSTRUCTION PART
11406 '017371':  '111117411'              U, S-LAST, Z                     " MACRO= LAST?
11407 '017372':  '120017413'                S= CORRECTION
11408 '017373':  '144340000'              N, M[B]+S                        ")ADAPT ADDRESS
11409 '017374':  '154240000'              Y, M[B]-S                        ")
11410 '017375':  '522017355'                GOTO(:LOOP3)
11411
11412 '017376':  '160017410'                SAVE = S
11413 '017377':  '174140000'                MINS (M[B])                    " CLEAR ADDRESS PART
11414 '017400':  '111017412'                S - JUJ1, Z                    " GOTO (LVAR1) ?
11415 '017401':  '120017410'                S = SAVE
11416 '017402':  '112200001'              Y, S - 1
11417 '017403':  '100000505'                S + BEGIN OF TRAFOTABLE
11418 '017404':  '126074400'                S = MS
11419 '017405':  '102200001'              Y, S + 1
11420 '017406':  '144040000'                M[B] + S                       " NEW ADDRESS PART
11421 '017407':  '522017355'                GOTO (:LOCP3)
11422 '017410':                  SAVE :     'SKIP' 1
11423 '017411':  '767700000'     LAST:      '767 700 000'
11424 '017412':  '520000000'     JUJ1:      '520 000 000'
11425 '017413':                  CORRECTION: 'SKIP' 1
11426
11427 '017414':  '410017216'     NEXT WORD: B-ASHIFT                       ")COUNT ONLY THE LAST
11428 '017415':  '440016215'                INSTR CNTR + B                 ")ZEROS OF THE ADMIWORD
11429 '017416':  '602000001'                G + 1                          "NEXT ADMIWORD
11430 '017417':  '020000072'                A = G
11431 '017420':  '011117220'              U, A-OLD INSTR CNTR, Z           "END OF ADMIWORDS?
11432 '017421':  '522317652'              N, GOTO(:LOOP)
11433 '017422':  '420017217'                B= APOS                        "RESTORE B
11434 '017423':  '126075377'                S= MC[-1]                      "FIRST ADMIWORD
```

```
11435 '017424':  '160016215'                 INSTR CNTR= S
11436 '017425':  '526475377'                 GOTOR(MC[-1])                    "EXIT
11437     117                                'END' ADAPT ADR
11438
11439 '017426':  '122000001'  NEXT ENTRY:        S = 1                        ") POINTER:= POINTER + 1
11440 '017427':  '160117227'                     PLUSS(PNTR)                  ")
11441 '017430':  '026074400'                     A = MS                       " CONTENTS OF CROSSTABLEWORD
11442 '017431':  '111100507'                  U, S - END OF CROSSTABLE,Z
11443 '017432':  '110000502'                     S - BEGIN OF CROSSTABLE      " NUMBER OF ENTRY IN S
11444 '017433':  '526075377'                     GOTO(MC[-1])                 " 6 INSTRUCTIONS
11445
11446 '017434':  '120000502'  INIT POINTER:      S = BEGIN OF CROSSTABLE
11447 '017435':  '160017227'                     PNTR    = S
11448 '017436':  '526475377'                     GOTOR(MC[-1])                " 3 INSTRUCTIONS
11449
11450     118                CRF:           'BEGIN'   CRF1, CRSS,  PSEUDO LVAR, LOOP, END
11451
11452 '017437':  '562417434'                  SUBC (: INIT POINTER)
11453 '017440':  '562417426'  CRF1:           SUBC(:NEXT ENTRY)
11454 '017441':  '522217467'               Y, GOTO(:PSEUDO LVAR)
11455 '017442':  '261117710'               U, A '*' D20, Z                     " PROCEDURE CALLED?
11456 '017443':  '562717445'               N, SUBC(:CRSS)                      " WORK OUT CROSSREFERENCE
11457 '017444':  '522017440'                  GOTO(: CRF1)
11458
11459     119                CRSS:          'BEGIN'  END, CYCLE
11460                                                                           ") CRSS EXPECTS IN S AN ENTRY
11461                                                                           ") OF THE CROSSTABLE AND BESIDES
11462                                                                           ") THAT IF CALLED FROM OUTSIDE IN
11463                                                                           ") A AN ENTRY OF THE INFOTABLE
11464 '017445':  '100000502'                  S + BEGIN OF CROSSTABLE
11465 '017446':  '066075400'                  MC = A
11466 '017447':  '026074400'                  A = MS
11467 '017450':  '261117711'               U, A '*' D21, Z                     " ALREADY NEEDED?
11468 '017451':  '522317465'               N, GOTO(:END)                       " THEN EXIT CRSS
11469 '017452':  '240017711'                  A '+' D 21
11470 '017453':  '066074400'                  MS = A                           " NOTICE PROCEDURE NEEDED
11471 '017454':  '127073402'  CYCLE:          S = MA[2], Z                     " ANY OTHER PROCEDURE NEEDED?
11472 '017455':  '522217465'               Y, GOTO(:END)                       " IF NOT EXIT CRSS
11473 '017456':  '362017777'                  S '*' 8191
11474 '017457':  '562417445'                  SUBC(:CRSS)
11475 '017460':  '126073402'                  S = MA[2]
11476 '017461':  '771000055'                  RUS(13), Z                       " ANY OTHER PROCEDURE NEEDED?
11477 '017462':  '562717445'               N, SUBC(:CRSS)
11478 '017463':  '002300001'               N, A + 1                            " NEXT ENTRY OF INFOTABLE
11479 '017464':  '522317454'               N, GOTO(:CYCLE)
11480 '017465':  '026075377'  END:            A = MC[-1]                       " RESTORE ENTRY OF INFOTABLE
11481 '017466':  '526475377'                  GOTOR(MC[-1])
11482
11483     119                                'END' CRSS
11484
11485 '017467':  '122000001'  PSEUDO LVAR:     S = 1
11486 '017470':  '160017222'                  RELADDR = S
11487 '017471':  '160017223'                  RELADDR1 = S
11488 '017472':  '100000436'                  S + TRANSPOSITION                "ACTUAL INSTR CNTR
11489 '017473':  '160116215'                  PLUSS (INSTR CNTR)
11490 '017474':  '022000000'                  A = 0                            ") LEADING ZERO OF
11491 '017475':  '066074377'                  MS [-1] = A                      ") PSEUDO LVARS
11492 '017476':  '562417434'                  SUBC (:INIT POINTER)
11493 '017477':  '562417426'  LOOP:           SUBC (:NEXT ENTRY)
11494 '017500':  '522217530'               Y, GOTO (:END)
```

```
11495 '017501':  '626173401'                G = MA [1]                          " LENGTH IN CORE
11496 '017502':  '261117711'             U, A '*' D21, Z
11497 '017503':  '261317714'             N, A '*' D24, Z
11498 '017504':  '522217477'             Y, GOTO (:LOOP)
11499 '017505':  '020000074'                A= S
11500 '017506':  '000000502'                A + BEGIN OF CROSSTABLE
11501 '017507':  '060000077'                D = A
11502 '017510':  '036000400'                A = - M0
11503 '017511':  '006001400'                A + M1                              " LENGTH ON DRUM
11504 '017512':  '000017222'                A + RELADDR
11505 '017513':  '010517223'             U, A - RELADDR1, P
11506 '017514':  '060217223'             Y, RELADDR1 = A                        " MAX. VALUE IN RELADDR1
11507 '017515':  '020000072'                A = G                               " LENGTH IN CORE
11508 '017516':  '040017222'                RELADDR + A
11509 '017517':  '022000002'                A= 2
11510 '017520':  '060116215'                PLUS A(INSTR CNTR)
11511 '017521':  '166073376'                MA[-2]= S                           " PSEUDO LVAR
11512 '017522':  '166073377'                MA[-1]= S                           " PSEUDO LVAR[1]
11513 '017523':  '100000500'                S + BEGIN OF TRAFOTABLE
11514 '017524':  '010000436'                A - TRANSPOSITION                   ")VIRTUAL
11515 '017525':  '012000002'                A - 2                               ")  ADDRESS OF
11516 '017526':  '066074400'                MS = A                              ")  PSEUDO LVAR
11517 '017527':  '522017477'                GOTO (: LOOP)
11518 '017530':  '020000436'  END:          A = TRANSPOSITION
11519 '017531':  '050016215'                INSTR CNTR - A                      " VIRTUAL AGAIN
11520 '017532':  '526475377'                GOTOR (MC[-1])
11521     118                               'END' CRF
11522
11523 '017533':  '026074400'  ADDR OF LVAR:  A = MS                              " NAMELIST[N]
11524 '017534':  '166075400'                MC = S
11525 '017535':  '120000073'                S = A
11526 '017536':  '272077777'                A '*' -32767
11527 '017537':  '362077777'                S '*' 32767                         " NUMBER OR ADDRESS OF PROCEDURE
11528 '017540':  '112507777'             U, S - 4095, P                         " NOT FIRST CALL?
11529 '017541':  '126275377'             Y, S = MC[-1]
11530 '017542':  '526675377'             Y, GOTOR (MC[-1])
11531 '017543':  '066075400'                MC = A
11532 '017544':  '100000500'                S + BEGIN OF TRAFOTABLE
11533 '017545':  '026074400'                A = MS
11534 '017546':  '006075377'                A + MC[-1]
11535 '017547':  '126075377'                S = MC[-1]
11536 '017550':  '066074400'                MS = A                              " NAMELIST [N]
11537 '017551':  '526475377'                GOTOR (MC[-1])
11538
11539 '017552':  '622017617'  PREPARE:       G = :LIBR CELL
11540 '017553':  '066072403'                MG[3] = A
11541 '017554':  '160000077'                D= S
11542 '017555':  '136000400'                S= - M0
11543 '017556':  '176072404'                MG[4]= - S                          "ADDRESS ON DRUM
11544 '017557':  '106001400'                S + M1
11545 '017560':  '166072401'                MG[1]= S                            " LENGTH INCLUDING ADMINISTRATION
11546 '017561':  '140016215'                INSTR CNTR + S                      " NEW INSTR CNTR
11547 '017562':  '526475377'                GOTOR(MC[-1])
11548
11549 '017563':  '010000436'  OBSERVE:       A - TRANSPOSITION                   "VIRTUAL INSTR CNTR
11550 '017564':  '012577777'             U, A - '77 777', P                     "BEYOND32K?
11551 '017565':  '526775377'             N, GOTOR(MC[-1])
11552 '017566':  '620100073'                G = A
11553 '017567':  '720116215'                INSTR CNTR= G                       "VIRTUAL AGAIN
11554 '017570':  '022000755'                A= 493
```

```
11555 '017571':  '610100434'                G - BEGIN OF PR AR              ") TOTALE LENGTH OF OBJECT-
11556 '017572':  '720016221'                VALUE OF CONSTANT=F             ") PROGRAM INTO ERRORM
11557
11558 '017573':  '522012772'                GOTO(:ERM)
11559
11560 '017574':  '00003014C'  BASE0:        :PROGRAM3
11561 '017575':  '000027023'                :CMPTL 3
11562 '017576':  '000030042'                :DECLST 3
11563 '017577':  '002000000'                A + 0
11564 '017600':  '002000000'                A + 0
11565 '017601':  '002000000'                A + 0
11566 '017602':  '562417437'                SUBC(:CRF)
11567 '017603':  '002000000'                A + 0
11568 '017604':  '002000000'                A + 0
11569 '017605':  '002000000'                A + 0
11570 '017606':  '562417536'                SUBC (:ADDR OF LVAR)
```