



Commodore BASIC 7.0
för Commodore 128
av Anders Hesselbom

Commodore BASIC 7.0

för Commodore 128

av Anders Hesselbom

Innehåll

| | |
|--|----|
| Introduktion..... | 3 |
| Om Commodore 128..... | 9 |
| Text | 13 |
| Grafik | 15 |
| Sprites | 17 |
| Ljud..... | 19 |
| Musik | 21 |
| Avancerade ljudeffekter | 23 |
| Användarinteraktion..... | 25 |
| 80-kolumnsläge | 27 |
| Commodore BASIC 7.0 DOS | 29 |
| Commodore 64-läge..... | 31 |
| CP/M..... | 33 |
| Appendix A: Ordförklaringar..... | 35 |
| Appendix B: En jämförelse mellan Commodore 128, Commodore 64 och VIC-20 | 37 |
| Appendix C: Maskinkod | 39 |
| Index | 41 |
| Bilder..... | 43 |

KAPITEL 1: INTRODUKTION

Introduktion

Commodore 128 är en av de mest mångsidiga och kapabla datorer som någonsin skapats. Maskinen har en avancerad BASIC (som är denna boks primära fokus), är kompatibel med Commodore 64, har två huvudprocessorer som används antingen inom eller utanför diskoperativsystemet CP/M, och har en generös uppsättning av inbyggda kommandon och verktyg.

Min bok om **Commodore BASIC 2.0 second release** handlar främst om det nämnda *språket*. Om du köpte en VIC-20 eller en Commodore 64 var det den BASIC-versionen du fick inbyggd i din dator. Commodore BASIC 2.0 second release saknar kommandon för multimedia, så ska du skriva program som utnyttjar datorns kapacitet för grafik och ljud är du hänvisad till att sätta minnesadresser eller rent av välja maskinkod istället för BASIC. Den boken är neutral till ditt val av dator.

Commodore BASIC 7.0 är ett språk framtaget för just Commodore 128, och den datorn har ungefär samma multimediakapacitet som Commodore 64. Den här boken är därför knuten till en specifik dator, nämligen just Commodore 128, och vänder sig till dig som vill bemästra den datorn, främst genom att lära sig dess BASIC – Commodore BASIC 7.0 – ett språk som erbjuder avancerade kommandon för multimedia.

Den här boken ger inte någon komplett bild av Commodore BASIC 7.0, utan fokuserar på datorn Commodore 128 och nyheterna i Commodore BASIC 7.0 jämfört med Commodore BASIC 2.0 second release.

Konventioner i boken

Indata som programrader eller kommandon skrivs med följande teckensnitt:

```
PRINT "HEJ"
```

Samma teckensnitt används för svaren från datorn.

Hänvisningar till tangenter på Commodore 128 skrivs med fetstil. Bilden visar till exempel **Return** till höger, **Run Stop** till vänster, och så vidare.



Figur 1: Tangentbordslayout på Commodore 128. Foto: Evan Amos

Den exakta tangentbordslayouten varierar beroende på vilken marknad du den dator du köpt är avsedd för. Bilden ovan visar en engelsk Commodore 128.

Ibland ska du trycka ner två tangenter. Om det står till exempel **Shift+A** ska **Shift** hållas nedtryckt medan **A** trycks ner.

Bildförklaringar och kodförklaringar skrivs i *kursiv stil*, som också används för att emfasera termer eller viktiga poänger. Även namn på felmeddelanden skrivs med kursiv stil.

Commodore BASIC 7.0

Commodore BASIC 7.0 är en vidareutveckling av Commodore BASIC 2.0 second release, och innehåller ungefär samma uppsättning av kommandon som Commodore BASIC 3.6. Förutom samtliga kommandon från 2.0 och några kommandon för flödeskontroll och felsökning handlar de flesta antingen om I/O eller multimedia.

Version 3.6 togs fram till en bärbar dator, *Commodore LCD*, som aldrig nådde marknaden. Men mycket arbete som Commodore gjorde, togs med till Commodore 128, däribland BASIC, som färdigutvecklat fick versionsnumret 7.0.

Kommandot `SYS`, har utökats för att vara mer mångsidigt.

För flödeskontroll har vi fått både ett `ELSE`-kommando och några nya sätt att skapa iterationer med `DO` och `LOOP`.

För den som vill analysera eller manipulera textsträngar finns t.ex. det nya kommandot `INSTR`.

För I/O finns en rik uppsättning kommandon för att skriva och läsa data, som t.ex. `BLOAD` och `BSAVE`.

För att hantera användarinteraktioner finns kommandon för att läsa av joystick och ljuspenna.

För högupplöst grafik finns inbyggda kommandon för att rita figurer som linjer, cirklar och rektanglar.

För rörlig grafik finns en uppsättning av kommandon för att hantera sprites.

För att skapa musik finns det mycket avancerade kommandot `PLAY`, och för den som vill utveckla egna ljudeffekter finns kommandot `SOUND`.

I princip finns det kommandon för att komma åt Commodore 128:s samtliga funktioner.

Versioner

Commodore BASIC finns i de versioner som presenteras nedan.

Version 1.0 för Commodore PET 2001 som baseras på Microsoft BASIC.

Version 2.0 för Commodore PET 2001 som är en vidareutveckling av version 1.0.

Version 4.0 för Commodore PET 4000 och CBM 8000 är den sista vidareutvecklingen av första version 2.0.

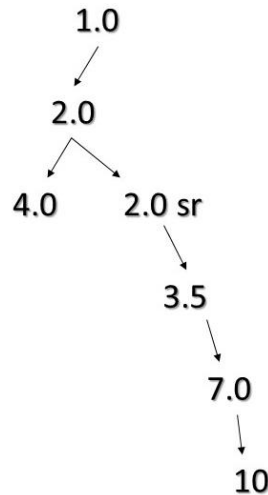
Version 2.0 second release som avhandlas i denna bok är buggrättad version av version 2.0 för PET 2001.

Version 4+ för Commodore CBM-II är en vidareutveckling på version 2.0 för PET 2001.

Version 3.5 för Commodore 16, Commodore 116 och Plus/4 är en vidareutveckling av Commodore BASIC 2.0 second release.

Version 7.0 för Commodore 128 är en vidareutveckling av version 3.5. Denna fanns även i prototypen Commodore LCD med versionsnumret 3.6.

Version 10 utvecklades för prototypen Commodore 65. Varken Commodore 65 eller Commodore LCD nådde någonsin konsumentmarknaden. Tyska *MEGA Museum of Electronic Games and Art* arbetar med att få ut en färdigställd Commodore 65-klon på marknaden.



Figur 2: Språkets utveckling.

Bokens innehåll

Den här boken innehåller, inklusive introduktionen, 13 kapitel och tre bilagor. Här följer en överblick över bokens kapitel, utöver detta första kapitel:

- Det andra kapitlet ger en övergripande beskrivning om datorn boken handlar om, **Commodore 128**.
- Kapitlet **Text** presenterar nya möjligheter att analysera och manipulera text.
- I kapitlet om **grafik** beskrivs hur högupplöst grafik kan skapas med Commodore BASIC 7.0.
- **Sprites** handlar om rörlig grafik och enklare animationer.
- Kapitlet om **ljud** visar hur man kan få Commodore 128 att spela upp enklare toner och effekter.
- I kapitlet om **musik** beskrivs hur melodier kan komponeras och framföras av flera röster.
- I kapitlet om **avancerade ljudeffekter** beskrivs tidigare odokumenterade funktioner för att skapa nya ljud med Commodore BASIC 7.0.
- Kapitlet om **användarinteraktion** beskriver hur man läser av tangentbordet, joysticks och ljuspennan.
- I kapitlet om **80-kolumnsläge** förklaras hur man kan dra nytta av datorns förmåga att dubblera antalet tecken som visas på skärmen.
- Därefter beskrivs de utökade möjligheterna att bevara data på disk i kapitlet om **DOS**.
- **Commodore 64-läget** beskrivs i det tolfte kapitlet.
- Det trettionde och sista kapitlet ger en introduktion till **CP/M**.

Här följer en beskrivning av bokens 3 bilagor, kallade *appendix A, B* och *C*:

- **Appendix A** förklarar de tekniska termer som används i boken.
- **Appendix B** jämför Commodore 128 med föregångarna Commodore 64 och VIC-20.
- **Appendix C** ger en introduktion till maskinkod.

KAPITEL 2: OM COMMODORE 128

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.

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.

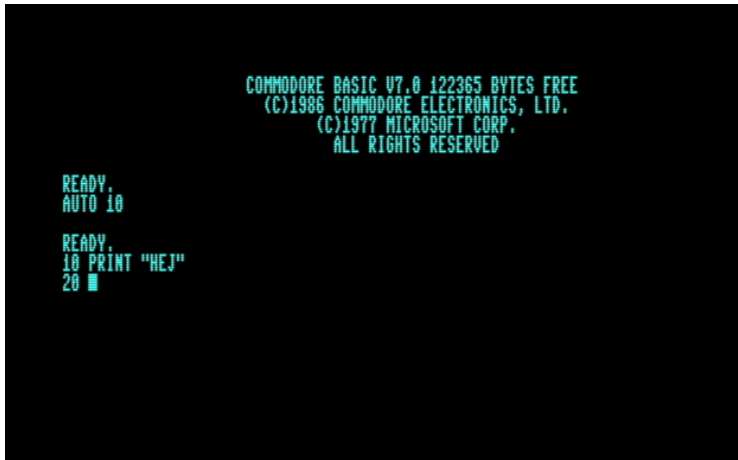
För den som inte vill köra operativsystem som likt CP/M styrs med textkommandon, finns möjligheten att köpa till det grafiska operativsystemet GEOS som kontrolleras med mus. GEOS (Graphic Environment Operating System) använder sig av rullgardinsmenyer, fönster och ikoner för att låta användaren kontrollera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

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 `$` (dollartecken) avser sträng och `%` (procenttecken) 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.

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
20 PRINT PEEK(4096)
30 BANK 1
40 POKE 4096,90
50 PRINT PEEK(4096)
60 BANK 0
70 PRINT PEEK(4096)
```

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

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

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

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

KAPITEL 3: TEXT

Text

XXX

Text till tal

En textsträng innehållande ett decimaltal (alltså ett tal som beskrivs av tecknen 0 till och med 9) konverteras till ett tal med funktionen VAL. Detta enkla program skriver värdet 40 på skärmen.

```
10 X=VAL ("20")      Rad 10 konverterar en sträng innehållande en tvåa och en nolla
20 PRINT X*2          till talet 20.
                      Rad 20 multiplicerar resultatet med 2 och skriver ut det (40) på
                      skärmen.
```

Exemplet visar hur en sträng innehållande ett tal kan konverteras till ett riktigt tal och användas i numeriska beräkningar. Detta är ganska standard inom BASIC, och har varit med länge i Commodore BASIC.

Utöver VAL kan Commodore 128 även jobba med strängrepresentationer av det hexadecimala talsystemet. TODO DEC

Formatera ett tal

TODO PUDEF

KAPITEL 4: GRAFIK

Grafik

XXX

KAPITEL 5: SPRITES

Sprites

KAPITEL 6: LJUD

Ljud

XXX

KAPITEL 7: MUSIK

Musik

XXX

KAPITEL 8: AVANCERADE LJUDEFFEKTER

Avancerade ljudeffekter

XXX

KAPITEL 9: ANVÄNDARINTERAKTION

Användarinteraktion

XXX

KAPITEL 10: 80-KOLUMNSLÄGE

80-kolumnsläge

XXX

KAPITEL 11: COMMODORE BASIC 7.0 DOS

Commodore BASIC 7.0 DOS

XXXX

KAPITEL 12: COMMODORE 64-LÄGE

Commodore 64-läge

XXX

KAPITEL 13: CP/M

CP/M

XXX

APPENDIX A: ORDFÖRKLARINGAR

Appendix A: Ordförklaringar

I/O

I/O är en förkortning av input/output och avser operationer som läser eller skriver från/till externa enheter som till exempel tangentbord, printer, skärm eller floppydisk.

Blitter

Ett chip som hanterar blitter ansvarar för snabb manipulering av data i RAM, vilket möjliggör rörlig datorgrafik.

Radnummer

BASIC-program på Commodore 128 använder tal för att hålla reda på vilket kommando som ska utföras i vilken ordning, och framför allt, vilka kommandon som ingår i ett program, genom att numrera programsatserna.

APPENDIX B: EN JÄMFÖRELSE MELLAN COMMODORE 128,
COMMODORE 64 OCH VIC-20

Appendix B: En jämförelse mellan Commodore 128, Commodore 64 och VIC-20

Tabellen nedan visar specifikationerna för VIC-20, Commodore 64 (C64) och Commodore 128 (C128).

| | VIC-20 | C64 | C128 |
|-----------------------------|---|---|---|
| Lanseringsår | 1980 | 1982 | 1985 |
| Programmeringsspråk | Commodore BASIC 2.0 second release | Commodore BASIC 2.0 second release | Commodore BASIC 7.0 |
| Längsta programsats | 88 tecken (4 rader) | 80 tecken (2 rader) | 160 tecken (4 rader i 40-kolumnsläge, 2 rader i 80-kolumnsläge) |
| Operativsystem ³ | - | GEOS (tillval) | GEOS (tillval), CP/M 3.0 |
| Huvudprocessor | MOS 6502 | MOS 6510 | MOS 8502, Z80B |
| Klockfrekvens ⁴ | 1,1 MHz | 0,99 MHz | 1-2 MHz, 4 MHz |
| ROM | 20 KB | 20 KB | 72 KB |
| RAM | 5 KB | 64 KB | 128 KB |
| Expansionsmöjlighet RAM | 32 KB ⁵ | 320 KB | 512 KB |
| Text | 22×23 | 40×25 | 40×25, 80×25 |
| Skärmupplösning | 176×184 plus border | 320×200 plus border | 320×200 plus border, 640×200 |
| Monokrom grafik | 176×184 | 320×200 | 320×200 |
| Flerfärgsgrafik | 88×184 | 160×200 | 160×200 |
| Videoutgång | Analog (A/V) | Analog (RF, A/V) | Analog (RF, A/V)/Digital (RGBI) |
| Ljud | Tre fyrkantsvågor och ett brusljud, tre kanaler | Konfigurerbar fyrkantsvåg, triangelvåg, sinusvåg, brus, filter, med mera. Tre kanaler | Konfigurerbar fyrkantsvåg, triangelvåg, sinusvåg, brus, filter, med mera. Tre kanaler |
| Maskinkodsmonitor | - | - | Ja |
| Sprites | - | 8 | 8 |
| Sprite-editor | - | - | Ja |

³ Utöver BASIC.

⁴ Processorns hastighet är aningen olika för PAL-anpassade datorer och NTSC-anpassade datorer. För Commodore 128 gäller 4 MHz när Z80B-processorn används.

⁵ Bryter kompatibiliteten.

APPENDIX C: MASKINKOD

Appendix C: Maskinkod

XXX

INDEX

Index

16-bitarstal, 23
AUTO, 10
blitter, 9, 35
border, 37
Commodore 128, 37

Commodore 64, 37
I/O, 35
radnummer, 10, 35
VAL, 13
VIC-20, 37

BILDER

Bilder

| | |
|---|----|
| Figur 1: Tangentbordslayout på Commodore 128. Foto: Evan Amos | 4 |
| Figur 2: Språkets utveckling..... | 6 |
| Figur 3: AUTO erbjuder automatisk inskrivning av radnummer..... | 10 |

Erkännanden:

Omslagsbilden föreställande en Commodore 128 är fotograferad av Evan Amos (CC BY-SA 3.0).