

INTRODUKTION:

PASCAL.....	2
PASCAL I SKOLAN.....	3
TURBO PASCAL+ FÖR PROGRAMUTVECKLING.....	3
PORTABEL PROGRAMVARA.....	4
MANUALEN.....	5
PROGRAMNOTATION.....	6
OLIKA VERSIONER AV TURBO PASCAL.....	7

KAPITEL 1 - 20**1. INSTALLATION OCH ANVÄNDNING AV TURBO PASCAL +**

1.1 COM och CMD-filer.....	1
1.2 KOPIERING AV PROGRAM.....	1
1.3 FILER.....	2
1.4 INSTALLATION.....	3
1.4.1 INSTALLATION AV SVENSKA TECKEN (CP/M-86,MS-DOS)...	3
1.4.2 INSTALLATION AV BILDSKÄRMEN.....	4
1.4.3 INSTALLATION AV EDITORNS KOMMANDON.....	5
1.5 SÅ ANVÄNDS TURBO PASCAL.....	6
1.5.1 L = arbetsdisk: X.....	7
1.5.2 A = aktuellt bibl(iotek): X MS-DOS, PC-DOS.....	7
1.5.3 W = arb(ets)fil.....	8
1.5.4 M = huv(ud)fil.....	8
1.5.5 E = edit(era).....	8
1.5.6 C = komp(iler).....	8
1.5.7 R = exekv(era).....	9
1.5.8 S = spar(a).....	9
1.5.9 X = annat program: CP/M-80.....	10
1.5.10 D = bibl(iotek).....	10
1.5.11 Q = slut.....	10
1.5.12 O = kompileringsval.....	10
1.5.13 UTVECKLING AV ETT PASCALPROGRAM.....	10
1.5.14 UTSKRIFT PÅ PRINTER.....	11
1.5.15 EXEMPEL: CALC/MC.....	11

1.6 EDITORN I TURBO PASCAL	12
1.6.1 AVSLUTA EDITERING	13
1.6.2 STYRNING AV MARKÖREN.....	13
1.6.3 INSÄTTNING OCH RADERING AV TEXT	14
1.6.4 BLOCKKOMMANDON.....	14
1.6.5 ÖVRIGA KOMMANDON.....	15
1.7 KOMPILERINGS DIREKTIV.....	16
1.8 SVENSKA TECKEN.	17
2. SPRÅKELEMENT	
2.1 TECKENUPPSÄTTNING.....	1
2.2 SAMMANSATTA SYMBOLER.....	1
2.3 AVVIKELSER I SVENSK TECKENSTANDARD.	1
2.4 RESERVERADE ORD.	3
2.5 STANDARDIDENTIFIERARE.....	3
2.6 PROGRAMRADER.....	3
3. SKALÄRA STANDARDTYPER	
4. EGENDEFINIERADE SPRÅKELEMENT	
4.1 IDENTIFIERARE.....	1
4.2 TAL.....	2
4.3 STRÅNGAR.	2
4.31 KONTROLLTECKEN.	2
4.4 KOMMENTARER.....	3
4.5 KOMPILERINGS DIREKTIV.....	3
5. PROGRAMHUVUD OCH PROGRAMBLOCK	
5.1 PROGRAMHUVUD.....	1
5.2 DEKLARATIONSDELEN.....	1
5.2.1 LABELDEKLARATIONEN.	2
5.2.2 KONSTANTDEKLARATIONEN.	2
5.2.3 TYPDEKLARATIONEN.....	3
5.2.4 VARIABELDEKLARATIONEN.....	3
5.2.5 PROCEDUR- OCH FUNKTIONSDEKLARATIONERNA.....	4
5.3 EXEKVERINGSDELEN.....	4
6. UTTRYCK	
6.1 OPERATORER.....	1
6.1.1 DIV OCH MOD VID NEGATIVA HELTAL.....	2
6.1.2 ARITMETISKA LOGISKA OPERATORER.....	2

7. PROGRAMSATSER	
7.1 ENKLA SATSER.....	1
7.2 SAMMANSATTA SATSER.....	1
7.2.1 CASE-SATSEN.....	2
8. SKALÄRA TYPER OCH DELINTERVALL	
8.1 EGENDEFINIERADE TYPER OCH DELINTERVALL	1
8.2 TYPKONVERTERING.....	1
8.3 KONTROLL AV GRÄNSER.....	2
9. STRÄNGHANTERING	
9.1 DEKLARATION AV STRÄNGTYPER.....	1
9.2 STRÄNGUTTRYCK.....	2
9.3 TILLDELNING AV STRÄNGAR.....	2
9.4 STRÄNGPROCEDURER.....	3
9.5 STRÄNGFUNKTIONER.....	3
9.6 STRÄNGAR OCH TECKEN.....	3
10. FÄLT	
10.1 ARRAY OF CHAR KONTRA STRING.....	1
10.2 KONTROLL AV INDEX.....	1
10.3 FÖRDEKLARERADE FÄLT.....	1
11. POSTER	
11.1 NÄSTLADE WITH-SATSER.....	1
11.2 POSTER MED VARIANTDEL.....	1
12. MÄNGDER	
12.1 MÄNGDOPERATORER.....	1
12.2 MÄNGDUTTRYCK.....	1
12.3 BITOPERATIONER MED MÄNGDER.....	2
13. TYPDEKLARERADE KONSTANTER	
13.1 ENKLA TYPDEKLARERADE KONSTANTER.....	1
13.2 STRUKTURERADE TYPDEKLARERADE KONSTANTER.....	2
13.2.1 FÄLTKONSTANTER.....	2
13.2.2 KONSTANTER AV POSTTYP.....	3
13.2.3 KONSTANTER AV MÄNGDTYP.....	4

14. FILHANTERING

14.1 DEKLARATION AV FILTYP.	1
14.2 SKILLNADER MELLAN TURBO PASCAL OCH STANDARD.	2
14.2.1 INTERNA OCH EXTERNA FILER.	2
14.2.2 DIREKTFILER.	2
14.2.3 HUR MAN LÄSER OCH SKRIVER PÅ FILER.	3
14.2.4 FELHANTERING.	3
14.2.5 LOGISKA ENHETER.	3
14.2.6 FILER UTAN ELEMENTTYP.	3
14.3 OPERATIONER PÅ FILER.	3
14.4 TEXTFILER.	6
14.4.1 BEARBETNING AV TEXTFILER.	7
14.4.2 FORMATTERING AV UTSKRIFT.	8
14.4.3 LOGISKA ENHETER.	9
14.4.4 STANDARDFILER.	10
14.4.4.1 KOPPLING LOGISKA OCH FYSISKA ENHETER.	10
14.4.4.2 Eof OCH Eoln.	11
14.5 FILER UTAN ELEMENTTYP.	12
14.6 FELHANTERING VID IN- OCH UTMATNING.	13

15. PEKARE

15.1 NEW - DYNAMISKA VARIABLER SKAPAS.	2
15.2 DISPOSE - MINNE ÅTERVINNS.	3
15.3 MARK/RELEASE - MINNE ÅTERVINNS.	3
15.4 ALLOKERING AV MINNE.	4

16. PROCEDURER OCH FUNKTIONER

16.1 BEGRÄNSNINGAR.	1
16.2 TILLÄGG.	1
16.2.1 STRÄNGPARAMETRAR AV GODTYCKLIG LÄNGD.	2
16.2.2 VARIABELPARAMETRAR UTAN TYP.	3
16.3 STANDARDPROCEDURER OCH FUNKTIONER I TURBO PASCAL	4
16.4 UTHOPP UR FUNKTIONER OCH PROCEDURER.	5
16.5 FUNKTIONER OCH PROCEDURER SOM PARAMETRAR.	6
16.6 REKURSION.	10
16.7 FORWARD DEKLARATIONER.	11
16.8 CONFORMANT ARRAYS.	12
16.9 SEPARAT KOMPILERING AV FUNKTIONER OCH PROCEDURER.	14

17. PROGRAMKOMPONENTER OCH STORA PROGRAMSYSTEM

17.1 GOD PROGRAMMERINGSTEKNIK.	1
17.2 FYSISKA BEGRÄNSNINGAR HOS HÄRDVARAN.	2

18. INCLUDEFILER

18.1 HUVUDFIL OCH INCLUDEFILER.....	1
18.2 LÄNKNING OCH INCLUDEHANTERING.....	2
18.3 KOMPILERINGSDirektiv I INCLUDEFILER.....	2
18.4 NÄSTLADE INCLUDEFILER.....	2
18.5 TLIST OCH INCLUDEFILER.....	3
18.6 EXEMPEL PÅ ANVÄNDNING AV PROGRAMKOMponenter.....	3

19. OVERLAYER OCH CHAINING

19.1 TYPFALL FÖR OVERLAYHANTERING.....	1
19.2 TYPFALL FÖR CHAINING.....	2
19.3 OVERLAYTEKNIK.....	2
19.3.1 BOK - ETT EXEMPEL MED OVERLAYER.....	3
19.3.2 PLANERING AV OVERLAYGRUPPER.....	5
19.3.3 DISKPLACERING AV OVERLAYFILER.....	6
19.3.4 RESTRIKTIONER FÖR OVERLAYER.....	6
19.3.5 EXEKVERINGSFEL I OVERLAYER.....	7
19.4 CHAINTEKNIK.....	7
19.4.1 CHAIN.....	7
19.4.2 EXECUTE.....	9

20. EXTERNA UNDERPROGRAM

20.1 EXTERNA UNDERPROGRAM.....	2
20.2 EXTERNALAREAN.....	2
20.3 INTERNA UNDERPROGRAM.....	3
20.4 LADDNING OCH ANROP AV EXTERNA UNDERPROGRAM.....	4
20.5 PARAMETRAR.....	5
20.6 BEGRÄNSNINGAR.....	5
20.7 FELMEDDELANDEN.....	6
20.8 UNDERHÅLL OCH FELSÖKNING.....	6
20.9 TABELLPROGRAM.....	7

APPENDIX**A. CP/M-80**

A.1 KOMPILERINGSVAL.....	1
A.1.1 KOMPILERING TILL COM-FIL.....	1
A.1.2 KOMPILERING TILL CHN-FIL.....	2
A.1.3 STARTADRESS.....	2
A.1.4 SLUTADRESS.....	2
A.1.5 FINN EXEKVERINGSFEL.....	3
A.1.6 AVSLUTA.....	3
A.2 ABSOLUTA VARIABLER.....	3
A.3 ADDR-FUNKTIONEN.....	4

A.4 FÖRDEKLARERADE FÄLT.....	4
A.5 NÄSTLADE WITH.....	4
A.6 OPTIMERING AV FÄLTINDEXERING.....	5
A.7 DYNAMISKT MINNE OCH PEKARE.....	5
A.8 EXTERNA UNDERPROGRAM.....	5
A.9 CHAIN OCH EXECUTE.....	6
A.10 INLINE MASKINKOD.....	6
A.11 SYSTEMANROP.....	7
A.12 SYSTEMVARIABLER.....	9
B. MS-DOS, PC-DOS, CP/M-86, CONCURRENT DOS	
B.1 GEMENSAMT FÖR 8086/8088-VERSIONERNA.....	1
B.1.1 KOD, DATA, STACK OCH HEAP.....	1
B.1.2 KOMPILERINGSVAL.....	2
B.1.2.1 KOMPILERING TILL MINNET.....	2
B.1.2.2 KOMPILERING TILL COM-FIL.....	3
B.1.2.3 KOMPILERING TILL CHN-FIL.....	4
B.1.2.4 MINSTA STORLEK KODSEGMENT.....	4
B.1.2.5 MINSTA STORLEK DATASEGMENT.....	4
B.1.2.6 MINSTA STORLEK PÅ STACK/HEAP.....	4
B.1.2.7 STÖRSTA STORLEK PÅ STACK/HEAP.....	5
B.1.2.8 FINN EXEKVERINGSFEL.....	5
B.1.2.9 AVSLUTA.....	5
B.1.3 ABSOLUTA VARIABLER.....	5
B.1.4 ABSOLUTA ADRESSFUNKTIONER.....	6
B.1.4.1 Addr.....	6
B.1.4.2 Ofs.....	6
B.1.4.3 Seg.....	7
B.1.4.4 Cseg, Dseg OCH Sseg.....	7
B.1.5 FÖRDEKLARERADE FÄLT.....	7
B.1.6 NÄSTLADE WITH.....	7
B.1.7 DYNAMISKT MINNE OCH PEKARE.....	7
B.1.8 EXTERNA UNDERPROGRAM.....	8
B.1.9 CHAIN OCH EXECUTE.....	9
B.1.10 INLINE MASKINKOD.....	9
B.1.11 PROCEDUREN Intr.....	10
B.1.12 SYSTEMVARIABLER.....	11
B.2 MS-DOS OCH PC-DOS.....	12
B.2.1 SYSTEMANROP.....	12
B.2.2 FILHANTERING.....	13
B.3 CP/M-86 OCH CONCURRENT DOS.....	13
B.3.1 SYSTEMANROP.....	14

C. STANDARDPROCEDURER, FUNKTIONER, FÖRDEKLARERADE VARIABLER OCH KONSTANTER

C.1 IN/UT-PROCEDURER.....	2
C.2 ARITMETISKA FUNKTIONER.....	4
C.3 SKALÄRA FUNKTIONER.....	5
C.4 TYPKONVERTERING.....	6
C.5 STRÄNGHANTERING.....	7
C.5.1 STRÄNGPROCEDURER.....	7
C.5.2 STRÄNGFUNKTIONER.....	8
C.6 FILHANTERING.....	9
C.6.1 PROCEDURER FÖR FILHANTERING.....	9
C.6.2 FUNKTIONER FÖR FILHANTERING.....	11
C.7 DYNAMISKA VARIABLER.....	13
C.7.1 PROCEDURER FÖR DYNAMISKA STRUKTURER.....	13
C.7.2 FUNKTIONER FÖR DYNAMISKA STRUKTURER.....	14
C.8 TERMINALPROCEDURER.....	15
C.9 ÖVRIGT.....	17
C.9.1 PROCEDURER.....	17
C.9.2 FUNKTIONER.....	18
C.10 FÖRDEKLARERADE KONSTANTER.....	21
C.11 FÖRDEKLARERADE VARIABLER.....	22

D. OPERATORER**E. KOMPILERINGSDIREKTIV****F. TURBO PASCAL KONTRA STANDARDPASCAL**

F.1 DYNAMISKA VARIABLER.....	1
F.2 FILHANTERING.....	1
F.2.1 BEGRÄNSNINGAR HOS Write OCH WriteLn.....	1
F.2.2 INTERNA FILER.....	2
F.2.3 Get OCH Put.....	2
F.2.4 Read/ReadLn.....	2
F.2.5 Eoln OCH Eof OCH RADSLUTTECKEN.....	2
F.2.6 FILPARAMETRAR.....	3
F.2.7 Page.....	3
F.3 GOTO.....	3
F.4 PACKADE VARIABLER.....	3
F.5 LOKALA VARIABLER VID REKURSION.....	3
F.6 FUNKTIONER/PROCEDURER SOM PARAMETRAR.....	4
F.7 CONFORMANT ARRAYS.....	4

G. KOMPILERINGSFEL**H. EXEKVERINGSFEL**

I. I/O-FEL**J. IBM PC/XT**

J.1 TextMode.	1
J.1.1 FÄRGER I TEXTMOD.	2
J.1.2 ADRESSERING AV MARKÖREN.	3
J.2 GRAFIKMODER.	3
J.2.1 Plot OCH Draw.	3
J.2.2 GraphColorMode.	4
J.2.3 GraphMode.	5
J.2.4 HiRes.	6
J.2.5 TEXT I GRAFISK MOD.	6
J.3 FÖNSTERHANTERING.	7
J.3.1 TEXTFÖNSTER.	7
J.3.2 GRAFIKFÖNSTER.	8
J.4 LJUD.	8

K. KORT LITTERATURFÖRTECKNING

K.1 TURBO PASCAL.	1
K.2 STANDARD PASCAL.	1
K.3 LÄROBÖCKER I PASCAL.	1
K.4 STRUKTURERAD PROGRAMMERING I PASCAL.	2
K.5 MODULA-2.	2
K.6 TIDSKRIFTER.	2
K.7 ANVÄNDARGRUPPER.	2

L. LITEN ORDLISTA**M. ISO-TECKENTABELLEN****N. UTSKRIFT MED TLIST**

N.1 PUNKTKOMMANDON.	1
N.2 INCLUDEFILER.	2
N.3 PRINTERSPECIFIKA KODER.	2

O. TERMINALINSTALLATION AV KOMPILERADE PROGRAM**SAKREGISTER**

INTRODUKTION

Detta är den svenska referensmanualen till **TURBO PASCAL+**.

TURBO PASCAL står för det internationellt välkända pascalsystemet från Borland International i USA, nu i *helsvensk tappning*.

+ står för en serie väsentliga och kraftfulla *tillägg*, som avsevärt förhöjer systemets användbarhet. Dessa tillägg är unika för den svenska versionen. Tilläggen har gjorts av DATABITEN.

TURBO PASCAL+ kommer med menyer, felmeddelanden, editor, installationsprogram, listprogram etc., helt på svenska. MS-DOS och CP/M-86 versionerna har installationsprogram som gör att du enkelt kan använda programmen för både standardiserad svensk ISO-kod, som IBM:s utvidgade ASCII-kod.

Den svenska versionen har förstärkts med biblioteksrutiner för

- separatkompilering av funktioner/procedurer till externa underprogram som kan anropas dynamisk under körning.
- funktioner och procedurer som parametrar.
- modifierad svensk standardfunktion UpCase.

I **TURBO PASCAL+** ingår också

- terminalinstallationsprogram för anpassning av redan kompilerade program till olika terminaler.
- en rad generella biblioteksmoduler, bl.a. QuickSort, anrop av aktuellt DOS, konvertering från decimaltal till binära tal och hexadecimala tal etc.

Den svenska manualen söker sin förebild hos det amerikanska originalet men är helt fristående och skiljer sig på vissa punkter i dispositionen. Många avsnitt är unika i den svenska versionen och innehåller information om systemet som inte har publicerats tidigare. De avsnitt som rör "+"-delen i **TURBO PASCAL** är naturligtvis helt igenom svenska. Flera fel som har insmugit sig i den amerikanska manualen har rättats.

Grundtankarna bakom den svenska manualen har varit att

- 1) hålla sig relativt kort och därför lägga tonvikten på att ta upp det som *skiljer* **TURBO PASCAL+** från minimistandarden för Pascal (enligt "Jensen- Wirth" och ISO) och som därför inte framgår av vanliga lärotexter i Pascal. Manualen gör alltså inte anspråk på att vara en lärobok i Pascal utan vi hänvisar till de läroböcker som listas i appendix K.

2) fungera som en lättanvänd och komplett referensmanual för "normalprogrammeraren".

Vi har därför lagt särskild vikt vid att göra innehållsförteckning och register detaljrika, men har undvikit att ta upp "låg nivå"-frågor som intern representation av data, parameteröverföring via stacken till externa rutiner, intern filstruktur, egendefinerade in/ut-procedurer, debugging med DDT, SID, DDT-86, DEBUG etc. Om detta kan du läsa mer i skriften "Avancerad programmering med TURBO PASCAL +".

3) peka på de skillnader som finns gentemot "standard" Pascal vid användning av svensk teckenstandard, samt andra skillnader mellan svenska TURBO PASCAL + och den amerikanska versionen.

4) i exemplen betona moderna strukturerade programmeringsprinciper och använda en modern notation för programtext.

Till TURBO PASCAL utvecklas både i Sverige och utomlands en rad hjälpverktyg som underlättar eget programbygge.

Från Borland International kan nämnas **TURBO TOOLBOX** med bl.a. en modul för indexerad sökning med B + träd.

DATABITEN har utvecklat ett mycket lättanvänt grafiskt paket, **TURBO GRAPHICS** för bildskärmsgrafik. TURBO GRAPHICS passar utmärkt vid utveckling av egna bildskärmslayouter och inte minst i dataundervisning. (TURBO GRAPHICS bygger på samma grafik som programspråket **LOGO**).

PASCAL

Programspråket Pascal utvecklades av professor Niklaus Wirth vid "Institutionen för informationsbehandling" i Zürich kring år 1970. Många tankegångar i språket är hämtade från det äldre språket ALGOL. Wirth har nyligen ur Pascal utvecklat ett nytt språk, **Modula-2**, som kan väntas få en stor betydelse under resten av 80-talet. Modula-2 bygger vidare på Pascaltraditionen.

Utmärkande drag för Pascal är att språket

- 1) är väl strukturerat
- 2) ger kraftfulla möjligheter att bygga egna datatyper
- 3) har en sträng typkontroll
- 4) är ett "litet" språk som möjliggör effektiva, minnessnåla, tillförlitliga och mycket snabba kompilatorer (TURBO PASCAL bevisar detta).

PASCAL I SKOLAN

Under 1970-talet och i synnerhet dess senare hälft har Pascal blivit det ledande undervisningsspråket vid högskolor och universitet över hela världen.

Som nybörjarspråk bland persondatoranvändare har Pascal dock befunnit sig i ett underläge gentemot främst BASIC, framför allt på grund av svåränvända editorer, otympliga och användarovänliga kompilatorer + länkare, samt bristande integration mellan utvecklingsystemets olika delar.

Med TURBO PASCAL finns nu för första gången ett Pascalsystem som tillåter ett *interaktivt* arbetssätt, mer lättanvänt än BASIC, utan att tumma på effektivitet och prestanda.

Den svenska versionen av TURBO PASCAL är ett mycket lämpligt system för undervisning i programmering vid högskolor/universitet, gymnasier, vuxenutbildning men även på högstadienivå.

Det finns numera ingen anledning att välja BASIC som nybörjarspråk!

TURBO PASCAL + FÖR PROGRAMUTVECKLING

TURBO PASCAL + är av flera skäl ett utmärkt redskap vid programutveckling för eget bruk och programutveckling för vidareförsäljning:

- Utvecklingstid och underhållskostnader av programvaran reduceras i hög grad genom den utmärkta editorn, sekundsnabba kompileringen och effektiva felhanteringsystemet.
- TURBO PASCAL kompilerar till mycket snabb och minnessnål maskinkod. TURBO PASCAL finns i 8087-version som ger extremt snabba flyttalsberäkningar.
- TURBO PASCAL ger portabel källkod mellan de tre stora operativsystemen idag, nämligen CP/M-80 (med Z80-processorn), CP/M-86/Concurrent DOS (8086, 8088, 80186, 80286) samt MS-DOS/PC-DOS (8086, 8088, 80186, 80286).
- TURBO PASCAL innehåller många viktiga tillägg till "standard"-Pascal som fri ordning mellan deklarerade och strukturerade konstanter, includerfiler, overlayhantering, dynamisk stränghantering, direktfilshantering, lågnivå-programmering som absolutadresserade variabler, typkonvertering, bit/byte-manipulering, logiska operationer på heltal, direkt läsning/skrivning till minne och dataportar, inline kodning av maskinspråksrutiner mm.

Med hjälpverktyg från DATABITEN för separatkompilering och dynamisk laddning, funktions/procedurparametrar, terminalinstallationsrutin, grafik mm. höjs kapaciteten avsevärt.

- Ingen licensavgift vid försäljning av program som har framställts med hjälp av TURBO PASCAL +.

PORTABEL PROGRAMVARA

Under 1970-talet utvecklades flera olika Pascaldialekter som alla i viss mån kan kallas "de facto standard". Av störst betydelse har dock varit den *minimistandard* som drogs upp av Niklaus Wirth och Kathleen Jensen i deras "Users Manual and Report" (Springer-Verlag 1975). De flesta Pascalsystem brukar mätas mot denna "Jensen-Wirth standard".

En annan dialekt som i hög grad har påverkat utvecklingen är **UCSD Pascal** från University Of California vid San Diego. UCSD Pascal utvecklades mot slutet av 1970-talet under ledning av Kenneth Bowles. I UCSD Pascal finns en rad tillägg utöver minimikraven hos Jensen-Wirth. Dessa tillägg har influerat senare Pascal-system. Tyvärr finns också en del avvikelser från "Jensen-Wirth".

En officiell internationell standard för Pascal togs inte förrän 1983 (ISO 7185-1983). Den svenska standarden (SS 63 61 12) från samma år är identisk med den internationella som i sin tur är identisk med den brittiska standarden från 1982. **ISO-standard**en följer "Jensen-Wirth" i stort men avviker på ett par punkter. Den internationella standarden är i likhet med "Jensen-Wirth" en **minimistandard** som även i framtiden lämnar fältet fritt för godtyckliga tillägg utöver standarden (under förutsättning att de inte bryter mot standarden). ISO-standardens har dessutom indelats i **två nivåer**, 0 och 1, med den skillnaden att nivå 1, förutom vad som ryms i nivå 0, även tillåter så kallade "conformant arrays".

Se kapitel 16 för detaljer.

TURBO PASCAL + följer med få undantag Jensen-Wirth och ISO-standardens. De viktigaste inskränkningarna är att underprogram (funktioner och procedurer) inte får förekomma som parametrar vid procedur- och funktionsanrop samt att "conformant arrays" inte är tillåtna som formella parametrar. I den svenska versionen av TURBO PASCAL har dessa problem undanröjts. Se kapitel 16.

Till minimistandarden har gjorts en rad tillägg (se bl.a. appendix C och D där tilläggen har noterats med "TURBO") varav många i enlighet med andra populära Pascaldialekter, i synnerhet UCSD-Pascal.

Vissa tillägg är knutna till ett visst operativsystem eller processor.

För att uppnå största möjliga **portabilitet** mellan olika *TURBO PASCAL- implementeringar* ska tillägg som är beroende av operativsystem och hårdvara undvikas, eller åtminstone isoleras till ett fåtal välkända programdelar.

För att uppnå största möjliga portabilitet mellan olika *Pascalssystem*, bör man begränsa sig till ISO-standarderna eller möjligen standarden enligt Jensen/Wirth. Många av de tillägg som finns i UCSD Pascal kan dock användas, fortfarande med god flyttbarhet.

Å andra sidan talar mycket för att man bör utnyttja sådana kraftfulla tillägg som **fri placering av deklarationer, typkonvertering, "open arrays", STRING-hantering** mm., som alla ligger i linje med Wirths utveckling av Pascal till Modula-2. Eftersom Modula-2 kan förväntas få stor spridning underlättas konvertering av programvara (och programmerare) om man redan nu börjar utnyttja flera av de finesser som ligger i TURBO PASCAL.

MANUALEN

Kapitel 1 beskriver installationen och användningen av TURBO PASCAL.

Kapitel 2 - 20 beskriver syntaxen hos TURBO PASCAL+ och är gemensamt för samtliga versioner.

Appendix A beskriver det som är speciellt för CP/M-80/Z80-versionen.

Appendix B beskriver först det som är speciellt för 8086/88-versionerna generellt. Därefter beskrivs också det som skiljer CP/M-86 (Concurrent DOS) från MS-DOS (PC-DOS).

Appendix C sammanfattar alla standardfunktioner/procedurer samt fördeklarade variabler och konstanter i TURBO PASCAL. Detta appendix är till stor hjälp vid programmeringsarbetet.

Appendix G - I beskriver felmeddelanden.

ξ För IBM PC-användare beskrivs appendix J de tillägg för grafik, fönsterhantering och ljud som finns i PC-DOS-versionen.

Etc.

PROGRAMNOTATION

Pascal ger stor frihet vid den typografiska utformningen av programtexten. Ett pascalprogram kan i princip skrivas som en enda lång sammanhängande ordrams och ändå godkännas av kompilatorn.

Under årens lopp har dock utvecklats en praxis för hur man skriver pascalprogram, eller strukturerade program överhuvud taget. Alla är eniga om att programstrukturen markeras genom **indragningar (indenteringar)** av texten. Hur *stora* indragningar som ska göras och andra detaljer är dock omtvistat. Vi kommer här att i stora drag följa den linje som rekommenderas av exv. Kernighan/Plauser (se appendix K) och av Wirth i Modula-2.

Reglerna kan kort beskrivas så här:

1. Alla reserverade ord i TURBO PASCAL skrivs med versaler (stora bokstäver), t.ex. **PROCEDURE**.
2. Standardtyper i TURBO PASCAL skrivs med versaler, t.ex. **INTEGER**.
3. Identifierare skrivs normalt med små bokstäver (ev. versal första bokstav).
Versaler används för att underlätta läsning av sammansatta namn, t.ex. **WriteLn**.
4. Satsparentesen **BEGIN..END** används i **IF..THEN..ELSE**, **FOR..DO**, **WHILE..DO** och **WITH..DO**-satser normalt så att **END** blir en *slutmarkering* på satsen.
I en kommentar efter **END** upprepas satstypen. Ex.

```
IF .. THEN BEGIN
  xxxx;
  yyyyy
END; (* IF *)
```

```
IF .. THEN BEGIN
  xxxx;
  yyyy END
ELSE BEGIN
  zzzz;
  vvvv
END; (* IF *)
```

5. Procedurer och funktioner avslutas på samma sätt med en kommentar som upprepar namnet. Ex:

```
PROCEDURE Init(x:REAL);  
BEGIN  
  ...  
  ...  
END; (* Init *)
```

OLIKA VERSIONER AV TURBO PASCAL

Skillnaderna mellan olika versioner av TURBO PASCAL är små. Där CP/M-80 versionen exv., ändå skiljer sig från övriga versioner anges detta. Följande beteckningar används:

Z80-versionen = 8-bitarsversionen = CP/M-80 versionen.

8086/(88)-versionen = 16-bitarsversionen = CP/M-86/Concurrent DOS samt MS-DOS/PC-DOS-versionerna.

MS-DOS-versionen inkluderar PC-DOS-versionen om inget annat sägs.

ANTECKNINGAR

KAPITEL 1: INSTALLATION OCH ANVÄNDNING AV TURBO PASCAL +

Detta kapitel beskriver installationen av TURBO PASCAL + systemet samt användningen av TURBO PASCAL, i synnerhet den inbyggda editorn.

1.1 COM och CMD-filer

Under operativsystemen CP/M-80 och MS-DOS/PC-DOS anges exekverbara (körbara) filer med typbeteckningen **.COM**. Under CP/M-86 (Concurrent CP/M, Concurrent DOS) markeras de istället med **.CMD**.

I denna manual används konsekvent beteckningen **.COM**. Om programvaran körs under CP/M-86 etc. ska alla **".COM"** således läsas som **".CMD"**.

1.2 KOPIERING AV PROGRAM

Innan du använder programvaran måste du *ovillkorligen* kopiera över innehållet på originaldisketten(erna) till ett antal arbetsdisketter. Förvara i fortsättningen originaldisketten på ett betryggande sätt. Den är din säkerhet vid svåra maskinfel etc. och ditt bevis på att du är berättigad att använda svenska TURBO PASCAL +. Kopieringen utförs på det sätt som beskrivs i manualen för ditt operativsystem.

Observera att disketten(erna) till TURBO PASCAL + av copyrightskäl inte är försedda med något operativsystem. Det är lämpligt att kopiera operativsystem och de viktigaste systemprogrammen till dina TURBO PASCAL + arbetsdisketter. Se systemmanualen.

OBS! Du får endast kopiera TURBO PASCAL + för din personliga användning och som backupskydd.

1.3 FILER

Försäljningsdisketten(erna) innehåller följande filer (filer inom parentes kan saknas hos vissa versioner).

(READ.ME)	Viktig information. Skriv TYPE READ.ME.
TURBO.COM	Svenska versionen av TURBO PASCAL (editor, kompilator, länkare). Måste finnas på arbetsskivan. (ca 30k)
(TURBO.OVR)	Overlay-fil (CP/M-80). Används endast då du anropar andra program från TURBO PASCAL + . Bör finnas på arbetsskivan.
TURBO.MSG	Textfil med alla felmeddelanden från kompilatorn.
(IBMTURBO.MSG)	Bör finnas på arbetsskivan.
(ISOTURBO.MSG)	
TLIST.COM	Utskriftsprogram för programtexter. Praktiskt att ha på arbetsskivan, men behöver inte finnas där.
(INSTIBM.xxx) och	
(INSTISO.xxx)	Installation av svenska tecken MS-DOS, CP/M-86
TINST.COM	Installationsprogram för TURBO.COM. Behövs bara vid installationen.
TINST.MSG	Textfil till installationsprogrammet.
(IBMTINST.MSG)	Behövs bara vid installationen.
(ISOTINST.MSG)	
TINST.DTA	Datafil för terminaler. Behövs bara vid installationen.
(TERMINST.COM)	Program för att terminalinstallera kompilerade program. Behövs inte på arbetsskivan.
SUBPAR.LIB och	
SUB.xxx	Includefil för funktioner/procedurer som parametrar. Bra att ha på arbetsskivan.
EXT.LIB,	
EXTnnn.xxx och	
ASSIGEXT.INC	Includefiler för dynamiska anrop av externa underprogram. Bra att ha på arbetsskivan.
LIB-filer	Generella biblioteksrutiner. Bra att ha på arbetsskivan, men behövs inte.
DOS-filer	Generella biblioteksrutiner för anrop av operativsystemet (DOS).
QUICK.SRT	Includefil för QuickSort. Behövs inte på arbetsskivan.
IO-filer	Generella biblioteksrutiner för in/ut-matning. Behövs inte på arbetsskivan.
PAS,	
INC-filer	Exemplifierande programtexter, t.ex. CALC.PAS/MC.PAS som kompilerar till ett enkelt VisiCalcliknande program. TAB.PAS demonstrerar separatkompilering och dynamisk laddning av externa underprogram.

Det är fullt möjligt att arbeta med ett *endisksystem*. TURBO PASCAL tar inte in- te stor plats, så det finns gott om utrymme för egna programfiler. Detta är en fördel vid skolundervisning där resurserna ofta är begränsade.

1.4 INSTALLATION

För att TURBO PASCAL ska fungera riktigt måste systemet anpassas till den termi- nal du arbetar med. Detta görs vid installationen.

Vid installationen kan också editorns kommandon ändras för att utnyttja funktions- tangenter etc. TURBO PASCAL är förinställt på de kommandon som finns i WordStar.

Om din version av TURBO PASCAL är "bunden" till en viss dator, så har installa- tionen redan gjorts av din leverentör. Du kan då hoppa över resten av sektion 1.4.

1.4.1 INSTALLATION AV SVENSKA TECKEN (CP/M- 86,MS-DOS)

Under MS-DOS och CP/M-86 används idag två skilda system för presentation av de svenska tecknen **Å, Ä, Ö, U, å, ä, ö** och **ü**. Den svenska ISO-standarder har uteslutet vissa andra tecken (hakparentes, klammer, uppåtpil) för att få plats med de svenska tecknen. Tyvärr används de uteslutna tecknen flitigt i Pascal. IBM har istället valt att utvidga teckenkoden för att få plats med alla internationella tecken tillsammans med de nationella tecknen.

Ta reda på vilken teckenuppsättning din maskin använder (om du kan skriva både hakparenteser och 'Å','Ä' så använder din terminal troligen IBM:s kod). Installera rätt tecken på följande sätt:

MS-DOS: Skriv **INSTISO** eller **INSTIBM**.

CP/M-86: Sätt din CP/M-systemskiva i drive A: och TURBO PASCAL + i drive B:. Kopiera över systemprogrammet PIP.CMD till TURBO-skivan. Skriv sedan

SUBMIT B:INSTISO eller **SUBMIT B:INSTIBM**.

Concurrent CP/M (Concurrent DOS): Systemskivan i B: och TURBO-skivan i A:. Skriv

SUBMIT INSTISO eller **SUBMIT INSTIBM**.

Nu kan du fortsätta med installation av bildskärm.

1.4.2 INSTALLATION AV BILDSKÄRMEN

Installationsprogrammet startas med

TINST <RET>

Efter en stund visas en meny där du kan välja mellan att installera bildskärm (**B**), tangentbord (**T**) eller avsluta (**Q**).

Om du väljer **B** presenteras efter en stund alla de förinställda bildskärmsterminaler som följer med systemet. I denna lista ingår några svenska terminaler (PC-DOS-versionen: endast IBM-skärmar). Om du har en av dessa terminaler är det bara att ange terminalens nummer. Installationsprogrammet frågar nu om du vill ändra på något i terminaldefinitionen. Normalt svarar du N. Du får då en fråga om din dators klockfrekvens. Här ska du ange den klockfrekvens i MHz (heltal) som framgår av manualen till datorn. Ett vanligt värde är 4 MHz.

Därmed är installationen klar.

Om din terminal inte finns med i listan, eller om du har valt en terminal ur listan men vill ändra definitionen på några punkter, svarar du J på frågan ovan. Installationsprogrammet går då igenom alla de kontrollkoder som används för att styra presentationen på bildskärmen. Vid varje fråga kan du svara med de kontrollkoder som används på din terminal, eller om funktionen saknas, med ett "-"tecken (bindestreck).

Kontrollkoder anges antingen genom att trycka ner CTRL-tangenten tillsammans med respektive tangent. (se manualen till din terminal samt appendix M) eller genom att ange kontrollkoden som ett decimalt eller hexadecimalt tal. Hexadecimala tal ska föregås av "\$"-tecknet (dollarstecken eller "sol"-tecken).

Ex. ESC CTRL-Z ger ASCII-koderna 27 26.

27 26 och 1B 1A ger samma kod.

Om flera kontrollkoder ingår så åtskiljs de av mellanslag. Man kan inte blanda kontrollkoder och tal utan måste hålla sig till *antingen* kontrollkoder *eller* tal.

Om du enbart slår <RET> så behålls den tidigare inställningen.

"-" raderar ev. föregående inställning.

För att markera **textblock** i editorn används exv. låg intensitet, inverterad text eller annan färg (LowVideo/NormVideo). Vissa terminaler saknar dock dessa möjligheter (eller ger konstiga effekter). Markera detta med "-". Man får då lära sig att leva utan *synliga* blockmarkeringar. Blockoperationerna fungerar dock som vanligt även om blocken inte markeras på skärmen.

1.4.3 INSTALLATION AV EDITORNS KOMMANDON

Alla editorkommandon är fördefinierade i enlighet med WordStar. Innan dessa ändras bör du åtminstone två ggr överlägga med dig själv (och kollegor) om en ev. egen "enklare" eller "smartare" terminalanpassning är värt den förlorade kompatibiliteten med WordStar.

För PC-DOS versionen gäller att du kan behålla kontrollkoderna i Wordstar samtidigt som du installerar alternativa sekvenser för vissa kommandon. Detta går bra så länge du inte försöker utnyttja Wordstarkoderna dubbelt för olika syften.

Om du beslutar dig för att ändra på editorns kommandon ska du välja alternativ **T** i menyn. Installationsprogrammet frågar nu för varje kommando i editorn vilka kontrollkoder du vill använda.

Det enklaste sättet att svara på denna fråga är helt enkelt att trycka ner den tangent eller sekvens av tangenter du vill använda.

Koderna kan också anges som ascii-koder, dvs. tal. Talen kan skrivas decimalt eller hexadecimalt (föregås av `0x`) och ska åtskiljas av mellanslag.

`<RET>` behåller tidigare värden.

Ett `"-"`tecken raderar tidigare definition.

B backar ett steg.

I sektion 1.6 beskrivs alla editorkommandon. Där finns också plats för att komplettera med dina egna definitioner.

När installationen är klar så avslutar du med **Q**. Du har nu skapat din egen version av TURBO.COM. Installationsfilerna (TINST.COM, TINST.MSG, TINST.DTA) behövs inte längre på arbetsskivan (men ska naturligtvis sparas för ev. senare bruk).

Installationen kan göras om när som helst ifall du exv. byter terminal eller vill ändra editorns styrkoder.

1.5 SÅ ANVÄNDS TURBO PASCAL

TURBO PASCAL startas upp med

TURBO <RET>

varvid pascalsystemet laddas in i datorns minne.

Om du har en dator med RAM-disk (ev. enhet C: eller M:) kan du först kopiera TURBO.COM till den snabbare ramdisken med den fördelen att det går snabbt att ladda in TURBO senare ifall du går ur systemet.

På bildskärmen finns nu en presentation och en fråga:

TURBO Pascal SW A.BCX NNNNNN
(c) 1983,xxxx BORLAND Inc / DATABITEN

Serienr SW.xxxxx

Terminal: xxxxxxxxxx

Felmeddelanden (J/N)? – J <RET>

Om du svarar **J** får du förutom felnummer även felmeddelanden i klartext på svenska.

Det rekommenderas. Vid stora program kan du emellertid spara ca 1.5 kbyte minne genom att inte ta med felmeddelandena (se appendix G).

HUVUDMENYN I TURBOPASCAL

L = arbetsdisk: A
(A = aktuellt bibl:)

W = arb.fil:
M = huv.fil:

E = edit C = komp R = exekv S = spar
(X = annat program) D = bibl Q = slut O = kompilersval

Text: 0 bytes
Kvar: xxxxxx bytes

>_

”A = aktuellt bibl(iotek)” finns endast på MS-DOS och PC-DOS versionerna.

”X = annat program” finns endast på CP/M-80 versionen.

Alla kommandon nås direkt genom att trycka på tangenten för den stora bokstav som står före likhetstecknet och kommandobeskrivningen, t.ex. L för ny disk. Inget <RET> som avslutning.

1.5.1 L = arbetsdisk: X

2 används för att ange vilken disk som är arbetsdisk. Du kan ändra detta utan att gå ur TURBO PASCAL genom att trycka L och ange ny disk (A,B,...).

Om ditt operativsystem är CP/M-80 ver 2.2 är det viktigt att du använder L om du byter diskett. Så fort luckorna på flexskiveenheten öppnas spärras nämligen disketten mot skrivning, vilket gör att du senare inte kan lagra dina program på den. Spärren tas bort med L.

1.5.2 A = aktuellt bibl(iotek): X MS-DOS, PC-DOS

MS-DOS och PC-DOS 2.0 och senare kan dela upp ett diskettbibliotek i flera underbibliotek. Se respektive manual för MS-DOS och PC-DOS.

1.5.3 W = arb(ets)fil

används för att ange vilken Pascaltext du vill jobba med. Eftersom TURBO PASCAL självt håller reda på arbetsfilen kan du redan från början ge den ett passande namn. Max 8 tecken (första bokstav) därefter ev. punkt och högst 3 ytterligare tecken (filtyp). Om du bara anger tecknen *före* punkt så lägger TURBO PASCAL självt till typbeteckningen

.PAS

Denna beteckning bör normalt användas för Pascalprogram.

Om du sätter ut punkten men inget efter den så får du en fil utan typbeteckning.

Ex. på godkända namn:

```
CIRKEL (blir CIRKEL.PAS på disken)
PROG3 (PROG3.PAS)
D840326 (D840326.PAS)
SORT.LIB
TEST. (blir TEST på disken)
```

1.5.4 M = huv(ud)fil

används vid större programsystem uppbyggda av flera pascaltexter (se includefiler kapitel 18). Med M väljs huvudfil. Man kan då editera includefiler och direkt kompilera varvid kompilatorn självt ser till att huvudfilen laddas in. Detta fungerar även automatiskt vid kompileringsfel.

1.5.5 E = edit(era)

används för att anropa editorn.

1.5.6 C = komp(iler)er

kompilerar "huvudfilen" eller om sådan inte finns "arbetsfilen". Kompileringen sker till minnet om inte kompilatorn har ställts om för kompilering till disk. Vid kompilering till disk kan man välja mellan att kompilera till en **.COM-fil** eller en **.CHN-fil**. Filnamnet blir detsamma som hos huvudfilen (arbetsfilen) bortsett från filtypen. En sammanfattning av olika kompileringssätt redovisas i följande tabell:

Källkod		Kompilerat program
---------	--	--------------------

Minnet	->	Minnet
Minnet	->	COM-fil
Minnet	->	CHN-fil
Disk	->	Minnet
Disk	->	COM-fil
Disk	->	CHN-fil

En COM-fil är ett fristående program som kan köras direkt från operativsystemet. CHN-filer utnyttjas av andra TURBO-program, dels vid "chaining", dels vid dynamisk laddning av externa underprogram. Detta förklaras närmare i kapitel 19-20 samt appendix A och B.

Kompilatorn sätter ingen maxgräns för hur stora filer som kan kompileras. Textfiler som inte ryms i arbetsminnet kan delas upp en huvudfil och ett antal includefiler. Vid kompileringen anges huvudfilen. De kan också kompileras från en diskfil.

Kompileringen kan avbrytas genom att trycka på tangenten J.

1.5.7 R = exekv(era)

kör det kompilerade programmet i minnet eller om kompileringssvalet är inställt på kompilering till disk så körs en .COM-fil.

(På 8086/88-versionerna måste man dock avsluta TURBO PASCAL och gå tillbaka till operativsystemet för att köra COM-filer. R kan endast användas vid program som har kompilerats till minnet).

Om programmet inte är kompilerat så görs detta först.

R kan alltså användas för att snabbt kompilera program (om detta inte har gjorts) och direkt exekvera programmet.

1.5.8 S = spar(a)

sparar programmet i minnet på en diskfil (OBS! bara programtexten sparas, inte det kompilerade programmet). Om det finns en äldre version av denna fil så döps den om till .BAK och den nya versionen ersätter den gamla versionen. Innan du provkör program bör du spara programtexten. Vid fel i programmet kan du kanske hamna i en loop som inte går att bryta utan att starta upp datorn på nytt (ifall du inte ha satt kompileringdirektivet U+ som gör att du kan bryta körningen med CTRL-C. Se appendix E).

1.5.9 X = annat program: CP/M-80

gör att du kan köra ett annat program utan att lämna TURBO PASCAL. T.ex. X TLIST <RET> anropar utskriftsprogrammet TLIST.COM och återgår sedan till TURBO PASCAL. Alla inställningar och textfiler du jobbade med före anropet av det andra programmet återfås. Kommandot fungerar för närvarande bara för CP/M-80 versionen av TURBO PASCAL.

1.5.10 D = bibl(iotek)

visar filerna på aktuell skiva (om inte en drivebeteckning anges t.ex. B:).

Urval av filer kan göras med tecknen * och ?, enligt samma principer som hos CP/M och MS-DOS.

Ex. *.PAS visar alla filer med typbeteckning .PAS.

1.5.11 Q = slut

avbryter TURBO PASCAL och återgår till operativsystemet. Om du har editerat en fil men inte sparat den till disk så frågar TURBO PASCAL först om du vill spara texten.

sn1.5.12 O = kompileringsval

(bokstaven O) används bl.a. för att ställa om kompileringen från minnet till disken. Se appendix A och B.

1.5.13 UTVECKLING AV ETT PASCALPROGRAM

1. Ange arbetsfil: T.ex. **W TEST <RET>** (ger TEST.PAS på skivan)
2. Anropa editorn: **E**
3. Editera pascaltexten
4. Gå ur editorn **CTRL-K D**
5. Spara TEST.PAS: **S**
6. Kompilera **C**

Vid fel ges feltext och uppmaning att trycka <ESC>.

När du gör detta går TURBO PASCAL direkt tillbaka till editorn och ställer markören vid felkällan. Rätta och upprepa från steg 3.

Om du vill ha en .COM-fil ska kompileringen först ändras från minnet till COM-fil. Görts med O och därefter C. Avsluta med Q.

7. Kör programmet: R

Vid fel under körning reagerar TURBO PASCAL på samma sätt som vid kompileringsfel. Med <ESC> går du alltså direkt in i editorn och kan rätta. Börja om från steg 3.

1.5.14 UTSKRIFT PÅ PRINTER

Pascaltexterna kan skrivas ut på printer med programmet **TLIST.COM**. Utskriften kan formateras med en serie **punktkommandon** för papperslängd, sidbrytning, rubrikhuvud etc. (se appendix N). TLIST startas direkt från operativsystemet med kommandot

TLIST

eller från TURBO PASCAL (endast CP/M-80 versionen) med menyval **X** för "Annat program".

1.5.15 EXEMPEL: CALC/MC

På en av distributionsdisketterna finns källkoden till programmet **CALC** (8086/88) eller **MC** (Z80/CP/M-80), ett kalkylprogram i stil med VisiCalc. Källkoden består av en huvudfil (CALC.PAS eller MC.PAS), samt för CP/M-80-versionen ett antal "include"-filer (MCxxx.INC). Dessutom ingår en hjälpfil (CALC.HLP/MC.HLP) till det färdiga programmet, samt ett exempel att provköra (CALCDEMO.MCS/MCDEMO.MCS).

Programmet är ett mycket lämpligt studieobjekt för hur man kan skriva stora program med TURBO PASCAL. CALC.PAS/MC.PAS ger också många inblickar i hur ett kalkylprogram byggs upp. Programmet kan du naturligtvis själv bygga vidare på om du vill.

Så här gör du för att kompilera programmet till en .COM-fil.

1. Ladda in TURBO och välj alternativ **M**:
Ange som huvudfil **CALC/MC** varvid CALC.PAS/MC.PAS laddas.
 2. Välj kompilersalternativ med **O**. (*)
Tryck **C** för kompilering till "COM-fil".
Gå tillbaka till huvudmenyn med **Q**.
 3. Starta kompileringen med **C**.
 4. Starta CALC.COM/MC.COM med **R**.
Du kan få hjälp genom att trycka **?**.
Kommandon föregås av **/**.
 5. **MC.COM/CALC.COM** kan också startas direkt från operativsystemet med **CALC/MC <RET>**.
-

*: Om du arbetar under CP/M-86, MS-DOS eller PC-DOS och har tillräckligt med minne (128 k) så kan CALC.PAS kompileras direkt till minnet.

1.6 EDITORN I TURBOPASCAL

Den text som editeras finns hela tiden i datorns minne. I Z80-versionen kan man arbeta med texter på maximalt ca 25 000 tecken.

8086/8088-versionen klarar textfiler på mer än 60 000 tecken. Större filer delas upp i en huvudfil och ett antal includefiler.

Vid editering ger översta raden information om markörens aktuella rad samt kolumn. Dessutom anges om editorn är i "Skjut in"-mod eller ej samt om den automatiska "Tab"-funktion är påkopplad eller ej, samt slutligen disk och namn på editerad fil.

I "Skjut in"-mod kommer texten som skrivs in att skjuta tidigare text framför sig. Annars skriver ny text över gammal text.

I "Tab"-mod kommer markören vid ny rad att hoppa fram till kolumnen rakt under det första tecknet på raden ovanför. I annat fall ställer sig markören vid radens början.

I följande avsnitt beskrivs alla styrkommandon hos editorn. Utrymme har lämnats för att du ska kunna fylla i de egna tangenter/tangentsekvenser som du ev. har valt. Som standard anges de förinställda WordStarsekvenserna.

Vissa kommandon är **dubbelkommandon**, t.ex. **CTRL-K D**. Det sista tecknet kan slås med eller utan CTRL-tangenten nedtryckt. Alla kombinationer nedan är alltså tillåtna och ger samma resultat:

CTRL-K D
CTRL-K CTRL-D
CTRL-K d
CTRL-K CTRL-d
CTRL-k D
CTRL-k CTRL-D
CTRL-k d
CTRL-k CTRL-d

1.6.1 AVSLUTA EDITERING

_____ **CTRL-K CTRL-D** avslutar editering och gör att du återvänder till huvudmenyn i TURBO PASCAL. Tryck ett extra <RET> för att ta fram menyn.

1.6.2 STYRNING AV MARKÖREN

_____ **CTRL-E** uppåt ett steg

_____ **CTRL-S** vänster ett steg

_____ **CTRL-D** höger ett steg

_____ **CTRL-X** nedåt ett steg

_____ **CTRL-A** vänster ett ord

_____ **CTRL-F** höger ett ord

_____ **CTRL-Q E** till skärmens första rad.

_____ **CTRL-Q X** till skärmens sista rad.

_____ **--> el. CTRL-I** tabulator - flyttar till de positioner som markeras av orden på raden ovanför.

_____ **CTRL-W** rulla bilden uppåt en rad. Markören flyttas inte.

_____ **CTRL-Z** rulla bilden nedåt en rad. Markören flyttas inte.

_____ **CTRL-R** uppåt en bildskärmsida

_____ **CTRL-C** nedåt en bildskärmsida

_____ **CTRL-Q R** början av texten

_____ **CTRL-Q C** slutet av texten

_____ **CTRL-Q S** början av raden

_____ **CTRL-Q D** slutet av raden

_____ **CTRL-Q B** början av markerat block

_____ **CTRL-Q K** slutet av markerat block

_____ CTRL-Q P till markörens föregående position

1.6.3 INSÄTTNING OCH RADERING AV TEXT

_____ CTRL-V Skiftar mellan "Skjut in"-mod och "skriv över"-mod.

_____ <-- el. CTRL < radera tecknet till vänster om markören

_____ CTRL-G radera tecknet till höger om markören

_____ CTRL-T radera ordet framför markören

_____ CTRL-Y radera hela raden som markören står på (VARNING!!!)

_____ CTRL-Q Y radera hela raden från markören och ut till höger.

_____ CTRL-N Skjut in en tom rad

1.6.4 BLOCKKOMMANDON

_____ CTRL-K B markera början på block
(Ett block markeras med låg intensitet, avvikande färg eller inverterad text, beroende på terminal. På vissa terminaler som inte kan ändra textpresentation är blocket osynligt).

_____ CTRL-K K markera slut på block

_____ CTRL-K T markera ett ord som block (det ord markören står på)

_____ CTRL-K H skifta mellan "dölj" block och "visa" block.

_____ CTRL-K C kopiera block till markörens position (originalet finns kvar)

_____ CTRL-K V flytta block till markörens position.

_____ CTRL-K Y radera block (VARNING!!!)

_____ CTRL-K R läs block från diskfil och skjut in texten med början vid markörens position.

_____ CTRL-K W skriv markerat block till en diskfil.

1.6.5 ÖVRIGA KOMMANDON

_____ CTRL-K D	Avsluta editering
_____ CTRL-Q I	Växlar mellan att sätta på och stänga av automatisk tabulering.
_____ CTRL-Q L	Återställ raden ("Ångerknapp") Kan rätta tillbaka den rad som markören står på ifall den har ändrats felaktigt. Men om du har tryckt "radera rad" så är texten tyvärr ohjälpligt förlorad.
_____ CTRL-Q F	sök (finn) text (ange söktext)
_____ CTRL-Q A	sök (finn) och byt ut text

Sökning börjar normalt från markörens position och framåt i texten.

Sökningar/sök-byt ut kan preciseras med ett eller flera av följande tillägg:

- B:** sök baklänges i texten
- G:** "global" sökning, dvs. sök från början av texten.
- n:** n = valfritt tal. **Sökning:** n:te förekomsten av söksträngen.
Sök-byt ut: Byter ut n ggr.
- N:** Endast vid Sök-byt ut: Byt utan att fråga.
- U:** ignorera små/stora bokstäver.
- W:** sök bara efter hela ord.

Användbar kombination:

GUW ger automatisk sökning av hela filen på hela ord och valfritt stora-små tecken.

Enbart <RET> ger inga finesser.

Vid frågan om söktext, byt-uttex samt tillägg kan valet vid föregående sökning/sök-byt ut upprepas genom att trycka **CTRL-F** (nästa ord).

_____ CTRL-L	upprepa sökning/sök-byt ut text
_____ CTRL-P	Prefix för att skriva kontrolltecken i textfilen. Kontrolltecken visas inverterade eller med låg intensitet på många terminaler.
_____ CTRL-U	Avbryt pågående kommando.

Ett tips:

Blockmarkeringar (**CTRL-K B** och **CTRL-K K**) kan också användas som markering av ett textavsnitt som man snabbt vill hitta tillbaka till. Du kanske måste förflytta dig till början av programtexten för att lägga till en variabeldeklaration:

Markera platsen du befinner dig med **CTRL-K B**.
Gå till textens början med **CTRL-Q R**.
Skriv till deklarationen.
Gå tillbaka till tidigare position med **CTRL-Q B**.

Om du vill testkompilera och sedan gå tillbaka till den tidigare positionen i texten kan du efter kompileringen trycka **CTRL-Q P**.

1.7 KOMPILERINGSDIREKTIV

Kompilatorn i TURBO PASCAL kan ställas in med en serie kompileringsdirektiv (se appendix E). Under programutveckling bör du alltid använda inställningen:

```
(*$R+,U+*),
```

som skrivs först i ditt program.

R+ betyder att kompilatorn kontrollerar att fältindex och variabler av egen-
definierade typer håller sig inom deklarerade gränser.

U+ gör det möjligt att bryta programmet när- och var som helst, även om du skulle råka hamna i en evig loop på grund av ett programmeringsfel.

Kontrollerna sänker exekveringshastigheten avsevärt, men vänta med att ta bort dem tills programvaran är fullt uttestad.

1.8 SVENSKA TECKEN

Den svenska teckenstandarden skiljer sig på några punkter från den internationella ISO-standard (ASCII). De skillnader som är av betydelse vid Pascal-programmering redovisas i kapitel 2.

ANTECKNINGAR

KAPITEL 2: SPRÅKELEMENT

2.1 TECKENUPPSÄTTNING

Följande teckenuppsättning används i TURBO PASCAL:

Bokstäver:	A till Z, a till z samt <u>—</u> (understreck)
Siffror:	0 till 9
Blanktecken:	
Specialtecken:	+ - * / = ↑ < > () [] { } . , ; ' # \$ @

Av dessa tecken hör följande *inte* till standardpascal:

— # \$ @

I Pascal görs ingen skillnad mellan stora och små bokstäver.

2.2 SAMMANSATTA SYMBOLER

TURBO PASCAL använder samma sammansatta **symboler** som standardpascal, dvs:

Tilldelningsoperatör:	:=
Relationsoperatorerna:	<> >= <=
Delintervallmarkering:	..
Klammer:	(. .) istället för [och] (tillåtet i ISO, ej Jensen-Wirth)
Kommentar:	(* *) istället för { och } (ISO, Jensen-Wirth)

2.3 AVVIKELSER I SVENSK TECKENSTANDARD

Den *svenska* teckenstandarden skiljer sig på några punkter från den internationella ISO-standarden (ASCII). De skillnader som är av betydelse vid Pascalprogrammering visas i följande tabell:

ISO-standard (ASCII)	Svensk standard	Alternativ	
[Ä	(.	TURBO/ISO-Pascal
]	Å	.)	TURBO/ISO-Pascal
{	ä	(*	TURBO/Wirth-Jensen/ISO
}	å	*)	TURBO/Wirth-Jensen/ISO
↑	Û	@	ISO (ej TURBO PASCAL!)

Vissa datorer (t.ex. IBM PC/XT) och skrivare har möjlighet att återge både de internationella tecknen och de svenska motsvarigheterna samtidigt, men de flesta datorer visar endast svenska tecken.

Detta gör att hakparenteser i Pascal (vid fält och mängder) eller klammerparenteser (kommentarer) måste bytas mot motsvarande svenska tecken. Som alternativ till **ÄÅ** vid hakparenteser kan (. .) användas (saknas i Jensen-Wirth men tillåtet i ISO-standardern).

Som alternativ till **ää** rekommenderas (* *). Klammrar bör överhuvudtaget endast användas som ett verktyg vid testning av program, då programdelar (inklusive kommentarer) kan behöva stängas av genom att placeras inom en kommentar. (I Modula-2 används *endast* (* *) för kommentarer).

Se upp med svenska tecken i identifierare! Kompilatorn upptäcker felaktig användning av **Ä, Å, ä** och **Û** men kan missuppfatta **ä** som då tolkas som en inledande klammerparentes, dvs. inledningen på en kommentarsats. Resten av programmet ses då som en kommentar, vilket ger det något kryptiska felmeddelandet

'91, Öväntat slut på källfilen'.

Vissa skrivare (t.ex. Epson) kan mjukvarumässigt ställas om till utskrift av internationella tecken med hjälp av styrkoder från datorn. Dessa koder kan då läggas direkt i en kommentar i Pascaltextern. Se appendix N.

En del svenska datorer saknar <ESC>-tangent (t.ex. ABC-maskinerna). Samma funktion fås med

CTRL-Ä.

2.4 RESERVERADE ORD

TURBO PASCAL har följande **reserverade ord**:

* ABSOLUTE	AND	ARRAY	BEGIN
CASE	CONST	DIV	DO
DOWNTO	ELSE	END	*EXTERNAL
FILE	FOR	FORWARD	FUNCTION
GOTO	IF	IN	*INLINE
LABEL	MOD	NIL	NOT
OF	OR	*OVERLAY	PACKED
PROCEDURE	PROGRAM	RECORD	REPEAT
SET	*SHL	*SHR	*STRING
THEN	TO	TYPE	UNTIL
VAR	WHILE	WITH	*XOR

De med * markerade orden är tillägg i TURBO PASCAL som saknas i standard-pascal.

I den svenska manualen skrivs alla reserverade ord med versaler.

2.5 STANDARDIDENTIFIERARE

Standardidentifierarna överensstämmer med Jensen-Wirth och ISO-Pascal med några få undantag och flera tillägg:

Följande identifierare saknas:

Get Pack Page Put Unpack (se appendix F)

Flera identifierare har tillkommit. Se kapitel 16 och appendix C.

2.6 PROGRAMRADER

En **programrad** får omfatta max **127 tecken**. Editorn skjuter automatiskt in en radslutmarkering vid för långa rader.

ANTECKNINGAR

KAPITEL 3: SKALÄRA STANDARDTYPER

TURBO PASCAL tillåter alla skalära standardtyper i Pascal samt en egen skalär typ, kallad **BYTE**.

Följande data och begränsningar gäller för TURBO PASCAL:

INTEGER	Tillåtet intervall -32768..32767 . Upptar 2 bytes av minnet. Observera att " <i>overflow</i> " vid beräkningar med heltal passerar oupptäckta. T.ex. 5*10000 DIV 2000 ger inte svaret 25, eftersom man får overflow vid beräkning av den första multiplikationen.
BYTE	BYTE är ett delintervall av INTEGER, 0..255 . Upptar 1 byte av minnet. BYTE-variabler är kompatibla med INTEGER-variabler.
REAL	1) 16-bitars 8087-versioner: Tillåtet intervall är 4.19E-307..1.67E + 308 , med en mantissa på upp till 16 signifikanta siffror. Upptar 8 bytes av minnet. 2) Z80 och 16-bitars versioner utan 8087: Tillåtet intervall är 1E-38..1E + 38 , med en mantissa på upp till 11 signifikanta siffror. Upptar 6 bytes av minnet. Vid <i>för stora</i> reella tal (absolutvärdet) avbryts exekveringen och man får ett felmeddelande om detta (se appendix H). Vid <i>för små</i> reella tal (absolutvärdet) fås resultatet 0. Observera de skilda formaten för 8087-versionen och övriga versioner.
BOOLEAN	TRUE eller FALSE. TRUE < FALSE. BOOLEAN upptar <i>alltid</i> 1 byte av minnet.
CHAR	antar tecken enligt 8 bitars ISO-kod. Ordningstalet får gå mellan 0..255 . Upptar 1 byte av minnet.

I den svenska manualen skriver vi de skalära standardtyperna med versaler (i enlighet med Modula-2).

ANTECKNINGAR

KAPITEL 4: EGENDEFINIERADE SPRÅKELEMENT

4.1 IDENTIFIERARE

Identifierare används för att namnge **labels** (etiketter), **konstanter**, **datatyper**, **variabler**, **procedurer** och **funktioner**.

I Pascal börjar identifierare alltid med en **bokstav** och åtföljs av en godtycklig kombination av **bokstäver** och **siffror**. I TURBO PASCAL får dessutom "—" (understreck) användas istället för bokstäver.

Identifierare är signifikanta till hela sin längd (dock ej längre än en hel rad d.v.s. 127 tecken).

Ex.

Kuben, sortera__lista

men *ej*

3World	får ej börja på siffra
sortera lista	inga blanktecken i identifierare

I Pascal skiljer man inte på stora och små bokstäver vid kompileringen varför exv. SORTERALISTA, sorteralista och SorteraLista betraktas som samma identifierare (till skillnad från t.ex. Modula-2). Vi rekommenderar dock att skriva sammansatta identifierare i blandad stil, i exemplet alltså

SorteraLista

vilket gör programmet mer lättläst.

Se upp med de svenska tecknen **Å, Ä, Ö, U, å, ä, ö, och ü** i identifierare. **ä** passerar som klammerparentes, dvs. kompilatorn tror att en kommentar påbörjas.

Ex:

Stjärna

kommer av kompilatorn att tolkas som

Stjärna.

4.2 TAL

I Pascal kan tal vara **heltalskonstanter** eller **flyttalskonstanter**.

Förutom vanliga decimala heltal tillåter TURBO PASCAL även **hexadecimala heltal**.

Hexadecimala tal inleds med tecknet '\$' (dollarstecken eller "sol"-tecken). Ex:

Decimaltalet 243 = \$F3.

Heltalsområdet -32768..32767 motsvarar det hexadecimala talområdet \$0000..\$FFFF.

Reella tal (flyttal) skrivs med decimalpunkt, dvs "3" betecknar ett heltal medan "3." betecknar ett reellt tal. Stora och små reella tal skrivs med tiopotenser, som i Pascal betecknas med bokstaven "E", t.ex.

3.24E17, -1.9E-4

4.3 STRÄNGAR

En **strängkonstant** skrivs inom **apostrofer**, t.ex. 'Pelle'. Om apostrofstecknet självt ska ingå i strängen skrivs det som två på varandra följande apostrofer.

Ex:

'O''brien'.

En sträng är kompatibel med en ARRAY OF CHAR av *samma längd*, samt med **alla** STRING-typer.

Strängar som innehåller endast *1 tecken* är kompatibla med CHAR.

Strängar utan tecken, dvs. "", är endast kompatibla med STRING-typer.

4.3.1 KONTROLLTECKEN

TURBO PASCAL tillåter att kontrolltecken (dvs tecken med ISO-värden under 32) bakas in i strängar. Detta kan göras på två olika sätt:

- 1) med nummersymbolen # åtföljt av en heltalskonstant inom intervallet 0..255.
- 2) med symbolen ↑ åtföljt av ett tecken.

Ex:

# 7	ASCII 7 (bel)
↑G	CTRL-G = ASCII 7 (bel)

Sekvenser av kontrolltecken kan läggas ihop till strängar genom att skriva dem utan avskiljande tecken. Ex:

```
# 10 # 13
↑G↑G↑G
```

Kontrolltecken kan också blandas med textsträngar. Ex:

```
'
Tiden är ute. "↑G↑G↑G ' Svara nu!'
```

4.4 KOMMENTARER

Klammerparenteser (ää på svenska skärmar) eller (* *) kan användas. Klammerparenteser kan inte skrivas inom klammerparenteser och (* *) kan inte skrivas inom (* *).

Däremot kan klammerparenteser skrivas inom (* *) och vice versa.

Detta är praktiskt vid programutveckling när man vill "kommentera bort" hela programavsnitt, inklusive de kommentarer som ingår. Det kan gälla program som har maskinspecifika avsnitt, vid felsökning, etc.

4.5 KOMPILERINGS DIREKTIV

Kompilatorn i TURBO PASCAL kan kontrolleras med ett antal kompileringsdirektiv, skrivna inom kommentarparenteser.

Ett kompileringsdirektiv börjar med tecknet \$, *direkt* åtföljt av minst ett direktiv, t.ex.:

```
(* $U + , R + *)
```

Se appendix E för en fullständig beskrivning av kompileringsdirektiven.

ANTECKNINGAR

KAPITEL 5: PROGRAMHUVUD OCH PROGRAMBLOCK

Ett pascalprogram består av ett **programhuvud** och ett **programblock**. Programblocket består i sin tur av en **deklarationsdel** och en **exekveringsdel**.

5.1 PROGRAMHUVUD

Programhuvudet behöver inte skrivas ut i TURBO PASCAL och tolkas inte av kompilatorn. Huvudet är emellertid av värde för att

- 1) ge programmet en **rubrik** och
- 2) ge information om **in-** och **utmatning** till och från programmet.

Programhuvudet måste också finnas för att garantera portabel källkod med pascal-system enligt Jensen-Wirth och ISO-standard. Ex.

```
PROGRAM test;  
PROGRAM cirkel(input,output);
```

Identifieraren för programhuvudet *behöver* inte vara densamma som filnamnet på programtexten.

5.2 DEKLARATIONSDELEN

I **deklarationsdelen** definieras labels, konstanter, typer, variabler samt funktioner och procedurer.

Enligt standardpascal kan de enskilda deklARATIONERNA endast förekomma *en* eller *ingen* gång och endast i följande **ordning**:

```
LABEL  
CONST  
TYPE  
VAR  
FUNCTION/PROCEDURE
```

I **TURBO PASCAL** får varje deklARATIONSRUBRIK förekomma hur många gånger som helst och i vilken ordning som helst!

Detta gör det möjligt att exv. bygga upp biblioteksrutiner (includefiler) som definierar egna **globala** konstanter, datatyper och variabler, och som inte nödvändigtvis måste placeras först i det program som utnyttjar dem. Detta är en kraftfull förstärkning gentemot standardpascal (och helt i linje med Modula-2).

5.2.1 LABELDEKLARATIONEN

Medan standardpascal endast tillåter labels med högst 4-siffriga heltal, tillåter TURBO PASCAL även **identifierare** som labels. Ex:

```
LABEL 10, 999, RETURN, error;
```

Etiketten **RETURN** kan användas som en tydlig markering av uthopp ur underprogram (procedurer och funktioner). Ex:

```
PROCEDURE calc;
LABEL RETURN;
...
BEGIN
...
IF IOrresult <> 0 THEN GOTO RETURN;
...
RETURN: END; (* calc *)
```

(Vissa Pascaldialekter kan avbryta underprogram med **EXIT**. I Modula-2 används **RETURN** för att bryta procedurer).

5.2.2 KONSTANTDEKLARATIONEN

Heltal, reella tal, tecken och teckensträngar kan deklarerars som konstanter.

```
CONST eps = 1.E-5;
      blank = ' ';
      operativsystem = 'MS-DOS';
```

TURBO PASCAL tillåter även kontrolltecken i konstantsträngar. Ex:

```
CONST bel = †G; (kan också deklarerars bel = †7;)
```

I likhet med Jensen-Wirth och ISO-Pascal är konstantuttryck *inte tillåtna* i TURBO PASCAL. Ex:

```
CONST AntalRader = 24;  
      MaxRad = AntalRader - 1; (ej tillåtet)  
      MaxRad = 23;           (korrekt)
```

I TURBO PASCAL kan konstanter också deklarerars med datatyp, sk. **typdeklarerade konstanter**. Se kapitel 13.

I TURBO PASCAL finns ett antal **fördeklarade konstanter**:

FALSE, **TRUE** och **MaxInt** enligt standard.
pi är ett tillägg i TURBO PASCAL. Se appendix C.

I PC-DOS-versionen finns ytterligare fördeklarade konstanter för **färg** mm.
Se appendix J.

5.2.3 TYPDEKLARATIONEN

TURBO PASCAL följer helt standardpascal. Ex på typdeklaration:

```
TYPE  CARDINAL = 0..$FFFF;  
      BITSET  = SET OF 0..7;
```

5.2.4 VARIABELDEKLARATIONEN

TURBO PASCAL följer helt standardpascal med ett tillägg. Alla tecken (dvs. upp till 127) är signifikanta (i enlighet med ISO-standarden). Kompilatorn skiljer däremot inte mellan stora och små bokstäver. Ex:

```
VAR  InquireWorkstation1:INTEGER;  
      InquireWorkstation2:INTEGER;      olika identifierare - alla tecken  
                                          signifikanta.  
      INQUIREWORKSTATION1:INTEGER;    Dubbeltydig. Samma variabel som ovan.  
                                          Ger kompileringsfel.
```

TURBO PASCAL tillåter också att variabler **ABSOLUTE**-deklarerats till

- 1) absoluta minnesadresser eller
- 2) till samma adresser som tidigare deklarerade variabler och parametrar.

Ex:

```
VAR s:STRING[80];
    Len: BYTE ABSOLUTE s;
```

Len överlagrar första längdbyten i s och har alltså samma värde som fås med funktionsanropet length(s).

5.2.5 PROCEDUR- OCH FUNKTIONSDEKLARATIONERNA

TURBO PASCAL följer standardpascal då det gäller procedur- och funktionsdeklARATIONER med följande tillägg:

1) Funktioner/procedurer kan deklaras **EXTERNAL**. Ex:

```
PROCEDURE Draw(x1,y1,x2,y2:INTEGER);EXTERNAL DrawAdr;(* CP/M-80 *)
PROCEDURE Draw(x1,y1,x2,y2:INTEGER);EXTERNAL 'DRAW';(* 8086/8088 *)
```

2) Funktioner/procedurer kan deklaras som **OVERLAY**. Ex:

```
OVERLAY PROCEDURE Menu(nr,choice:INTEGER);
```

Funktioner/Procedurer kan **FORWARD**-deklaras i enlighet med standardpascal.

Ex:

```
FUNCTION tan(x:REAL):REAL; FORWARD;
```

5.3 EXEKVERINGSDELEN

Exekveringsdelen är programblockets sista del. Programblocket inleds av det reserverade ordet

BEGIN

och avslutas med

END.

Mellan BEGIN och END finns en lista av **programsatser** åtskilda av ;. Listan kan vara tom.

