

Om Commodore 128

Commodore 128 introducerades på marknaden år 1985, och såldes fram till och med år 1989, när 16-bitarssystemen som Atari ST och Amiga började vinna mark. Som namnet indikerar har datorn 128 kilobyte (KB) RAM (som kunde utökas till 640 KB) vilket räcker ganska långt för många olika typer av program, men det är kanske lite i underkant för mer avancerade animationer och avancerad multimedia.

Detta kapitel ger en snabb överflygning över datorns chips, hur BASIC-program matas in, vilka datatyper BASIC hanterar och vilken minneshanteringskapacitet som BASIC erbjuder.

Datorn har två huvudprocessorer. MOS 8502 klarar samma instruktioner som de processorer som satt i bl.a. Commodore 64 (MOS 6510 eller MOS 8500) och VIC-20 (MOS 6502). Det är denna som normalt driver runt din Commodore 128 med en arbetshastighet på 1-2 megahertz (MHz). Den andra processorn är en Zilog Z80 på 4 MHz. Det är denna som driver runt din dator när du arbetar i CP/M-läge.

Det finns ett antal olika operativsystem för Commodore 128. Datorn kan köras med eller utan operativsystemet CP/M. CP/M (*Control Program for Microcomputers*) behöver läsas in från diskett (att "boota" operativsystemet). Datorn levererades med version CP/M Plus version 3.0, och ger tillgång till avancerad mjukvara som t.ex. Turbo Pascal eller Microsoft Basic.

För den som inte vill köra operativsystem som likt CP/M styrs med textkommandon, finns möjligheten att köpa till det grafiska operativsystemet GEOS som kontrolleras med mus. GEOS (Graphic Environment Operating System) använder sig av rullgardinsmenyer, fönster och ikoner för att låta användaren kontrollera datorn.

Commodore 128 har samma ljudkapacitet som Commodore 64, som drivs av MOS-chippet 6581 (8580 i senare modeller). Dessa kallas kort och för SID (Sound Interface Device).

För grafik har Commodore 128 en VIC-II E med ungefär samma kapacitet som VIC-II som satt i Commodore 64, vilket innebär en skärmapplösning på 320×200 punkter (pixlar), 16 färger, 8 sprites och raster. Förutom detta har version E även stöd för blitter och ett extra grafikläge med en skärmapplösning på 640×200 punkter.

Om inget operativsystem startas, används själva Commodore BASIC 7.0 för att kontrollera datorn genom BASIC-kommandon.

För att programmera Commodore 128 används företrädesvis just Commodore BASIC, som i utförande 7.0 är väldigt kraftfullt. Den stora nackdelen med BASIC är dess undermåliga prestanda. BASIC kan göra allt du önskar men om prestanda är en faktor måste du titta på något annat, t.ex. maskinkod. Maskinkod presenteras överskådligt i appendix C.

Inmatning av BASIC-program

Språket har två lägen. Det ena kallas *direkt*, och innebär att man skriver en instruktion utan radnummer, som exekveras direkt när man trycker på **Return**. Det andra kallas *runtime*. Instruktioner som får ett radnummer, exekveras i runtime, alltså när programmet körs med (normalt) RUN. Om inget annat anges, kan alla kommandon användas både i direktläge och i runtime-läge. För den intresserade bjuds en hel del trevliga tricks, som t.ex. AUTO, som visas på bilden.



Figur 3: AUTO erbjuder automatisk inskrivning av radnummer.

Kommandot AUTO tar ett argument, och det är avståndet från nuvarande rad till nästa. Genom att skriva AUTO 10 så säger man till datorn att nästa radnummer ska vara nuvarande plus 10, vilket gör att datorn föreslår 20 efter att ett kommando matats in på rad 10, och så vidare.

För att stänga av automatiskt förslag på radnummer, skriv `AUTO` utan några parametrar och tryck **Return**.

Om du vill infoga ett kommando mellan, säg, rad 10 och rad 20, kan du kalla den nya raden för 15. Genom att skriva in dessa tre rader...

```
10 PRINT "A"  
20 PRINT "B"  
15 PRINT "C"
```

...så får du följande program, som kan visas med `LIST`:

```
10 PRINT "A"  
15 PRINT "C"  
20 PRINT "B"
```

Om du infogar tillräckligt många rader mellan rad 10 och 20, kommer utrymmet att ta slut. Kommandot `RENUMBER` justerar avståndet mellan existerande rader så att den första raden börjar på 10 och att avståndet mellan alla rader. Exempel:

```
RENUMBER
```

```
READY.  
LIST
```

```
10 PRINT "A"  
20 PRINT "C"  
30 PRINT "B"
```

Det finns även en möjlighet att själv ange startnummer och steg. Genom att anropa `RENUMBER 3,2` kommer programmet att starta på rad 3 och efterföljande rader kommer med ett avstånd på 2.

```
RENUMBER 3,2
```

```
READY.  
LIST
```

```
3 PRINT "A"  
5 PRINT "C"  
7 PRINT "B"
```

Du kan även ange `RENUMBER` endast ska agera på rader som är lika med eller större än ett visst befintligt radnummer genom att komplettera med ett

startnummer. `RENUMBER 1000,10,5` säger att kodrader i programmet ska starta på rad 1000 och efterföljande rader kommer på ett avstånd på 10, men det gäller bara raderna som nu är större än eller lika med 5 (alltså inte rad 3).

```
RENUMBER 1000,10,5
```

```
READY.  
LIST
```

```
3 PRINT "A"  
1000 PRINT "C"  
1010 PRINT "B"
```

Om du använder `RENUMBER` i ett program så uppstår felet *direct mode only*, och `RENUMBER` får inte användas när programmet har en `GOTO` eller `GOSUB` som inte pekar på en existerande rad, då uppstår felet *unresolved reference*. Om programmet slår över taket (högsta tillåtna radnummer är 63999) uppstår felet *line number too large*.

En liten varning! I Commodore BASIC definieras subrutiner av ett radnummer, vilket innebär att du behöver kontrollera vilka nya radnummer dina subrutiner har fått efter att du använt `RENUMBER`. Annars kan det bli svårt att anropa dessa på nytt.

Datatyper

Commodore BASIC 7.0 har stöd för tre datatyper. Dessa är *realtal*, *heltal* och *strängar*. Realtal använder punkt som decimalavgränsare och strängar omges av citattecken. Variablernas typ deklarerar med ett postfix på variabelnamnet, där `§` (dollarstecken) avser sträng och `%` (procentstecken) avser heltal. Realtal är variabler som saknar ett avslutande tecken. Apropå variabelnamn så identifieras variabler endast av de två första bokstäverna i sitt namn. En mer ingående beskrivning av datatyper och variabelnamn finns i boken *Commodore BASIC 2.0 second release*¹.

¹ Finns att ladda hem i PDF-format eller EPUB-format här:
<https://ahesselbom.se/pages/commodorebasic20.html>

Minneshantering

Commodore 128 arbetar med s.k. minnesbanker. En minnesbank är en fördefinierad minneskonfiguration, och du bestämmer vilken minnesbank som är tillgänglig för processorn genom att använda kommandot `BANK`. Som argument tar `BANK` ett tal mellan 0 och 15. Exempel:

```
BANK 4
```

Du kan hoppa mellan nio olika banker. Dessa har nummer 0, 1, 4, 5, 8, 9, 12, 13, 14 och 15. 2 är samma som 0, 3 är samma som 1, 6 är samma som 4, 7 är samma som 6, 10 är samma som 8 och 11 är samma som 9. Val av minnesbank påverkar kommandon som använder minnet direkt. Dessa kommandon är `SYS`, `PEEK`, `POKE` och `WAIT`. Minnesbank 0 är den som är förvald.

Vill du se en konsekvens av att växla mellan minnesbanker, kan du köra ett enkelt testprogram².

10 POKE 4096,75	<i>Rad 10 skriver värdet 75 i aktuell minnesbank (vilket är 0 eller 3 om inget annat har sagts).</i>
20 PRINT PEEK(4096)	<i>Rad 20 konstaterar det skrivna värdet.</i>
30 BANK 1	<i>Rad 30 växlar minnesbank till 1.</i>
40 POKE 4096,90	<i>Rad 40 skriver 90 till samma adress, fast i bank 1 istället för bank 0 eller 3.</i>
50 PRINT PEEK(4096)	<i>Rad 50 konstaterar det skrivna värdet.</i>
60 BANK 0	<i>Rad 60-70 växlar tillbaka till bank 0 (eller 3) och konstaterar att adressen fortfarande innehåller 75.</i>
70 PRINT PEEK(4096)	

Utöver detta, kommandot `STASH` och kommandot `FETCH` används för att kopiera data mellan minnesbanker och `SWAP` används för att låta två minnesbanker byta data mellan varandra.

Avslutningsvis, om du vill läsa av minnesadressen till en specifik variabel används funktionen `POINTER`. Exempel:

```
PRINT POINTER(A)
```

Resultatet av ovanstående är att minnesadressen till `A` skrivs på skärmen.

² Kom ihåg att använda kommandot `NEW` för att radera eventuellt befintligt BASIC-program ur minnet, innan du skriver in ett nytt program.

Tre datorer i en

Din dator kan köras i tre lägen:

- Commodore 128-läge
- CP/M-läge
- Commodore 64-läge

Commodore 128 är egentligen tre datorsystem i en enda låda. Vid normal uppstart hamnar man i det som kallas **Commodore 128-läge**, och det är där vi har Commodore BASIC 7.0 som primärt kontrollprogram.

Om datorn startas med boot-disken för CP/M 3.0, hamnar man i **CP/M-läge**. I detta läge är det inte längre MOS-processorn (MOS 8502) som är huvudprocessor, utan Z80-processorn. I CP/M-läge kan du utföra avancerade I/O-instruktioner och köra program som Turbo Pascal eller Microsoft BASIC. Du kan växla till CP/M-läget från Commodore 128-läget genom att skriva `BOOT` (givet att rätt diskett är laddad i diskettstationen³).

CP/M-läget beskrivs i ett eget kapitel (sida 56).

Om du håller ner Commodore-tangenten medan du startar datorn, hamnar du i **Commodore 64-läge**. Detta läge är mycket attraktivt för sitt gigantiska utbud av mjukvara, främst spelprogram. Du kan växla till Commodore 64-läget från Commodore 128-läget genom att skriva:

```
GO 64
```

(Skrivs ofta `GO64`.) Commodore 64-läget beskrivs i ett eget kapitel (sida 54).

³ Jag använder ordet diskdrive synonymt med ordet diskettstation i den här boken.