

Aflever individ på magnetbånd.

1. Baggrund og formål.
2. Konventioner.
3. Kode.

1. Baggrund og formål.

Ved en stor del opgaver, især indenfor kontorautomatisering arbejder man med meget store datamængder, som man må lagre på magnetbånd for at kunne processe samlet efter indlæsningen.

Kodning af dette er et rent rutinearbejde og virker derfor ualmindelig kedsommeligt samtidig med at det (i kraft af håndkodning) kan indeholde kodefejl (hvem har ikke haft checksumfejl uden teknisk forklaring?).

Denne sekvens imødekommer begge disse kedelige ting, idet den afleverer individer på magnetbånd samtidig med at den tæller disse og danner helordsvis checksum af dem.

Sekvensen har endnu en fordel, idet den efter afbrydelse af indlæsningen ved fortsættelse selv finder frem til på hvilken blok og hvor denne lagringen skal fortsætte og fortsætter korrekt.

2. Konventioner.

- 2.1. Løs snak.
- 2.2. Udførlig beskrivelse.

2.1. Løs snak.

Sekvensen forudsætter følgende:

- a. Båndstationen startet.
- b. BA - 2 lagret fra hac 0.
- c. En indholdsfortegnelse over magnetbåndet, lagret et nærmere angivet sted i ferritlageret.

Sekvensen har 3 indhop nemlig:

1. Trimming.
2. Aflever individ.
3. Aflever sidste blok.

Kodelængde 0 - 82

hac 34 - 82 anvendes kun under trimningen og er til fri disposition efter denne.

## 2.2. Udførlig beskrivelse.

### 2.2.1. Indholdsfortegnelse.

#### 2.2.2. Indhop.

### 2.2.1. Indholdsfortegnelse.

hac 0 c bb F

" 1 p A 35

hec 2 c an A

" 4 c checksum A

hac b B A 1c

" 7 ba A 00

" 8 2 1 A 00

bb = begyndelsesblok for lagring af data.

p = startadresse indenfor blokken på næste individ.

an = antal individer lagret på magnetbåndet.

checksum = helordsvis checksum af lagrede individer.

B = Båndstation nr.

ba = ferritlagerplads for blok (128 hac.)

l = antal hec pr. individ.

bb, p, an og checksum ændres af sekvensen.

Ved start af lagring af data skal p, an og checksum sættes lig 0.

Efter afbrydelse i indlæsning skal disse ikke nulstilles.

### 2.2.2. Indhop.

#### 2.2.2.1. Trimming.

2.2.2. 2. Aflever individ.

2.2.2. 3 Aflever sidste blok.

2.2.2. 1. Trimming.

navn: trimaf

funktion: sætter magnetbåndet i udgangsposition og sætter adresser i afleverer individ.

indgang: AR irrelevant.

indhop: 16 trimaf (34 A 8 16)

parametre: 75 indhf (indhf A 35)

udgang: AR ikke retableret.

Fejludskrifter kun gennem BA - 2.

Bemærkning.

Ved indlæsning af en datamængde over to eller flere kørsler skal bb, p, an og checksum (se indholdsfortegnelse) kun sættes første gang. De efterfølgende gange retablerer trimming situationen som den så ud umiddelbart før sidste hop til aflever sidste blok. Det forudsættes at indholdsfortegnelsen har været gemt siden da.

2.2.2. 2. Aflever individ.

navn: af

funktion: afleverer individ til mellemlager og, hvis dette fyldt, mellemlager til magnetbånd.

an: = an + 1 og akkumuler individ i checksum.

indgang: AR irrelevant.

indhop: 16 af (0 A 8 16)

parametre: 75 individ, i (individ I 75)

udgang: AR ikke retableret

Fejludskrifter kun gennem BA - 2.

Bemærkning: i (I) kan være A, B, C eller D.

2.2.2. 3. Aflever sidste blok.

navn: af sidste.

funktion: afleverer sidste individer i mellemlager til magnetbånd.

indgang: AR irrelevant.

indhop: 16 af sidste (28 A 8 16)

parametre: ingen

udgang: AR ikke retableret

Fejludskrifter kun gennem BA - 2.

3. Kode.

.....  
begin; Aflevering af individ paa magnetbaand

in af: 74 d  
34 d-1  
37 1,d  
74 lø+1  
60 arbc  
26 lø+1  
37 [indhf+1]  
75 [-2x1]  
lø: 75 2,d  
40 [individ+2x1-2],d  
08 [ba],b  
06 [indhf+4]  
35 2,b  
73 lø  
40 2041  
06 [indhf+2]  
75 [-entier(64/1)x2x1],b  
73 d-2  
60 2041  
a: 26 [indhf]  
35  
00 [B]  
16 7  
b: 1f [ba]  
34 [indhf+1]  
35  
d: 75  
.. 10 2,d  
in afsidste: 34 d-1  
75 -1,d  
74 d  
37 [indhf+1]  
10 b-2

arbc:	00	
<u>in</u> trimaf:	74	d
	37	1,d
	74	a
	75	1,d
	74	lø-2
	74	arbc-2
	74	d-2
	75	1,d
	74	a-4
	75	2,d
	74	lø+3
	60	2,d
	28	a+2
	28	h1
	28	h2
	28	h3
	60	3,d
	29	lø+2
	29	b
	29	h2+2
	61	4,d
	20	2040
	29	lø-1
	60	4,d
	21	to
	29	arbc
	60	a128
	21	4,d
	11	k-1
	20	4,d
	21	a128
	20	2040
	29	a-3
	60	-4,d
	28	h1+2
	21	2041
	28	h3+2
h1:	00	
	16	1
h2:	00	
	16	5
	1d	[ba]
h3:	00	
	16	1
	00	
	10	d
to:	00	2
a128:	00	128
...		
<u>end</u>		