Rialto, den 20.5.64

ALGOL-procedure til 1604-A

Navn: TYRK

Koder: E. Petersen

Formål: Proceduren er beregnet til trykning af heltal, tekst samt tegn i opgivne områder af en liniebuffer med 120 skrivepositioner, samt at trykke denne linie på en valgt pseudoenhed.

Funktion: Ved første indhop samt efter hver linieskift eller formularskift fyldes hele bufferen med mellemslag, så brugeren behøver kun at interessere sig for de positioner, hvori tekst eller tal ønskes trykt. Da intet trykkes på outputmediet før signal om vognretur afgives, kan linien opbygges i vilkårlig orden, ligesom det er muligt at skrive oven i et allerede benyttet område.

Ved skrivning på hulkorthulleren, hulles naturligvis kun de første 80 tegn fra liniebufferen.

Magnetbånd skrives med ulige paritet (binært) og med én linie pr. blok (dvs. 15 ord). Ved bufferlængde- eller paritetsfejl på bånd skrives om indtil skrivningen foregår fejlfrit, idet ca. 5' bånd overspringes ved hvert forsøg.

Mødes 'end of tape mark' skrives et fil-slut-mærke på båndet, hvorefter dette tilbagespoles med lås. På skrivemaskinen udskrives:

monter nyt outputbånd

Når nyt bånd er sat på stationen fortsættes skrivningen.

Kald: TYRK (FKT, ADR, FRAPOS, TILPOS, B).

Parametre: FKT er en integer, som angiver, hvilken funktion proceduren skal udføre. Der er følgende muligheder:

FKT	= 0	giver	indhop til	sluttegn
-	= 1	-	-	trykhel
-	= 2	-	-	tryktegn
-	= 3	-	-	tryktekst
-	= 4	-	-	tryklinie

Da betydningen af de øvrige parametre til dels afhænger af FKT, beskrives de særskilt for hver værdi af FKT.

FKT = 0: ADR er her en integer, hvis værdi erstatter BCD-symbolet ▯, som sluttegn i lagrede tekster. Se tryktekst (FKT = 3). De øvrige parametre er irrelevante, og kan udelades.

FKT = 1: ADR er af typen integer og angiver navnet på den variabel, som ønskes trykt.

FRAPOS og TILPOS er begge integers, medens B er en boolean. Hvis relationen

$$1 \leq \text{FRAPOS} \leq \text{TILPOS} \leq 120$$

ikke er opfyldt sørger proceduren for at dette bliver tilfældet således:

```
FRAPOS := if FRAPOS ≤ 0 then 1 else if FRAPOS > 120
          then 120 else FRAPOS.
TILPOS := if TILPOS ≤ 0 then 1 else if TILPOS > 120
          then 120 else if TILPOS < FRAPOS then FRAPOS
          else TILPOS.
```

Hvis B = false trykkes decimalpunkt før næstsidste anslag.
Der trykkes højst

```
if B then TILPOS - FRAPOS + 1 else TILPOS - FRAPOS
```

betydende cifre og TILPOS - FRAPOS + 1 anslag.
Fortegn trykkes umiddelbart før første ciffer og er, hvis tallet er negativt et minustegn ellers mellemslag.
Er der ikke plads til fortegn udelades det.
Tallet 0 trykkes enten som 0 eller 0.00.

FKT = 2: ADR er en integer. Det til ADR.s værdi svarende tegn (modulo 64) trykkes i FRAPOS. De øvrige parametre er irrelevante og kan udelades.
Kaldet kan i dette tilfælde se således ud:

TYRK(2, ADR, FRAPOS)

FKT = 3: ADR kan enten være en integer eller en string. FRAPOS og TILPOS er integers, hvorom gælder samme begrænsninger i de tilladte værdier som beskrevet under trykhel (FKT = 1).
B er en boolean.
Teksten skal være lagret med 8 BCD-tegn pr. celle.
B = false.
Den tekst, som er lagret med begyndelsesadressen ADR trykkes fra FRAPOS og indtil BCD-symbolet ▯ +) (decimal værdi 60) eller TILPOS nås, dvs. højst TILPOS - FRAPOS + 1 anslag.
B = true.
I dette tilfælde er ADR irrelevant, idet sekvensen nu fortsætter trykningen fra det sted i teksten, hvortil den nåede ved forrige kald af TYRK med FKT = 3. Teksten trykkes som i første tilfælde fra FRAPOS frem indtil sluttegn eller TILPOS nås.

FKT = 4 ADR er en integer, som angiver nummeret på den pseudoenhed, hvorpå den i bufferen trykte linie ønskes skrevet.
FRAPOS angiver antallet af vognreturer, der ønskes trykt. Ekstra vognreturer trykkes efter at linien er skrevet
Er FRAPOS = 0 fås formularskift.
Bemærk! Ved formularskift trykkes intet.
I øvrigt gælder her som ved sluttegn og tryktegn at de to sidste parametre kan udelades.

Bemærkninger Da trykningen foregår ved hjælp af WRITE* - der som bekendt benytter position 1 til særlige formål, såfremt outputenheden er printeren - må brugeren benytte første skriveposition med en vis forsigtighed.
Hvor parameteren B forekommer er det tilladt i stedet for 'TRUE' at skrive et vilkårligt negativt tal. 'FALSE' kan tilsvarende erstattes med et positivt tal.

Erik Petersen

+) På printer og hulkort svarer ▯ til en rund højre-parentes

Rialto, den 12.5.64

ALGOL-procedure til 1604-A

Der har længe eksisteret en procedure ved navn STOP blandt programmerere på RC, Rialto. Denne procedure var oprindelig tænkt som en parameterløs procedure, der virkede på den måde, at maskinen stoppede, hvorefter man ved tryk på START fik udhop til næste statement. Da det viste sig, at man ikke kan have parameterløse externals i OAK-RIDGE-ALGOL, blev det almindelig praksis at benytte denne procedure med en dummy-parameter, der almindeligvis sættes til nul. Da denne praksis af mange ikke anses for 'good ALGOL style', har jeg omkodet proceduren STOP til følgende specifikationer:

Navn.....: STOP

Koder ....: H.B. Hansen

Kald .....: STOP (L)

Parameter.: Der er 2 muligheder for parametren L:

L = 0: Maskinen stopper. Ved tryk på START fortsættes med næste statement.

L ≠ 0: Maskinen stopper. Ved tryk på START fortsættes med det statement, der har L som label.

Programmering 1

H.B. Hansen

H.B. Hansen

Rialto, 11.5.64

ALGOL-procedure til 1604-A.

Brugere af BUFPROC vil være bekendt med, at denne procedure kræver en god del flytning af tal mellem de forskellige buffere. Disse flytninger vil som regel blive udført i for-sætninger med indicerede variable. Da det jo i de fleste tilfælde drejer sig om flytning af tal, der i maskinen er lagret konsekutivt (step 1), er det åbenlyst, at der må gå megen tid til spilde ved den gentagne beregning af de indicerede variables adresser i for-sætningen på grund af denne sætnings mere generelle karakter. Den foreliggende procedure er kodet med henblik på nedsættelse af flyttetiden i de tilfælde, hvor man ønsker at flytte en vektor til en anden vektor. Målinger ved hjælp af REALTIME har vist, at forholdet F mellem flyttetiden i en for-sætning og flyttetiden i FLYTORD stort set er givet ved:

$$F = 0{,}14 \times N + 1{,}09$$

hvor N er antallet af ord, der skal flyttes.

Navn.....: FLYTORD
Koder....: H.B. Hansen
Kald.....: FLYTORD (A, B, N)
Parametre:

A: Begyndelseselementet i den talsuite, der skal flyttes fra (normalt en indiceret variabel. Typen irrelevant).

B: Begyndelseselementet i den talsuite, der skal flyttes til (normalt en indiceret variabel. Typen irrelevant).

N: En integer, der angiver det antal tal, der skal flyttes.

OBS! Ved flytninger til eller fra 2-dimensionale arrays skal man huske, at arrays i OAK-RIDGE-ALGOL lagres søjlevis.

Programmering 1
H.B. Hansen
H.B. Hansen

REGNECENTRALEN
Rialtoejen 28.1.64
RIALTO-AFDELINGEN

Programopdeling i 1604-A

På grund af 1604'erens store ferritlager har der hidtil ikke været de samme problemer med denne maskine som med f.eks. GIER, hvor ferritlageret jo er så lille, at spørgsmålet om at dele programmet mellem forskellige former for lagre (f.eks. ferritlageret og tromlen) altid er aktuelt. Imidlertid er der efterhånden fremkommet programmer til 1604 af en størrelsesorden, der kan berettige, at spørgsmålet om programopdeling tages op til overvejelse. Også af andre grunde kan en programopdeling være hensigtsmæssig - f.eks. vil det være en fordel at have så lidt program som muligt i ferritlageret under sorteringsfasen i et administrativt program. Nu er det jo så heldigt, at CO-OP-monitoren har indbyggede faciliteter til administration af programmer lagret på magnetbånd, men da kendskabet til disse faciliteter synes at være ret begrænset, følger hermed en nødtørftig beskrivelse af systemet samt nogle eksempler på brugen.

Filosofien bag systemet er følgende:

Et program tænkes opdelt i et hovedprogram og diverse underprogrammer, der igen kan være opdelt i under-underprogrammer. De 3 trin i dette hieraki betegnes MAIN, OVERLAY og SEGMENT. Til at begynde med kan hele programmet tænkes at være lagret på et magnetbånd, hvor MAIN står forrest efterfulgt af OVERLAYS og SEGMENTS efter følgende system:

MAIN
OVERLAY 1
SEGMENT 1 i OVERLAY 1
SEGMENT 2 i OVERLAY 1
.
.
.
.
OVERLAY 2
SEGMENT 1 i OVERLAY 2
SEGMENT 2 i OVERLAY 2
.
.
.
.
.

ADR.: FALKONER ALLÉ 1 . KØBENHAVN F . TLF.: (01) FA 9911
TELEX: 5468 . TELEGR.-ADRESSE: RIALTOCENTRAL
GIRO: 63976 . BANK: AARHUUS PRIVATBANK 1752

Programmet startes ved at læse MAIN ned i ferritlageret (det gør monitoren). MAIN forbliver i ferritlageret under hele gennemløbet og skal altså kodes som en administration af de forskellige OVERLAYS. Da alle OVERLAYS lagres på samme sted i ferritlageret, kan der kun være eet OVERLAY i ferritlageret ad gangen. Hvis et OVERLAY er opdelt i SEGMENTS, gælder endvidere, at der kun kan være eet SEGMENT nede ad gangen. Et OVERLAY må altså kun referere til ENTRY POINTS eller COMMONS i sig selv eller MAIN, mens et SEGMENT kun må referere til ENTRY POINTS eller COMMONS i sig selv, sit OVERLAY eller MAIN.

Nedhentning fra magnetbåndet til ferritlageret af de forskellige OVERLAYS og SEGMENTS sker ved hop til monitorsekvenserne OVERLAY og SEGMENT. I CODAP-1 kaldes disse sekvenser således:

```
          RTJ     OVERLAY
 +        ZRO     L (P)
          ZRO     L (X)
          ZRO     L (O)
          ZRO     O
```

hhv.:

```
          RTJ     SEGMENT
 +        ZRO     L (P)
          ZRO     L (X)
          ZRO     L (O)
          ZRO     L (S)
```

Her er:

P: Pseudoenhedsnummeret for det bånd, hvorpå det ønskede OVERLAY eller SEGMENT er lagret. Det er muligt at have program på indtil 4 bånd på samme tid.

X: Navnet på en variabel, hvis værdi ønskes overført til det nye program.

O: Nummeret på det ønskede OVERLAY. Indenfor hvert P skal O antage værdierne 1 og opefter.

S: Nummeret på det ønskede SEGMENT. Indenfor hvert O skal S antage værdierne 1 og opefter.

Ved et sådant kald af OVERLAY eller SEGMENT sker følgende:

Den ønskede programstump findes og transporteres til ferritlageret; derefter hoppes til programstumpen med X i A-registret. Ved udhop fra programstumpen kommer man tilbage efter det kald af OVERLAY eller SEGMENT, der bevirkede transport af programstumpen.

Tildelingen af pseudoenhedsnumre m.v. til de enkelte OVERLAYS og SEGMENTS sker under indlæsningen af programmet ved hjælp af særlige styrekort. Af disse findes i hovedsagen 3, der beskrives i det følgende:

MAIN-kort

Dette kort har formen:

```
11
 0
 7  MAIN,P.
 9
```

P angiver pseudoenhedsnummeret på det bånd, hvorpå MAIN ønskes skrevet. Der må kun forekomme eet MAIN-kort pr. program.

OVERLAY-kort

Dette kort har formen:

```
11
 0
 7  OVERLAY,P,O.
 9
```

P er pseudoenhedsnummeret og O er løbenummeret på det følgende OVERLAY.

SEGMENT-kort

Dette kort har formen:

```
11
 0 SEGMENT,P,O,S.
 7
 9
```

P er pseudoenhedsnummeret, O er OVERLAY-nummeret og S er det følgende SEGMENT's løbenummer.

For yderligere at anskueliggøre fremgangsmåden ved dannelse af OVERLAY-bånd følger sluttelig nogle eksempler på kortstakke indeholdende de nys beskrevne styrekort. For nemheds skyld er hulkombinationen 7,9 kaldt v, mens hulkombinationen 11,0,7,9 er kaldt w.

Eksempel 1. Dannelse af OVERLAY-bånd uden udførelse.

Delprogrammerne tænkes alle at foreligge som binære kort.

```
vCOOP,1069,HBH,S/MT10,10,1000.
vLOAD,50.
wMAIN,10.
```
Hovedprogram

```
benævnt TRA-kort
wOVERLAY,10,1.
```
1. OVERLAY

```
benævnt TRA-kort
wSEGMENT,10,1,1.
```
1. SEGMENT i 1. OVERLAY

```
benævnt TRA-kort
wSEGMENT,10,1,2.
```
2. SEGMENT i 1. OVERLAY

```
benævnt TRA-kort
wOVERLAY,10,2.
```
2. OVERLAY

```
benævnt TRA-kort
TRA-kort
```

Eksemplet viser indlæsning via hulkort af et program bestående af et hovedprogram og 2 OVERLAYS, hvoraf det første er opdelt i 2 SEGMENTS. Programmet skrives på pseudoenhed 10, der - som det fremgår af COOP-kortet - skal være LOW DENSITY.

Eksempel 2. Udførelse af båndet fra eksempel 1.

```
vLOADMAIN,1069,HBH,I/MT10,8,4000.
vMAIN,10.
DATA
```

I stedet for det normale COOP-kort bruges her et kort, der har eksakt samme form, men som indeholder ordet LOADMAIN i stedet for COOP.

Eksempel 3. Opdeling af et ALGOL-program.

OVERLAY-systemet er ikke umiddelbart tilgængeligt i ALGOL men vil blive det gennem de to procedurer PROGTRNS og DATATRNS, der vil blive beskrevet og udgivet for sig. Disse 2 procedurer er kodet i CODAP-1, og det følgende er den kortstak, der benyttedes til test af disse procedurer. ALGOL-programmet består af et hovedprogram og 3 delprogrammer.

```
vCOOP,1069,HBH,S/1S/2S/MT1,10,1000.
vBINARY,56.
wMAIN,1.
vCODAP1,L,E.
PROGTRNS
   END
DATATRNS
   END
   FINIS
vALDAP,,,56.
```

```
HOVEDPROGRAM
    EOP
    FINIS
vBINARY,56
wOVERLAY,1,1.
vALDAP,,,56.

1. DELPROGRAM
    EOP
    FINIS
vBINARY,56.
wOVERLAY,1,2.
vALDAP,,,56.

2. DELPROGRAM
    EOP
    FINIS
vBINARY,56.
wOVERLAY,1,3.
vALDAP,,,56.

3. DELPROGRAM
    EOP
    EOP
    FINIS
vEXECUTE.

DATA
```

En oversættelse med PROGTRNS og DATATRNS som binære kort sker ved at indsætte de binære stakke i stedet for de med klamme mærkede kort. Der findes mange andre måder at stakke et sådant job på; det eneste, man hele tiden skal holde sig for øje, er, at hent-og-kør-båndet skal slutte med at have samme udseende som kortstakken i eksempel 1.

Programmering 1

H.B. Hansen

H.B. Hansen

Rialto, 28.4.64

ALGOL-procedure til 1604-A

Navn.....: PROGTRNS

Koder....: H.B. Hansen

Formål...: At gøre COOP-monitor-sekvensen OVERLAY tilgængelig i ALGOL.

Kald.....: PROGTRNS (P,N)

Parametre:

P: En integer, der angiver pseudoenhedsnummeret på det bånd, hvorpå det ønskede delprogram er lagret.

N: En integer, der angiver delprogrammets løbenummer på den angivne pseudoenhed.

PROGTRNS kalder OVERLAY som boskrevet i publikationen "Programopdeling i 1604-A". Opmærksomheden henledes på, at tilbagehoppet efter et kald af PROGTRNS først sker, når delprogrammet er udført.

Programmering 1

H.B. Hansen

H.B. Hansen

Rialto, 28.4.64

ALGOL-procedure til 1604-A

Navn.....: DATATRNS

Koder....: H.B. Hansen

Formål...: At muliggøre datatransport mellem delprogrammer i ALGOL.

Kald.....: DATATRNS (F,A,N)

Parametre:

F: En integer, der kan antage værdierne 1 og 2.

F=1 betyder GEM DATA

F=2 betyder HENT DATA

A: En integer, der angiver begyndelseselementet for de data, der skal gemmes hhv. hentes. A vil normalt være en indiceret variabel.

N: En integer, der angiver antallet af helceller, der skal gemmes hhv. hentes. Proceduren kræver $N \leq 100$.

Ved et kald af DATATRNS med F=1 sker der det, at de angivne data flyttes til et internt array i DATATRNS. Indholdet i dette array kan igen hentes frem ved at kalde DATATRNS med F=2. Proceduren er tænkt anvendt i forbindelse med PROGTRNS til overførsel af data mellem forskellige delprogrammer. Opmærksomheden henledes på, at DATATRNS altid gemmer og henter "forfra" i det interne array; de data, der skal gemmes, må derfor før kaldet af DATATRNS være anbragt i et array i forlængelse af hinanden, og må igen hentes ud i et array.

Programmering 1

H.B. Hansen

H.B. Hansen

REGNECENTRALEN

Rialto, den 31.3.64

Nye systemer på 1604-A.

Indhold:

Fra Palo Alto har vi modtaget reviderede systemer til 1604-A og et systembånd omfattende disse vil blive taget i brug mandag den 6. april kl. 0900. Grunden til en så nøjagtig angivelse af overgangstidspunktet er at systemernes konventioner, især hvad angår styrekort er ændret så meget, at adskillige af de tidligere anvendte styrekort ikke vil kunne anvendes. Dette gælder især ALGOL og CODAP1. I det følgende vil systemerne blive omtalt enkeltvis og tilsidst anføres en oversigt over de af de af Regnecentralen kodede sekvenser og procedurer der er optaget på systembåndet.

ALGOL

Ændringerne har udadtil følgende virkning:

1. Det oversatte program anvender input-outputsekvenserne fra Fortran-63 og adskillige funktioner fra Fortran-63. Virkningerne spores især i FORMAT, der i Fortran-63 er mere omfattende end i Fortran-62, der hidtil har været anvendt. Der henvises til FORTRAN 63/REFERENCE MANUAL.

2. ALGOL-styresystemet er afskaffet. Såvel ALGO som ALDAP kaldes nu via COOP. Brugen af navnet ALGOL vil medføre fejludskrift LOADER ERROR UN.

3. I udskriften af oversatte ALDAP-programmer er der følgende ændringer.
   a. Alle variable, konstanter og arbejdsceller adresseres relativt til en given adresse. Eksempelvis vil en variabel, der tidligere hed V1 nu hedde V + 1.
   b. Symbolet OWNARRAY, der har været entry-point i ALDAP programmer og externt symbol i ALDAP proceduren, er nu sat lig RELOCOM*

4. De ovenfor nævnte ændringer bevirker, at binære stakke, der er resultat af tidligere oversættelser, ikke kan udføres med dette systembånd.

COBOL

Systembåndet indeholder version 3 af amerikansk COBOL. Der er ingen ændringer udadtil i forhold til den hidtil benyttede version 2. Vdr kendte,endnu ikke rettede fejl henvises til skrivelse fra Programmering 1 af 7.2.64. COMPUTE er medtaget og en beskrivelse af metoden er tilgængelig hos Programmering 1.

FORTRAN

Systembåndet indeholder Fortran 63. Der henvises til FORTRAN 63/ REFERENCE MANUAL.

CODAP

CODAP er blevet ændret væsentlig og den ændrede version findes på systembåndet. De hidtil gældende konventioner er,bortset fra styrekortet, en undermængde af de nye konventioner, således at 'gamle' CODAP-programmer stadig vil kunne oversættes, når blot styrekortet udskiftes. De nye konventioner er:

7
9 CODAP1, parametre.

Felt 1: 7-9 hul i kolonne 1 umiddelbart efterfulgt af CODAP1. Kortet er frit felt efter kolonne 2. Parametre kan komme i vilkårlig orden efterfulgt af komma. Ukendte parametre og overflødige tegn ignoreres. Parametrene afsluttes med punktum. Er der ingen parametre, udskrives kun fejl og de faste overskrifter. Enhver parameter kan forkortes til kun at bestå af det første tegn.

Eksempler:

7
9CODAP1, L, E, P.
7
9CODAP1, LIST = 1, E = 10.

Enhver parametre kan efterfølges af = n, hvor n er et pseudoenhedsnummer, der skal benyttes ved den til parameteren knyttede operation.

Parametre:		n ≠ 0
LIST	Udskriv det oversatte program	Udskriv det oversatte program.
PUNCH	Hul relative, binære kort på pseudoenhed nr 52.	Hul binære kort på pseudoenhed n.
EXECUTE	Skriv hent-og-kør bånd på pseudoenhed 56	Skriv 'hent-og-kør' på n.
INPUT	Læs symbolske kort fra 50. Samme hvis parameteren er udeladt	Læs symbolske kort fra n.
SYMBOL	Dimensioner oversætterens symboltabel til 2048 ord. Udelades parameteren, regnes med 1024 ord.	Dimensioner tabellen til max (n, 1024) ord.

REFERENCES	Undertryk oversætterens krydsreferencetabel. Udelades parametren, skrives krydsreferencetabellen	Undertryk tabellen.
NULLS	Undertryk udskriften af ubenyttede symboler; udelades parametren, udskrives ubenyttede symboler	Undertryk udskriften af ubenyttede symboler.

(Er n = 0, fortolkes parametren som om den var udeladt)

Endvidere er CODAP1 udvidet med adskillige nye pseudooperationer. En nærmere beskrivelse af disse findes i Programming System Bulletin nr. 10, der indtil et tilstrækkeligt antal eksemplarer findes, kan studeres hos Programmering 1.

Sekvenser og procedurer på systembåndet.

Følgende af RC kodede sekvenser og procedurer vil findes på systembåndet

MBADM	*	**
CHECKMB	*	**
KOMPPROC	*	
BINTAL	*	
BCDTAL	*	
BCDBIN	*	
BINBCD	*	
TEGNIND	*	
HELIND	*	
LAESHEL	*	
LAESTEGN	*	
EQSPROC		
VENT		
VENTEJAS		
EOFFREM		
EOFBAK		
DATE		
LAESTEMB		
SKRIVMB*		
TALTEGN		
LST5		
LST8		
BESKED		
POLYSORT	**	
BUFPROC	**	
UNLOAD		
PLADS		

De med * mærkede fandtes på det tidligere systembånd. Samtlige procedurer og sekvenser er beskrevet tidligere, idet dog de med ** er ændrede m.h.t konventioner og/eller funktion.

Ændring til MBADM.

Konventionerne for MBADM er uændrede.Som følge af overgangen til Fortran-63 sekvenser for læsning og skrivning er funktionen af MBADM ændret, således at der i tilfælde af fejl foretages omlæsning eller omskrivning.

1. Skrivning.
   Ved paritets- eller bufferlængdefejl foretages indtil 5 omskrivninger, hver gang med overspringelse af ca 5'' bånd.
2. Læsning.
   Ved paritetsfejl foretages indtil 3 omlæsninger. Ved bufferlængdefejl foretages intet yderligere.

I alle tilfælde gælder, at såfremt operationen ikke lykkes indenfor det ovenfor specificerede antal gange, fortsættes kørslen, men ved førstkommende kald af CHECKMB fås fejlreaktion.

Ændring til CHECKMB.

Konventionerne er ændret, således at CHECKMB nu er en integer procedure, der ved udhoppet indeholder længden af sidst læste eller skrevne blok. (se skema)

Parameteren B3 er ændret til type integer, men værdierne er valgt således, at gamle programmer kan køre uden ændringer. I tilfælde af paritetsfejl får B3 værdien -2 , i tilfælde af bufferlængdefejl værdien -1 og ellers værdien 0.

Sammenhængen mellem værdierne af B1, B2, B3 og CHECKMB fremgår af følgende tabel:

B1	B2	B3	CHECKMB	situation
false	irr.	irr.	-1	ikke klar.
true	true	irr	0	EOF/EOT
true	false	-2	bloklængde	paritetsfejl
true	false	-1	bloklængde	bufferlængdefejl
true	false	0	bloklængde	alt ok

Det bemærkes, at såfremt der checkes en ikke aktiveret enhed eller der checkes mere end én gang på samme enhed, får CHECKMB værdien 0, B1 true, B2 false og B3 0.

I tilfælde af bufferlængdefejl med læste eller skrevne blok længere end forlangt, får CHECKMB den forlangte bloklængde som værdi. (B3 = -1).

Ændring til BUFFPROC

Som følge af, at den nye MBADM selv udfører omlæsninger og -skrivninger i tilfælde af båndfejl, er BUFFPROC blevet ændret således, at den kalder FEJLPROC, så snart en båndfejl observeres.

Ændring til POLYSORT

Parametren N i POLYSORT, der angiver det omtrentlige antal ledige celler i ferritlageret, er blevet overflødiggjort af den af J. Hald udgivne procedure PLADS, hvorfor den er strøget af parameterlisten for POLYSORT. På dens plads i parameterlisten er kommet en ny parameter af typen integer med følgende betydning:

Lad os kalde den nye parameter for IK.

Hvis IK = 0 ved kaldet af POLYSORT, er virkningen eksakt som anført i beskrivelsen.

Hvis IK ≠ 0 ved kaldet af POLYSORT, skal værdien af IK angive nummeret på den jumpkey, man ville have benyttet mellem første og anden fase af sorteringen (IK skal altså være 1, 2 eller 3). Der sker da det, at udskriften mellem første og anden fase overspringes, idet maskinen med det samme går videre med den til IK-værdien svarende reaktion. Specielt gælder, at hvis IK = 1 bliver inputbåndet afspolet, hvorefter maskinen venter på jumpkey 1.

OBS! Gamle programmer, der benytter POLYSORT, kan altså køre med den nye udgave ved blot at sætte N = 0.

Programmering 1.

Algol-procedure til 1604-A

Navn: LAEST.

Koder: Finn Larsen.

Formål: At indlæse èn integer eller real af gangen fra hulkort uanset antal og placeringen af de tal, som er hullet i kortet.

Funktion: LAEST er en real procedure, der fungerer omtrent som læst i Gier-algol og har en parameter P, en integer, der optræder både som indgangs- og udgangsparameter.

LAEST læser til første terminator efter betydende cifre, idet cifre mellem et eventuelt decimalpunkt og terminator opfattes som decimaler, og får derefter tilknyttet værdien af det indlæste tal med fortegn (plustegn overflødigt).

Bortset fra 0123456789.+- virker alle tegn og nyt kort som terminatorer, dvs. at en terminator er overflødig efter det sidste tal på kortet. Mellemslag og terminatorer i mellem tallene overspringes, dvs. at det er muligt at skrive kommentarer på datakortene i mellem tallene.

Hvis et tal indeholder mere end ét minus eller ét decimalpunkt får LAEST tilknyttet værdier af de indlæste cifre indtil fejlen og følgende udskrift kommer på printeren:

"forbudt tegn hullet i kolonne xx i kort xxxx..." (alle 80 tegn).

Kald: LAEST(P).

Parameter: P skal være en integer og må, da den optræder som udgangsparameter, naturligvis ikke være en konstant.

Før første kald af LAEST skal P nulstilles.

Ved udhoppet er der følgende muligheder:

P=1: Tallet ok.
P=2: Filslut er læst. LAEST=0.
P=3: Talfejl. Fejludskrift på printer. LAEST= værdien af cifrene før fejlen.

Bemærkninger: Hvis LAEST assignes til en integer afrundes det indlæste tal naturligvis til et heltal.

Finn Larsen

Rialto 31.3.64

ALGOL-procedure til 1604-A.

Navn ...: UNLOAD

Koder...: H.B. Hansen

Formål..: At afspole et magnetbånd.

Kald ...: UNLOAD (P)

Virkningen af et kald er, at båndet på pseudoenhed nr. P bliver afspolet.

H.B. Hansen

Marts. 1964

Algol-procedurer til 1604-A

Ved en del sorteringsopgaver vides det på forhånd, at det antal forskellige værdier, sorteringskriteriet kan antage, er begrænset og væsentlig mindre end antallet af individer i datamaterialet.

Dersom man i sådanne tilfælde kan sammensmelte alle individer med samme kriterium til eet (evt. udvidet) individ, uden at det får uheldige konsekvenser for den senere behandling af de sorterede individer, kan det komme på tale at foretage en kombineret sortering og individ-kummulering i ferritlageret.

Den hertil nødvendige plads kan nedskæres til p × (L + 1), hvor p er antal forskellige kriterier i datamaterialet og L er individlængden for et kummuleret individ.

De i det følgende beskrevne procedurer har til formål at administrere en sådan proces. Den anvendte metode er i korthed følgende:

Der opereres med en stak, som indeholder et individ for hver kriterieværdi, der er forekommet indtil aktuelle tidspunkt.

Et individ består af det af brugeren specificerede individ, hvortil er hægtet et referenceord, som indeholder to referencer. Individerne i stakken er kædet sammen til adressekæder ved hjælp af disse to referencer, idet der gælder følgende konvention for disse: den ene, f-referencen, henviser, hvis den ikke er <empty>, til et individ, som går forud for aktuelle i sorteringsfølgen. Den anden, e-referencen, henviser til et individ, som skal befinde sig efter aktuelle.

Når et nyt individ afleveres til stakken, sker det efter følgende procedure. Det nye individ sammenlignes med første individ i stakken. Viser sammenligningen, at det skal før aktuelle individ sker næste sammenligning med det individ, som udpeges af f-referencen, skal det nye individ efter aktuelle, sker næste sammenligning med det individ, e-referencen udpeger. Har det nye individ samme kriterium som aktuelle, afsluttes søgningen, idet man jo nu har fundet det individ, der kummuleres i for aktuelle kriterieværdi.

Dersom det under søgningen konstateres, at den reference, der skal anvendes er <empty>, findes det søgte kriterium ikke i stakken. Der oprettes derfor et nyt individ i stakken, hvorefter den tomme reference forsynes med det nye stakindivids adresse, således at dette nu er tilsluttet en kæde.

Den ene af de to følgende procedurer udfører en opbygning af stakken efter ovennævnte princip. Den anden procedurer udtager stakkens individer i sorterings rækkefølgen.

NAVN INDSTAK

KODER JENS HALD

FUNKTION AT AFLEVERE ET INDIVID I SORTERINGS STAKKEN.

KALD INDSTAK (BOO, TRAE, PROC)

PARAMETRE

BOO En boolean , som er true ved trimmeindhop og false ved normalindhop.

TRAE Et array, som benyttes som arbejdsplads. Array'et er een dimensionalt og har grænserne 1 og n.

Dersom BOO er true skal ved kaldet A[1] indeholde n, A[2] skal indeholde individlængden (inclusive een referencecelle, som i TRAE placeres bagved hvert egentligt individ).

PROC En af brugeren specificeret integer procedure, som skal være erklæret som følger:

integer procedure proc (index) ; integer index ;

hvor index peger på et individ i TRAE. Procedurens værdi ved udhop skal opfylde flg. regler.

Hvis aktuelle individ skal bagved det individ, index udpeger, skal funktionsværdien være < 0. Skal det foran skal funktionsværdien være > 0. Dersom kriterierne for de to individer er sammenfaldende og de skal sammensmeltes skal funktionsværdien være 0.

INDSTAK er en integer procedure. Der gælder følgende regler for funktionsværdien.

1. Et positivt heltal n.

TRAE[n] er da det første helord i et stakindivid, der har den søgte kriterieværdi.

2. Et negativt heltal -n.

TRAE[n] er det første helord i et stakindivid, som er oprettet, da søgte kriterium ikke fandtes. Brugeren må selv oprette individet, idet INDSTAK kun sætter referencerne, som ikke må berøres af brugeren.

3. Værdi 0.

Dette betyder, at søgte kriterium ikke fandtes, og at stakken er fyldt således, at nye individer ikke kan oprettes.

NAVN	UDSTAK
KODER	JENS HALD
FORMÅL	AT UDTAGE INDIVIDER AF SORTERINGSSTAKKEN I ORDNET RÆKKEFØLGE
FUNKTION	UDPEGER NÆSTE INDIVID I SORTERINGSFØLGEN.
KALD	UDSTAK (BOO)

PARAMETRE

BOO en boolean, som skal være true ved første kald og false i alle følgende kald.

UDSTAK er en integer procedure, hvis værdi er et positivt heltal eller 0. Dersom værdien er 0 er stakken tømt. Er værdien $\neq 0$ angiver den index i TRAE på første helord i det udpegede individ.

Det bemærkes, at den anvendte metode er mest effektiv, når materialet er helt tilfældigt ordnet. Herved bliver adressekæderne af ensartet længde. Et materiale, som er næsten sorteret eller næsten ordnet i modsat følge af den ønskede vil give en skæv opbygning af stakken med lange adressekæder.

Endvidere skal det bemærkes, at en nedbrydning af stakken kun kan ske een gang, idet adressekæderne ødelægges under udtagningen af individerne.

Jens Hald

Programmering 1
programmering 2
Programmering 3
Heise
Aastrup
Pust

Rialto, den 15.1.64
MP/BJ/PO

Vedr.: Strimmel-bånd-omsætter

Med den nye strimmel-bånd-omsætter vil det for de fleste programmer være gældende, at operatøren efter sidste datastrimmel må indlæse en lille strimmel med et sluttegn på, så den senere proces ved, at det var sidste strimmel.

For ikke at få et virvar af små strimler, foreslås det, at man i sin kodning anvender tegnet 67 (vr + 3) som slutsymbol. Dette tegn eksisterer normalt ikke.

M. Pust

CDC 1604-A.

Endelige konventioner for LAESTEMB.

1. Almindelig orientering.

Programmet LAESTEMB indeholder sekvenser til læsning af tegn fra magnetbånd, der er fremstillet af hulstrimmel-magnetbånd-omsætteren ud fra 8-kanal-hulstrimler.
LAESTEMB er opbygget således, at konventionerne kun afviger fra LAESTEGN's konventioner, hvor dette er nødvendigt på grund af magnetbåndsadministrationen. Hvis et program er kodet sådan, at der hoppes til læsning af tegn et centralt sted i programmet, vil det derfor være muligt, ved indsættelse af ret få ordrer at ændre programmet fra at indlæse tegn fra hulstrimmel til at indlæse tegn fra magnetbånd.

Programmet LAESTEMB har to indhop, nemlig til sekvenserne TILBSPOL og LAESTEMB. I TILBSPOL udføres den indledende eller afsluttende båndadministration samt en trimning af sekvenserne i LAESTEMB, således at der ved det første følgende hop til LAESTEMB udføres en læsning af en blok på magnetbåndet til et i sekvensen reserveret område af lageret. I LAESTEMB bringes ved hvert indhop det næste relevante tegn frem til brugeren, og når tegnene i den i lageret værende blok alle er afleveret, sørger sekvensen for at læse næste blok fra magnetbåndet, og aflevere første relevante tegn til brugeren.

I det følgende opdeles samtlige flexowritertegn i følgende mængder:

1. COBOL-tegn: et tegn fra COBOL-tegnsættet, dvs. et tegn valgt ud af den tegnmængde, der er anført i 'Foreløbig beskrivelse af COBOL for 1604-A', afsnit 3.1, side 5.
2. Særtegn: et tegn af mængden: CARRET, TAB, STOPCODE.
3. Benyttede tegn: mængden af COBOL tegn og særtegn.
4. Ubenyttede tegn: mængden af alle øvrige flexowriter-tegn.

Iøvrigt henvises til afsnit 7.

2. Hulstrimmel-magnetbånd-omsætter.

Omæstteren afleverer hvert tegn fra 8-kanal-hulstrimlen (altså også ALLHOLES og TAPEFEED, men ikke NOHOLES)i to tegn (12 bits) på et magnetbånd.med lav tæthed.Benævnes hulrækkerne på strimlen fra højre til venstre H1-H8 (hvor vognreturhulrækken altså er H8), og bittene i de to tegn på båndet fra højre mod venstre B1-B12, overføres H1 til B1, H2 til B2, ... og H8 til B8, mens B9, B10, B11 og B12 nulstilles.

Paniteten på magnetbåndet er ulige.

Informationen opbygges på båndet i blokke å 32 ord å 4 hulstrimmeltegn å 12 bits. Alle blokke har samme længde. Indeholder strimlen ikke tegn svarende til et helt antal blokke, fyldes sidste blok op med NOHOLES. Når en strimmel er indlæst, skrives end-of-file-mark på magnetbåndet.

3. TILBSPOL.

Indhoppet er et returhop med 3 programparametre som følger:

```
+ RTJ TILBSPOL
    FC  A     PSEUDOENHED
    ALTERNATIVT
    ZRO       INDIKATOR
    NORMALT
```

Parametrene har følgende betydning:

FC = 00	betyder ingen tilbagespoling.
= 07	betyder tilbagespoling til LOAD POINT;
= 10	betyder tilbagespoling til LOAD POINT med lås (dvs. at båndet spoles helt af hjælpespolen).
A = 0	betyder åbning; det vil sige, at LAESTEMB trimmes, således at første følgende indhop til LAESTEMB medfører læsning af første blok på båndet på PSEUDOENHED.
= 1	betyder lukning; det vil sige, at den eksisterende trimning af LAESTEMB ophæves.
PSEUDOENHED	er pseudoenhedsnummeret på den ydre enhed, man ønsker som indgangsenhed for TILBSPOL og LAESTEMB.

Udhoppet sker til NORMALT, såfremt ingen tilbagespoling ønskes (FC=00), eller tilbagespolingen er udført fejlfrit. I begge tilfælde er INDIKATOR irrelevant.

Udhoppet sker til ALTERNATIVT, hvis der er båndadministrationsfejl. Når tilbagespoling skal udføres (FC=07 eller 10), foretages før denne et check-only indhop til READ *, således at der i TILBSPOL ventes på, at den aktuelle kanal og pseudoenhed er parat, før tilbagespolingen igangsættes. Derimod afventer TILBSPOL ikke, at denne tilbagespoling fuldføres (men venteordre findes før båndadministration i LAESTEMB). Check-only indhoppet kan ikke give anledning til udhop fra TILBSPOL, og udhop til ALTERNATIVT kan derfor kun forekomme ved fejl under tilbagespoling uden eller med lås. De relevante INDIKATOR-visninger må søges i oversigten i afsnit 5.

4. LAESTEMB.

Indhoppet er et returhop uden programparametre som følger:

```
        RTJ  LAESTEMB
+       ALTERNATIVT 1
        ZRO    INDIKATOR 1
        ALTERNATIVT 2
        ZRO INDIKATOR 2
        NORMALT
```

Udhoppet sker til NORMALT, hvis det læste tegn var et COBOL tegn; i dette tilfælde er indholdet af såvel INDIKATOR 1 som INDIKATOR 2 irrelevant. Tegnet findes oversat til BCD i de sidste 6 positioner i A (pos 5-0). Resten af A indeholder oktale nuller. For bogstaver og mellemrum skelnes ikke imellem UPPER CASE og LOWER CASE. Har dette alligevel betydning, kan informationen hentes i celle MBCASE, der indeholder + 0, hvis tegnet var i UPPER CASE, og -0, hvis tegnet var i LOWER CASE.

Udhoppet sker til ALTERNATIVT 1, hvis det tegn der læstes ikke var et COBOL-tegn; INDIKATOR 2 er irrelevant og INDIKATOR 1 indeholder information om det læste tegn, idet:

INDIKATOR 1 = +0	betyder, at tegnet var et særtegn. Særtegnets decimale ækvivalent findes i A LOWER.
INDIKATOR 1 = -0	betyder, at tegnet havde forkert paritet. 8 bit repræsentationen for det gale tegn findes i A LOWER.
INDIKATOR 1 < -0 (=-1)	betyder, at det læste tegn var et ubenyttet tegn i LOWER CASE. Tegnets decimale ækvivalent står i A LOWER.
INDIKATOR 1 > +0 (=+1)	betyder, at det læste tegn var et ubenyttet tegn i UPPER CASE. Tegnets decimale ækvivalent står i A LOWER.

Udhoppet sker til ALTERNATIVT 2, hvis trimning mangler, og som følge af fejl i magnetbåndsadministrationen. INDIKATOR 1 er irrelevant, og INDIKATOR 2 indeholder oplysning om den indtrufne fejl; se afsnit 5. Specielt bemærkes det, at udhoppet ved filslut sker til ALTERNATIVT 2 med INDIKATOR 2-visning = 4.
Når læsning fra magnetbånd skal udføres, starter LAESTEMB med et check only indhop til READ *, således at der i sekvensen ventes på, at den aktuelle pseudoenhed og kanal er parat, før læsningen påbegyndes. Denne venten er ligeledes indført før udhop til NORMALT eller ALTERNATIVT 1. Check-only indhoppet kan ikke give anledning til udhop fra LAESTEMB, og udhop til ALTERNATIVT 2 kan derfor kun skyldes fejl under den i afsnit 5 nævnte læseprocedure eller manglende trimning.

TAPEFEED, ALL HOLES og NOHOLES overspringes af sekvensen.

Sekvensen holder selv regnskab med i hvilket case, man befinder sig. Ved trimningen i TILBSPOL og før udhop til ALTERNATIVT 2 med INDIKATOR visning = 4 stilles celle MB CASE = -0 (LOWER CASE), idet det forudsættes, at indlæsningen påbegyndes i LOWER CASE. Når et CASE tegn læses, sker der altså kun noget internt i sekvensen, hvorefter næste tegn læses. Et CASE tegn kan derfor ikke være udgangstegn, men information om det aktuelle CASE kan hentes i celle MBCASE, der altså indeholder +0, når man befinder sig i UPPER CASE og -0, hvis man befinder sig i LOWER CASE.

5. Magnetbåndsadministration.

Når magnetbåndsoperationer skal udføres, starter såvel TILBSPOL som LAESTEMB med check-only indhop, således at der i sekvenserne ventes på, at den aktuelle kanal og pseudoenhed er parat. Når magnetbåndsoperationer er udført, foretages i LAESTEMB check-only indhop før udhop til NORMALT eller ALTERNATIVT 1 og i de nedenfor nærmere omtalte tilfælde ved udhop til ALTERNATIVT 2 og INDIKATOR 2-visning lig 4 eller 2o48, mens der ikke foretages check only indhop før udhop fra TILBSPOL.

Hvis læsning fra magnetbåndet (i LAESTEMB) mislykkes, spoler sekvensen båndet en blok tilbage og gentager læsningen. Læsningen opgives først efter tre mislykkede forsøg. Såfremt der sker fejl under arbejdet med at retablere læsesituationen, sker udhoppet til ALTERNATIVT 2 med INDIKATOR 2-visningen 1o24. Hvis kun selve læseprocessen er mislykket tre gange, udføres ALTERNATIVT 2 udhop med INDIKATOR 2-visning 2o48. Den aktuelle blok er da gået tabt, men kan den undværes, kan kørslen altså fortsættes.

Fra TILBSPOL kan udhop til ALTERNATIVT kun finde sted ved fejl i den ønskede tilbagespoling med eller uden lås. Indikatorvisningerne er de samme som i READ *, se følgende oversigt.

Fra LAESTEMB kan udhop til ALTERNATIVT 2 kun finde sted såfremt trimning mangler, d.v.s. INDIKATOR 2 visning = o, såfremt end-of-file mark er mødt d.v.s. INDIKATOR visning = 4, såfremt der sker fejl under forsøg på at retablere læsesituationen, d.v.s. INDIKATOR 2 visning = 1o24, eller såfremt der sker fejl under selve læseprocessen, d.v.s. INDIKATOR 2 visning = 2o48.

Under indkørsel kan det være aktuelt at undersøge indikatorerne i indhoppene fra LAESTEMB til READ *. Nedenfor følger derfor den fuldstændige oversigt over indikatorvisninger. Med stjerne er angivet de for brugeren af TILBSPOL og LAESTEMB aktuelle tilfælde.

TILBSPOL UDHOP	LAESTEMB UDHOP	INDIKATOR ELLER INDIKATOR 2	BETYDNING
NORMALT		IRRELEVANT	Alt vel.
	NORMALT	IRRELEVANT	Alt vel.
	ALTERNATIVT 1	IRRELEVANT	Alt vel.
ALTERNATIVT	ALTERNATIVT 2		
*	*	0	Trimning ikke udført, fejl i programparametrene.
		1	Paritetsfejl på båndet.
		2	Bufferlængdefejl ved skrivning.
		2	Bufferlængdefejl ved læsning.
	*	4	End-of-file passeret
	*	4	End-of-tape passeret.
*		8	Tom operation ønsket.
*		16	Unormal operation ønsket.
		32	Forbudt tegn ved korthulning.
		32	Forbudt tegn ved kortlæsning.
		64	Kanal i brug.
*		128	Den ydre enhed er oversprunget i følge styrekortet.
	*	1o24	Fejl under forsøg på at retablere læsesituationen.
	*	2o48	Fejl i tre forsøg på læsning (1 blok tabt.)

Kun en del af disse INDIKATOR-visninger vil kunne forekomme under normale forhold. Med hensyn til hvad der forstås ved tom operation og unormal operation, henvises til skemaet side 37 i 'CO-OP MONITOR / Programmers Guide'. Mere end een fejl kan indtræde samtidigt; INDIKATOR eller INDIKATOR 2 vil da indeholde summen af pointene for de indtrufne fejl.

6. Eksempel.

I programmet skal indlæses tegn fra filer på to magnetbånd på båndstationerne med pseudoenhedsnumrene lo og 12 i følgende rækkefølge:

1. fil, A, fra pseudoenhed lo.
2. fil, B, - - lo.
1. fil, C, - - 12.
3. fil, D, - - lo.
2. fil, E, - - 12.

Derefter skal båndet på pseudoenhed lo spoles af, og båndet på pseudoenhed 12 spoles til LOAD POINT.

Indlæsningen af tegn foretages centralt:

```
BAAND       SLJ        * *
+           ZRO        o          INDHOP TIL TILBSPOL
            ZRO        o
            RTJ        MBUNORM    MAGNETBAANDSFEJL
            ZRO        * *        INDIKATOR
            SLJ        BAAND
AABBAAND    RTJ        TILBSPOL   PARAMETERCELLER
            oo           o
AABNSPOL    RTJ        TILBSPOL
            o7           o
LUKSPOL     RTJ        TILBSPOL
            o7    1      12
LUKLAAS     RTJ        TILBSPOL
            lo    1      lo
```

```
IND/UD          SLJ          * *          TILBAGE TIL MONITOR
                LIU  6       IND/UD
IGEN            RTJ          LAESTEMB     HOP TIL LAESTEMB
+               SLJ          FORBI        IKKE COBOL TEGN
                ZRO          * *
                SLJ          MBFEJL       MAGNETBAANDSFEJL
                ZRO          * *
                STA          TEGN         GEM TEGN
                SLJ          IND/UD       HOP UD
FORBI           LIL  1       IGEN +1      INDIKATOR TIL IR1
                ENQ  1         0          INDIKATOR TIL Q
                QJP  N       IGEN         HOP HVIS UBENYTTET TEGN
                QJP  P       IGEN         HOP HVIS SÆRTEGN
                SLJ          IGEN +3      HOP, PARITETSFEJL
MB FEJL         LIL  1       IGEN +2      INDIKATOR TIL IR 1
                ENA  1         0          INDIKATOR TIL A
                INA            -4
                AJP  Z       FILSLUT      HOP, INDIKATOR = 4
                INA          -2044
                AJP  Z       IGEN         HOP, INDIKATOR = 2048
                SLJ          TAPEFEJL     HOP FOR FEJL
FILSLUT         ENA  6       * *          OVER TIL NY FIL
                SAU          FILSLUT +1
                SLJ          * *          HOP TIL NY FIL
TAPEFEJL                                  FORHOLDSREGLER VED
                                          MAGNETBAANDSFEJL
TEGN                                      LAGERCELLE
```

Programmet ser herefter således ud:

```
FILA    LDA     AABNSPOL    AABNING M. TILBAGESPOLING
        STA     BAAND +1
        ENA     lo
        SAL     BAAND +1    PSEUDOENHED lo
        ENA     7
        SAU     FILSLUT     TILBAGEHOPADR.
        RTJ     BAAND
+       RTJ     IND/UD
+       ¦.....              BEHANDLING AF COBOL TEGN
        .....               SAMT PARITETSFEJL
FILB    ENA     11
        SAU     FILSLUT     TILBAGEHOPADR.
        RTJ     IND/UD      NY FIL, SAMME PSEUDOENHED
+       .....
        .....
FILC    ENA     12
        SAL     BAAND +1    NY PSEUDOENHED (12)
        ENA     7
```

```
                SAU     FILSLUT     TILBAGEHOPADR.
                RTJ     BAAND       AABNING M.TILBAGESPOLING
+               RTJ     IND/UD
                :::::::::::::::::
FILD            ENA     11          TILBAGEHOP ADR.
                SAU     FILSLUT
                LDA     AABBAAND    SKIFT PSEUDOENHED
                STA     BAAND +1    UDEN TILBAGESPOLING
                ENA     10
                SAL     BAAND +1
                RTJ     BAAND
+               RTJ     IND/UD
                :::::::::::::::::

FILE            ENA     5           TILBAGEHOP ADR.
                SAU     FILSLUT
                ENA     12          SKIFT PSEUDOENHED
                SAL     BAAND +1    UDEN TILBAGESPOLING
                RTJ     BAAND
+               RTJ     IND/UD
                :::::::::::::::::

                LDA     LUKSPOL     LUK PSEUDOENHED 12
                STA     BAAND +1
                RTJ     BAAND
+               LDA     LUKLAAS     LAAS PSEUDOENHED 10
                STA     BAAND +1
                RTJ     BAAND

FAERDIG                             VIDERE
```

7. Behandling af tegn.

De pr. tegn modtagne bits er følgende:

BAAND: B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1

STRIMMEL: H8 H7 H6 H5 H4 H3 H2 H1

Værdierne TEGN og T defineres som følger, idet bit = hul = 1:

TEGN: $H8 \times 2 \uparrow 7 + H7 \times 2 \uparrow 6 + H6 \times 2 \uparrow 5 + H5 \times 2 \uparrow 4 + H4 \times 2 \uparrow 3 + H3 \times 2 \uparrow 2 + H2 \times 2 \uparrow 1 + H1 \times 2 \uparrow 0$

T = $H8 \times 2 \uparrow 6 + H7 \times 2 \uparrow 5 + H6 \times 2 \uparrow 4 + H4 \times 2 \uparrow 3 + H3 \times 2 \uparrow 2 + H2 \times 2 \uparrow 1 + H1 \times 2 \uparrow 0$

Tegnene behandles som følger:

1. B12: =: B11: =: B10: = B9: = 0.
2. TEGN = 0 (NO HOLES) : nyt tegn.
3. - = 255 (ALL HOLES) : nyt tegn.
4. - = 127 (TAPEFEED) : nyt tegn.
5. - = 124 (UPPER CASE): MB CASE := 0; nyt tegn.
6. - = 122 (LOWER CASE): MB CASE := -0; nyt tegn
7. HULANTAL LIGE: PARITETSFEJL
8. B5 FJERNES : B8 B7 B6 B4 B3 B2 B1.
   $(0 \leq T \leq 127)$.
9. $T \geqq 65$: UBENYTTET TEGN.
10. ALM. UNDERSØGELSE: $0 \leq T \leq 64$.

For $0 \leq T \leq 64$ behandles tegnene som anført i nedenstående skema:

TEGN	T	LC	LC	LC + UC	UC	UC
16	0		COBOL	SPACE	COBOL	
1	1	1	COBOL		UBEN	
2	2	2	COBOL		UBEN	
19	3	3	COBOL		COBOL	/
4	4	4	COBOL		COBOL	=
21	5	5	COBOL		UBEN	
22	6	6	COBOL		UBEN	
7	7	7	COBOL		UBEN	
8	8	8	COBOL		COBOL (	(
25	9	9	COBOL		COBOL	)
26	10		UBEN		UBEN	
11	11		SAER	STOPCODE	SAER	
28	12		COBOL	'	COBOL	
13	13		COBOL	Å	COBOL	
14	14		UBEN		UBEN	
31	15		UBEN		UBEN	
32	16	0	COBOL		UBEN	
49	17		UBEN		UBEN	
50	18		COBOL	S	COBOL	
35	19		-	T	-	
52	20		-	U	-	
37	21		-	V	-	
38	22		-	W	-	
55	23		-	X	-	
56	24		-	Y	-	

TEGN	T	LC	LC	LC + UC	UC	UC
41	25		COBOL	Z	COBOL	
42	26		UBEN		UBEN	
59	27	,	COBOL		UBEN	
44	28		UBEN		UBEN	
61	29		UBEN		UBEN	
62	30		SAER	TAB	SAER	
47	31		UBEN		UBEN	
64	32	-	COBOL		COBOL	+
81	33		COBOL	J	COBOL	
82	34		-	K	-	
67	35		-	L	-	
84	36		-	M	-	
69	37		-	N	-	
70	38		-	O	-	
87	39		-	P	-	
88	40		-	Q	-	
73	41		COBOL	R	COBOL	
74	42		UBEN		UBEN	
91	43		COBOL	Ø	COBOL	
76	44		COBOL	*	COBOL	
93	45		UBEN		UBEN	
94	46		-		-	
79	47		UBEN		UBEN	
112	48		COBOL	Æ	COBOL	
97	49		-	A	-	
98	50		-	B	-	
115	51		-	C	-	

TEGN	T	LC	LC	LC + UC	UC	UC
100	52		COBOL	D	COBOL	
117	53		-	E	-	
118	54		-	F	-	
103	55		-	G	-	
104	56		-	H	-	
121	57		COBOL	I	COBOL	
122	58		'UBEN'	[LCASE]	'UBEN'	
107	59		COBOL		UBEN	
124	60		'UBEN'	[UCASE]	'UBEN'	
109	61		UBEN		UBEN	
110	62		'UBEN'		'UBEN'	
127	63		'UBEN'	[TAPEFEED]	'UBEN'	
128	64		SAER	CR	SAER	

Januar 1964 J.J. Damgaard

REGNECENTRALEN

RIALTO-AFDELINGEN

3.1.64

Vdr. TAPEMCC.

Dette er den officielle forside til beskrivelsen af TAPEMCC. Folk, der ikke gider se ordet vdr. mere kan afrive denne side og arkivere den i papirkurven.

ADR.: FALKONER ALLÉ 1 . KØBENHAVN F . TLF.: (01) FA 9911
TELEX: 5468 . TELEGR.-ADRESSE: RIALTOCENTRAL
GIRO: 63976 . BANK: AARHUUS PRIVATBANK 1752

A. Identifikation.

Titel: TAPEMCC.

CO-OP identifikation: CODA TAPEMCC.

Kategori: Hjælpeprogram.

Dato: 22. maj 1963.

B. Formål:

At give mulighed for kopiering, redigering, udskrift og fletning af bånd.

C. Benyttelse.

1. Betjening.

Hovedsekvensen er skrevet i FORTRAN-63. Der er også en CODAP-1 sekvens, der indlæser data til benævnt COMMON og tager sig af blokke af variabel længde i in- og output.

Behandlingen af båndene styres af styrekort, der specificerer følgende mulige processer.

REW Tilbagespol båndet angivet i indgangsfeltet.

EOF Skriv fil-slut på båndet angivet i udgangsfeltet.

SKIPF Skip antal filer angivet i fil-feltet på båndet angivet i indgangsfeltet.

SKIPR Skip antal individer angivet i individ-feltet på båndet angivet i indgangsfeltet.

COPYF Kopier antal filer angivet i fil-feltet fra båndet angivet i indgangsfeltet til båndet angivet i udgangsfeltet.

COPYR Kopier antal individer angivet i individ-feltet fra båndet angivet i indgangsfeltet til båndet angivet i udgangsfeltet.

EDIT — Læs næste individ fra båndet angivet i indgangsfeltet og erstat ord i individet med oktale ord læst fra dette kort, idet der begyndes med det (oktale) ord, der er angivet i adressefeltet og fortsættes konsekutivt med det antal ord, der er angivet i ordtælleren. Hvis næste styrekorts procesfelt ikke er lutter blanke, skriv individet på båndet angivet i udgangsfeltet. Er næste korts procesfelt lutter blanke, gem udgangsfeltet.

"blanke" — Proces adressefelt, ordtæller og oktale ord som en EDIT-proces.

DMPOF
DMPOR — Læs det specificerede antal filer (eller individer) fra det specificerede indgangsbånd med den angivne repræsentation, og skriv individet på oktal form på standardudgangsmediet. Vdr. formen af indskriften, se C9.

DMPCF
DMPCR — Læs det specificerede antal filer (individer) fra det specificerede indgangsbånd med den angivne repræsentation, og skriv individet på BCD-form på standard-udgangsmediet. Vdr. formen af udskriften, se C9.

LISTS — Læs det specificerede antal filer fra det specificerede indgangsbånd med den angivne repræsentation, skriv individet på BCD-form med enkelt linieafstand og tæl antal paritetsfejl. Der skrives kun de første 119 tegn i et individ.

END — Dette angiver slut på programmet og afslutter processerne.

3. Nødvendig plads: 1656_8 ord.

4. Arbejdsområder: 20000_{10} ord.

7. Fejlstop.
Fejlene i det følgende vil bevirke afslutning af kørslen med den angivne fejludskrift på standard-udgangsmediet:

Fejludskrift.	Betydning.
MODE NOT 1,2 OR 3	Den angivne repræsentation var ikke 0,1 eller 2 (Der gælder: repræsentation = RM i Programmers Guide -1). Bemærk: Alle kort skal indeholde repræsentation, også selv om den ikke benyttes.
ILLEGAL OPERATION	En operation, der ikke var angivet i C1, var specificeret.
ILLEGAL FILE COUNT	Fil-feltet indeholder et negativt tal eller nul.
ILLEGAL RECORD COUNT	Individ-feltet indeholder et negativt tal eller nul.
UNRECOVERABLE INPUT ERROR	Der er stadig paritetsfejl efter tre omlæsninger.
EOF DURING xxxxx OPERATION	Fil-slut læst under individ-proces. xxxxx angiver processen.
EDIT CONTINUE OUT OF ORDER	Et styrekort med blankt procesfelt efter et kort, der ikke var EDIT.
EDIT WORD COUNT ERROR	Ordtæller større end 4 forekommet ved EDIT.

Programmets eneste stop betyder, at bånd-slut er nået på udgangsbåndet, der kan erkendes som det bånd, der tilbagespoles med aflåsning. En filslut vil være skrevet på båndet før tilbagespolingen. Efter påsætning af nyt bånd og betjening af START, fortsætter processen.

8. Ind- og Udgangsbånd.

Disse styres af specificationerne på MCS-styrekortet og programmets styrekort. Programmet er helt båndorienteret, men kortlæser, korthuller og linieskriver kan substitueres ved ekvivaleringer på MCS-kortet. Pseudo-enhed nr. 19 skal altid ekvivaleres med 54 (kommentarer til operatøren).

9. Form af input og output.

Længden af individer er kun begrænset af, at den skal være mindre end 20000_{10}, og at BCD-bånd skal være et helt multiplum af 8 tegn ved kopiering.

Form af styrekort.

Kolonne.	Indhold.
1-5	Procesfelt (se C1)
6-7	Indgangsfelt } pseudo-enhedsnr.
8-9	Udgangsfelt } pseudo-enhedsnr.
10	Repræsentation.
11-15	Fil-felt, individ-felt eller adresse i EDIT.
16	Ordtæller (antal ord på dette kort)
17-32	Første oktale ord.
33-48	Andet oktale ord.
49-64	Tredie oktale ord.
65-80	Fjerde oktale ord.

Form af oktalt output.

1. Linie i hvert individ:

RECORD NUMBER xxxxx, FILE NUMBER xxxxx {PARITY / NO} ERROR

Hver datalinie:

xxxxx xxxxxxxxxxxxxx (gentaget indtil 3 gange)

Sidste linie:

xxxxx WORDS IN RECORD

hvor x'erne er decimale cifre undtagen i datalinie, hvor alle er oktale.

Form af BCD-output.

1. linie samme som for oktal.
2. linie i hvert individ:

   1.....* 10.....* 20 etc. *.....80

   Hver datalinie indeholder indtil 80 tegn.

   Næstsidste linie er som 2. linie.

   Sidste linie er som for oktal.

For begge former gælder, at fil-slut bevirker udskrift af END OF FILE xxxxx

E. Bemærkninger.

COPYF Hvis mere end 1 fil skal kopieres, vil alle mellemliggende fil- slut-mærker også kopieres. Ønskes fil-slut efter sidste fil også, må EOF-processen benyttes.

EDIT Adressen må svare til adressen, der fås fra den oktale udskrift (dvs. første ord betragtes som adresse 0).

Oversat og udgivet ved

Bent Bagger.

Programmering 1 (FD).