

NN31545.1302

NOTA 1302 ^{II}

juli 1981

Instituut voor Cultuurtechniek en Waterhuishouding
Wageningen

ASPECTEN VAN INFORMATIEVERWERKING

27

TOEPASSING EN GEBRUIK VAN LIBRARIES OP EEN CYBER COMPUTER

J.R. Maassen en J.H. Vink

**BIBLIOTHEEK
STABINGSGEDUW**

Nota's van het Instituut zijn in principe interne communicatie-middelen, dus geen officiële publikaties.

Hun inhoud varieert sterk en kan zowel betrekking hebben op een eenvoudige weergave van cijferreeksen, als op een concluderende discussie van onderzoeksresultaten. In de meeste gevallen zullen de conclusies echter van voorlopige aard zijn omdat het onderzoek nog niet is afgesloten.

Bepaalde nota's komen niet voor verspreiding buiten het Instituut in aanmerking

15N 165778-122



0000 0386 4481

ASPECTEN VAN INFORMATIEVERWERKING

27

De nota's handelende over Aspecten van Informatieverwerking bevatten inlichtingen over de ontwikkeling van de informatieverwerking binnen het Instituut. Naast meer concluderende en toelichtende beschouwingen zal aandacht worden besteed aan het gebruik van programma's en programmapakketten en zullen zakelijke inlichtingen over praktijkervaring met en toepassing van de informatieverwerking worden gegeven

I N H O U D

	Blz.
1. INLEIDING	1
2. HET IN WERKING STELLEN VAN EEN LIBRARY	1
3. HET SYSTEEMPROGRAMMA EDITLIB	3
4. HET NOS COMMANDO EDITLIB	4
5. EDITLIB OPDRACHTEN	5
6. NAAMGEBRUIK	8
7. ALFABETISCH GERANGSCHIKTE EDITLIB OPDRACHTEN	9
ADD	10
CONTENT	12
DELETE	13
ENDRUN	14
FINISH	15
LIBRARY	16
LISTLIB	17
RANTOSEQ	18
REPLACE	19
REWIND	21
SETAL	23
SETFL	24
SETFLO	25
SEQTORAN	
SKIPB	26
SKIPF	27
8. ENKELE SAMENVATTENDE OPMERKINGEN	28

REFERENTIES

Voorbeeld 1. Het aanmaken van een library bestaande uit subroutines	
2. Het aanmaken van een library bestaande uit zelfstandige programma's	40
3. Het aan een library toevoegen van programma's uit een andere library	49
4. Het vervangen van programma's in een library	55
5. Het verwijderen van programma's uit een library	61
6. Het kopiëren van libraries in random format	67

1. INLEIDING

In deze nota in de reeks Aspecten van Informatieverwerking zal worden ingegaan op de toepassingsmogelijkheden van gebruikers-libraries.

Een gebruikers-library is een bibliotheek van vertaalde programma's of subroutines die een gebruiker vanuit zijn eigen programma kan aanroepen.

Van libraries kan nuttig gebruik worden gemaakt wanneer men over een programmapakket beschikt. In een library kunnen vertaalde programma's of subroutines worden opgenomen, die allemaal iets te maken hebben met een bepaald onderwerp. Deze kunnen dan op een enkele file worden bewaard.

De hier besproken mogelijkheden hebben betrekking op het gebruik van libraries op CYBER-computers, zoals van IWIS-TNO te 's-Gravenhage. De in deze nota gebruikte programma's en subroutines dienen ter verduidelijking van de afzonderlijke mogelijkheden.

2. HET IN WERKING STELLEN VAN EEN LIBRARY

Bij het gebruik van libraries wordt onderscheid gemaakt tussen zelfstandige programma's en subroutines.

Het in werking stellen van een reeds gecreëerde library gaat als volgt:

```
ATTACH,lib,libnaam,ID=itl  
LIBRARY,lib
```

Wanneer men een zelfstandig programma uit een library wil gebruiken kan worden volstaan met het aanroepen van het programma. De com-

puter zoekt vervolgens het betreffende programma op in de library en voert het uit.

Voorbeeld

ATTACH,PROGLIB,PROGRAMLIBRARY,ID=JRM	- aanroepen van de library
LIBRARY,PROGLIB	- laden van de library
FACUL	- uitvoeren van het programma FACUL

Wanneer men een subroutine uit een library wil gebruiken moet de subroutine in het zelfstandige programma worden aangeroepen. De computer zoekt vervolgens de betreffende subroutine op in de library en behandelt de subroutine verder als ware het een subroutine van het eigen (hoofd)-programma.

Voorbeeld

ATTACH,PROG,PROGRAMMA ,ID=VIN	- aanroepen van het programma
ATTACH,SUBLIB,SUBLIBICW,ID=VIN	- aanroepen van de library
LIBRARY,SUBLIB	- laden van de library
FTN,I=PROG	- vertalen van het hoofdprogramma
LGO	- uitvoeren van het programma

Stel, er is een library waarin de subroutine FACUL voorkomt. Deze subroutine betekent de faculteit van een op te geven getal. In het programma wordt deze subroutine als volgt in werking gesteld:

```
PROGRAM REKEN(INPUT,OUTPUT)
.
.
CALL FACUL(inp,out)
.
.
PRINT *,out
.
.
END
```

3. HET SYSTEEMPROGRAMMA EDITLIB

Het systeemprogramma EDITLIB verzorgt het hele gebeuren rond gebruikers-libraries.

Een gebruikers-library, aangemaakt door EDITLIB, bevat:

- programma's en/of subroutines in ASSEMBLER-taal
- tabellen, betrekking hebbend op entry points
 - external references
 - programmanummers
 - programmanamen

Een gebruikers-library mag maximaal 2047 programma's,
2047 entry points
en 2047 external references

bevatten.

In een programma mogen maximaal 124 entry points
en 124 external references

voorkomen.

Een library kan zowel op disc als op tape worden opgeslagen.
Wanneer een library op tape wordt opgeslagen, komen de tabellen voor
de programma's te staan:

```

I           I           I           I           I
I tabel 1 I tabel 2 I tabel ... I programma I.....
I-----I-----I-----I-----I-----
```

De opslag is in sequentieel format.

Wanneer echter een library op disc wordt opgeslagen, gebeurt dit
in random format. Hierbij moet erop worden gelet dat de naam van de
library aan het permanent-file-device wordt gegeven, voordat EDITLIB
wordt aangeroepen.

Bij het kopiëren van libraries van disc naar tape en omgekeerd,
dient men respectievelijk de EDITLIB-opdrachten RANTOSEQ en SEQTORAN
te gebruiken. Een gewone COPY-opdracht werkt niet correct!

4. HET NOS COMMANDO EDITLIB

Aangezien EDITLIB niet interactief werkt, moeten alle manipulaties met een gebruikers-library worden uitgevoerd in een batch-job. Het EDITLIB-commando komt hierbij voor in het besturings-record van de job. Het EDITLIB-commando ziet er als volgt uit:

EDITLIB(user, I=lfnl,L=lfn2).

Hierin is user - een naam, maximaal 7 letters of cijfers, beginnend met een letter, die de library onderscheidt van systeemlibraries. Default is USER;

lfnl - logical file naam, maximaal 7 letters of cijfers, beginnend met een letter. Deze file bevat EDITLIB-opdrachten. Default is INPUT, terwijl I staat voor Input;

lfn2 - logical file naam, maximaal 7 letters of cijfers, beginnend met een letter. Op deze file wordt output weggeschreven. Default is OUTPUT, terwijl L staat voor List.

Voorbeeld:

```
BIJOB.  
ACCOUNT,in,projcode,cd.  
ATTACH,LIB,ID=ITL.  
FTN,I=PROG,B=BINPROG,SYSEDIT.  
EDITLIB(USER).  
EXTEND,LIB.  
eor  
  
FORTRAN programma  
  
eor  
  
EDITLIB-opdrachten die het vertaalde programma toe-  
voegen aan library LIB  
  
eof
```

5. EDITLIB OPDRACHTEN

Elke opdracht die EDITLIB moet uitvoeren, moet in de eerste 72 posities van een ponskaart of regel van het beeldscherm worden gespecificeerd. Spaties mogen naar willekeur worden gebruikt. Zij worden door EDITLIB verwijderd, tenzij het om een stukje commentaar gaat. Het format van de EDITLIB-opdrachten is gelijk aan dat van NOS opdrachten:

KEYWORD. of KEYWORD(PARAMETER LIST).

Een parameter list na een keyword moet altijd tussen haakjes staan. Parameters met een optioneel karakter mogen in willekeurige volgorde voorkomen en hebben het volgende format:

PARAMETER = waarde

Alle andere parameters moeten in een vaste volgorde voorkomen, eventueel gevolgd door niet verplichte parameters.

In het kort wordt hier een overzicht gegeven van toegestane EDITLIB-opdrachten.

I-----I	I-----I	I
I Opdracht	I Betekenis	I
I-----I	I-----I	I
I	I	I
I LIBRARY(libname,-- ^{OLD} --)	I Definieert een nieuw aan te maken	I
I ^{NEW}	I library(NEW) of een te wijzigen	I
I	I library(OLD) met de naam libname	I
I	I	I
I FINISH.	I Beëindigt manipulaties met de	I
I	I onderhavige library	I
I	I	I
I ENDRUN.	I Beëindigt het uitvoeren van	I
I	I EDITLIB opdrachten	I
I	I	I
I ADD	I Voegt een nieuw programma toe aan	I
I	I een library	I
I	I	I
I REPLACE	I Vervangt een programma in een	I
I	I bestaande library	I
I	I	I
I DELETE	I Verwijdt een programma uit een	I
I	I library	I
I	I	I
I SETAL	I Wijzigt het acces-level van een	I
I	I programma	I
I	I	I
I SETFL	I Wijzigt de field-length van een	I
I	I programma	I
I	I	I
I SETFLO	I Plaatst een override-bit ten	I
I	I behoeve van INTERCOM	I
I	I	I
I LISTLIB	I Geeft informatie omtrent een	I
I	I programma in een library	I
I	I	I
I REWIND	I Zet pointer op beginpunt van file	I

I CONTENT	I Zie LISTLIB	I
I	I	I
I SKIPF	I Skipt files of records	I
I	I vooruit	I
I	I	I
I SKIPB	I Skipt files of records terug	I
I	I	I
I */	I Voegt commentaar tussen	I
I	I output	I
I	I	I
I RANTOSEQ	I Maakt van een random file	I
I	I een sequential file	I
I	I	I
I SEQTORAN	I Maakt van een sequential	I
I	I file een random file	I
I	I	I
I-----I	I-----I	I

6. NAAMGEBRUIK

Namen van programma's en/of subroutines mogen niet langer zijn dan 7 characters. Ieder door het systeem geaccepteerd character mag worden gebruikt.

Als er in de naam characters voorkomen die als scheidingstekens worden gebruikt, moet de gehele naam tussen dollartekens (\$) worden geplaatst.

Toegestane scheidingstekens zijn:

\$ () - + = . , / blank

Wil men een dollarteken in de naam opnemen, dan moet dit teken worden voorafgegaan door nog een dollarteken.

Een sterretje kan programmanamen vervangen, EDITLIB behandelt dan alle programma's van de desbetreffende file.

Als men manipulaties met een reeks aaneenvolgende programma's wil uitvoeren, is het niet noodzakelijk elk programma afzonderlijk te noemen. Men kan volstaan met het vermelden van het eerste programma gevolgd door een plusteken (+) en de naam van het laatste programma van de reeks. B+D in de reeks A,B,C,D,E,F betekent dus B, C en D.

Omgekeerd is het mogelijk om van een reeks programmanamen een aaneengesloten reeks niet bij de manipulaties te betrekken. B-D in de reeks A,B,C,D,E,F betekent dus dat alleen A, E en F worden verwerkt.

Enkele voor EDITLIB acceptabele namen zijn:

Opgegeven naam	Wordt door EDITLIB verwerkt tot
PROG12	PROG12
\$PROG12\$\$\$	PROG12\$
\$I-O\$	I-O
AA BB	AABB
\$AA BB\$	AA BB
3AB	3AB

Namen van files mogen nooit beginnen met ZZZZZ omdat deze combinatie gereserveerd is voor systeemprogramma's.

7. ALFABETISCH GERANGSCHIJKTE EDITLIB OPDRACHTEN

In het hierna volgende worden alle EDITLIB-commando's in alfabetische volgorde behandeld.

Programma toevoegen aan een library

ADD(prog,lfn,AL=level,FL=fl,FLO={⁰₁},LIB)

Hierin zijn de parameters prog en lfn verplicht, alle andere parameters kunnen naar keuze worden toegepast.

prog	- Naam van aan de library toe te voegen programma c.q. reeks van programma's.
lfn	- Local file, die de vertaalde programma's bevat. De naam bestaat uit maximaal 7 letters en/of cijfers, beginnend met een letter.
AL=level	- Acces Level, bestaande uit een oktaal getal van maximaal 4 cijfers. AL geeft aan of een INTERCOM-gebruiker een zelfstandig hoofdprogramma uit een library mag gebruiken: acces level is even - programma kan niet worden gebruikt acces level is oneven - programma kan wel worden gebruikt Default waarde van AL is 0, wat als een even getal wordt beschouwd.
FL=fl	- Field Length, nodig voor het laden en uitvoeren van het programma(0 - 377777 oktaal). Als Fl=0 wordt de laatst opgegeven field length gebruikt. Default waarde van FL is 0.
FLO={ ⁰ ₁ }	- Field Length Override bit. FLO = 1 overschrijft de laatste opgegeven field length. FLO = 0 overschrijft de laatst opgegeven field length niet. Default waarde van FLO is 0.
LIB	- Geeft aan dat de parameter lfn een bestaande library is.

Indien men AL, FL, of FLO waarden in een nieuwe library tabel wil opnemen moeten de waarden expliciet worden opgegeven in de EDITLIB opdracht. Dit geldt ook wanneer programma's aan de library worden toegevoegd, die reeds onderdeel zijn van een andere gebruikers-library.

Wil men de waarden van deze parameters in een bestaande library veranderen dan moeten de opdrachten SETAL, SETFL en SETFLO worden gebruikt.

Voorbeelden:

OPDRACHT	RESULTAAT
ADD(*,HUIS)	Alle programma's vanaf de positie van de pointer tot de end-of-file van file HUIS worden aan de library toegevoegd.
ADD(BOOM,BOS,FL=12300)	In file BOS wordt programma BOOM opgezocht. Er wordt een field length van 12300 gereserveerd voor het draaien van programma BOOM.
ADD(STOOM-IJS,WATER)	Alle programma's van file WATER, behalve de programma's STOOM t/m IJS worden aan de library toegevoegd.
ADD(**KOE,VEE)	Van file VEE worden alle programma's vanaf de positie van de pointer t/m programma KOE aan de library toegevoegd.
ADD(GROEN/GEEL,KLEUR)	Van file KLEUR worden de programma's GROEN en GEEL aan de library toegevoegd.
ADD(COBOL,TALEN,LIB)	Vanaf de bestaande library TALEN wordt programma COBOL aan de onderhavige library toegevoegd.

Listen van programma's

CONTENT(prog,lfn)

|
| prog - Naam van te listen programma of reeks van programma's.
|
| lfn - Local file naam waarop de te listen programma's staan.
|

Voorbeelden:

I-----I	I-----I
I OPDRACHT	I RESULTAAT
I-----I	I-----I
I	I
I CONTENT(BLAUW,KLEUR)	I Van file KLEUR wordt programma BLAUW
I	I gelist.
I	I
I CONTENT(ROOD+GEEL,KLEUR)	I Van file KLEUR worden de programma's
I	I ROOD t/m GEEL gelist.
I	I
I-----I	I-----I

Zie ook bij LISTLIB.

Verwijderen van programma's

DELETE(prog)

|
| prog - Naam van te verwijderen programma of reeks van programma's.
|

Voorbeelden:

I-----I	I-----I	I
I OPDRACHT	I RESULTAAT	I
I-----I	I-----I	I
I	I	I
I DELETE(VOORN)	I Programma VOORN wordt uit de library	I
I	I verwijderd.	I
I	I	I
I DELETE(**HARING-SNOEK)	I Alle programma's vanaf de positie van	I
I	I de pointer t/m programma HARING,	I
I	I behalve programma SNOEK, worden uit de	I
I	I library verwijderd.	I
I	I	I
I-----I	I-----I	I

Het beëindigen van EDITLIB manipulaties

ENDRUN.

ENDRUN beëindigt de uitvoering van library manipulaties.

In de meeste gevallen zal het dus de laatste EDITLIB opdracht zijn. Het is echter mogelijk de ENDRUN eerder in de reeks opdrachten te plaatsen; dit heeft tot gevolg dat de daarna komende opdrachten niet worden uitgevoerd, wel wordt er door EDITLIB gecontroleerd of de opdrachten correct zijn. Wanneer men in een EDITLIB job als eerste opdracht een ENDRUN geeft dan vindt er totaal geen executie plaats maar is het geheel een testrun.

Het beëindigen van file manipulaties

FINISH.

Met de opdracht FINISH wordt aangegeven dat er geen verdere veranderingen in de library worden aangebracht.

Wel kunnen dan nog programma listings worden verkregen en file formats worden veranderd.

Definiëren van een library

LIBRARY(libname, $\frac{\text{old}}{\text{new}}$)

- | libname - Naam van de library waarmee gewerkt zal gaan worden.
- | old - Wordt gebruikt wanneer de library reeds bestaat.
- | new - Wordt gebruikt wanneer er een nieuwe library wordt gecreëerd.

De plaats van de LIBRARY opdracht in een EDITLIB job is voor de eerste opdracht die de inhoud van een library verandert.

Commentaar en file-format veranderende opdrachten mogen aan de LIBRARY opdracht vooraf gaan.

Elke LIBRARY opdracht moet later in de job gevolgd worden door een FINISH.

Listen van library files

LISTLIB(prog, lfn)

|
| prog - Naam van te listen programma of reeks van programma's.
|
| lfn - Naam van de file waar de te listen programma's op staan.
|

LISTLIB kan alleen files in library format listen, CONTENT heeft deze beperking niet.

Voorbeelden:

I-----I	I-----I	I
I OPDRACHT	I RESULTAAT	I
I-----I	I-----I	I
I	I	I
I LISTLIB(PAARD,BEEST)	I Van file BEEST wordt programma PAARD	I
I	I gelist.	I
I	I	I
I LISTLIB(MEI-JUNI,JAAR)	I Van file JAAR worden alle programma's	I
I	I behalve MEI t/m JUNI gelist.	I
I	I	I
I-----I	I-----I	I

Het converteren van random format naar sequentieel format.

Library files kunnen alleen naar tape gecopieerd worden indien zij sequentieel zijn.

RANTOSEQ maakt van een random library file een sequentiële library file die dezelfde programma's bevat.

RANTOSEQ(rlfn,slfn)

	rlfn	- File naam van de random library file.
	slfn	- File naam van de sequentiële library file.

Vervangen van programma's in een library

REPLACE(prog,lfn,AL=level,FL=f1,FLO= $\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$),LIB)

- prog - Naam van het te vervangen programma c.q. de reeks te vervangen programma's. Oude programma's met dezelfde naam als de nieuwe programma's worden verwijderd voordat de nieuwe programma's erin gezet worden. Wanneer nieuwe programma's niet dezelfde naam hebben als de oude programma's, volgt een boodschap en worden de nieuwe programma's alsnog in de library gezet. De oude programma's worden dan niet verwijderd.
- lfn - Naam van de file waarop de nieuwe programma's staan.
- AL=level - Acces Level, bestaande uit een oktaal getal van maximaal 4 cijfers.
AL geeft aan of een INTERCOM-gebruiker een zelfstandig hoofdprogramma uit een library mag gebruiken:
acces level is even: programma mag niet worden gebruikt
acces level is oneven: programma mag wel worden gebruikt
Default waarde van AL is 0, wat als een even getal wordt beschouwd.
- FL=f1 - Field Length, nodig voor het laden en uitvoeren van het programma(0 - 377777 oktaal).
Als F1=0 wordt de laatst opgegeven field length gebruikt.
Default waarde van FL is 0, de eerste maal wordt dan de field length genomen zoals deze in de jobkaart voorkomt.
- FLO= $\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$ - Field Length Override bit.
FLO = 1 overschrijft de laatst opgegeven field length.
FLO = 0 overschrijft de laatst opgegeven field length niet.
Default waarde van FLO is 0.
- LIB - Geeft aan dat de parameter lfn een bestaande library is.

Voorbeelden:

I-----I	I-----I	I-----I
I	OPDRACHT	I
I		RESULTAAT
I-----I	I-----I	I-----I
I REPLACE(IJS,WATER,FLO=1)	I	I Het bestaande programma IJS wordt verwijderd,
I	I	I waarna van file WATER het nieuwe programma IJS
I	I	I aan de library wordt toegevoegd. Parameter FLO
I	I	I krijgt hierbij de waarde 1, de andere parameters
I	I	I krijgen de default waarde.
I	I	I
I REPLACE(LOGAR,FUNCTIE,LIB)	I	I Het bestaande programma LOGAR wordt verwijderd,
I	I	I waarna programma LOGAR van de library FUNCTIE
I	I	I wordt toegevoegd.
I	I	I
I REPLACE(GROEN/GEEL,KLEUR)	I	I De bestaande programma's GROEN en GEEL worden
I	I	I verwijderd, waarna de nieuwe programma's GROEN en
I	I	I GEEL van file KLEUR worden toegevoegd.
I	I	I
I-----I	I-----I	I-----I

Rewinden van files

REWIND(lfn) of REWIND(lfn1/lfn2/.../lfnx)

|
| lfn - Naam van de te rewinden file(s).
|

Het converteren van sequentieel format naar random format

Om library files op disc te kunnen bewaren is het nodig dat ze random zijn.

SEQTOTAN maakt van een sequentiële library file een random library file die dezelfde programma's bevat.

SEQTORAN(slf,rlf)

| slf - File naam van de sequentiële library file.

| rlf - File naam van de random library file.

Veranderen van acces level

SETAL(prog,level)

prog	- Naam van programma of reeks van programma's.
level	- Nieuw acces level bestaande uit max. 4 cijfers (oktaal).

Veranderen van field length

SETFL(prog,f1)

| prog - Naam van programma of reeks van programma's.

| f1 - Nieuwe field length (0-377777 oktaal).

Aanbrengen van field length override bit ten behoeve van INTERCOM

SETFLO(prog, {⁰₁})

| prog - Naam van programma of reeks van programma's.

| 0 - Er wordt geen override bit geplaatst (default).

| 1 - Er wordt wel een override bit geplaatst.

Het terug skippen van records en files

SKIPB laat de pointer in de library een of meer records c.q. files terug skippen. De pointer gaat hierbij naar het begin van het record of van de file. Indien er bij de skip opdracht een aantal is opgegeven, wordt bij een begin-of-information (boi) of een end-of information (eoi) gestopt met skippen. Wanneer bij de skip opdracht een naam is opgegeven wordt zonodig de gehele file doorzocht in de opgegeven richting. Het opgeven van een naam is alleen mogelijk bij sequentiële files.

1. Het terug skippen van records.

SKIPB($\begin{matrix} n \\ \text{prog} \end{matrix}$, lfn)

	n - Aantal terug te skippen records (>0).
	prog - Naam van het programma waar naar toe geskapt wordt.
	lfn - File waarop het programma staat.

2. Het terug skippen van files.

SKIPB (n, lfn, F)

	n - Aantal terug te skippen files (>0).
	lfn - File waar naar toe geskapt wordt.
	F - Aanduiding dat er files geskapt moeten worden.

Het vooruit skippen van records en files

SKIPF laat de pointer in de library een of meer records c.q. files vooruit skippen. De pointer gaat hierbij naar het begin van het record of van de file. Indien er bij de skip opdracht een aantal is opgegeven, wordt bij een begin-of-information (boi) of een end-of-information (eoi) gestopt met skippen. Wanneer bij de skip opdracht een naam is opgegeven wordt zonodig de gehele file doorzocht in de opgegeven richting. Het opgeven van een naam is alleen mogelijk bij sequentiële files.

1. Het vooruit skippen van records.

SKIPF(ⁿ_{prog},lfn)

- | n - Aantal vooruit te skippen records (>0).
- | prog - Naam van het programma waar naar toe geskipt wordt.
- | lfn - File waarop het programma staat.

2. Het vooruit skippen van files.

SKIPF(n,lfn,F)

- | n - Aantal vooruit te skippen files (>0).
- | lfn - File waar naar toe geskipt wordt.
- | F - Aanduiding dat er files geskipt moet worden.

8. ENKELE SAMENVATTENDE OPMERKINGEN

- Een library wordt aangemaakt door de library te definiëren in een LIBRARY opdracht, gevolgd door manipulatie opdrachten, afgesloten met een FINISH opdracht. Binnen 1 EDITLIB deck mogen meer dan 1 set opdrachten voorkomen die beginnen met een LIBRARY opdracht en eindigen met een FINISH opdracht. Na de laatste FINISH opdracht moet een ENDRUN opdracht voorkomen. Indien geen ENDRUN opdracht voorkomt wordt door EDITLIB een ENDRUN geplaatst na de laatste FINISH.
- Bestaande libraries in random format worden gewijzigd door de ADD, REPLACE en DELETE opdrachten. De opdrachten SETAL, SFTFL en SETFLO wijzigen in bestaande libraries de tabel met entry points van de programma's. De plaats van al deze opdrachten is tussen een LIBRARY en een FINISH opdracht.
- Het format van een library kan gewijzigd worden met de opdrachten RANTOSEQ en SEQTORAN.
- EDITLIB zet alle files, behalve INPUT, op hun beginpunt voordat met de eerste opdracht wordt begonnen. Wanneer random library files worden beschreven worden zij na de laatste opdracht weer op hun beginpunt gezet. Bij sequentiële library files staat, na de laatste opdracht, de pointer na het laatst behandelde programma in de library.
- Informatie over programma's of library files wordt verkregen door middel van de CONTENT en LISTLIB opdrachten. Deze informatie bevat:
 - programma naam
 - datum en tijd van compilatie
 - operating system van de computer
 - entry points
 - external references
 - AL en FL waarden
 - programma lengte (oktaal)
 - type programma: absolute of relocatable

REFERENTIES

CONTROL DATA, NOS/BE Version 1, Reference Manual

Control data corporation, 1980. St. Paul, Minnesota.

Voorbeeld 1. Het aanmaken van een library bestaande uit subroutines

De afzonderlijke onderdelen zijn:

1. batchjob met input voor EDITLIB
2. subroutines
3. output door EDITLIB

```

*****
*
*   BATCHJOB VOOR HET AANMAKEN VAN EEN LIBRARY
*
*   RESTAANDE UIT SUBROUTINES
*
*****

```

CVINK.

ACCOUNT,43,77101058,2S.

REQUEST,LIB,*PF.

ATTACH,LOG,SUBLOG,ID=VIN.

ATTACH,WOR,SUBWOR,ID=VIN.

ATTACH,FAC,SUBFAC,ID=VIN.

ATTACH,TAN,SUBTAN,ID=VIN.

FTN,I=LOG,R=3,B=BINLOG,SYSEDIT.

FTN,I=WOR,R=3,B=BINWOR,SYSEDIT.

FTN,I=FAC,R=3,B=BINFAC,SYSEDIT.

FTN,I=TAN,R=3,B=BITAN,SYSEDIT.

EDITLIB(USER).

CATALOG,LIB,LIBICH,ID=VIN,RP=999.

ZEOR

LIBRARY(LIB,NEW)

ADD(*,BINLOG)

ADD(*,BINWOR)

ADD(*,BINFAC)

ADD(*,BINTAN)

FINISH.

ENDRUN.

ZEOF

```

1      SUBROUTINE LOGAR
      COMMON ICW
      10 WRITE(3,100)
      100 FORMAT(1H,'VOER EEN POSITIEF GETAL IN : ')
      READ(3,*) X
      IF(X.LT.0.0000001) GOTO 10
      XLOG=ALOG10(X)
      101 WRITE(3,101) X,XLOG
      101 FORMAT(1H,'DE LOGARITHME VAN "F10.3" = "F10.5"')
      102 WRITE(3,102)
      102 FORMAT(1H,'WILT U J/N NOG EEN LOGARITHME UITGEREKEND HEBBEN? ')
      103 READ(3,103) ANTW
      103 FORMAT(A1)
      IF(ANTW.EQ.1HJ) GOTO 10
      RETURN
      END

```

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS	DEF LINE	REFERENCES
1 LOGAR	1	22

VARIABLES	SN	TYPE	RELOCATION	REFS
77 ANTW		REAL		18
0 ICW		INTEGER	/ /	2
75 X		REAL		8
76 XLOG		REAL		12
				10

FILE NAMES	MODE	WRITES	4	12	15	READS	7	18
TAPE3	MIXED							

EXTERNALS	TYPE	ARGS	REFERENCES
ALOG10	REAL	1	10

STATEMENT LABELS	DEF LINE	REFERENCES
2 10	4	8
25 100	5	4
43 101	13	12
54 102	16	15
70 103	19	18

COMMON BLOCKS	LENGTH	MEMBERS - BIAS NAME(LENGTH)
/ /	1	0 ICW (1)

```

1      SUBROUTINE WORTEL
      COMMON ICW

10     WRITE(3,100)
100    FORMAT(/1H,'VOER EEN POSITIEF GETAL IN : ')

      READ(3,*) X
      IF(X.LT.0.) GOTO 10
      XWOR=SQRT(X)

      WRITE(3,101) X,XWOR
101    FORMAT(/1H,'DE WORTEL VAN 'F10.3' = 'F10.5)

15     WRITE(3,102)
102    FORMAT(/1H,'WILT U J/N NOG EEN WORTEL UITGEREKEND HEBBEN? ')

      READ(3,103) ANTW
103    FORMAT(A1)
      IF(ANTW.EQ.1HJ) GOTO 10

      RETURN
      END
    
```

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS	DEF LINE	REFERENCES		
1 WORTEL	1	22		
VARIABLES	SN	TYPE	RELOCATION	
75 ANTW	REAL			REFS 18
0 ICW	INTEGER	/		REFS 2
73 X	REAL			REFS 8
74 XWOR	REAL			REFS 12
FILE NAMES	MODE			REFS 10
TAPES	MIXED			
EXTERNALS	TYPE	ARGS	REFERENCES	
500	REAL	1 LIBRARY	10	
STATEMENT LABELS	DEF LINE	REFERENCES		
2 10	4	8		20
24 100	5	4		
42 101	13	12		
53 102	16	15		
67 103	19	18		
COMMON BLOCKS	LENGTH	MEMBERS	- RIAS NAME(LENGTH)	
/	1	0	ICW (1)	


```

1  SUBROUTINE TANGENS
COMMON ICW
1 WRITE(3,100)
100 FORMAT(1H ,%GEEF NU DE GROOTTE VAN DE HOEK OP IN RADIALEN %/* HET
5  *PROGRAMMA ANTWOORDT HET DE TANGENS%/* HOEK*)
READ(3,101) HOEK
IF(EOF(3).NE.0) GOTO 2
101 FORMAT(F10.0)
TANG=SIN(HOEK)/COS(HOEK)
10 WRITE(3,102) HOEK,TANG
102 FORMAT(1H ,%DE TANGENS VAN%F10.3* RAD IS%F10.7/)
WRITE(3,103)
103 FORMAT(1H ,% TIK EOF ALS U WILT STOPPEN%/)
GOTO 1
15 2 RETURN
END

```

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS	DEF LINE	REFERENCES			
1 TANGENS	1	15			
VARIABLES	SN	TYPE	RELOCATION	REFS	
101 HOEK		REAL		2*9	10 DEFINED 6
0 ICW		INTEGER	/	2	
102 TANG		REAL		10	DEFINED 9
FILE NAMES	TAPES	MODE			
		FMT	WRITES	3	
				10	12 READS 6

EXTERNALS	TYPE	ARGS	REFERENCES
COS	REAL	1 LIBRARY	9
EOF	REAL	1 LIBRARY	7
SIN	REAL	1 LIBRARY	9

STATEMENT LABELS	DEF LINE	REFERENCES
2 1	3	14
22 2	15	7
31 100	4	3
52 101	8	6
61 102	11	10
72 103	13	12

COMMON BLOCKS	LENGTH	MEMBERS	BIAS NAME(LENGTH)
/	1	0	ICW (1)

STATISTICS	PROGRAM LENGTH	CM BLANK	COMMON LENGTH	CM USED
	1038	18	67	1
				560008

***** LIBRARY(LIB,NEW)

1 LIBRARY(LIB,NEW)

***** ADD(*,BINLOG)

2 ADD(*,BINLOG)

***** ADD(*,BINWOR)

3 ADD(*,BINWOR)

***** ADD(*,BINFAC)

4 ADD(*,BINFAC)

***** ADD(*,BINTAN)

5 ADD(*,BINTAN)

***** FINISH.

6 FINISH.

***** ENDRUN.

7 ENDRUN.

EDITLIP VERSION 2.4

PROCESSING DIRECTIVE NUMBER 1

PROCESSING DIRECTIVE NUMBER 2

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

LOGAR

SYSTEM NAME NOS/BE 1.309/07/81 10.05.24 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I INSTRUCTION 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 132 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

LOGAR

EXTERNAL REFERENCES

ALOGIO.

PROCESSING DIRECTIVE NUMBER 3

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

WORTEL

SYSTEM NAME NOS/BE 1.309/07/81 10.05.26 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I INSTRUCTION 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

WORTEL

EXTERNAL REFERENCES

INPFI.

PROCESSING DIRECTIVE NUMBER 4

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

FACUL

SYSTEM NAME NOS/BE 1.309/07/81 10.05.29 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I INSTRUCTION 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 143 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

FACUL

EXTERNAL REFERENCES

INPFI.

PROCESSING DIRECTIVE NUMBER 5

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

INPFI.

EXTERNAL REFERENCES

ALOGIO.

PREFIX TABLE INFORMATION

PROGRAM NAME

TANGENS

SYSTEM NAME ' NOS/BE 1.309/07/81 10.05.31

PROCESSOR NAME FTN 4.8 518

INSTRUCTION 64 4

DEPENDENCIES HARDWARE I

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 137 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

TANGENS

EXTERNAL REFERENCES

COS. EOF INPCI. OUTCI. SIN.

PROCESSING DIRECTIVE NUMBER 6

PROCESSING DIRECTIVE NUMBER 7

MFB 09/07/81 IWS-TNO NOS/BE 1.4MFB - 027 LEVEL 518
10.05.20.CUINKGA FROM MFA/BI
10.05.20.IF 00000192 WORDS FILE INPUT ,0081 04
10.05.20.CUINK.
10.05.20.ACCOUNT,43,77101058,*,
10.05.21.REQUEST,LIB,XPF,
10.05.22.ATTACH,LOG,SUBLOG,ID=VIN,
10.05.22.AT CY= 001 SN=SYSTEM
10.05.22.ATTACH,WDR,SUBWDR,ID=VIN,
10.05.23.AT CY= 001 SN=SYSTEM
10.05.23.ATTACH,FAC,SUBFAC,ID=VIN,
10.05.23.AT CY= 001 SN=SYSTEM
10.05.24.ATTACH,TAN,SUBTAN,ID=VIN,
10.05.24.AT CY= 001 SN=SYSTEM
10.05.24.FTN,I=LOG,R=3,R=BINLOG,SYSEDIT,
10.05.26. .217 CP SECONDS COMPILATION TIME
10.05.26.FTN,I=WDR,R=3,B=BINWDR,SYSEDIT,
10.05.29. .219 CP SECONDS COMPILATION TIME
10.05.29.FTN,I=FAC,R=3,B=BINFAC,SYSEDIT,
10.05.31. .296 CP SECONDS COMPILATION TIME
10.05.31.FTN,I=TAN,R=3,B=BIN TAN,SYSEDIT,
10.05.33. .216 CP SECONDS COMPILATION TIME
10.05.33.EDITLIB(USER),
10.05.36.CATALOG,LIB,LIBICW,ID=VIN,RP=999,
10.05.40.INITIAL CATALOG
10.05.43.CT ID= VIN PFN=LIBICW
10.05.43.CT CY= 001 SN=SYSTEM 0000896 WORDS.
10.05.44.DF 00061920 WORDS FILE OUTPUT ,0081 40
10.05.44.WS 7296 WORDS (38016 MAX USED)
10.05.44.CPA 1.358 SEC. 1.357 ADJ.
10.05.44.ID 7.518 SEC. 11.277 ADJ.
10.05.44.CM 4.239 KWS. 4.239 ADJ.
10.05.44.SS 16.875
10.05.44.FP 29.263 SEC. DATE 09/07/81
10.05.44.MAX FL USED 0560008
10.05.44.NT .000 SEC. (10)
10.05.44.AVERAGE FL/ CP-SEC (OCTAL) 47467
10.05.44.AVERAGE FL/ ID-SEC (OCTAL) 24465
10.05.44.EJ END OF JOB, BI

Voorbeeld 2. Het aannemen van een library bestaande uit zelfstandige programma's

De afzonderlijke onderdelen zijn:

1. batchjob met input voor EDITLIB
2. programma's
3. output door EDITLIB

```

*****
*
*   BATCHJOB VOOR HET AANMAKEN VAN EEN LIBRARY
*
*   BESTAANDE UIT ZELFSTANDIGE PROGRAMMA'S
*
*****

```

```

BIJRM.

ACCOUNT,43,77101058,2S.

REQUEST,PROGLIB,*PF.

ATTACH,SIN,SINUS ,ID=JRM.

ATTACH,WOR,WORTEL ,ID=JRM.

ATTACH,FAC,FACULTEIT,ID=JRM.

FTN,I=SIN,B=BISIN,SYSEDIT.

FTN,I=WOR,B=BIMOR,SYSEDIT.

FTN,I=FAC,B=BIFAC,SYSEDIT.

EDITLIB(USER).

CATALOG,PROGLIB,PROGLIB,ID=JRM,RP=999.

```

```

ZEUR

LIBRARY(PROGLIB,NEW)

ADD(*,BISIN,AL=1)

ADD(*,BIMOR,AL=1)

ADD(*,BIFAC,AL=1)

FINISH.

CONTENT(SINUS+FACUL,PROGLIB)

ENDRUN.

$EOF

```

```

1      PROGRAM SINUS(INPUT,OUTPUT)
      INTEGER ANTW
      CALL CONNEX(5LINPUT)
      CALL CONNEX(6LOUTPUT)
10     PRINT*, " GEEF EEN HOEK OP IN RADIALEN : "
      READ*, X
      XSIN=SIN(X)
      PRINT*, " DE SINUS VAN 'X,' RADIALEN = ',XSIN
20     PRINT*, " WILT U NOG EEN SINUS UITGEREKEND HEBBEN? "
      PRINT*, " (1=JA,0=NEE)
      READ*, ANTW
      IF (ANTW.NE.1.AND.ANTW.NE.0) GOTO 20
      IF (ANTW.EQ.1) GOTO 10
      END

```

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS					
4137 SINUS					
VARIABLES		SN	TYPE	RELOCATION	
4223 ANTW			INTEGER	4224 X	REAL
4225 XSIN			REAL		
FILE NAMES		MODE	2054	OUTPUT	FREE
0 INPUT		FREE			
EXTERNALS		TYPE	ARGS	SIN	REAL
CONNEX			1		1 LIBRARY
STATEMENT LABELS					
4144 10			4153	20	
STATISTICS					
PROGRAM LENGTH			253R	171	
BUFFER LENGTH			4006B	2054	
56000B CM USED					

```

1      PROGRAM WORTEL(INPUT,OUTPUT)
      INTEGER ANTW
5      CALL CONNEX(6+OUTPUT)
      CALL CONNEX(5+INPUT)
10     PRINT *, 'VOER EEN POSITIEF GETAL IN : '
      READ*,X
      IF(X.LT.0.) GOTO 10
      XWOR=SQRT(X)
15     PRINT*, 'DE WORTEL VAN ',X,' = ',XWOR
20     PRINT*, 'WILT U NOG EEN WORTEL UITGEREKEND HEBBEN? '
      PRINT*, '(1=JA,0=NEE)
      READ*,ANTW
      IF(ANTW.NE.1.AND.ANTW.NE.0) GOTO 20
      IF(ANTW.EQ.1) GOTO 10
      END

```

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
4137 WORTEL

VARIABLES	SN	TYPE	RELOCATION	4225 X	REAL
4224 ANTW		INTEGER			
4226 XWOR		REAL			

FILE NAMES	MODE	2054 OUTPUT	FREE
0 INPUT	FREE		

EXTERNALS	CONNEX	TYPE	ARGS	1	SOORT	REAL	1 LIBRARY

STATEMENT LABELS	4144 10	4154 20

STATISTICS	251B	169	4006B	2054
PROGRAM LENGTH				
BUFFER LENGTH				
56000B CM USED				

```

1  PROGRAM FACUL(INPUT,OUTPUT)
    DOUBLE FAC
    INTEGER ANTW
5  CALL CONNEX(5,INPUT)
    CALL CONNEX(6,OUTPUT)
10 PRINT*, ' GEEF EEN GEHEEL GETAL TUSSEN 0 EN 175 OP : '
    READ*,IX
    FAC=1
    IF(IX.EQ.0) GOTO 25
15 DO 20 I=1,IX
        FAC=I*FAC
20 CONTINUE
25 PRINT*, ' DE UITKOMST VAN ',IX,' * ',FAC
30 PRINT*, ' WILT U NOG EEN FACULTEIT UITGEREKEND HEBBEN? '
    PRINT*, ' (1=JA,0=NEE)
    READ*,ANTW
    IF(ANTW.NE.1.AND.ANTW.NE.0) GOTO 30
    IF(ANTW.EQ.1) GOTO 10
30 END

```

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
4137 FACUL

VARIABLES	SN	TYPE	RELOCATION	4236 FAC	DOUBLE
4240 ANTW		INTEGER		4241 IX	INTEGER
4242 I		INTEGER			

FILE NAMES	MODE	2054 OUTPUT	FREE
0 INPUT	FREE		

EXTERNALS	TYPE	ARGS
CONNEX		1

STATEMENT LABELS	0	20
4144 10		
4166 30		

4164 25

LOOPS LABEL	INDEX	FROM-TO	LENGTH	PROPERTIES
4156 20	I	16 19	58	OFT

```
***** LIBRARY(PROGLIB,NEW)
      1 LIBRARY(PROGLIB,NEW)
***** ADD(*,BISIN,AL=1)
      2 ADD(*,BISIN,AL=1)
***** ADD(*,BIWOR,AL=1)
      3 ADD(*,BIWOR,AL=1)
***** ADD(*,BIFAC,AL=1)
      4 ADD(*,BIFAC,AL=1)
***** FINISH.
      5 FINISH.
***** ENDRUN.
      6 ENDRUN.
```

PROCESSING DIRECTIVE NUMBER 1

PROCESSING DIRECTIVE NUMBER 2

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

SINUS

SYSTEM NAME NOS/BE 1.309/07/81 10.41.03 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 237 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 1

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

SINUS

EXTERNAL REFERENCES

END.

CONNEC

SIN.

INPFI.

OUTFI.

Q2NTRY.

PROCESSING DIRECTIVE NUMBER 3

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

WORTEL

SYSTEM NAME NOS/BE 1.309/07/81 10.41.04 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 235 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 1

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

WORTEL

EXTERNAL REFERENCES

END.

CONNEC

SORT.

INPFI.

OUTFI.

Q2NTRY.

PROCESSING DIRECTIVE NUMBER 4

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

FACUL

SYSTEM NAME NOS/BE 1.309/07/81 10.41.13 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 252 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 1

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

FACUL

EXTERNAL REFERENCES

END.

CONNEC

INPFI.

OUTFI.

Q2NTRY.

EDITLIS VERSION 2.4

PROCESSING DIRECTIVE NUMBER 5

PROCESSING DIRECTIVE NUMBER 6

MFB 09/07/81 IWS-TNO NOS/BE 1.4MFB - 027 LEVEL 518
 10.41.01.B1JRM15 FROM MFA/BI
 10.41.01.1P 00000192 WORDS FILE INPUT ,00BI 04
 10.41.01.B1JRM.
 10.41.01.ACCOUNT,43,77101058,*.
 10.41.02.REQUEST,PROGLIB,*.PF.
 10.41.02.ATTACH,SIN,SINUS,ID=JRM.
 10.41.02.AT CY= 001 SN=SYSTEM
 10.41.03.ATTACH,WOR,WORTEL,ID=JRM.
 10.41.03.AT CY= 001 SN=SYSTEM
 10.41.03.ATTACH,FAC,FACULTEIT,ID=JRM.
 10.41.03.AT CY= 001 SN=SYSTEM
 10.41.03.FTN,I=SIN,B=BISIN,SYSEDIT.
 10.41.06. .262 CP SECONDS COMPILE TIME
 10.41.06.FTN,I=WOR,B=BWOR,SYSEDIT.
 10.41.13. .251 CP SECONDS COMPILE TIME
 10.41.13.FTN,I=FAC,B=BIFAC,SYSEDIT.
 10.41.15. .319 CP SECONDS COMPILE TIME
 10.41.15.EDITLIB(USER).
 10.41.18.CATALOG,PROGLIB,PROGLIB,ID=JRM,RP=999.
 10.41.18.INITIAL CATALOG
 10.41.18.CT ID= JRM PFN=PROGLIB
 10.41.18.CT CY= 001 SN=SYSTEM 0000960 WORDS.
 10.41.19.OP 00001344 WORDS FILE OUTPUT ,00BI 40
 10.41.19.MS 7296 WORDS (31424 MAX USED)
 10.41.19.CPA 1.197 SEC. 1.197 ADJ.
 10.41.19.IO 2.870 SEC. 4.305 ADJ.
 10.41.19.CM 2.748 KWS. 2.748 ADJ.
 10.41.19.SS. 8.251
 10.41.19.FF. 29.625 SEC. DATE 09/07/81
 10.41.19.MAX FL USED 0560008
 10.41.19.NT .000 SEC. (IO)
 10.41.19.AVERAGE FL/ CP-SEC (OCTAL) 47336
 10.41.19.AVERAGE FL/ ID-SEC (OCTAL) 42103
 10.41.19.EJ END OF JOB, BI

Voorbeeld 3. Het aan een library toevoegen van programma's uit een
andere library

De afzonderlijke onderdelen zijn:

1. batchjob met input voor EDITLIB
2. output door EDITLIB

```

*****
*
*   BATCHJOB VOOR HET TOEVOEGEN VAN PROGRAMMA'S
*
*   UIT EEN ANDERE LIBRARY
*
*****

```

BIVIN.

ACCOUNT,43,77101128,2S.

ATTACH,LIB1,LIBICW ,ID=VIN.

ATTACH,LIB2,LIBICWNR2,ID=VIN.

EDITLIB(USER).

EXTEND,LIB1.

ZEOR

LIBRARY(LIB1,OLD)

ADD(SINUS,LIB2,LIB)

FINISH.

LISTLIB(*,LIB1)

ENDRUN.

ZEUF

```
***** LIBRARY(LIB1,OLD)
,      1 LIBRARY(LIB1,OLD)
***** ADD(SINUS,LIB2,LIB)
      2 ADD(SINUS,LIB2,LIB)
***** FINISH.
      3 FINISH.
***** LISTLIB(*,LIB1)
      4 LISTLIB(*,LIB1)
***** ENDRUN.
      5 ENDRUN.
```

PROCESSING DIRECTIVE NUMBER 1

PROCESSING DIRECTIVE NUMBER 2

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

SINUS

SYSTEM NAME NOS/BE 1.302/07/81 10.48.27 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

SINUS

EXTERNAL REFERENCES

INPFI.

OUTCI.

SIN.

PROCESSING DIRECTIVE NUMBER 3

PROCESSING DIRECTIVE NUMBER 4

LISTING OF LIBRARY LIB1 FOLLOWS

FOLLOWING PROGRAM LISTED

PREFIX TABLE INFORMATION

PROGRAM NAME

LOGAR

SYSTEM NAME NOS/BE 1.309/07/81 10.05.24 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 132 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

LOGAR

EXTERNAL REFERENCES

INPFI.

INPFI.

BUTCI.

FOLLOWING PROGRAM LISTED

PREFIX TABLE INFORMATION

PROGRAM NAME

WORTEL

SYSTEM NAME NOS/BE 1.309/07/81 10.05.26 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

WORTEL

EXTERNAL REFERENCES

INPFI.

OUTCI.

SGRT.

FOLLOWING PROGRAM LISTED

PREFIX TABLE INFORMATION

TIME 10.53.58.

DATE 09/07/81

VERSION 2.4

EDITLIB

PROGRAM NAME

FACUL

SYSTEM NAME NOS/BE 1.309/07/81 10.05.29 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES , HARDWARE I INSTRUCTION 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 143 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

FACUL

EXTERNAL REFERENCES

INPFI.

INPFI.

OUTCI.

FOLLOWING PROGRAM LISTED

PREFIX TABLE INFORMATION

PROGRAM NAME

TANGENS

SYSTEM NAME NOS/BE 1.309/07/81 10.05.31 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES , HARDWARE I INSTRUCTION 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 137 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

TANGENS

EXTERNAL REFERENCES

COS.

EOF

INPFI.

OUTCI.

SIN.

FOLLOWING PROGRAM LISTED

PREFIX TABLE INFORMATION

PROGRAM NAME

SINUS

SYSTEM NAME NOS/BE 1.302/07/81 10.48.27 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES , HARDWARE I INSTRUCTION 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

SINUS

EXTERNAL REFERENCES

INPFI.

INPFI.

OUTCI.

SIN.

PROCESSING DIRECTIVE NUMBER 5

MFB 09/07/81 IWS-TWO NOS/BE 1.4MFB - Q27 LEVEL 518
 10.53.55.BIVINJC FROM MFA/BI
 10.53.55.IF 0000192 WORDS FILE INPUT ,0081 04
 10.53.55.RVIN.
 10.53.56.ACCOUNT,43,77101128,*.
 10.53.57.ATTACH,LIB1,LIBICW,ID=VIN.
 10.53.57.AT CY= 001 SN=SYSTEM
 10.53.58.ATTACH,LIB2,LIBICWNR2,ID=VIN.
 10.53.58.AT CY= 001 SN=SYSTEM
 10.53.58.EDITLIB(USER).
 10.54.01.EXTEND,LIB1.
 10.54.01.EX ID= VIN PFN=LIBICW
 10.54.01.EX CY= 001 SN=SYSTEM 00001408 WORDS.
 10.54.01.OF 0000704 WORDS FILE OUTPUT ,0081 40
 10.54.01.MS 3648 WORDS (13888 MAX USED)
 10.54.01.CPA .369 SEC.
 10.54.01.ID .840 SEC. 1.261 ADJ.
 10.54.01.CH .387 KWS. .387 ADJ.
 10.54.01.SS 2.018
 10.54.01.PP 5.577 SEC. DATE 09/07/81
 10.54.01.MAX FL USED 045008
 10.54.01.NT .000 SEC. (IO)
 10.54.01.AVERAGE FL/ CP-SEC (OCTAL) 32575
 10.54.01.AVERAGE FL/ IO-SEC (OCTAL) 17546
 10.54.01.EJ END OF JOB, BI

Voorbeeld 4. Het vervangen van programma's in een library

De afzonderlijke onderdelen zijn:

1. batchjob met input voor EDITLIB
2. output door EDITLIB

```

*****
*
*   BATCHJOB VOOR HET VERVANGEN VAN PROGRAMMA'S
*
*   IN EEN BESTAANDE LIBRARY
*
*****

```

```

BIVIN.

```

```

ACCOUNT,43,77101128,25.

```

```

ATTACH,LIB,LIRICW,ID=VIN.

```

```

ATTACH,FAC,SUBFAC,ID=VIN.

```

```

FTN,I=FAC,L=0,B=BINFAC,SYSEDT.

```

```

EDITLIB(USER).

```

```

EXTEND,LIB.

```

```

%EOF

```

```

LIBRARY(LIB,OLD)

```

```

REPLACE(FACUL,BINFAC)

```

```

FINISH.

```

```

LISTLIB(*,LIB)

```

```

ENDRUN.

```

```

%EOF

```

```
***** LIBRARY(LIB,OLD)
,      1 LIBRARY(LIB,OLD)
***** REPLACE(FACUL,BINFAC)
      2 REPLACE(FACUL,BINFAC)
***** FINISH.
      3 FINISH.
***** LISTLIB(*,LIB)
      4 LISTLIB(*,LIB)
***** ENDRUN.
      5 ENDRUN.
```

PROCESSING DIRECTIVE NUMBER 1

PROCESSING DIRECTIVE NUMBER 2

FOLLOWING PROGRAM DELETED
FACUL

FOLLOWING PROGRAM ADDED
PREFIX TABLE INFORMATION
PROGRAM NAME

SYSTEM NAME NOS/BE 1.309/07/81 11.01.49 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 143 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS
FACUL

EXTERNAL REFERENCES
INPFI.
OUTCI.

PROCESSING DIRECTIVE NUMBER 3

PROCESSING DIRECTIVE NUMBER 4
LISTING OF LIBRARY LIB FOLLOWS

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION
PROGRAM NAME

SYSTEM NAME NOS/BE 1.309/07/81 10.05.24 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 132 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS
LOGAR

EXTERNAL REFERENCES
ALOG10.
INPFI.
OUTCI.

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION
PROGRAM NAME

SYSTEM NAME NOS/BE 1.309/07/81 10.05.26 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS
WORTEL

EXTERNAL REFERENCES
INPFI.
OUTCI.
SORT.

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION
PROGRAM NAME

TANGENS
SYSTEM NAME NOS/RE 1.309/07/81 10.05.31 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 137 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS
TANGENS
EXTERNAL REFERENCES EOF INPCI. OUTCI. SIN.
COS.

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION
PROGRAM NAME

SINUS
SYSTEM NAME NOS/RE 1.302/07/81 10.48.27 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS
SINUS
EXTERNAL REFERENCES INPFI. OUTCI. SIN.

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION
PROGRAM NAME

FACUL
SYSTEM NAME NOS/RE 1.309/07/81 11.01.49 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 143 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS
FACUL
EXTERNAL REFERENCES INPFI. OUTCI.

PROCESSING DIRECTIVE NUMBER 5

```

HEB 09/07/81 IWS-IND NOS/BE 1.4MEB - Q27 LEVEL 518
11.01.46.RIVINJF FROM MFA/BI
11.01.46.IF 0000192 WORDS FILE INPUT ,00BI 04
11.01.46.RIVIN.
11.01.47.ACCOUNT,43,77101128,*.
11.01.47.ATTACH,LIB,LIRICW,ID=VIN.
11.01.48.AT CY= 001 SN=SYSTEM
11.01.48.ATTACH,FAC,SUBFAC,ID=VIN.
11.01.49.AT CY= 001 SN=SYSTEM
11.01.49.FIN,I=FAC,L=0,B=BINFAC,SYSEDI.
11.01.51. .261 CP SECONDS COMPILATION TIME
11.01.51.EDITLIB(USER).
11.01.53.EXTEND,LIB.
11.01.54.EX ID= VIN PFN=LIBICW
11.01.54.EX CY= 001 SN=SYSTEM 00001920 WORDS.
11.01.54.OP 0000768 WORDS FILE OUTPUT ,00BI 40
11.01.54.MS 3648 WORDS ( 17536 MAX USED)
11.01.54.CPA .624 SEC.
11.01.54.ID 1.368 SEC.
11.01.54.ID 1.134 KWS.
11.01.54.CM 1.134 KWS.
11.01.54.SS 3.810
11.01.54.FP 7.398 SEC. DATE 09/07/81
11.01.54.MAX FL USED 056000R
11.01.54.NT .000 SEC. (IO)
11.01.54.AVERAGE FL/ CP-SEC (OCTAL) 42430
11.01.54.AVERAGE FL/ IO-SEC (OCTAL) 35174
11.01.54.EJ END OF JOB, BI

```

Voorbeeld 5. Het verwijderen van programma's uit een library

De afzonderlijke onderdelen zijn:

1. batchjob met input voor EDITLIB
2. output door EDITLIB

***** LIBRARY(LIB,OLD)

1 LIBRARY(LIB,OLD)

***** DELETE(SINUS)

2 DELETE(SINUS)

***** FINISH.

3 FINISH.

***** LISTLIB(*,LIB)

4 LISTLIB(*,LIB)

***** ENDRUN.

5 ENDRUN.

PROCESSING DIRECTIVE NUMBER 1

PROCESSING DIRECTIVE NUMBER 2

FOLLOWING PROGRAM DELETED
SINUS

PROCESSING DIRECTIVE NUMBER 3

PROCESSING DIRECTIVE NUMBER 4
LISTING OF LIBRARY LIB FOLLOWSFOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION
PROGRAM NAME

LOGAR
SYSTEM NAME NOS/BE 1.309/07/81 10.05.24 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 132 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS

LOGAR

EXTERNAL REFERENCES

ALOG10.

INPCI.

INPFI.

OUTCI.

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION

PROGRAM NAME

WORTEL

SYSTEM NAME NOS/BE 1.309/07/81 10.05.26 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS

WORTEL

EXTERNAL REFERENCES

INPCI.

OUTCI.

SGRT.

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION

PROGRAM NAME

TANGENS

SYSTEM NAME NOS/BE 1.309/07/81 10.05.31 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 137 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS

TANGENS

EXTERNAL REFERENCES

COS.

EOF

INPCI.

OUTCI.

SIN.

MFB 09/07/81 IMIS-TNO NOS/RE 1.4NFB - 027 LEVEL 518
 11.11.23.RIVINUM FROM MFA/BI
 11.11.24.IP 00000192 WORDS FILE INPUT '00BI 04
 11.11.24.BIVIN.
 11.11.24.ACCOUNT,43,77101128,*.
 11.11.25.ATTACH,LIB,LIBICW,IO=VIN.
 11.11.25.AT CY= 001 SN=SYSTEM
 11.11.26.EDITLIB(USER).
 11.11.27.EXTEND,LIB. VIN PFN=LIBICW
 11.11.31.EX ID= VIN PFN=LIBICW
 11.11.31.EX CY= 001 SN=SYSTEM 00002304 WORDS. 40
 11.11.32.OP 00000512 WORDS FILE OUTPUT '00BI 04
 11.11.32.NS 3648 WORDS (13888 MAX USED)
 11.11.32.CFA ,314 SEC. ,312 ADJ.
 11.11.32.IO ,633 SEC. ,950 ADJ.
 11.11.32.CH ,324 KMS. ,324 ADJ.
 11.11.32.SS ,1.587
 11.11.32.PP 8.061 SEC. DATE 09/07/81
 11.11.32.MAX FL USED 045000B
 11.11.32.NT ,000 SEC. (IO)
 11.11.32.AVERAGE FL/ CP-SEC (OCTAL) 32535
 11.11.32.AVERAGE FL/ IO-SEC (OCTAL) 21306
 11.11.32.EJ END OF JOB, BI

EDITLIB VERSION 2.4

DATE 09/07/81 TIME 11.11.26.

PAGE 3

FOLLOWING PROGRAM LISTED
PREFIX TABLE INFORMATION
PROGRAM NAME

FACUL
SYSTEM NAME NOS/BE 1.309/07/81 11.01.49 PROCESSOR NAME FTN 4.8 518
DEPENDENCIES HARDWARE I INSTRUCTION 64 64 4
THIS IS A RELOCATABLE PROGRAM.
BINARY LENGTH = 143 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
ACCESS LEVEL IS 0
FIELD LENGTH MAY NOT BE INCREASED.
ENTRY POINTS

EXTERNAL REFERENCES
INPCT.

INPFI.

OUTCI.

PROCESSING DIRECTIVE NUMBER 5

Voorbeeld 6. Het kopiëren van libraries in random format

De afzonderlijke onderdelen zijn:

1. batchjob met input voor EDITLIB
2. output door EDITLIB

```
*****
*
*   BATCHJOB VOOR HET COPIEREN VAN LIBRARIES
*
*   IN RANDOM FORMAT
*
*****
```

BIVIN.

ACCOUNT,43,77101128,2S.

REQUEST,LIB2,*PF.

ATTACH,LIB1,LIBICW,ID=VIN.

EDITLIB(USER).

CATALOG,LIB2,LIBICWNR2,ID=VIN,RP=999.

ZEOR

LIBRARY(LIB2,NEW)

ADD(*,LIB1,LIB)

FINISH.

LISTLIB(*,LIB2)

ENDRUN.

ZEOP

***** LIBRARY(LIB2,NEW)

1 LIBRARY(LIB2,NEW)

***** ADD(*,LIB1,LIB)

2 ADD(*,LIB1,LIB)

***** FINISH.

3 FINISH.

***** LISTLIB(*,LIB2)

4 LISTLIB(*,LIB2)

***** ENDRUN.

5 ENDRUN.

TIME 11.20.10.

DATE 09/07/81

VERSION 2.4

EDITLIB

PROCESSING DIRECTIVE NUMBER 1

PROCESSING DIRECTIVE NUMBER 2

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

LOGAR

SYSTEM NAME NOS/RE 1.309/07/81 10.05.24 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 132 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

LOGAR

EXTERNAL REFERENCES

ALOG10.

INPCI.

OUTCI.

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

WORTEL

SYSTEM NAME NOS/RE 1.309/07/81 10.05.26 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

WORTEL

EXTERNAL REFERENCES

INPCI.

INPFI.

SORT.

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

TANGENS

SYSTEM NAME NOS/RE 1.309/07/81 10.05.31 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 137 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

TANGENS

EXTERNAL REFERENCES

COS.

EOF

INPCI.

OUTCI.

SIN.

FOLLOWING PROGRAM ADDED

PREFIX TABLE INFORMATION

PROGRAM NAME

FACUL

SYSTEM NAME NOS/RE 1.309/07/81 11.01.49 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

INSTRUCTION 64 64 4

THIS IS A RELOCATABLE PROGRAM.
 BINARY LENGTH = 143 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
 ACCESS LEVEL IS 0
 FIELD LENGTH MAY NOT BE INCREASED.
 ENTRY POINTS

EXTERNAL REFERENCES
 INPCI.
 OUTCI.

PROCESSING DIRECTIVE NUMBER 3

PROCESSING DIRECTIVE NUMBER 4
 LISTING OF LIBRARY LIB2 FOLLOWS

FOLLOWING PROGRAM LISTED
 PREFIX TABLE INFORMATION
 PROGRAM NAME
 LOGAR

SYSTEM NAME NOS/BE 1.309/07/81 10.05.24 PROCESSOR NAME FTN 4.8 518
 DEPENDENCIES HARDWARE 1 INSTRUCTION 64 64 4
 THIS IS A RELOCATABLE PROGRAM.
 BINARY LENGTH = 132 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
 ACCESS LEVEL IS 0
 FIELD LENGTH MAY NOT BE INCREASED.
 ENTRY POINTS

EXTERNAL REFERENCES
 ALGOL10.
 INPCI.
 OUTCI.

FOLLOWING PROGRAM LISTED
 PREFIX TABLE INFORMATION
 PROGRAM NAME
 WORTEL

SYSTEM NAME NOS/BE 1.309/07/81 10.05.26 PROCESSOR NAME FTN 4.8 518
 DEPENDENCIES HARDWARE 1 INSTRUCTION 64 64 4
 THIS IS A RELOCATABLE PROGRAM.
 BINARY LENGTH = 126 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
 ACCESS LEVEL IS 0
 FIELD LENGTH MAY NOT BE INCREASED.
 ENTRY POINTS

EXTERNAL REFERENCES
 INPCI.
 OUTCI.
 SQRT.

FOLLOWING PROGRAM LISTED
 PREFIX TABLE INFORMATION
 PROGRAM NAME
 TANGENS

SYSTEM NAME NOS/BE 1.309/07/81 10.05.31 PROCESSOR NAME FTN 4.8 518
 DEPENDENCIES HARDWARE 1 INSTRUCTION 64 64 4
 THIS IS A RELOCATABLE PROGRAM.
 BINARY LENGTH = 137 AND EXECUTION FIELD LENGTH = 0 (OCTAL).
 ACCESS LEVEL IS 0
 FIELD LENGTH MAY NOT BE INCREASED.
 ENTRY POINTS

SIN.

OUTCI.

INPCI.

EOF

EXTERNAL REFERENCES

FOLLOWING PROGRAM LISTED

PREFIX TABLE INFORMATION

PROGRAM NAME

FACUL

SYSTEM NAME NOS/BE 1.309/07/81 11.01.49 PROCESSOR NAME FTN 4.8 518

DEPENDENCIES HARDWARE I

INSTRUCTION 64 64 4

THIS IS A RELOCATABLE PROGRAM.

BINARY LENGTH = 143 AND EXECUTION FIELD LENGTH = 0 (OCTAL).

ACCESS LEVEL IS 0

FIELD LENGTH MAY NOT BE INCREASED.

ENTRY POINTS

FACUL

EXTERNAL REFERENCES

INPCI.

OUTCI.

INPFI.

PROCESSING DIRECTIVE NUMBER 5

```

RFB 09/07/81 IWS-TND NOS/BE 1.4MFB - Q27 LEVEL 518
11.20.06.RIVINJT FROM MFA/BI
11.20.06.IF 00000192 WORDS FILE INPUT ,00B1 04
11.20.07.RIVIN.
11.20.07.ACCOUNT,43,77101128,**.
11.20.08.REQUEST,LIE2,*PF.
11.20.09.ATTACH,LIE1,LIBICW,ID=VIN.
11.20.10.AT CY= 001 SN=SYSTEM
11.20.12.EDITLIB(USER).
11.20.15.CATALOG,LIE2,LIBICWNR2,ID=VIN,RP=999.
11.20.13.INITIAL CATALOG
11.20.19.CT ID= VIN PFN=LIBICWNR2
11.20.19.CI CY= 001 SN=SYSTEM 00000896 WORDS.
11.20.19.OF 00000832 WORDS FILE OUTPUT ,00B1 40
11.20.19.MS 8768 WORDS ( 17536 MAX USED)
11.20.19.CPA .448 SEC.
11.20.19.IO 2.814 SEC.
11.20.19.CM .779 KWS.
11.20.19.SS 5.450
11.20.19.FP 13.838 SEC. DATE 09/07/81
11.20.19.MAX FL USED 045000B
11.20.19.NT .000 SEC. (IO)
11.20.19.AVERAGE FL/ CP-SEC (OCTAL) 31753
11.20.19.AVERAGE FL/ IO-SEC (OCTAL) 12002
11.20.19.EJ END OF JOB, BI

```