

LÄS & SKRIV

Kompletterande dokumentation av BASREGISTER
samt beskrivning av hur egna program kan införas.

Innehåll:

1. Inledning
2. Dataformat mm
3. Layoutformat på skärmen
4. Globala variabler
5. Gosubadresser i BASREGISTER
6. Rutinerna LÄS och SKRIV
7. Exempel på användning av LÄS
8. Exempel på användning av SKRIV
9. Fler än ett register
10. Assembler-calls

1. Inledning

Med denna avancerade version av BASREGISTER kan man göra specialtillämpningsprogram för eget bruk. Tillstånd för kopiering och försäljning kan avtalas med undertecknad.

Följande möjligheter erbjuds:

- & Du kan läsa eller skriva valfri post på skivan.
- & Du kan bearbeta en eller flera poster i taget. Exempelvis beräkningar eller jämförelser av olika slag.
- & Du får tillgång till en dokumentation över programmets uppbyggnad. Dvs GOSUB-adresser, variabler mm. Samt några programmerade exempel på egna tillämpningar.

Några vanliga behov är följande:

- ! Egen inmatningsrutin, som passar just Din tillämpning.
- ! Eget utskriftsprogram, som summerar kolumnerna.
- ! Uppdatering av viss grupp av poster. Exempel Prislista: Höj priset med 5% på alla varor som kostar under 50 kronor.
- ! Uppläggning av fler än ett register på dataskivan.

2. Dataformat mm

Dataskivan innehåller inga filer. Istället delas sektorerna in i ett antal block, numrerade från 1. Blockets längd=L%. Startsektor för blockräkning=S%. (Vid avancerad tillämpning kan man alltså byta dataarea genom att ändra S%). Varje post disponerar ett block. Det finns en katalog (adressvektor) A%(), vars värde är ett blocknummer. Vid sortering mm flyttas aldrig posterna utan istället uppdateras katalogen. Om A%(18)=99 betyder detta att den post som har ordningsnummer 18 ligger på block 99. Posten ligger i A\$(4). A\$(4) innehåller rad 4 i posten. Internt samsas raderna om plats i posten genom att de ligger i en enda lång följd.

3. Layoutformat på skärmen

I	I		I I		I I		
I	I		I I		I I		
I	I		I I		I I		
I	Nr	I	Rubrik	I I	Post	I I	H%
I	I		I I		I I		
I	I		I I		I I		
I	I		I I		I I		
3	G%		1		F%		2

4. Globala variabler

L%	Postlängd	Z%	Callvariabel, status
N%	Aktuellt postantal	R%	Radnr
M%	Maximalt postantal	P%	Parameter vid input
H%	Radantal	D\$()	Radrubriker
G%	Radrubrikvidd	N\$	Basens namn
F%	Postens radvidd	Y%	Akt. postnr
B%	Postnr = blocknr	C\$	Chain-namn
S%	Startsektor för data	V\$(0)	Bitvektor
E%	Enhet vid utskrift	P	Pekare i dosbuffer 0
A\$()	Rader	B\$()	Rader föreg. post
A%()	Adressvektor för poster	G\$	Gamla sökordet
I\$	Inputsträng	P1%	Gamla sökraden
J\$	Editeringssträng	P\$	Printerkod

5. Gosubadresser i BASREGISTER

- 121 Läs kommando. Anropar 148. Om första tecknet är en bokstav elimineras resten. Liten bokstav göres stor. Eventuell efterföljande siffra läggs i P%.
- 148 Inputrutin. Resultatet i I\$ vid RETURN. Före anrop kan J\$ laddas med text som skall editeras.
- 226 Packa upp. Buf0-pekare ges i P. Efter anrop har den post, som börjar i adress P packats upp i A\$(1).....A\$(H%).
- 244 Packa ned. Motsatsen till 226. Om texten är för lång skäres fr.o.m. hela den felaktiga raden.
- 730 Sök plats. Nästa lediga block på skivan erhålles i B%.
- 739 Märk ut plats. Boka plats B% i bitvektorn V\$(0%).
- 745 Friställ plats. Motsats till 739.
- 751 Skriv. A\$() nedpackas och skrivs på block B% på skivan.
- 763 Läs. Hämtar block B% och sätter startposition P i buf0.
- 772 Läs bas. Läser de konstanter som bestämmer databasens konfiguration. Läser även namn, radrubriker och aktuellt antal poster.
- 823 Läs katalog. Läser in katalogen över vilka block, som respektive post disponerar.
- 853 Skriv katalog. Skriver katalogen samt uppgift om aktuellt postantal.
- 883 Mall. Skriver ut radrubrikerna på skärmen.
- 895 Nollställ virtuellt minne. Tvingar floppyn att göra skivaccess nästa gång en post läses.

6. Rutinerna LÄS och SKRIV

Dessa rutiner används för att göra egna program som skall kopplas till databasen.

"SKRIV" innehåller läs- och skrivkommandon så att data kan hämtas, uppdateras och läggas tillbaks igen. Ytterligare poster kan också fyllas på.

"LÄS" saknar möjlighet till återläggning, men är i gengäld ett kortare program.

7. Exempel på användning av LÄS:

1. LOAD LÄS
2. Skriv följande program, som skriver ut en lista över vara och pris inkl. moms samt summan. Varunamn antas ligga på rad 1 och pris på rad 2:

```
1000 REM
1100 OPEN P$ ASFILE 3% : S$="0"
1200 ;#3"Namn" TAB(40) "Pris" : ;#3
1300 FOR I%=1% TO N%
1400 B%=A%(I%) : REM SLÅ I KATALOG
1500 GOSUB 763 : REM LÄS POST
1600 GOSUB 226 : REM PACKA UPP POST
1700 X$=MUL$(A$(2),"1.2063",2)
1800 ;#3A$(1) TAB(40) X$
1900 S$=ADD$(S$,X$,2)
2000 NEXT I%
2100 ;#3 : ;#3TAB(40)S$
2200 RETURN
```

3. Spara hela programmet på programskivan t.ex. "SAVE SUMMA".
4. Vid körning av databasen kan man från huvudmenyn skriva "Ö" följt av "E" och därefter skriva programnamnet (i detta fall "SUMMA"). Härvid köres det egna programmet. Sedan återkommer man automatiskt till huvudmenyn.

8. Exempel på användning av SKRIV:

1. LOAD SKRIV
2. I ett fastighetsregister skall man höja hyran (rad 9) med X kronor per kvm. Ytan står på rad 8: Skriv följande rader:

```
1000 REM
1100 ; "ANTAL KRONOR"; : INPUT X$
1200 FOR J%=1% TO N%
1300 B%=A%(J%) : GOSUB 763 : GOSUB 226
1400 Y$=MUL$(X$,A$(8),0)
1500 A$(9)=ADD$(A$(9),Y$,0)
1600 GOSUB 751 : REM SKRIV PÅ DATASKIVAN
1700 NEXT J% : RETURN
```

3. Spara programmet med t.ex. SAVE HYRA.
4. Gör som punkt 4 ovan.

9. Fler än ett register

Det enklaste sättet att använda fler än ett register är att lägga dem på samma dataskiva. Då bör man på t.ex. rad 1 ha en kod som anger vilket register det är. Vid sökning i registret testas man på rad 1 och beroende på vilket register det var går man till respektive subrutin för presentation på skärmen. Radantal och radrubriker kan på så sätt vara olika i de olika registren. Du måste alltså kunna programmera lite själv. Men Du har god hjälp av GOSUBadresserna på sidan 3.

1. LOAD BAZ80
2. Ta bort gamla och lägg till nya programavsnitt.
3. SAVE SPECIAL
4. Vid körning: Se punkt 4 under LÄS på sidan 4.

10. Assembler-calls

Vissa assemblerprogram avslöjas här nedan:

BLINK:

Anrop: Z%=CALL(64500,F)

F är felnumret enligt tabell nedan. Efter anropet blinkar en text fem gånger mitt på skärmens nedersta rad.

Felnr	Text
1	Fel i programmet
2	Otillåtet kommando
3	Otillåtet tal
4	Finns ej
5	För lång text
6	Oläsbara data
7	Läsfel från skivan
8	Skrivfel på skivan
9	Skivan full

QSORT:

Anrop: Ex: P9\$="A\$,A%,N%" : Z%=CALL(-3703%)

Härefter sorteras strängvektorn A\$() samt även dess adressvektor A%(). Totalt sorteras bara de N% första elementen i strängvektorn. Argument 2 och 3 i P9\$ kan slopas.

PDATA

Hökvägen 113
222 48 Lund
040-464000