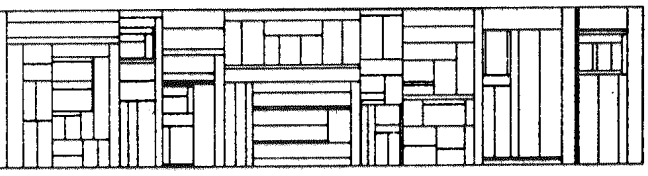
BRUGERVEJLEDNING
DAIMI / GIER

af
Fred Mosekjær Madsen

DAIMI MD-1
August 1973

Matematisk Institut Aarhus Universitet
DATALOGISK AFDELING
Ny Munkegade - 8000 Aarhus C - Danmark
Tlf. 06-12 83 55



Denne vejledning beskriver de udvidelser af det system, som findes ved DAIMI/GIER. Desuden er visse dele af help3 forklaret mere udførligt og eksemplificeret. Vejledningen kan ikke stå alene, men er kun et supplement til eksisterende manualer og rapporter.

INDHOLDSFORTEGNELSE

Hjælp 3 systemet ved GIER

For at kunne kommunikere med datamaten GIER må man kende lidt til help3. En beskrivelse findes i A Manual of Help3 ed. Søren Lauesen, udgivet af Regnecentralen, København 1967. Det anbefales at benytte manualen som reference.

Help3 er et system af hjælpeprogrammer, som hjælper brugeren af datamaten med at rette, oversætte og køre programmer.

Eksempel 1:

Man vil have indlæst og oversat et algolprogram, som er hullet på papirstrimmel. På strimlen står:

```
algol<
begin
   .
   .
   .
program
   .
   .
   .
end
```

Hulstrimlen lægges i læseren, og man skriver følgende hjælp3 information på skrivemaskinen:

```
r<
```

Virkningen af dette vil være, at strimmellæseren aktiveres (r=reader) og indlæsningen startes. Det første der indlæses er algol< . Indlæsningen standses, og GIER algol 4 compileren hentes frem fra baggrundslageret. Derpå fortsættes indlæsningen af det egentlige algolprogram som oversættes til maskinkode.

□

Hjælp3 bestyrer, hvorfra der skal indlæses, og hvortil der skal udskrives. I ovenstående eksempel, kunne man også have tænkt sig at programmet var lagret på GIER's baggrundslager, under navnet abcde, og vi ønsker derfor at algol oversætteren skal tage input derfra. Hjælp3 informationen til skrivemaskinen bliver da:

abcde<

□

Hjælp3 kataloget

For at kunne administrere input og outputmedierne, som altså omfatter både ydre medier (strimmellæser, skrivemaskine, perforator, linieskriver m. m.) og indre medier (navngivne, reserverede områder på baggrundslagrene), samt hjælpeprogrammerne, har hjælp3 et katalog. Dette katalog indeholder en række navne, nemlig:

1) navne på ydre input-output enheder
2) navne på hjælpeprogrammer
3) navne på reserverede områder på baggrundslageret.
4) navne på konstanter

Foruden navne indeholder kataloget også oplysninger om, hvor man kan finde et navngivet område etc.

Eksempel på katalog

Navn:	Kommentar:
free	det frie areal i baggrundslageret. Hertil indlæses der når man vil have reserveret og navngivet et areal.
work	et arbejdsareal. Det er her det oversatte algolprogram bliver lagret.
date	en konstant, som indeholder oplysning om dato og nr. hjælp-kald den dag.
image	et areal, hvor indholdet af ferritlageret gemmes af vejen ved et kald af hjælp3.
r	strimmellæser (r=reader)
t	skrivemaskine som input (t=typer)
p	perforator
l	linieskriver
w	skrivemaskine som output (w=writer)
ga4	gier algol 4-oversætteren
algol	et hjælpeprogram til at starte processen, at oversætte et algolprogram.
run	et hjælpeprogram til kørsel af allerede oversatte programmer.
edit	et hjælpeprogram til rettelse af tekster (herunder programmer)
res	et hjælpeprogram som reserverer den del af free der er booked, under det navn, der er givet som parameter

Input - Output medier

t beskriver input fra skrivemaskinen ved GIER.

w beskriver output på skrivemaskinen ved GIER

p beskriver output på perforator ved gier.

l beskriver output på printer (linieskriver).

lp beskriver output på både printer og perforator.

r1 beskriver input fra læseren ved GIER. stopper ikke på paritetsfejl.

r2 beskriver input fra læseren ved converteren, under forudsætning af at den er koblet til GIER på omkoblingskassen. Stopper ikke på paritetsfejl.

□

Karaktersæt på printer

Sammenhængen mellem talværdier og typografiske symboler er den samme som for flexowriter dog, at der kun trykkes store bogstaver og iøvrigt med følgende undtagelser og tilføjelser:

Talværdi:	Funktioner:
10	mellemrum
11	mellemrum
12	Ü
15	% &
26	mellemrum
28	mellemrum
29	mellemrum
30	tabulering forberedes. Er næste tegn talværdi n vil skrivepositionen flyttes til kolonne 2n.

(fortsættes)

Talværdi:	Funktioner:
31	mellemrum
42	mellemrum
44	* ' (apostrof)
45	mellemrum
46	mellemrum
47	mellemrum
61	mellemrum
62	mellemrum
63	mellemrum

Talværdier større end 64 initierer trykning og papirfremføring. Der er fire formularstyringskanaler som vælges af hver sin af de fire mindst betydende bits af talværdien medens den femte blokerer papirfremføringen.

sy 64: Før papiret een linie frem

sy 65: Før papiret frem styret af kanal 2
(linieskift med overspringelse af tværperforeringen)

sy 66: Før papiret frem styret af kanal 4
(formularskift)

sy 68: Før papiret frem styret af kanal 6
(ledig)

sy 72: Før papiret frem styret af kanal 8
(ledig)

sy 80: Trykning uden papirfremføring

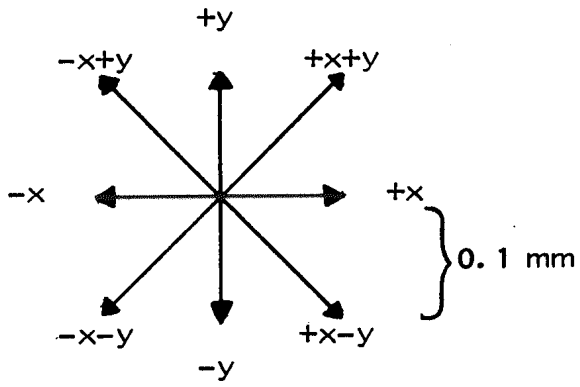
Kanalvalget kan ske logisk additivt, således at f. eks. sy 67 vil føre papiret frem til der mødes et hul i styrestrimlen i kanal 2 eller 4.

□

Plotteren

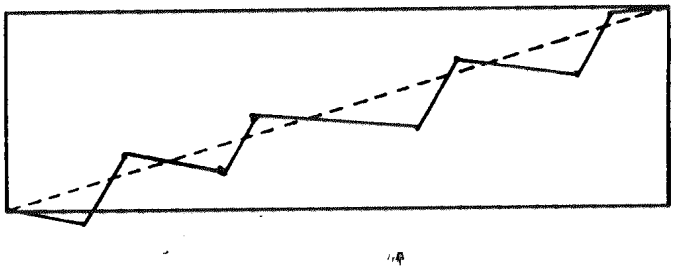
Datalogisk Afdeling råder over en Calcomp kurvetegner, model 563. Denne er tilsluttet, så den kan køres direkte af GIER eller styres off-line af et magnetbånd via vor converter.

Enhver tegning på plotteren består af små rette liniestykker af størrelsesorden 0,1 mm. Plotteren kan i en enkelt operation bevæge pennen i forhold til papiret i enten x-aksens eller y-aksens retning eller i begge retninger således:



x-retningen kontrolleres af en stepmotor, der bevæger pennen vandret hen over papiret; y-retningen kontrolleres af en anden stepmotor, der fører papiret frem under pennen.

Da plotteren foruden disse retlinede step kun kan udføre de to operationer PEN OP og PEN NED, må GIER i almindelighed i stedet for at tegne rette linier lave en zigzag tilnærmelse som f. eks.



Normalt vil man dog ikke kunne se forskellen - dels på grund af de små step og dels på grund af pennens tykkelse.

Plotterens maksimale hastighed er 300 step pr. sekund, papirets bredde ca. 75 cm. Normalt køres med kuglepenne, men vore operatører kan montere tuschpenne til skønskrift.

□

Hjælp 3 information

Hjælp 3 information består i det væsentlige af en liste af navne. Desuden kan forekomme understregede bogstaver, samt heltal. Listens elementer adskilles af komma, og listen afsluttes med < (udtales hak).

Eksempel 2:

r, edit, o free<

Hjælp 3 informationen indlæses af hjælp's løbende inputmedium, som i almindelighed vil være skrivemaskinen. Hvis dette ikke er tilfældet (den røde lampe lyser ikke) kan man trykke på en knap ved GIER's kontrolbord, HP-knappen (=hjælpknappen). Herved hentes hjælp 3 maskinkodeprogrammet ind i ferritlageret og startes, noget af det første der sker er, at skrivemaskinen vælges som løbende input, og hjælp 3 information kan nu indlæses.

Efter indlæsningen, som slutter når der mødes et < , vil hjælp 3 udfra kataloget identificere listens navne et for et. Hvis et navn ikke findes i kataloget, vil der komme en udskrift på skrivemaskinen (med rødt): undef. Ellers sker der følgende:

Hvis navnet er:

1) en outputenhed, vælges denne som hjælp's outputmedium, og hjælp 3 fortsætter i listen.

2) en inputenhed eller et reserveret område i baggrundslageret, vælges dette som hjælp's løbende inputmedium, og hjælp fortsætter i listen.

3) et hjælpeprogram, vil resten af listen blive flyttet til en parameterkanal, og hjælp 3 vil kalde det pågældende hjælpeprogram ind

i ferritlageret, og overgive kontrollen til dette. Hvad der vil ske med resten af listen, afhænger af det pågældende hjælpeprogram, idet resten af listen opfattes som parametre til hjælpeprogrammet. Når hjælpeprogrammet er færdigt, venter GIER på hjælp 3 information fra hjælp's løbende inputmedium.

Møder hjælp 3 et < i listen uden at have overgivet kontrollen til et hjælpeprogram, vil hjælp's løbende inputmedium blive valgt som inputmedium, og ny hjælp 3 information vil blive læst herfra.

□

Eksempel 3:

samme som eksempel 1 (side 1)

Der skrives r< på skrivemaskinen. navnet r tydes af hjælp 3, som et legitimt navn på et inputmedium, nemlig strimmellæseren. Denne vælges som hjælp's løbende inputmedium, og da der står et hak efter r'et, starter hjælp 3 nu indlæsning fra strimmellæseren. I dette tilfælde må papirstrimlen indledes med hjælp 3 information. Forrest på strimlen står algol<, som medfører et kald af hjælpeprogrammet algol, der igen kalder hjælpeprogrammet ga4, der indlæser og oversætter programmet. Hvis algolprogrammet blev oversat, ville der på skrivemaskinen komme udskriften "ok", og hjælpeprogrammet er færdigt. Hjælp venter nu på mere information fra løbende input (i dette tilfælde strimmellæseren) og det havde derfor været hensigtsmæssigt, hvis papirstrimlen havde haft følgende udseende:

algol< begin ..... end run<

thi da ville programmet straks efter oversættelsen blive kørt. Kræver programmet data fra strimmel, og er der flere datasæt, kan det være hensigtsmæssigt at lade programstrimlen se således ud:

algol< begin ..... end t<

efter endt oversættelse og ok vil GIER læse t< og derfor vælge skrivemaskinen (typewriter) som løbende inputmedium. Man lægger nu sin datastrimmel i læseren, hvor datastrimlen har følgende udseende:

run< data1, data2 .....

man taster nu r< på skrivemaskinen, hvorefter programmet bliver kørt. □

Vigtigt!

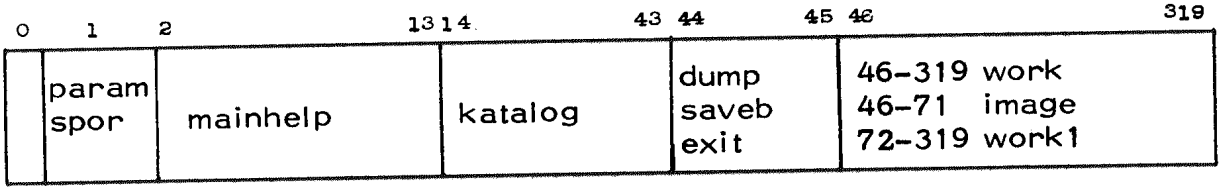
Man bør altid tilrettelægges sin kørsel på GIER således at man kun skal taste r< på skrivemaskinen, derved sparer man megen tid for sig selv, og mange irritationsmomenter for de øvrige brugere, der venter på at komme til.

Baggrundslagre i GIER

Foruden ferritlageret, som er på 1024 ord á 40 bit, og bufferlageret, som er på 4096 ord, har GIER tilknyttet følgende baggrundslagre:

Tromlen

består af 320 blokke af 40 ord, tromlen er organiseret på følgende måde:



Der er direkte forbindelse mellem GIER's ferritlager og tromlen.

Disk

Der er til GIER tilknyttet to diskunits, hvoraf den ene unit (unit 9) bærer systemkittet og den anden unit bærer private kits. Et kit består af 10 plader á 203 spor (blokke) á 600 ord.

Båndstationerne

anvendes ikke fast af hjælp 3, og er altså frit tilgængelige for brugerne. Man kan skrive og læse på båndene med to forskellige tætheder: 556 bpi (bit pr. inch), som er det normale, og 800 bpi. Endvidere kan båndstationerne switches til konverteren.

Diskunits og båndstationer kaldes buffermedier, fordi læsning og skrivning foregår via bufferen.

□

Diskadministration

Et diskkit indeholder to kataloger:

et brugerkatalog (spor 0-2) over de brugere der må benytte det pågældende kit, og

et filekatalog (spor 3-6) over de områder af kittet der er benyttet.

Beskrivelserne i filekataloget af de områder en bruger har er linket sammen, og knyttet til brugerens navn i brugerkataloget. For at få adgang til sine områder må man benytte hjælpeprogrammet login.

login

kald:
login, <ident><
<ident>::=<empty>|<ident><ident>

hvis parameterlisten er tom, får brugeren adgang til en række "ufarlige" hjælpeprogrammer, såsom algoloversættere, run, edit og forskellige udskrifts og redigeringsprogrammer, men ingen programmer der kan oprette eller slette beskrivelser i kataloget.

Hvis der er parametre, får brugeren også adgang til programmer, der kan oprette og slette beskrivelser i kataloget, samt til de områder der knyttet til brugernavne i parameterlisten. Første navn i parameterlisten huskes, og kaldes i det følgende logged-in-user.

□

res

kald:

res, <length>, <ident>, <bits>, <entries><

<length>::=<number>| <empty>

<bits>::=o|c|p|i|s| <helpnumber>|d<helpnumber>| <empty>| <bits><bits>

<entries>::=<name>| <name><helpnumber>| <entries><entries>

<number>: se hjælp 3 manual p. 40.

<bits>:

o	reservation dirigeres til private (own) kit.
c	reservation dirigeres til system (common) kit.

hvis o og c mangler dirigeres datoreservationer (d) til privat kit, og alle andre reservationer til systemkit.

<ident>:

b<user>	reservationen knyttes til pågældende user
<empty>	reservationen knyttes til logged-in-user.

Se iøvrigt hjælp 3 manual p. 40.

res benyttes til at reservere områder af free. Det navn (eller de navne), man vil tildele sit område, skal gives som parametre til res, og res vil begynde med at sikre sig at brugeren ikke allerede har et område med dette navn. Det kan anbefales at man lader de første bogstaver være ens - f. eks. refererende til den pågældende opgave, hvis man følger denne regel kan man få megen glæde af et andet hjælpeprogram kaldet lis.

Antag nu at man har et program som løser kannibalproblemet, og at dette program er indlæst til free, så kan man få det reserveret til senere brug under navnet kan1 ved kaldet:

res,kan1<

Man kan nu regne med at programmet findes til næste morgen. Der findes dog en mulighed til at angive at man ønsker at bevare sit område til en bestemt dato. Hvis man f. eks. mener at skulle bruge sit område indtil juleaften 1973 ser kaldet således ud:

res, d24. 12. 73, kan1 <

i forbindelse med reservation må man være opmærksom på 2 ting.

For det første vil der altid være risiko for, at området alligevel bliver ødelagt, enten ved operatørfejl eller - mere sandsynligt - af rent tekniske årsager, og man må derfor altid sørge for, at man er i stand til at reetablere hvad man har stående på disken.

For det andet er pladsen ikke ubegrænset, så det vil ikke være god tone, hvis man blot ukritisk reserverer det ene område efter det andet i adskillige måneder.

Ovenstående eksempler på reservationer knyttes til logged-in-user, hvis man ønsker området knyttet til en anden bruger angives dette ved b<user> i parameterlisten til res.

□

redate

kald:
redate, <ident>,
<vilkårlig premutation af elementerne <form><værdi>
og <navneliste>><
<ident>: se res ovenfor
<form> se under list i hjælp 3 manualen
<værdi>::=<empty>|d<helpnumber>
<navneliste> må ikke være tom.

For hvert navn i parameter listen sker følgende:

1) Hvis navnet refererer til et reserveret område, fjernes eventuelle forekomster af p, i og s bit i arealordet. Hvis der findes et sekundært ord indsættes seneste forekommende <værdi> heri, eller 0, hvis der ingen har været. Katalogindgangen udskrives i overensstemmelse med seneste form, hvis der har været nogen

2) Hvis navnet er work, ændres <work-as-output> så det begynder med den kanal, hvis nr. står i de sidste 9 bit af <helpnumber> i den seneste <værdi>. Hvis kanalnr ligger udenfor området 46-319 indsættes dog 1. kanal for image. Udskrift som ovenfor.

3) I andre tilfælde kommer udskrift som ovenfor efterfulgt af udskriften no change.

Eksempler:

Ønsker en bruger der kører under brugernavnet dat, at ændre datoreservationen af sit område med navn kan1, sker dette på følgende måde:

redate,b,dat,d 1. 12. 73,kan1<

området bliver da fjernet den 2/12-73 om morgenen.

kaldet:

redate,b,dat,d5. 10. 73,kan1,kan2,kan3,d23. 12. 73,kan4<

vil bevirke at områderne kan1, kan2, kan3 vil blive stående til 5/10-73, og at området kan4 vil blive stående til 23/12-73.

Advarsel!

Man kan kun bruge redate på områder der oprindeligt er reserveret med dato.

□

clear

kald:
clear,b<ident>,<name><
<ident>: brugernavn
<name>: hjælp 3 navn

clear fjerner beskrivelsen af det reserverede område knyttet til brugernavnet

□

Båndadministration

I visse tilfælde ønsker en bruger at gemme meget store mængder af information eller at gemme information over et par måneder. Man kunne f. eks. tænke sig, at en bruger har et sæt standardprogrammer, der skal køres hyppigt i uændret form, eller at han ved hver kørsel skal have adgang til nogle meget store tabeller. I sådanne tilfælde vil et magnetbånd være meget mere pålideligt end disken.

Udlevering af magnetbånd

Ved henvendelse til en af instituttets operatører kan en bruger få udleveret et bånd til eget brug. Ved samme lejlighed kan han få en instruktion i den manuelle betjening af båndstationerne.

Beksrivelse af båndområder

Ligesom de reserverede områder skal båndområderne være beskrevet i kataloget. Dette sker ved hjælp af hjælpeprogrammet set, det kaldes således:

set, 3, 1, st. nr. , file nr. , 0, navn<

st.nr.' er nummeret på den båndstation man ønsker at benytte - normalt vil 2 være at foretrække. file nr. angiver, hvor mange områder der ligger før det nyoprettede område, file nr. skal altså gennemløbe værdierne 0, 1, 2, 3 ..... .

Som man vil se er det en ret kompliceret sag at beskrive et båndområde, imidlertid er flere af parametrene faste, og da move og edit altid ajourfører længden af et båndområde, hvortil de har leveret output, og da femte parameter, der angiver hvor mange blokke der skal overspringes, normalt skal være 0, og da file nr. normalt skal være 1 større end højeste tidligere benyttede filenr., findes det er hjælpeprogram med en

betydelig lettere administration, der kan erstatte set. Dette program hedder stp.

stp

benyttes i stedet for set (se manualen) hvis ovenstående betingelser er opfyldt. Hvis der i kataloget er beskrevet områder til og med file nr. 4 på båndstation nr. 3, vil

stp, 3, peter<

være ækvivalent med

set, 3, 1, 3, 5, 0, peter<

stp kan indsætte alle former for beskrivelser. Således kan en ga4-oversætter på magnetbånd laves ved:

stp, pis 0, tapega4, 11. 199. 199. 0<

idet stp benytter station 2, når andet ikke er angivet.

For at undgå at andres beskrivelser af båndområder får stp til at indsætte for store værdier af filenr. bør man indlede sin kørsel med et kald af outtp (se dette), og så siden benytte excl og incl.

□

cattap

Før man kan benytte områder på et bånd, skal dette være forsynet med en cattap-label. Dette gøres med hjælpeprogrammet cattap.

cattap, 3< skriver en label på station nr. 3
cattap< skriver en label på station nr. 2

NB! Der skal ikke trykkes på hp, som ved cattap fra strimmel.

□

outtp

outtp fjerner fra kataloget alle beskrivelser vedrørende den ønskede båndstation. Man bør altid kalde outtp før en kørsel der involverer stp eller excl. De andre brugeres båndområder kan ellers gribe forstyrrende ind.

outtp, 1<	fjerner beskrivelser vedrørende station nr. 1
outtp, <	fjerner beskrivelser vedrørende station nr. 2

□

excl

opsamler alle beskrivelser vedrørende den ønskede station, og gemmer dem bagest (efter sidste beskrevne file) på båndet. Samtidig fjernes beskrivelserne fra kataloget,

excl, 4<	arbejder med båndstation nr. 4
excl<	arbejder med båndstation nr. 2

□

incl

søger på den ønskede båndstation, til den møder en stump hjælp 3-katalog, der er lavet af excl. Derpå gåes alle navne efter, og findes et navn ikke allerede i det egentlige hjælp 3-katalog, bliver det indsat med sin beskrivelse, eller bliver det skrevet på skrivemaskinen med rødt. Er KA tændt, vil alle navnene blive skrevet ud med sort.

incl, 3<	henter katalog på båndstation nr. 3
incl<	henter katalog på båndstation nr. 2

□

rewind

o

spoler båndet på den ønskede station tilbage til loadpoint.

o, 1<	spoler station nr. 1 tilbage
o<	spoler station nr. 2 tilbage

□

eof

skriver et filemark på den ønskede båndstation.

eof, 3<	skriver et filemark på station nr. 3
eof<	skriver et filemark på station nr. 2

Som det ses arbejder alle båndadministrationsprogrammerne på station nr. 2, når intet andet er angivet. For yderligere oplysninger om båndadministration se bind 3 af GIER manualen, samt manualen om sambasystemet.

□

copymt

kald:

copymt, <helpnumber>, <wordlength>, <parity><

<helpnumber>::=<inputst>. <inputfile>. <outputst>. <outputfile>| <empty>

Hvis helpnumber er udeladt (<empty>) kopieres fra øjeblikkelig stilling på station til øjeblikkelig stilling på station nr. 2.

Hvis file = 0 kopieres til eller fra øjeblikkelig stilling på input eller outputstation.

<wordlength>::=<empty>| s

s der kopieres med afkortet ordlængde. (default: normal ordlængde.)

<parity>::=<empty>|e

e der kopieres med lige paritet. (default: ulige paritet.)

Formål:

At kopiere magnetbånd. Programmet kan endvidere benyttes når man ønsker at dumpe et program der benytter tapeinput.

Udhop:

Programmet hopper ud når en af følgende situationer forekommer:

1) filemark på inputbånd.
2) ændret bloklængde på inputbånd.
3) end-of-tape på input eller outputbånd.

Eksempler:

copymt<

kopiering fra station 1 til station 2 uden tilbagespoling af båndene. Normal ordlængde. Ulige paritet.

copymt, 2. 0. 3. 0, e<

kopiering fra station 2 til station 3, ingen tilbage spoling af båndene. Normal ordlængde. Lige paritet.

copymt, 5. 2. 3. 1, s, e<

kopiering fra station 5 file 2 til station 3 file 1. Afkortet ordlængde, Lige paritet.

NB! Hvis man ønsker at starte ved loadpoint må man benytte hjælpeprogrammet o.

□

Eksempler på anvendelse af magnetbånd

Antag at man har 3 Algol 4-programmer, der skal køres hyppigt uden ændringer. De oversatte programmer gemmes på bånd på station nr. 2 således:

```
outtp<
cattap<
stp, tp2prg1<
stp, tp2prg2<
stp, tp2prg3<
a1, prg1, n<
move, work, tp2prg1<
a1, prg2, n<
move, work, tp2prg2<
a1, prg3, n<
move, work, tp2prg3<
excl<
t<
```

Ovenstående er naturligvis skrevet på strimmel, så at man kun behøver at taste r< på skrivemaskinen.

De oversatte programmer ligger nu på båndet, og på den bageste file findes en stump hjælp 3-katalog, som beskriver områderne.

Når man senere vil køre f. eks. prg1 gør man følgende:

1. monterer båndet på station nr. 2
2. taster incl<
3. taster y, tp2prg1<

Som tidligere nævnt, sker der det når man taster incl< at der søges på båndet på station nr. 2 efter en stump hjælp 3-katalog, og når den er

fundet kopieres den ind i det rigtige hjælp 3-katalog. Da beskrivelsen imidlertid altid står bagest på båndet (efter den sidste file) kan det tage nogen tid (2-10 min. afhængig af hvor mange og hvor lange områderne er), at finde beskrivelsen af båndområderne, dette kan imidlertid undgås hvis følgende er opfyldt:

1) alle programnavne starter med de samme bogstaver.

2) ingen andre programmer i kataloget starter med disse bogstaver.

Man kan da benytte hjælpeprogrammet lis:

```
lis rtp2<
outparam, t<
```

For at undgå at GIER hænger på læseren, genereres da en strimmel, der indeholder de til områderne hørende beskrivelser, i form af kald af hjælpeprogrammet set. Denne strimmel gemmer man sammen med båndet, og når man så skal bruge sit program, gøres følgende:

1. monterer båndet på station nr. 2
2. indsætter strimlen i læseren og taster r<.
3. taster y, tp2prg1<

Man sparer derved 2-10 minutter.

Advarsel!

Hver gang man flytter information til et båndområde, skal man huske at det område der står lige efter også vil blive ødelagt. Dette hænger sammen med at f. eks. en bloklængde på 200 ord ikke behøver at fylde lige mange centimeter bånd, hvis den lægges op to gange, det afhænger ganske af den hastighed hvormed båndet passerer skrivehovedet. Hvis man således også gemmer uoversatte programmer på bånd, må man være

klar over at det eneste område man kan rette i uden risiko er det sidst beskrevne område, idet man så kun får ødelagt kataloget (på båndet), men det reetableres jo igen når man taster excl <.

Programafvikling

algol

kalder en algol 4 oversætter med hjælp's løbende inputmedium, som programkilde. Man kan kalde Algol med forskellige parametre, f. eks.

algol, n<
algol, 10<

n betyder at algolprogrammet skal køres uden indexcheck på de indicerede variable (benyttes kun ved indkørte programmer, man sparer ca. 0.3 msec. pr. indiceret variabel).

10 bevirker, at der kommer udskrift af hver 10. linie i algolprogrammet (en hjælp ved fejlsøgning - andre positive heltal kan benyttes).

Normal vil den kaldte oversætter være standardudgaven ga4, men man kan vælge en anden udgave ved at give dennes navn, som første parameter til kaldet af algol. Der findes i øjeblikket følgende muligheder:

algol<	henter standardudgaven ga4
algol, pl<	henter oversætter med plotteradministration
algol, samba<	henter oversætter med båndadministration
algol, sambaplot<	henter oversætter med bånd og plotteradministration
algol, buffer<	henter oversætter, der kan udføre visse mindre algolprogrammer hurtigere end ga4.

Eksempel:

Hvis man har et program, der anvender plotteren, stående på området pax og man ønsker det oversat med udskrift på printeren af hver linie, og uden indexcheck, er den nødvendige hjælp 3-information altså

1, pax, algol, pl, n, 1<

Lidt om bufferoversætteren

Denne oversætter er i princippet en almidelig algol 4-oversætter uden plotteradministration. Den udmærker sig ved, at det færdigoversatte program, når det startes, begynder med at flytte sig selv over i bufferen fra celle 0 og fremefter. Det betyder, at den operation at hente et kanal segment ind i ferritlageret i stedet for at tage 20 millisek. nu tager 0.5 millisek., så hvis perogrammet tidligere anvendte megen tid på at vente på tromletransporter, kan man spare en del køretid. At programmet venter på tromletransporter, kan ses ved, at hele den nederste række lamper på det store kontrolbord lyser uden at YE-lampen lyser.

Da bufferen jo også anvendes til lagring af indicerede variable, kan det let blive et problem at få plads til det hele, Størrelsen af det oversatte program kan man finde ved at liste work lige efter at programmet er oversat, og antallet af indicerede variable kan man finde ud fra en programudskrift. Viser det sig at programmet ikke mangler meget i at kunne være i bufferen, vil det måske være ulejligheden værd at få det presset lidt mere sammen. I bedste (værste) fald kan programmet komme til at køre på en trediedel tid.

Plotteroversætteren

For at lette brugen af plotteren i algol 4-programmer er der lavet en speciel oversætter med navnet pl, dvs. den skal i hjælp 3 kaldes ved f. eks.:

```
algol,pl<
l,algol,pl,n,1<
```

eller lignende. Oversætteren beslaglægger ekstra ca. 120 celler i ferritlageret og 441 celler i bufferen. Den indeholder foruden de nedenfor beskrevne standardvariable og -procedurer en sekvens til at foretage den tidligere beskrevne zigzag-approximation og eventuelt levere output til en båndstation.

STANDARDVARIABLE

integer xxx, yyy;

Disse indeholder til enhver tid pennens position i forhold til origo, målt i hele step. Origo er pennens position ved start af programmet, men kan ændres enten ved kald af plotaxes eller ved, at man ændrer værdien af xxx eller yyy. Vil man således udråbe pennens øjeblikkelige position til origo gøres dette ved sætningen xxx:=yyy:=0;

□

Boolean magnetbånd

Er denne sand, vil alle plotterprocedurer levere output til en båndstation. Den initieres ved sætningen

```
magnetbånd:=ka on;
```

der udføres i det øjeblik, man kalder run.

Hvis man lader plotteroutput gå til en båndstation og slår plotteren fra på krydsfeltet, vil GIER kunne plotte 18cm/sek i stedet for 3cm/sek. Desuden får man muligheden for flere tegninger, hvis f. eks. pennen løber tør.

□

boolean stationnumber;

Denne indeholder i de første 10 bits nummeret på den båndstation, hvortil eventuelt output skal leveres, og i de næste 10 bits, det antal filemarks, der skal overspringes inden skrivning startes. Inden skrivning startes vil maskinen spole båndet tilbage og sikre sig, at det ikke er et hjælp3-bånd eller et samba-bånd; I så fald kommer udskriften nytape på skrivemaskinen, og man må enten montere et andet bånd eller, hvis man er sikker på at man må skrive på båndet, taste et c. Ved start udføres sætningen:

stationnumber:= 10 1 10 0;

dvs. skrivning på station 1 uden overspringelse af nogen file.

□

real scalex, scaley;

Disse er faste skalafaktorer og indeholder længden i centimeter af en enhed enhed i henholdsvis x-aksens og y-aksens retning. Ved start vil plotteroversætteren udføre sætningen:

scalex := scaley := 1.0;

svarende til at en enhed i plotterprocedurerne giver en centimeter på papiret. Vær forsigtig, hvis pennen ikke står i origo når scalex og scaley ændres.

□

STANDARDPROCEDURER

procedure plotline(x0,y0,x1,y1);
value x0,y0,x1,y1; real x0,y0,x1,y1;

Denne tegner den (næsten) rette linie fra (x0,y0) til x1,y1). Hvis første og andet punkt er ens, vil pennen blive flyttet til dette punkt, men ikke blive sænket. Dette kan f. eks. benyttes, hvis man ønsker, at tegningen skal starte 10 centimeter fra papirets venstre kant, således

plotline(-76,0,-76,0); xxx:=0; plotline(10,0,10,0); xxx:=0;

□

```
procedure plotgraph(x0, x1, x, y, dx);
value x0, x1; real x0, x1, x, y, dx;
```

Tegner funktionen y=y(x) i intervallet [x0, x1] ved at interpolere lineært mellem funktionsværdierne i x0, x0+dx, x0+2xdx ----- .

Funktionen y=sin(1/x) kan således tegnes ud i intervallet [0.01, 10] ved sætningen

```
plotgraph(0.01, 10, x, sin(1/x), 0.1);
```

eller

```
plotgraph(10, 0.01, x, sin(1/x), if x<1 then 0.01 else 0.5);
```

der tegner den anden vej og benytter større nøjagtighed nær nul eller

```
for i:=1 step 1 until 1000 do
y[i]:=sin(10/i); scalex:=0.01;
plotgraph(1, 1000, x, y[x], 1.0);
```

Bemærk, at vi ikke kan benytte en integer som dummy variabel, og at steplængden også skal være af type real.

□

```
procedure plotcurve(x, y, t, t1, t2, n);
value t1, t2, n; real x, y, t, t1, t2; integer n;
```

Tegner kurven x=x(t), y=y(t) for t løbende i intervallet [t1, t2] ved at dele dette interval og i n delintervaller og interpolere lineært mellem de fremkomne n+1 punkter. En regulær 17 kant med centrum i (8, 8) og radius 15 centimeter kan laves således:

```
plotcurve(8+15xcos(t), 8+15xsin(t), t, 0, 2xpi, 17);
```

□

```
procedure plotcond(x, y, t, e1, e2, b1, b2);
real x, y, t, e1, e2; boolean b1, b2;
```

Tegner samme kurve som plotcurve, men administrationen er mere generel. e1 er startværdien for t; e2 er et udtryk af formen t+dt, hvor dt opfattes som en skridtlængde, der igen kan afhænge af t. Proceduren vil kun tegne den del af kurven, for hvilken b2 er sand, og den hopper ud i det øjeblik b1 bliver falsk. Administrationen kan udtrykkes i sætningen:

```
for t:=e1, e2 while b1 do
if b2 then TEGN KURVEN;
```

Vi kan tegne den del af den ovenfor beskrevne 17-kant, der ligger i første kvadrant, ved sætningen:

```
plotcond(8+15×cos(t), 8+15×sin(t), t,
0.0, t+2×pi/17, t≤2×pi, cos(t)>-8/15∧sin(t)>-8/15);
```

□

```
procedure plotaxes(x min, x max, y min, y max, boo);
value x min, x max, y min, y max;
integer x min, x max, y min, y max; boolean boo;
```

Tegner et retvinklet koordinatsystem med origo i pennens øjeblikkelige position, x-akse fra x min til x max og y-akse fra y min til y max. Hvis boo er sand, vil plotaxes desuden sætte tal på aksene. Længden af en enhed er stadig styret af scalex og scaley. Hvis man f. eks. kun vil have tal på x-aksen, kan det gøres således:

```
plotaxes(-10, 10, -5, 5, yyy=0);
```

□

```
real procedure plotchar(t, x, y, h);
value t, x, y, h; integer t; real x, y, h;
```

Svarer til writechar. Tegnet med talværdi t tegnes, så det har nederste venstre hjørne i punktet (x, y). Uppercasetegn fås ved at addere 128 til

tegnets talværdi eller ved explicit at kalde plotchar med t=60. h er højden i centimeter af et stort bogstav uanset værdien af scalex og scaley. Værdien af plotchar er x-koordinaten til det næste tegn på papiret. Et program, der skriver en strimmel ud indtil første CR kan se således ud:

```
x:=0; y:=1;
for t:=lyn while t≠64 do x:=plotchar(t, x, y, 0.3);
```

□

```
procedure plottext(s, x, y, h, d);
value h, d; real x, y, h, d; string s;
```

Svarer til writetext. s er en textstring, der vil blive tegnet ud (med plotchar) med første karakter i (x, y). h og d er henholdsvis bogstavhøjde og linieafstand i centimeter, uafhængigt at scalex og scaley. Efter et kald af plottext vil (x, y) være koordinaterne til næste karakter på papiret; deraf følger, at der på kaldsstedet skal stå simple variable på x's og y's pladser.

Eksempel:

```
x:=5; y:=-3;
plottext({<Jeg græmmes}, x, y, 0.3.0);
```

□

```
real procedure plotinteger(t, x, y, h);
value t, h; integer t; real x, y, h;
```

Eneste forskel fra plotchar er, at selve talværdien af t tegnes ud. Tallet vil fylde så mange anslag, som der er cifre i t plus eventuelt et fortegn.

□

integer procedure define symbol(b1, b2);
value b1, b2; boolean b1, b2;

Indsætter et nyt tegn i alfabetet. Et tegn beskrives således: Vi nummererer punkterne (0, 0), (0, 1)....., (0, 5), (1, 0)....., (4, 5) med tallene fra 0 til 29. Et tegn kan da bestå af op til 14 liniestykker mellem disse 30 punkter. Numrene på de indgående punkter pakkes i b1 og b2 med 5 bits pr. nummer. Desuden skal de to første tal angive, hvor mange punkter vi skal igennem, og hvor vi skal starte.

Sætningen:

```
lille e:=define symbol(5 11 5 25 5 18 5 12 5 7 5 8
                       5 15 5 21 5 26 5 19 5 13 5 8);
```

vil bevirke, at beskrivelsen af et e indsættes på en ledig plads i tabellen over tegn, og i lille e indsættes en pointer. Herefter vil sætningen

```
plotchar(lille e, .....
```

tegne et e. Efter at have udført sætningen

```
KV:=define symbol(5 5 5 0 5 1 5 7 5 6 5 0, false);
```

kan vi ved at variere bogstavhøjden tegne et kvadrat af vilkårlig størrelse ved hjælp af plotchar. Vil man løfte pennen undervejs, indskyder man det fiktive punkt 31. Et lighedstegn hedder således

```
(5 5 5 8 5 20 5 31 5 21 5 9, false).
```

□

procedure drejplot;

Hvert kald af drejplot vil rotere koordinatsystemet 90 grader i positiv retning omkring pennens øjeblikkelige position.

□

procedure slutplot;

Retter koordinatsystemet op, hvis drejplot har været anvendt. Hvis der har været plotteroutput til en båndstation flyttes pennen helt ud til venstre side, og der sættes et filemark på båndet. Da operatøren selv kan beslutte at plotte via tape - og normalt vil gøre det - skal alle plotterprogrammer slutte med at kalde slutplot.

□

procedure plotstop;

Vil ved off-line plotning medføre, at converteren standser; nyttigt, hvis man vil tegne i flere farver.

□

ANDRE PROCEDURER

Foruden de beskrevne procedurer, der alle er bygget ind i plotteroversætteren, findes på disken yderliger to procedurer, der kan kopieres ind i et program ved sætningen

```
copy plot<
```

```
procedure plotfunct(f, x, y, g, h, s, t, m, n, d);
value m. n. d; integer m, n, s, t; real f, x, y, g, h, d;
```

Denne procedure tegner de konturer, der svarer til, at funktionen f=f(x, y) antager værdierne 0, ±d, ±2d .... x og y, der angiver koordinaterne på papiret, beregnes ud fra henholdsvis g og h, der igen er funktioner af s og t. Idet s og t gennemløber værdierne fra nul til henholdsvis m og n får vi således planen indelt i et (ikke nødvendigvis rektangulært) net, indenfor hvis masker plotfunct vil anvende lineær interpolation. Niveaukurver for funktionen x↑2-(y-2)↑2 for x i intervallet [-5, 5] og y i intervallet [∩, 5] fås ved sætningen

```
plotfunct(x↑2-(y-2)↑2, x, y, -5+0. 5×x, 0. 5×t, s, t, 21, 10, 0. 5);
```

□

```
procedure plotdim(f, x, y, a, R, u, tværs, n);
value R, a, u, tværs, n; integer n;
real f, x, y, a, R, u; boolean tværs;
```

Tegner en perspektivisk tegning af funktionen f(x, y) for x og y liggende i intervallet [-a, a]. R er afstanden fra øjet til forkanten af dette kvadrat, og u er vinklen mellem denne afstand og planen. Hvis tværs er falsk, tegnes n lodrette snit i funktionen parallelt med x-aksen; er tværs sand, indtegnes også snit vinkelret på x-aksen; Skalafaktorerne i planen justeres, så at kvadratets forkant fylder 2a enheder på papiret.

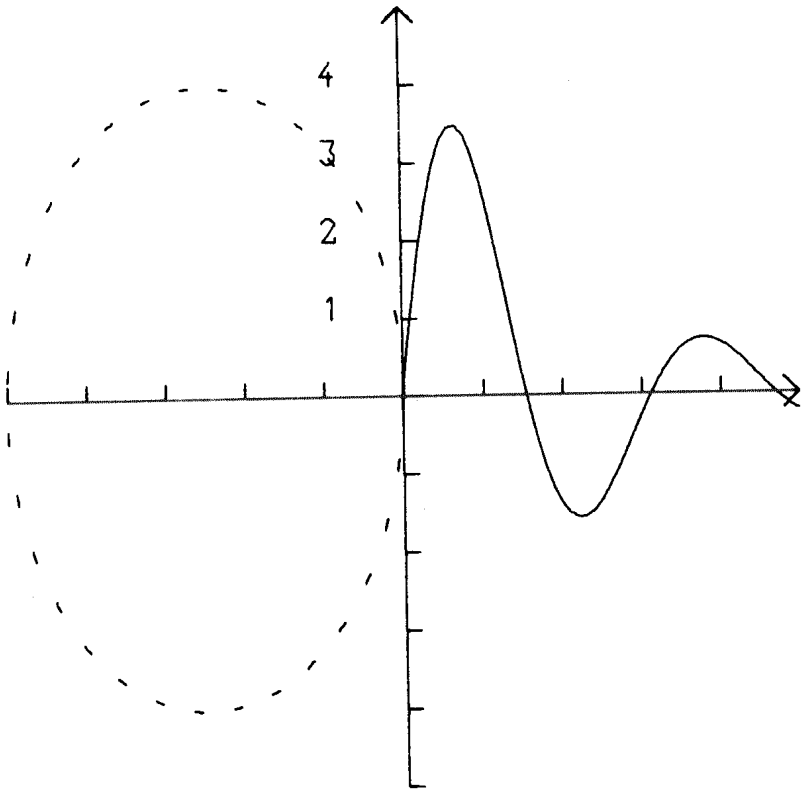
□

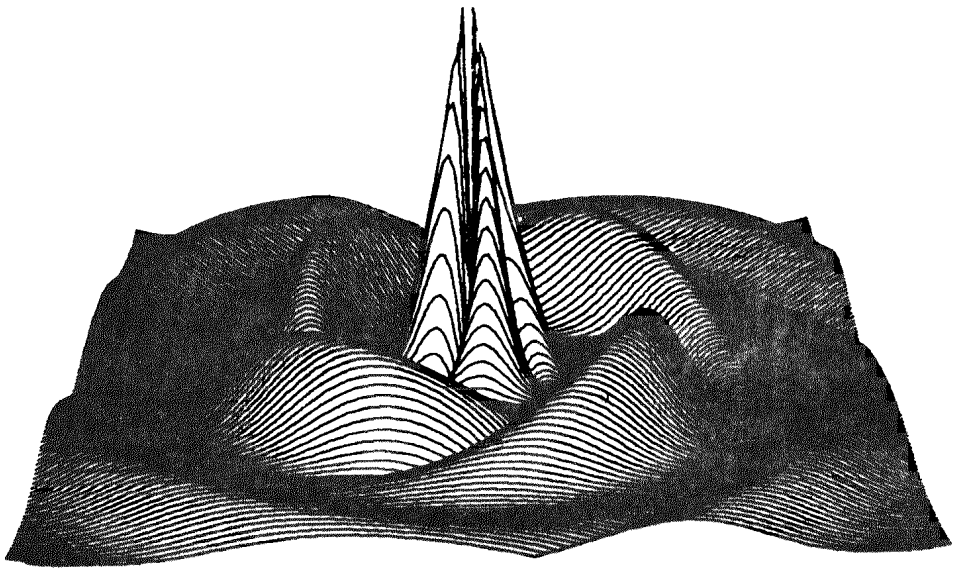
Output fra følgende plotterprogram kan ses på næste side.

```
a,pl<
begin integer i,p,q;
   real x,y,t,r,fi;
   copy plot<
   real procedure polær;
   begin
      r:=sqrt(x↑2 + y↑2);
      fi:=arctan(y/(x + 10-12));
      polær:=8.0×cos(r)↑2×abs(sin(2×fi + r))/(1.5 + r↑(1.3))
   end polær;

   plotline(-78,0,-78,0); xxx:=0;
   plotline(11,0,11,0);
   plotaxes(-5,5,-5,5,yyy>0);
   x:=-5.0; y:=6.5;
   plottext(‡<Demonstration af plotterprocedurer‡,x,y,0.3,1);
   x:=-5.0; y:=6.3;
   for i:=1 step 1 until 34 do  x:=plotchar(32,x,y,0.3);
   plotgraph(0.0,5.0,x,5.0×exp(-x/2)×sin(2×x),0.1);
   plotcond(-2.5+2.5×cos(t),4.0×sin(t),t,0.0,t+0.04,t≤6.3,
            entier(t×10)mod2=0);
   plotline(0,-10,0,-10); yyy:=0;
   plotfunct(sin(x+y)/(1+x),x,y,p/10.0,q/10.0,p,q,50,40,0.1);
   plotline(0,-9,0,-9); yyy:=0;
   plotdim(polær,x,y,6.0,30.0,0.5,false,150);
   slutplot
end
t<
```

Demonstration af plotterprocedurer





TAPEOUTPUT

Programmer der leverer meget output eller skal køre lang tid uden operatør, kan med fordel benytte tapeoutput. Tapen skrives ud over converteren efter endt kørsel ved hjælp af katalogstrimlen SB-PT.

De nødvendige initialiseringer sker ved et kald af proceduren tapeoutput, der tænkes at have følgende hoved:

```
procedure tapeoutput (stationsnummer, a);
value stationsnummer; integer stationsnummer; array a;
```

a skal være erkjæret fre 0 til 2xn, hvor n er et heltal mellem 1 og ca. 200; a vil blive anvendt som dobbeltbuffer, så at hver blok på båndet vil indeholde 4xn tegn. Jo større n, jo hurtigere vil pakningen ske. Den øvre grænse sættes af at converteren ikke kan behandle blokke med mere end ca. 800 tegn. Udover at erklære arrayet a må programmøren overhovedet ikke røre arrayet a.

Virkningen ved et kald af tapeoutput er, at der hentes en sekvens frem til skrivning på bånd, desuden spoles båndet helt tilbage til loadpoint, og maskinen sikrer sig at det monterede bånd ikke er et hjælp3-bånd (i så fald skal Gier startes med et space, når et nyt bånd er monteret). Ingen anden test foretages.

select

Så snart tapeoutput har været kaldt, kan man vælge båndstationen helt parallelt med de øvrige outputenheder (perforator, printer mm.) ved at addere 128 til selectet. Således vil select(135) give output på bånd og printer, og input fra skrivemaskinen.

endtape

Ligesom ved plotning via magnetbånd, skal båndet afsluttes. Det sker ved et et kald af proceduren endtape, der desuden fjerner båndadministrationen fra ferritlageret. Har tapeoutput ikke været kaldt er endtape virkningsløs.

Har man et program med et extremt stort printeroutput med mange formularskift, og korte linier, kan det betale sig at benytte bånd til gemning af printeroutput. Med n=200 kan der således være ca. 1000 sider printeroutput på et helt lille bånd.

(tapeoutput fortsat)

De tre tegn med talværdi 45, 46, 47, der alle er ubenyttede i algol, anvendes til henholdsvis 64, 65 (linieskift med overspringelse af tværperforeringen) og 66 (formularskift).

Båndadministrationen optager 40 celler i ferritlageret. Programmer med en meget speciel struktur kan derfor komme til at køre meget langsommere end med output direkte til de ydre enheder. Desuden tager det i snit 0.8 msek. pr. tegn at skrive på magnetbåndet; er båndet ikke valgt gennem select, er der ingen forsinkelse.....

Endelig skal det bemærkes, at den beskrevne metode, er den eneste mulighed for at få perforatoroutput, hvis man vil lade maskinen køre alene uden opsyn. Perforatoren må jo ikke køre når der ikke er en operatør tilstede.

□

TAPEINPUT

Ligesom der er mulighed for at få output på tape, kan man også få input fra tape. Det sker ved hjælp af proceduren tapeinput. tapeinput er ikke en standardprocedure og er derfor ikke med i algoloversætteren. Vil man derfor benytte tapeinput, må man mellem sine erklæringer i yderste blok have:

```
copy tapeinput<
```

De nødvendige initialiseringer sker ved et kald af proceduren tapeinput som tænkes at have følgende hoved:

```
procedure tapeinput(stnr,a);
value stnr; integer stnr; array a;
```

a skal være erklæret fra 0 til 2×n, hvor n er et heltal mellem 1 og ca. 200. a vil blive anvendt som dobbelt-buffer. Udover at erklære dette array, må brugeren ikke røre ved dette array.

paritet

Hvis man lægger 128 til stnr læses med lige paritet, ellers læses med ulige paritet.

bcd

Er det bånd man vil læse, et bcd-bånd, vil et kald af proceduren bcd gøre det muligt at læse normalt, altså:

10 ændres til 16
16 ændres til 0

har man først kaldt bcd, har man ingen mulighed for at vende tilbage til normal karakteropfattelse.

tapein

endin

Det indskærpes at man kun må kalde tapeinput en gang i løbet af kørslen. Så snart man har kaldt tapeinput, vil næste indlæsning foregå fra tape, ved hjælp af proceduren endin vender man tilbage til sædvanlig indlæsning via sidste kald af select. Hvis man senere vil læse på tape igen, sker dette ved et kald af proceduren tapein. Valg af outputmedium påvirkes ikke af tapein og endin.

□

Kontroludskrifter fra et kørende Algol 4-program

Som et experiment er der lavet en Algol 4-oversætter med navnet tr (med tilhørende plotter-variant trp), der i et opgivet linieinterval kan give udskrift af næsten alle assignments, der bliver udført under kørslen af et program.

Det ønskede linieinterval opgives som parametre kaldet af oversætteren ligesom ved kald af hjælpeprogrammet line, ønsker man udskrift af alle assignments i linierne 13 til 35 og 75 til 87 i programmet cha gøres det ved hjælp 3-informationen:

```
cha, a, tr, 13..35, 75..87<
```

Metoden er denne: Under indlæsning af programmet vil oversætteren hver gang, den møder et begin eller et semikolon i det opgivne linieinterval begynde at opsamle de indlæste tegn i en lille intern buffer. Viser den påbegyndte sætning sig at være en simpel aritmetisk sætning, bliver hele den tekst der udgjorde venstresiden, gemt og benyttet i et kald af proceduren writetr, der automatisk bliver kopieret ind i starten af programmet. De to sætninger

```
i:=1;
A[i+5]:=1.0;
```

vil komme ud af pass 1 således:

```
i:=1; writetr (boolean i, {<i});
A[i+5]:=1.0; writetr (boolean A[i+5], {<A[i+5]});
```

og under kørslen vil der komme følgende udskrift på printeren:

```
          ⇒ i
1.000000 ⇒ A[i+5]
```

Da hele mekanismen skulle lægges ind i pass 1, hvor der er meget lidt plads har det været nødvendigt at fjerne en del sjældent benytte mekanisker fra denne passage:

1) Alle fejludskrifter er fjernet

2) Punch off og Punch on er blinde

3) Endcode, Clearcode og Sumcode er blinde

4) message behandles som comment

5) Fede kommaer er ikke tilladt.

6) Ved input fra skrivemaskinen kan man ikke rette en linie ved at taste fire case-skift.

Derudover medfører den anvendte mekanisme følgende restriktioner:

1) Sætninger umiddelbart efter then, else eller do vil ikke blive "fanget".

2) Hvis der på venstre side af := findes et procedurekald, vil denne procedure blive kaldt en extra gang. Der må derfor ikke være sætninger af typen A[readinteger]:=.....; i det specificerede linieinterval.

3) Der må ikke ske et assignment til procedurenavnet i en type-procedure.

4) Hvis mekanismen anvendes i et case statement, hvor der findes simple statements, vil de indsatte kald af writetr få maskinen til at tælle galt.

På trods af de nævnte begrænsinger vil den beskrevne metode i en del tilfælde være den hurtigste vej til at finde f. eks. en indexfejl eller et splill.

På den anden side må man ikke udelade sine egne kontroludskrifter, da erfaringen viser, at tracing af et program giver så store outputmængder, at man let drukner i overflødig information.

□

run

u

Disse to programmer er helt identiske. De undersøger om der findes et oversat program på work, er dette tilfældet, startes programmet, ellers kommer fejludskriften "not present". Udover at starte programmet kommer der også en overskrift på printeren, der angiver dato og sted samt en eventuel meddelelse til brugerne.

y

er helt identisk med run og u, bortset fra overskriften, som er udeladt.

Alle tre programmer udfører det samme som beskrevet under run i help 3-manualen + følgende:

Hvis der er en parameter til run eller u eller y, og denne parameter beskriver et buffermedium, vil området (med move) blive flyttet til work (work vil starte på spor 72 lige efter image), hvorefter kontrol overgives til det kaldte program. Work er naturligvis derved blevet ændret.

Alarmudskrifter

kind	hverken tromle eller bufferareal er kaldt.
undef	det kaldte område er ikke beskrevet i kataloget
not present	det kaldte område var ikke et oversat algolprogram.

Eksempel:

Et oversat program, der er reserveret under navnet paxo på disken ønskes kørt. Der ønskes ikke overskrift på printeren, hjælp 3-informationen er da:

y, paxo<

Vigtigt!

Man må ikke trykke på hp medens run, u eller y arbejder!

□

dump

Ved hjælp af dump, kan man afbryde et kørende program, for senere at gøre kørslen færdig. Visse betingelser skal dog være opfyldt. Disse er:

1) Programmet skal være et Algol 4-program eller et slip-program.

2) Celle 1-9 må ikke røres (kun relevant for slip-programmer).

3) Kanal 46-71 må ikke røres (dvs. et oversat program må højst fylde 248 tromlekanaler).

4) Programmer, der anvender put og get skal behandles med omhu. Hvis et program anvender put og get direkte på free, er dumping ikke mulig, og da operatøren kan blive nødt til at dumpe en brugers program er det i brugerens egen interesse altid at reservere sig et område til brug for put og get. Da imidlertid put og get anvender absolutte adresser på disken, må man ikke flytte de benyttede områder. Et dumpet program, der benytter disken, kan således ikke i almindelighed fortsættes, hvis der i mellemtiden har været benyttet compress eller clean. Der er dog mulighed for at klare dette problem, snak med datalogisk afdelings medarbejdere om dette.

Hvis et program opfylder de nævnte 4 betingelser, kan det dumpes ved, at man trykker på hp (aldrig reset), og derefter taster f. eks.:

dump, d3. 11. 71, alprgd<

Virkningen af dette kald af dump er følgende:

Først gemmes bufferens indhold på bufim. dernæst undersøges, om det opgivne navn allerede findes i kataloget, er dette tilfældet, flyttes bufim, image og work til det angivne område, og der indsættes i kataloget en oplysning om works placering, hvis området ikke findes i forvejen, reserverer dump et passende stort område og fortsætter som ovenfor.

Hvis programmet ikke har læst færdigt fra strimmellæseren, når det bliver afbrudt, er man nødt til at kopiere de resterende data over på en ny strimmel ved hjælp af copy. Man kan nemlig ikke vide, hvor man skal starte den oprindelige datastrimmel når programmet fortsættes.

Hvis et program benytter magnetbånd, skal man selv sørge for at båndet ikke røres før programmet fortsættes.

Et dumpet fortsættes ved hjælp af programmet cont.

□

cont

benyttes til at starte et dumpet program med. Kaldet er som følger:

cont, <navnet på det dumpede program><

følgende fejludskrifter kan komme i forbindelse med dump og cont:

no work spec
Man forsøger at dumpe over i et område, hvortil der ikke i kataloget er afsat plads til en beskrivelse af work. Man kan altså kun dumpe oven i et område der oprindelig er lavet af dump

improper work spec.
Man har sagt cont til et område der ikke indeholder et dumpet program.

overlap
Man forsøger at dumpe oven i et område der er blevet for lille.

□

saveb
exitb

De to programmer vil henholdsvis gemme bufferens indhold på diskområdet bufim og hente bufim frem i bufferen. De har ingen parametre.

Hovedformålet med programmerne er at muliggøre anvendelsen af hjælpeprogrammer i forbindelse med et kørende algol 4-program. Vil man under kørslen af et program se indholdet af bufferens celle 3000-4020 på integer-form kan dette gøres således:

1. tryk hp
2. tast saveb<
3. tast l, print, bufim, w 5, i, 5. 0. 420<
4. tast exitb<
5. tast exit<

□

Et kørende Algol 4-program kan benytte move til at flytte disk-området data over på båndområdet tape således:

```
code i;
2, 44
hs 1
hv re1
t saveb;
0c
e1: hs 1
hv re2
t move;
t data;
t tape;
0c
e2: hs 1
hv re3
t exitb;
0c
e3: e;
```

□

Editering og udskrift

edit

e

er et retteprogram. Det læser rettelser ind fra hjælp's løbende inputmedium, retter teksten (som specielt kan være et algolprogram) i et nærmere angivet inputareal og lagrer den rettede tekst i et nærmere angivet outputareal, hvis intet angives som inputareal vælger edit strimmellæseren, og hvis intet angives som outputareal vælger edit perforatoren.

Eksempel:

r, edit<

I dette eksempel læses rettelserne fra hjælp's løbende inputmedium (r=strimmellæser). Den tekst der skal rettes læses også fra strimmellæseren, og den rettede tekst kommer ud på perforatoren.

free, edit, opax<

I dette eksempel læses rettelserne fra det frie areal på disken (free), den tekst der skal rettes læses fra strimmellæser, og den rettede tekst kommer til at ligge på et område kaldet "pax".

edit, ipax, ofree<

rettelserne læses fra hjælp's løbende inputmedium, som når intet specificeres er skrivemaskinen. Den tekst der skal rettes befinder sig på området pax og den rettede tekst lægges ud på free.

Rettelselisten

Rettelselisten har form af en liste af enkeltrettelser, som skal komme i den rækkefølge, hvori de forekommer i programmet.

Enkeltrettelse består af tekststrenge adskildt af .

<kopier til og med> . <indsæt> . <slet til og med> .

listen af rettelser afsluttes med en stop code eller et å. Bemærk at cr og space er blinde symboler i strengene <kopier til og med> og <slet til og med>. Det tilrådes af hensyn til edit at afslutte sit program med en stop code.

Eksempel:

rettelsesstrimmel:

```
edit,o free<
({.<Jens.Jens.
STOP CODE
```

programstrimmel:

```
algol<
begin integer i,j;
...
select(8);
writetext({Jens Olsens program});
i:=readinteger;
...
end t<
STOP CODE
```

Udover det her beskrevne anvendelsesområde, har edit flere andre nyttige funktioner. en beskrivelse af disse kan findes i hjælp 3-manualen.

e1

er identisk med edit (e) bortset fra at mens edit vender tilbage til skrivemaskinen, vender e1 tilbage til løbende input.

□

e2, a2,
ok,nonok

Hjælp 3 giver brugerne mulighed for at tilrettelægge mere indviklede kørsler så at operatøren kun behøver at taste r< på skrivemaskinen. Derved har man opnået væsentlig sikkerhed mod operatørfejl, men i visse situationer har virkningen været den stik modsatte.

Eksempel:

Man har lavet en rettelsesstrimmel, der skal foretage en helt enfoldig rettelse i et af ens områder på disken og lader derfor edit aflevere det rettede program på samme område som det tog input fra. Imidlertid har man lavet en lille fejl i sin rettelsesstrimmel, og rettelsen går galt, så at man har mistet hele sit program. Hvad man kunne ønske sig er en mulighed for først at rette programmet, derpå, hvis rettelserne gik godt, da flytte den nye version tilbage på de oprindelige område, dette kan opnås ved anvendelsen af e2, a2, ok, nonok.

e2

e2 afviger fra edit på to punkter. For det første vender det ikke tilbage til skrivemaskinen efter at have udført rettelserne, men vender tilbage til løbende inputmedium (derved ligner det edit-varianten e1), for det andet slutter e2 med at sætte en intern boolean til true, hvis alle rettelser blev fundet og det rettede program blev afleveret i hel tilstand (dog ikke overlab) ellers sættes den interne boolean til false.

a2

a2 er identisk med hjælpeprogrammet algol (a) udover, at a2 sætter omtalte boolean til sand, hvis oversættelsen gik godt ellers false.

ok og nonok anvendes herefter til at spørge på den interne boolean. Ethvert hjælp 3 kald kan indledes med enten ok eller nonok, og resten af hjælp 3-kaldet vil da kun blive udført, hvis den interne boolean er sand henholdsvis falsk.

Eksempel:

Se eksemplet side 59

□

out

Formål:
at udskrive tekstområder på linieskriver. out vender tilbage til løbende inputmedium.

kald:
out, <input><
<input>::=<drum>| <buffer medium>

Eksempel:

out, free<

teksten på free udskrives på linieskriver.

jum3, out, jum3<

teksten i jum3 udskrives på linieskriver, og derefter oversættes jum3 (jum3 har naturligvis de dennødvendige help3 kald).

Virkemåde

Linieskriveren kan normalt modtage karaktererne lige så hurtigt som GIER kan aflevere dem. En undtagelse er karakteren lige efter et linieskift. Ved

at opbygge en hel linie ad gangen i GIER, medens linieskriveren skriver den foregående linie, og dernæst hurtigt afleverer linien, er der sparet megen tid (dobbel buffer princippet).

Ved skrivning af linier der har færre end 10 karakterer er edit og out lige hurtige, i alle andre tilfælde er out hurtigere.

□

pæn

Formål:
at udskrive algolprogrammer med understregninger og gennemstregninger sat som ved flexowriterudskrifter. pæn virker fuldstændig som out, men tager dobbelt så lang tid om en udskrift. Det henstilles at man indskrænker brugen af pæn til det absolut minimale.

□

alged
ae

Formålet med dette program er at kunne få en printerudskrift af et algolprogram, hvoraf begin - end strukturen tydeligt fremgår. Da det er ønskeligt at udskriften leveres hurtigst muligt, antager hjælperpogrammet at algolprogrammet opfylder følgende:

1) der er ikke compound-fejl i beg eller end.

2) der forekommer ikke beg eller end i messages eller i code-comments, og beg forekommer ikke i implicitte comments efter end.

Alle indrykninger der måtte forekomme i programmet i forvejen fjernes. Dvs. hvis strings indeholder cr (vognretur) ødelægges opstillingen i programudskriften. Trods disse mangler og restriktioner har programmet

vis sig særdeles anvendeligt, når man ønsker overblik over strukturen i et algolprogram, og ikke selv laver denne ved hulning af programmer og evt. påfølgende rettelser.

For at kunne udskriften til opsøgning af evt. syntaktiske fejl i programmet er udskriften forsynet med linienummerering, som dog kan udelades, hvis det ønskes.

Programmet har skrivebuffer i ferritlageret ligesom out, og er kun 5% langsommere end dette.

Det er muligt at begrænse udskriften til intervaller, som ved line.

kald:

ae,<input>,<forms>,<lines><

<input>::=<empty>/<area>

<empty>: input fra strimmel

<area> : input fra området <area>

<forms>::=<empty>/<form>/<form><form>

<form>::= s/=

<empty>: udskrift med linienumre

s : udskrift med overspringelse af perforering

= : udskrift uden linienumre

<lines>::=<empty>/l<linelist>

<empty>:hele området udskrives

<linelist>: se under line

Der kan ikke vælges outputmedium, hvilket naturligvis skyldes, at man kun har brug for indrykning på printeroutput, og at det er ganske unødvendigt at have hundredevis af overflødige spacer på strimmel eller interne områder.

□

line

Formål:
at udskrive et algolprogram med linienumre i udvalgte intervaller. Input fra alle medier, output på ydre medier. Vender tilbage til løbende inputmedium, som ethvert andet hjælpe gram. Udskrift i overensstemmelse med algols linienummerering.

Kald:
line, <input>, <linelist><
<input>::=<external>| <drum>| <buffermedium>| <empty>
<linelist>::=<helpnumber>| <helpnumber>, <linelist>| <empty>
<helpnumber> er af formen first line .. last line
hvis <input>=<empty>, vælges strimmellæser som input
hvis <linelist>=<empty> afbrydes programmet og hjælp fortsætter på løbende input. Hvis first line = last line, kan man nøjes med at skrive last line, når det ikke er første helpnumber.

Eksempel:

line, free, 15. . 18<

skriver line 15 til line 18 ud på løbende outputmedium af algolprogrammet på free.

w, line, jum3, 1000<

skriver de første 1000 linier ud på skrivemaskinen af algolprogrammet på jum3.

l, jum3, line, jum3, 13. . 15, 20, 30, 100. . . 125, 200<

Dette kald skriver linierne 13-15, 20, 30, 100-125, 200 ud på linieskriveren derefter kaldes jum3 for at oversætte (idet jum3 naturligvis begynder med a, n< eller lignende).

Alarmudskrifter

param	<linelist> er forkert
undef	<input> eksisterer ikke
text end	teksten er ikke så lang som teksten fordrer.

Bemærkning

Hvis man benytter copy< i sit algolprogram, passer linienumrerne ikke.

□

outmt

Formål:
Programmer eller andre tekstområder på internt medium der ønskes udskrevet på perforator lægges over på magnetbånd og udskrives over konverteren for at spare tid på GIER.

Kald:
outmt, <input>, <numberlist><
<input> er navnet på det område der ønskes udskrevet
<numberlist>::=<empty>| <helpnumber>| <helpnumber>, <helpnumber>

Virkning af kaldet:
Teksten på inputområdet lægges over på magnetbånd således:

<numberlist>	station no.	file no.
<empty>	2	0
i	2	i
i,j	i	j

Eksempel:

outmt, free<	skriver free på station no. 2 file no. 0
outmt, atm, 3<	skriver atm på station no. 2 file no. 3
outmt, pax, 4, 1<	skriver pax på station no. 4 file no. 1

Udskrivning:

Ved udskrivning benyttes SB - PT katalogstrimlen.

Fejludskrifter:

Udover de sædvanlige hjælp 3 udskrifter kan følgende forekomme:

file protect

Skrivning ikke tilladt på det monterede bånd.

illegal tape

Der forsøges skrevet på et bånd med {<cattap} label, når et nyt bånd er monteret vil et SPACE på skrivemaskinen få programmet til at starte forfra.

Programmet spoler selv båndet tilbage til loadpoint før skrivning, der sættes filemark efter hvert områder der er lagt op på båndet.

NB! NB! NB!

Man opfordres kraftigt til at benytte outmt, når mans kal have tekstområder ud på puncher, istedet for edit, ofree<å eller lignende.

□

print

Print er et uhyre nyttigt hjælpeprogram, som kan udskrive et område fra baggrundslagrene, ferritlageret eller bufferen på enhver ønskelig form. Se hjælp 3 manual p. 38.

□

lis

kald:
lis, <parameterliste><

hvor <parameterliste> er den samme som for hjælpeprogrammet list (se hjælp 3 manual), med den begrænsing at intet navn må være længere end 6 tegn.

Virkningen af et kald af lis er følgende:

For hvert navn i parameterlisten udskrives alle katalogindgange, hvori forekommer et navn, hvis første bogstaver er identiske med navnet i parameterlisten.

Udskriften sker i overensstemmelse med <form>.

□

move

Virker som beskrevet i hjælp 3 manualen. Følgende alarmer kan forekomme udover de sædvanlige:

length:

Den i kaldet specificerede længde er længere end inputområdets længde, eller inputområdet har længden 0 (kan under normale omstændigheder ikke forekomme i kataloget, men kan være tilfældet, hvis input specificeres som b<helpnumber>).

end of tape input:

end-of-tape-mark er passeret på inputbåndet.

end of tape output:

end-of-tape-mark er passeret på outputbåndet.

illegal move:

der må ikke flyttes til outputområdet (kan f. eks. være et hjælpeprogram)

Der kan forekomme følgende meddelelse, der ikke er en alarm (dvs der vendes tilbage til løbende input):

filelength:

Forekommer kun ved tapeinput og betyder, at inputområdet er kortere end angivet i kataloget. Hele filen, (dvs. indtil filemark) er flyttet, og ved output på free er booked justeret, ved output på work er work as input justeret og ved output på tape er outputområdets længde justeret (hvis output er specificeret ved navn).

Dette kan man drage nytte af, hvis man har et hjælp3-bånd, men ikke har nogen beskrivelse af de skrevne files. Man sætter blot i kataloget en beskrivelse af hver file men med en meget stor bloklængde (så stor at den med sikkerhed er større end filelængden på båndet) og flytter efter tur hver file til free og reserverer. Hvis meddelelsen filelength ikke kommer, har bloklængden været for lille (eller evt. passet præcis). En anden mulighed er at kopiere båndet med move og gemme beskrivelserne af båndområderne på det nye bånd, dvs. bruge excl eller lis,r<

Det bemærkes til sidst at de fejl, der var i de to tidligere versioner af move, er rettet, dvs. beskrivelsen af work ødelægges aldrig, når move bruges (move, work, free <res,nno<y< virker altså altid); b ved tape-

output fører heller ikke mere tiludefinerede situationer; men af sikkerhedsmæssige grunde kan det ikke indskærpes kraftigt nok, at man undlader at bruge b-faciliteten ved output.

□

moveb

kald:
moveb, <areaname><

moveb flytter den del af free der er kooked til <areaname>, idet det først udregnes hvormange blokke á 40 ord der er booked, og derefter kaldes move med de rigtige parametre.

Eksempel:

Antag at man med move eller edit har lagt information ud på free, og at denne information fylder 5 diskspor, booked vil da være sat til 5. Denne information ønsker man nu at flytte til et område kaldes pax3, som også fylder 5 diskspor.

moveb, pax3<

genererer da kaldet

move, free, pax3, 75<

□

f

Hjælpeprogrammet laver så mange formularskift på printeren (via 6 på krydsfeltet), som parameteren angiver.

f, 2< to formularskift
f< et formularskift

□

cr

Laver så mange linieskift på printeren som parameteren angiver.

□

blank

Laver så mange blanke (fremføringshuller), på papirstrimmel, som parameteren angiver:

blank, 50< 50 blanke
blank< 100 blanke.

□

Eksempel:

Vi er nu i stand til at give et eksempel på en veltilrettelagt kørsel. Antag at en bruger har et algolprogram på et område kalde alprg. Han ønsker nu at rette i dette program, oversætte den rettede version, og hvis alt er gået godt, at gemme det oversatte program under navnet alprgo. For at genere de øvrige brugere mindst mulig er kørslen tilrettelagt således at der kun skal rastes r< på skrivemaskinen. Al anden information findes på strimmel:

move, alprg, free<	alprg kopieres til evt. senere brug.
e2, ialprg, oalprg<	der rettes med kildetekstinput fra alprg, og den
<rettelser>	rettede version lægges tilbage på alprg.
<STOPCODE>	rettelsesliste slut.
nonok, moveb, alprg<	hvis rettelsen ikke gik godt, kopieres free tilbage
nonok, t<	til alprg, og skrivemaskinen vælges som inputmedium.
l, algprog, a2<	det rettede program oversættes.
nonok, l, line, alprg, 1000<	hvis oversætt. ikke gok godt, uskrives program-
	met med linienr. og skrivemaskinen vælges som
nonok, t<	hjælp's løbende inputmedium.
move, work, free<	det oversatte program kopieres til free.
res, d1. 10. 71, alprgo<	det oversatte program reserveres til den 1. okt. 71.
t<	skrivemaskinen vælges som løbende input.

□

Specielle fejludskrifter:

SUM (med rødt og stort)

mainhelp er ødelagt. Indlæs hjælp 3 disk system (findes på strimmelbakken eller i nederste skuffe under læseren) med et space. Hvis hp-udskriften ikke kommer nu,tilkaldes en systemoperatør.

catalog (med rødt og lille)

kataloget er ødelagt. Indlæs hjælp 3 disksystem med <. Hvis hp-udskriften ikke kommer, tilkaldes en systemoperatør.

sum (med rødt og lille)

man har prøvet at kalde et program med sumcheck, summen er imidlertid forkert (fordi programmet er ødelagt), hvis det kaldte program er enten saveb, dump eller exit, indlæses hjælp 3 disk system. I alle andre tilfælde tilkaldes en systemoperatør.

image

hvis udskriften kommer efter kørsel af et algolprogram, betyder det at det oversatte program er så stort at det nør ind i image (kanal 46-71 på tromlen). Der tastes et space og situationen er normal. Advanced users information: Se "A Manual of help 3".

Hvis systemet iøvrigt gør alvorlige knuder tilkaldes i arbejdstiden en operatør, ellers gøres følgende:

1) På øverste hylde ved siden af højre båndstation udvælges omhyggeligt det store bånd der har den ældste dato.

2) Dette bånd monteres på en magnetbåndstation, der tildeles nummeret 1.

3) Den valgte station tilkobles GIER på omkoblingskassen.

4) På strimmelbakken eller i nederste skuffe under læseren, findes en strimmel med navnet' hent eller gem. Denne strimmel indlæses med <

5) Når programmet spørger hent eller gem tastes g.

6) Når gem er færdig afmonteres båndet, og det afleveres hurtigst muligt til en operatør, sammen med en fyldestgørende beskrivelse af uheldet, samt bilag i form af de sidste sider skrivemaskine-output.

7) På hylden beskrevet under pkt. 1 findes nu det bånd med den nyeste dato.

8) Dette bånd monteres på den nys brugte station, og hent eller gem indlæses.

9) Når programmet spørger hent eller gem tastes h.

10) Når programmet er færdigt kan SUM eller catalogudskriften forekomme.

11) Hvis systemet ikke er i orden nu, pakker man sin taske sammen og efterlader en besked til næste bruger om at systemet er nede.

Når GIER er slukket:

0) Læs denne vejledning grundigt igennem før noget andet. Alle pkt. skal udføres i den beskrevne rækkefølge.

HVIS DETTE IKKE GØRES KAN DER SKE STOR SKADE PÅ DISK OG BUFFER

1) Start GIER (bag højre paneldør idet første skab), hvis fejllampen lyser - tryk reset.

2) Tænd tromlen (i det lelle skab bagved GIER under vinduet). Der er tændt hvis normal-lampen lyser, ellers åbnes lågen og der trykkes på den lille hvide kanp nederst til venstre. Luk lågen.

3) Tænd bufferen (bag højre paneldør i skabet bagved båndstationerne). Tryk på on.

4) Vent 5-10 sek.

5) Tænd disken ved at trykke på start.

6) Tryk reset-hp ved skrivemaskinen, hvis der ikke kommer udskrift kan det være fordi printeren er slukket - tænd den, kommder der stadig ingen udskrift, kontroller da, at tromlen er tændt (se pkt. 2).

7) Kald et hjælpeprogram (f. eks. list, a free<) hvis der intet sker gøres følgende:

8) Reset adapteren i bufferen: Midterste metallåge på bagsiden af bufferskabet åbnes. Ca. midt i skabet sidder en lille-bitte kontakt (udfor mærkaten på indersiden af lågen) tryk på denne og udfør pkt. 6 og pkt. 7.

9) Sker der stadig intet under pkt. 7 åbnes venstre paneldør på forsiden af bufferskabet. Den øverste metallåge aftages og foroven til højre (udfor mærkaten) sidder en lille bladfjederkontakt. Tryk på denne. Udfør pkt. 6 og 7. Hvis der stadig intet sker er der intet at gøre og man lægger en besked til næste bruger.

10) Iøvrigt kan nogle af de specielle fejl-udskrifter forekomme, disse behandles i overensstemmelse med opslaget på GIER.

11) Luk alle låger.

12) MAN MÅ ALDRIG SLUKKE FOR GIER BUFFER ELLER DISK!

CONVERTEREN

RC 3000 converter er en maskine, der muliggør off-line kopiering. Den kan i det væsentlige udføre 3 operationer, som den gentager i det uendelige i følgende rækkefølge:

1) Læs et tegn fra et af inputmedierne strimmellæser, eller magnetbånd.

2) Omsæt ved hjælp af et katalog det læste tegn til et andet.

3) Skriv det nye tegn på et af outputmedierne plotter, printer, perforator eller magnetbånd.

Kun kombinationen magnetbånd til magnetbånd er ikke tilladt!

Kataloget i punkt 2 er placeret forrest i converterens lager, som består af 1024 celler á 8 bit. Er det indlæste tegn f. eks. et a (talværdi 49), bliver det nye tegn indholdet af celle 49. Kopieringen sker altså ligesom ALGOL-sætningerne:

```
L: writechar (katalog lyn); go to L;
```

Hvis man eksempelvis har fyldt katalogets første celler og svarende til:

```
for i:=0 step 1 until 127 do katalog [i]:=i
```

vil vi kunne benytte converteren til kopiering af strimler.

Foruden den beskrevne tegn-for-tegn kopiering kan man give udvalgte tegn en særlig behandling. Således kan man indsætte SKIP i celle 63 og STOP i celle 11; derved før man fjernet tapefeed, og converteren stopper, når den kommer til stopcode.

□

TILKOBLING AF YDRE ENHEDER

Valget af input- og outputmedier sker via den omkoblerkasse, der står ved siden af GIER's store kontrolbord. På kassen er monteret 8 kontakter

der tager fra venstre svarer til venstre båndstation, højre båndstation, printeren, plotteren, converterens strimmellæser og converterens perforator (den sidste kontakt er i reserve). Hver kontakt har 3 stillinger: i øverste stilling er den pågældende enhed tilkoblet GIER, i mellemste stilling er enheden koblet helt fra, og i nederste stilling er den tilsluttet converteren. Converterens perforator kan dog ikke sættes til GIER. INGEN MÅ RØRE OMKOBLINGSMASSEN UDEN FØRST AT SPØRGE DEN, DER KØRER MED GIER, HVILKE ENHEDER DER ER LEDIGE. ER DEN PÅGÆLDENDE IKKE TIL STEDE, MÅ MAN UNDER INGEN OMSTÆNDIGHEDER RØRE ENHEDER, DER ER SAT PÅ GIER.

Først når man har sikret sig, at de ønskede enheder er ledige, kan de kobles til converteren. Af de øvrige enheder skal strimmellæseren også sættes til converteren, medens de ubenyttede enheder kobles helt fra.

I forbindelse med omkoblingen kan det nævnes, at sættes strimmellæseren til GIER, kan man køre med begge læsere
på GIER. I help 3 har den "nye" læser navnet r2; et ALGOL-program kan f. eks. indlæses ved, at man taster r2, algol<. I ALGOL 4 vælges den "ny" læser ved, at man adderer 4 til tallet i select; select (12) vil algså give input fra den "ny" læser og output på printeren.

□

KATALOGER

På hylden under plotteren ligger på hulstrimmel de kataloger der normalt kan blive brug for. De skal her beskrives ganske kort:

MT-PLOTTER	anvendes ved udskrivning af plotteroutput, der er gemt på magnetbånd.
SB-PT	anvendes ved udskrift af et magnetbånd skrevet af tapeoutput, outmt eller SAMBA systemet.
COPY NORMAL	bruges, hvis man ønsker kopi eller udskrift af en normal (skrevet i flexowritercode) hulstrimmel. Kan ikke kopiere binære strimler eller katalogstrimler. Springer over tabulator og tapefeed.
PT-PT	kopierer enhver form for strimler - selv strimler med paritetsfejl. Bør derfor ikke anvendes til normal kopiering. COPY-PAPERTAPE ONLY skal være tændt.

INDLÆSNING AF KATALOG

Når de ønskede enheder er koblet til og gjort klar, sker indlæsningen af kataloget ved at man

1) trykker på knappen READ KATALOG og
2) sætter katalogstrimlen i læseren og trykker reset.

converteren går herefter straks igang med arbejdet.

ANDRE KATALOGER

En del små opgaver kan løses, hvis man laver lidt om på de eksisterende kataloger. Vil man f. eks. have fjernet alle endcode fra sit program, sker der ved at:

1) kopiere COPY NORMAL ved hjælp af PT-PT,
2) finde endcode på den nye strimmel og håndhulle den om til SKIP.

(SKIP hulles som et tegn med talværdi 104=64+32+8, STOP hulles som et tegn med talværdi 100=64+32+4): Iøvrigt findes der et algol4-program til fremstilling af kataloger

BETJENINGSPANEL

Converteren er på forsiden forsynet med en række hvide kontrollys og røde fejllamper. De hvide lys angiver, hvorfra og hvortil kopieringen sker, idet dog PRINTER kan dække over både printer, plotter og perforator. De røde lys har følgende betydning:

L-ERROR	fejl i katalogstrimmel
ERROR	paritetsfejl under læsning af en strimmel.
STOP	converteren har mødt et af de tegn, der i kataloget er angivet ved stop. Trykkes knappen ned, fortsættes kopieringen. Knappen kan også benyttes af operatøren til at standse converteren: trykkes den ind, vil STOP tændes, og maskinen stopper i løbet af nogle sekunder.
MT OFF	der er ikke sat nogen båndstation til converteren, eller båndstationen står i LOCAL, eller båndet mangler skrivering.
MT ERROR	paritetsfejl ved læsning fra magnetbånd. Converteren bliver ved med at forsøge at læse den fejlramte blok, indtil man trykker MT ERROR ind. Converteren går så videre til næste blok på båndet.
COPY PAPERTAPE ONLY	dette er ligesom STOP-knappen en såkaldt hverandengangskontakt. Den bruges til at undertrykke forskellige former for kontrol i converteren (bl. a. paritetskontrol), og skal være tændt når man benytter kataloget PT-PT. Ellers skal den være sl slukket.

□

Relevante rapporter og manualer

[1] Peter Naur, Revised report on the algorithmic language algol 60, A/S Regnecentralen, København 1964.

[2] Peter Naur, A manual of Gier algol 4, A/S Regnecentralen, København 1967.

[3] Søren Lauesen, A manual of Help3, A/S Regnecentralen, København 1967.

[4] Chr. Andersen og Chr. Gram, A manual of Gier programming, A/S Regnecentralen, København 1963.

[5] RC 3000 Users manual, GSL 390, A/S Regnecentralen.

[6] Katalogfremstilling til RC 3000, GSL 498, A/S Regnecentralen.