

CMPAS/R0125/MAN/1140

C O M P A S
Common Pascal Language System
Version 2.1

B R U G E R M A N U A L

Mangfoldiggørelse af denne manual - helt eller delvis - er ifølge loven om ophavsret forbudt, hvis der ikke foreligger en skriftlig tilladelse fra Poly-Data microcenter ApS. Dette gælder enhver form for mangfoldiggørelse, f.eks. fotokopiering, eftertryk, mikrofotografering, etc.

Copyright (C) 1982,1983
Poly-Data microcenter ApS
Strandboulevarden 63
2100 København Ø

9

0,

0,

0,

INDHOLDSFORTEGNELSE

0	Indledning	2
	0.1 Systemkrav	3
	0.2 Distributionsdisketten	3
	0.3 Manualens notationer	3
1	Opstart og kommandolinier	4
	1.1 Opstart af COMPAS	4
	1.2 Kommandolinier	4
2	Ind- og udlæsning af kildetekster	6
	2.1 LOAD kommandoen	6
	2.2 SAVE kommandoen	6
3	Editoren	7
	3.1 MOVE mode	8
	3.2 EDIT mode	10
	3.3 Editorens fejlmeddelelser	10
4	Compileren	12
	4.1 COMPILE kommandoen	12
	4.2 RUN kommandoen	13
	4.3 PROGRAM kommandoen	13
	4.4 OBJECT kommandoen	14
	4.5 FIND kommandoen	14
	4.6 WHERE kommandoen	15
5	Andre kommandoer	16
	5.1 DIR kommandoen	16
	5.2 NEW kommandoen	16
	5.3 MEMORY kommandoen	16
	5.4 ZAP kommandoen	17
	5.5 QUIT kommandoen	17
A	Systemkonfigurering	18

Kapitel 0

Indledning

COMPAS er et dansk udviklet programudviklingssystem baseret på det blokstrukturerede programmeringssprog Pascal. COMPAS kan køres på alle mikrodatamater med Z-80 processor og CP/M operativsystem.

COMPAS har alle de faciliteter, der behøves for editering, oversættelse, og kørsel af programmer skrevet i Pascal. Systemet består af en run-time programdel, en skærmorienteret editor og en Pascal compiler, og fylder ialt kun 22K bytes.

COMPAS Pascal følger nøje definitionen af Standard Pascal, der gives af K. Jensen og N. Wirth i bogen "Pascal User Manual and Report". I COMPAS Pascal findes desuden en del udvidelser, der yderligere forbedrer sproget.

Denne manual beskriver opstart og brug af COMPAS systemet. I programmerings spørgsmål henvises der til COMPAS Pascal programmeringsmanualen. I manualen refereres der løbende til navnene CP/M, Z-80 og COMPAS Pascal. Disse er registrerede varenavne for henholdsvis Digital Research, Zilog Inc. og Poly-Data microcenter ApS.

COMPAS Pascal systemet og den tilhørende dokumentation er forfattet af Anders Hejlsberg.

0.1 Systemkrav

De følgende krav må være opfyldt af dit computersystem før du kan bruge COMPAS:

- o Z-80 microprocessor.
- o CP/M 2.2 (eller senere) operativsystem.
- o Mindst en diskettestation.
- o Mindst 32K bytes RAM lager.

Bemærk, at COMPAS Pascal ikke kan anvendes på systemer med en 8080 eller en 8085 processor.

0.2 Distributionsdisketten

De følgende filer findes på distributionsdisketten:

COMPAS.COM Denne fil indeholder selve COMPAS systemet, dvs. run-time programdelen, den skærmorienterede editor, og Pascal compileren.

COMPAS.ERM Denne fil indeholder de fejlmeddelelser, der bruges af compileren, når den rapporterer fejl under oversættelser.

CONFIG.COM Dette program bruges til at indlægge systemafhængige parametre i COMPAS. Hvis COMPAS er leveret som en ukonfigureret version, skal dette program køres før systemet tages i brug. CONFIG kan desuden anvendes til at rette enkeltparametre i en præ-konfigureret version.

CONFIG.PAS Kildeteksten til CONFIG.COM.

Desuden kan distributionsdisketten eventuelt indeholde en række demonstrationsprogrammer i tekstform (altså som '.PAS' filer).

0.3 Manualens notationer

Betegnelsen "auto-diskette" bruges om den diskettestation, der vælges, hvis intet andet angives (CP/M viser auto-diskettens identifikator, når systemet er klar til at modtage en kommando. For eksempel angiver 'A', at A er auto-disketten).

Betegnelsen "filnavn" angiver et CP/M filnavn. Det generelle format af CP/M filnavne er:

.<disk>:<navn>.<type>

hvor <disk> er diskettens identifikator (A-P), <navn> er filnavnet (op til otte bogstaver eller talcifre), og <type> er filtypen (op til tre bogstaver eller talcifre). Hvis <disk> identifikatoren og det efterfølgende kolon udelades, vælges auto-disketten. Hvis <type> feltet og det foranstående punktum udelades, vælger systemet '.PAS' for kildetekstfiler, '.COM' for programfiler og '.OBJ' for objektfiler.

Hexal skrives med et '\$' tegn foran, for eksempel \$16E0.

Kapitel 1

Opstart og kommandolinier

1.1 Opstart af COMPAS

COMPAS opstartes fra CP/M med kommandolinien:

COMPAS

COMPAS bør altid startes fra auto-disketten. Hvis COMPAS systemet for eksempel er lagret på diskette B, så giv kommandoen 'B:' før COMPAS startes. Ved opstart udskriver systemet:

```
COMPAS V2.XY (CP/M Z-80 version)
Copyright (C) 1982,1983 by
Poly-Data microcenter ApS
Include error messages (Y/N)?
```

hvor XY er versionsnummeret. Tast 'Y' for at indlæse COMPAS.ERM filen eller ethvert andet tegn for at undgå indlæsningen. Hvis COMPAS.ERM indlæses, vil compileren udskrive en fejlmeddelelse, når den finder en fejl under en oversættelse. I modsat fald udskrives kun fejlens nummer (fejlteksterne findes i appendix H i COMPAS Pascal programmeringsmanualen).

Efter en koldstart (som beskrevet ovenfor) kan COMPAS varmstartes fra CP/M med kommandolinien:

COMPAS *

Dette kræver naturligvis, at der ikke i mellemtiden er kørt programmer, der har overskrevet dele af arbejdslageret.

1.2 Kommandolinier

Når COMPAS er klar til at modtage og udføre en kommandolinie, udskrives en dobbelt vinkel ('>>'). Under indtastning af kommandolinier kan de følgende editeringsnøgler anvendes:

BACKSPACE	Sletter det sidst indtastede tegn. Denne funktion findes normalt på tastaturet som BS, BACKSPACE eller en venstrepil, men den kan altid genereres ved at taste CTRL/H.
DEL	Samme som BACKSPACE. På de fleste tastaturer findes denne funktion som DEL eller RUBOUT.
CTRL/X	Sletter hele indtastningen.
RETURN	Afslutter indtastningen. Denne funktion findes normalt på tastaturet som RETURN, ENTER eller NEWLINE.

De nedenfor viste kommandoer genkendes af COMPAS (hver enkelt kommando beskrives fuldt ud i de følgende kapitler). Alle kommandoer kan forkortes til deres første bogstav.

LOAD	Læs kildetekst fra disk.
SAVE	Gem kildetekst på disk.
EDIT	Start editor.
COMPILE	Oversæt kildetekst og gem kode i lageret.
RUN	Start kørsel af program.
PROGRAM	Oversæt kildetekst og gem kode i programfil.
OBJECT	Oversæt kildetekst og gem kode i objektfil.
FIND	Find kørselsfejl i kildetekst.
WHERE	Find compilerfejl i include-fil.
DIR	Udskriv en diskettes bibliotek.
NEW	Aktiver nyindsat diskette.
MEMORY	Udskriv størrelse af kildetekst og arbejdslager.
ZAP	Slet kildeteksten.
QUIT	Afslut COMPAS og returner til CP/M.

Kapitel 2

Ind- og udlæsning af kildetekster

2.1 LOAD kommandoen

LOAD kommandoen bruges til at indlæse en kildetekst fra en diskfil. Kommandoliniens format er:

```
LOAD <filnavn>
```

Hvis filtypen udelades, vælger systemet automatisk '.PAS'. Når en kildetekstfil indlæses, bliver den lagt i forlængelse af den kildetekst, der allerede findes i arbejdslageret. Hvis filen ikke findes udskriver systemet:

```
No such file
```

Hvis der ikke er plads i arbejdslageret til at indlæse filen, udskriver systemet:

```
File too big
```

Den kildetekst, der allerede er i lageret, berøres ikke af en af de ovenstående fejl.

2.2 SAVE kommandoen

SAVE kommandoen bruges til at gemme kildeteksten i arbejdslageret ud i en diskfil. Kommandoliniens format er:

```
SAVE <filnavn>
```

Hvis filtypen udelades, vælger systemet automatisk '.PAS'. Filer af samme navn, som det der angives, får deres type ændret til '.BAK', før den nye fil oprettes (dette kan ændres således, at file af samme navn blot bliver slettet. Yderligere detaljer herom findes i Appendix A). Hvis diskettens bibliotek er fuldt, udskriver systemet:

```
Directory is full
```

Hvis disketten er fuld, udskriver systemet:

```
Disk is full
```

Hvis en af de ovenstående fejl rapporteres, så indsæt en anden disk, giv en NEW kommando og brug derefter SAVE igen.

Kapitel 3

Editoren

COMPAS editoren bruges til at indtaste og rette kildetekster og startes med kommandoen:

EDIT

Editoren er specielt designet til brug på en skærmterminal. Når editoren startes, vises den tekst, der ligger i lageret, på skærmen. Hvis teksten er "længere" end skærmen, hvilket den normalt er, er det kun begyndelsen af teksten, der vises. Skærmen er i realiteten et "vindue" til teksten. Hele teksten findes i lageret, og kan bearbejdes af editoren, men det er kun en del af teksten, der kan ses gennem vinduet.

Arbejds punktet, dvs. det sted i teksten der bearbejdes, angives af markøren. Markøren er altid indenfor vinduet, og hvis man prøver at flytte cursoren "udenfor" vinduet, er det i stedet vinduet, der flyttes.

Der er ingen begrænsning på linjelængder i editoren. Hvis en linie er længere en skærmens bredde, fortsætter den ud over "kanten" af skærmen og ned på den næste linie. Når en CR/LF (carriage return, line feed) sekvens mødes i teksten, startes der på en ny linie, og når et TAB tegn mødes i teksten, starter de efterfølgende tegn i den næste multiplum-af-otte position.

Da der næppe findes to tastaturer, der er helt ens, kan de tegn (eller tegnfølger), der bruges til at aktivere editorens kommandoer, variere med de forskellige versioner af COMPAS. Derfor bruges der i denne manual symbolske navne (så som <LEFT>, <UP>, <FIND>, osv.) til at beskrive editorens funktioner. Ved hjælp af tabellen i tillæget til brugermanualen kan disse symboler omsættes til egentlige taster. Man kunne selvfølgelig have valgt et fast sæt kontroltegn til styring af editoren, men så ville det ikke være muligt at nyde godt af de specialnøgler, så som markørpile, der findes på mange tastaturer.

Editoren arbejder i to forskellige modes: MOVE mode og EDIT mode. MOVE mode bruges, når der flyttes rundt i teksten, og EDIT mode når der indsættes eller slettes tegn eller linier. Ved start vælger editoren automatisk MOVE mode. <MODE> tasten bruges til at skifte mellem de to modes: Når <MODE> tastes i MOVE mode, startes EDIT mode, og når <MODE> tastes i EDIT mode, returneres der til MOVE mode.

Hvergang EDIT mode bruges til at slette tegn, bliver disse tegn gemt i en edit-buffer. Ved hjælp af <INSERT> funktionen (i MOVE mode) kan edit-bufferens indhold kopieres ind i teksten på ethvert sted. Dette er umiddelbart anvendeligt, når man fortryder, at man slettede noget, men det kan også bruges til at flytte og kopiere dele af teksten.

Når større blokke af tekst skal slettes, flyttes eller kopieres, bruges <MARK> og <ZAP> kommandoerne. <MARK> bruges til at markere en bestemt position i teksten, og <ZAP> bruges derefter til at slette alle tegn mellem markøren og denne position. Før blokken slettes, gemmer editoren dog en kopi af den i edit-bufferen. Ved hjælp af <INSERT> funktionen kan denne kopi derefter genindsættes en eller flere gange på ethvert sted i teksten.

De fleste programmører anvender normalt "indrykning" af linierne i et program for at gøre det mere overskueligt. Et almindelig anvendt princip er, for eksempel, at rykke to tegn ind for hver BEGIN/END blok i programmet. Ved hjælp af auto-tabulatoren i COMPAS editoren kan arbejdet med at lave indrykninger lettes betydeligt. Når auto-tabulatoren er aktiveret, laver editoren automatisk en indrykning hver gang en ny linie startes. Indrykningen svarer altid til indrykningen på linien ovenover, men den kan rettes med <SPACE> eller <BACKSP>.

Den øverste linie på skærmen er altid reserveret til COMPAS editorens statuslinie. Statuslinien viser løbende hvilket mode (MOVE, EDIT eller FIND), der arbejdes i, og om auto-tabulatoren er aktiv eller passiv, samt hvormange bytes der er anvendt og ledige.

3.1 MOVE mode

MOVE mode bruges, når markøren skal flyttes omkring i teksten. Ved start af editoren bliver MOVE mode automatisk valgt. De følgende funktioner kan bruges i MOVE mode:

<MODE>	Start EDIT mode.
<BACKSP>	Flyt markøren en position mod venstre.
<SPACE>	Flyt markøren en position mod højre.
<LEFT>	Flyt markøren en position mod venstre.
<RIGHT>	Flyt markøren en position mod højre.
<TAB>	Flyt markøren til den næste multiplum-af-otte position.
<RETURN>	Flyt markøren til begyndelsen af den næste linie, eller, hvis auto-tabulatoren er aktiv, til den første position, der ikke er blank.
<REMOVE>	Flyt markøren til begyndelsen af den nuværende linie, eller, hvis markøren allerede er ved begyndelsen af en linie, til begyndelsen af den forrige linie.
<AUTO>	Skift auto-tabulatorens stilling (hvis OFF, så skift til ON, og omvendt).
<UP>	Flyt markøren en linie op.
<DOWN>	Flyt markøren en linie ned.
<PREV>	Flyt markøren en side op.

- <NEXT> Flyt markøren en side ned.
- <BEGIN> Flyt markøren til begyndelsen af teksten.
- <END> Flyt markøren til slutningen af teksten.
- <FIND> Indlæs søgestreng og erstatningsstreng, og find den første forekomst af søgestrengen. Idet der tages <FIND>, slettes alle positioner efter markøren, og et spørgsmålstegn udskrives. Indtast nu søgestrengen, og afslut med et tryk på <MODE> tasten. Eventuelle fejlindtastninger kan rettes med <BACKSP> tasten, og tabulatortegn kan indsættes med <TAB> tasten. Når søgestrengen er indtastet, udskrives der endnu et spørgsmålstegn. Indtast nu erstatningsstrengen, og afslut som før med et tryk på <MODE> tasten. Teksten gennemses herefter for den første forekomst af søgestrengen. Hvis en sådan findes, bliver markøren flyttet dertil. I modsat fald bliver markøren stående. Søgningen omfatter kun teksten efter markøren. Hvis hele teksten skal gennemses, kan <BEGIN> funktionen eventuelt bruges før <FIND>.
- <CONT> Fortsæt søgning. Der søges efter den næste forekomst af den sidst indtastede søgestreng.
- <REPLACE> Erstat søgestrengen med erstatningsstrengen. Den netop fundne søgestreng udskiftes med erstatningsstrengen. Denne funktion virker kun, hvis den bruges umiddelbart efter <FIND>, <CONT> eller <RPLCON>.
- <RPLCON> Erstat og fortsæt søgning. Denne funktion svarer til <REPLACE> efterfulgt af <CONT>.
- <MARK> Marker position. Markørens nuværende position lagres i tekstmærkerregisteret, der bruges af <ZAP> og <LIST> kommandoerne, til at afgøre hvor den blok, der arbejdes på, slutter. Der kan kun defineres et tekstmærke (når et mærke defineres, "glemmes" den tidligere position), og der er ingen synlige tegn på, hvor tekstmærket står (hvis du ikke kan huske positionen, så sæt blot mærket igen). Bemærk, at når EDIT mode bruges, bliver tekstmærkerregisteret nulstillet.
- <ZAP> Slet blok. Alle tegn mellem markøren og tekstmærket slettes. Bemærk, at cursoren skal stå tidligere i teksten end tekstmærket. En kopi af blokken bliver gemt i edit-bufferen, før blokken slettes. Denne kopi kan senere genindsættes med <INSERT> kommandoen.
- <INSERT> Kopier edit-buffer. Edit-bufferens indhold kopieres ind i teksten ved markørens nuværende position. Edit-bufferen indeholder de tegn, der blev slettet sidst EDIT mode eller <ZAP> kommandoen blev brugt.
- <LIST> Udskriv blok. Blokken, der starter ved markøren og slutter ved tekstmærket, udskrives på printeren (CP/M systemets LST: enhed). Sørg altid for, at printeren er klar til at modtage tegn, da systemet ellers "hænger" og må RESETtes.

<QUIT> Afslut editor. Editoren afsluttes, og der returneres til kommandoniveauet.

3.2 EDIT mode

EDIT mode bruges, når der skal indsættes og/eller slettes tegn eller linier. Når <MODE> funktionen bruges i MOVE mode, startes EDIT mode, og samtidigt slettes alle positioner efter markøren. Der kan nu indsættes tegn blot ved at taste dem på tastaturet. Hvergang et tegn indsættes, flyttes markøren en position mod højre. De følgende funktioner kan bruges i EDIT mode:

<MODE> Returner til MOVE mode. Før der returneres, bliver alle positioner efter markøren opdateret.

<BACKSP> Slet tegn. Sletter tegnet umiddelbart foran markøren, og flytter markøren en position mod venstre.

<LEFT> Samme som <BACKSP>.

<RIGHT> Genindsæt tegn/linie. Genindsætter det tegn eller den linie, der lige er blevet slettet.

<TAB> Indsæt et HT tegn. Markøren flyttes til den næste multiplum-af-otte position.

<RETURN> Indsæt et lineskift (CR/LF sekvens). Markøren flyttes til en ny linie. Hvis auto-tabulatoren er i brug, vil den nye linie automatisk starte med en indrykning, der svarer til linien ovenover.

<REMOVE> Slet linie. Alle tegn foran markøren op til begyndelsen af linien slettes. Hvis markøren allerede står ved begyndelsen af en linie, slettes hele den foregående linie.

<AUTO> Skift auto-tabulatorens stilling (hvis OFF, så skift til ON, og omvendt).

3.3 Editorens fejlmeddelelser

Editorens fejlmeddelelser vises altid på skærmens øverste linie (der hvor statuslinien normalt står). Et eksempel:

ERROR: No room to insert. Press <SPACE>

Fejlmeddelelsen fjernes igen ved et tryk på mellemrumstangenten. Editoren har fire forskellige fejlmeddelelser. Disse er:

No room for edit buffer

Denne fejlmeddelelse rapporteres i EDIT mode, hvis man prøver at slette flere tegn, end der kan være i edit-bufferen. Efter fejlen returneres der til MOVE mode. Hvis tegnene ikke skal genindsættes senere, kan man blot gå tilbage i EDIT mode og fortsætte med at slette. Fejlen opstår også, når <ZAP> kommandoen bruges, og der ikke er plads til en kopi af blokken i edit-bufferen.

○ - No room to insert

Denne fejl rapporteres i EDIT mode, hvis man prøver at ind-sætte tegn, når der ikke er mere ledigt lager. Efter fejlen, returneres der til MOVE mode. Fejlen rapporteres også hvis <FIND> funktionen bruges, når der er mindre end 256 bytes ledigt lager, hvis <REPLACE> eller <RPLCON> forsøges, når der ikke er lager nok til at foretage en erstatning, eller hvis man prøver at bruge <INSERT> når der ikke er nok lager.

Marker not set

Denne fejl rapporteres af <ZAP> og <LIST>, hvis tekstmærket ikke er sat.

Invalid marker position

Denne fejl rapporteres af <ZAP> og <LIST>, hvis tekstmærket står før markøren i teksten.

Kapitel 4

Compileren

Compileren er hjertet i COMPAS systemet. Ved hjælp af denne kan Pascal programmer oversættes til Z-80 maskinkode.

Når compileren startes fra en COMPILE eller en RUN kommando, bliver objektkoden gemt direkte ud i lageret umiddelbart efter kildeteksten. På denne måde arbejder compileren uhyre hurtigt (op til 5000 linier pr. minut), men til gengæld kræves der også en del arbejdslager, da både kildetekst og objektkode skal være til stede på een gang.

Når compileren startes fra en PROGRAM eller en OBJECT kommando, bliver objektkoden gemt ud i en diskfil. Denne metode er naturligvis noget langsommere end den ovenstående, men den kræver til gengæld ikke så meget arbejdslager, og muliggør desuden generering af programfiler (.COM filer), der kan startes fra CP/M blot ved at taste deres navn.

Ved hjælp af FIND kommandoen kan compileren bruges til at finde det sted i kildeteksten, der var skyld i en kørselsfejl i objektkoden. Denne facilitet er en uvurderlig hjælp under fejlsøgning af programmer.

Hvis compileren finder en fejl under en oversættelse, stopper den og viser et fejlnummer på skærmen. Hvis COMPAS.ERM filen blev indlæst under opstart, vises desuden en fejlmeddelelse:

```
    Error 04: Dubleret identifier. Press <SPACE>
```

Ved et tryk på mellemrumstangenten startes editoren, og markøren flyttes automatisk til det sted, hvor fejlen blev fundet. Fejlen kan nu rettes, og en ny oversættelse kan derefter foretages. Bemærk, at editoren ikke bliver startet, hvis compileren finder en fejl i en include-fil. I stedet vil fejlmeddelelsen angive yderligere informationer om, hvor i filen fejlen findes (læs mere herom i afsnit 4.6).

4.1 COMPILE kommandoen

Når compileren startes fra en COMPILE kommando, bliver objektkoden gemt direkte ud i arbejdslageret, umiddelbart efter kildeteksten. Ved start udskriver systemet:

Compiling

Efter endt oversættelse (forudsat at der ikke blev fundet en fejl) informerer systemet om størrelsen af objektkoden og data-lageret, samt antallet af ledige bytes:

```
Code: rrrrr bytes (aaaa-bbbb)
Free: sssss bytes (cccc-dddd)
Data: ttttt bytes (eeee-ffff)
```


De i parentes viste tal er start- og slutadresserne på det givne område (i hexnotation).

Bemærk, at når editoren startes bliver en eventuel objektkode automatisk slettet.

4.2 RUN kommandoen

RUN kommandoen bruges til at starte kørsel af programmet. Hvis der ikke findes en objektkode, når kommandoen gives, bliver compileren først startet for at oversætte kildeteksten. Forudsat at der ikke bliver fundet en fejl under oversættelsen, eller hvis objektkoden allerede er tilstede, udskriver systemet:

Running

hvorefter programmet startes. Når programmet slutter, bliver kontrollen tilbagegivet til COMPAS systemet.

Hvis der opstår en kørselsfejl under udførelsen af programmet, bliver det stoppet, og en fejlmeddelelse udskrives:

```
EXECUTION ERROR nn AT PC=aaaa
Program terminated
```

hvor nn er fejlens nummer, og aaaa er fejlens relative adresse (i forhold til startadressen på objektkoden). Ved hjælp af FIND kommandoen kan den sætning i kildeteksten, der var skyld i fejlen, findes.

4.3 PROGRAM kommandoen

Når compileren startes fra en PROGRAM kommando, bliver objektkoden lagret i en diskfil. Mulige kommandolinier er:

```
PROGRAM <filnavn>
PROGRAM <filnavn>,<start>
PROGRAM <filnavn>,<start>,<slut>
PROGRAM <filnavn>,,<slut>
```

<start> og <slut> er hex-adresser (uden det foranstillede '\$' tegn). Hvis filtypen udelades, bliver '.COM' automatisk valgt.

<start> angiver startadressen på objektkoden. Hvis <start> udelades, bliver slutadressen på run-time programdelen automatisk valgt. <start> værdier må aldrig være mindre end slutadressen på run-time programdelen (denne adresse kan findes ved at bruge PROGRAM kommandoen uden <start> parameteren).

<slut> angiver adressen på toppen af programmets arbejdslager. Hvis <slut> udelades, bliver den nuværende adresse på toppen af arbejdslageret valgt (denne er bestemt af CP/M operativsystemet, og står i adresse 6-7). Eftersom compileren reserverer lager til variable fra toppen af arbejdslageret og nedefter, kan programmer, der er oversat for en given størrelse af arbejdslageret, ikke køres på systemer med mindre arbejdslager.

Førend kildeteksten oversættes, skriver COMPAS en kopi af run-time programdelen ud i diskfilen. Run-time programdelen optager altid den første del af en programfil. Området fra slutadressen på run-time programdelen til startadressen på programkoden berøres aldrig af programmet, og det er derfor et passende sted til EXTERNAL specificerede maskinkoderutiner, der for eksempel kan indsættes med DDT programmet (DDT programmet medfølger på den originale CP/M systemdisk).

Efter at have gemt run-time programdelen ud i diskfilen starter compileren på at oversætte kildeteksten. Ved denne lejlighed udskrives:

Compiling

Efter endt oversættelse (forudsat at der ikke blev fundet en fejl) informerer systemet om størrelsen af objekt-koden og data-lageret, samt antallet af ledige bytes:

```
Free: rrrrr bytes (aaaa-bbbb)
Code: sssss bytes (cccc-dddd)
Free: ttttt bytes (eeee-ffff)
Data: uuuuu bytes (gggg-hhhh)
```

Den første linie vises kun, hvis en <start> adresse blev angivet på kommandolinien.

4.4 OBJECT kommandoen

OBJECT kommandoen er magen til PROGRAM kommandoen, bortset fra, at run-time programdelen ikke medtages i den diskfil, der genereres af compileren. Desuden er det ikke '.COM' typen der automatisk bliver valgt, men i stedet '.OBJ'. Objektfiler kan kun startes ved et kald til chain proceduren fra en anden COMPAS Pascal programfil (de kan altså ikke startes direkte fra CP/M). Yderligere detaljer herom findes i kapitel 18 i COMPAS Pascal programmeringsmanualen.

4.5 FIND kommandoen

FIND kommandoen bruges til at finde den sætning i kildeteksten, der svarer til en given adresse i objekt-koden. Når compileren startes fra en FIND kommando, genererer den ingen objekt-kode. Kommandolinien kan være en af to muligheder:

```
FIND
FIND <reladr>
```

hvor <reladr> er den relative adresse i objekt-koden på det sted, der ønskes fundet. Hvis objekt-koden for eksempel starter i adresse \$1E80, vil kommandoen FIND 348 finde det sted i kildeteksten, der svarer til den maskinkodeinstruktion i objekt-koden, der ligger på adresse \$21C8. Hvis <reladr> ikke specificeres, vælger systemet automatisk adressen på den sidst forekomne kørselsfejl. Således kan man, blot ved at indtaste FIND efter en kørselsfejl, finde det sted i kildeteksten, der er skyld i fejlen. Førend søgningen indledes, udskriver compileren:

Searching

Hvis compileren finder den adresse, der angives, stopper den og udskriver:

```
Target address found. Press <SPACE>
```

Ved et tryk på mellemrumstangenten startes editoren, og markøren flytter automatisk til det sted i kildeteksten, der blev fundet. Hvis den givne adresse ikke findes, udskriver compileren:

```
Target address not found.
```

hvorpå den returnerer til kommandoniveauet.

4.6 WHERE kommandoen

Hvis compileren finder en fejl i en include-fil under en oversættelse, vil fejludskriften ikke blot vise fejlens nummer og betydning, men også include-filens navn og fejlens tegnadresse. Et eksempel:

```
Include file A:GRAFIK.LIB at CC=07B2  
Error 41: Uens typer i udtryk
```

Editoren startes ikke, da fejlen er i include-filen og ikke i den programtekst, der findes i lageret. Tegnadressen (CC) er det antal tegn (i hex), der blev læst fra include-filen, før fejlen opstod. Den følgende metode kan bruges til at rette fejlen i include-filen: Gem først den programtekst, der findes i lageret, ved hjælp af SAVE kommandoen (medmindre den ikke er blevet ændret, siden den sidst blev gemt på disketten). Slet herefter teksten med ZAP kommandoen, indlæs include-filen med LOAD kommandoen, og giv tilsidst en WHERE kommando (uden argument) - denne vil starte editoren og flytte markøren til det sted, fejlen blev fundet.

WHERE kommandoen kan anvendes på to måder:

```
WHERE  
WHERE <tegnadr>
```

hvor <tegnadr> er et hexstal, der angiver tegnadressen på det sted i teksten, der ønskes fundet. Hvis <tegnadr> udelades, vælges tegnadressen på den sidst forekomne fejl i en include-fil.

Når editoren forlades, gemmes markørens tegnadresse som den adresse, WHERE kommandoen anvender, hvis intet andet angives. WHERE kommandoen kan således bruges til at komme tilbage til det sted i teksten, man forlod, da <QUIT> kommandoen blev givet.

Kapitel 5

Andre kommandoer

5.1 DIR kommandoen

DIR kommandoen bruges til at udskrive en diskettes bibliotek. Formatet af kommandolinien er:

```
DIR <ffn>
```

hvor <ffn> er et flertydigt filnavn, med et format magen til det, der anvendes af CP/M systemets DIR kommando, dvs. et filnavn, hvori spørgsmålstegn (?) og stjerner (*) kan anvendes.

Et spørgsmålstegn indikerer, at denne position kan indeholde ethvert tegn, og en stjerne indikerer, at resten af tegnene i det pågældende felt (navnet eller typen) kan være hvad som helst.

Hvis både navnet og typen udelades, altså hvis kun drive-identificeren og det efterfølgende kolon angives, udskrives navnene på alle filer på den pågældende diskette. Hvis <ffn> udelades fuldstændigt, udskrives navnene på alle filer på auto-disketten.

5.2 NEW kommandoen

NEW kommandoen bruges til at aktivere en nyindsat diskette og til at ændre auto-disketten. Kommandolinien kan være en af to muligheder:

```
NEW  
NEW <disk>:
```

hvor <disk> er en diskidentificer (A-P). En NEW kommando skal udføres hvergang der skiftes diskette i en af diskettestationerne (ellers risikerer man, at CP/M giver en R/O fejl når man prøver at skrive på disketten). Hvis <disk> angives i en NEW kommando, bliver auto-disketten ændret til denne diskette.

5.3 MEMORY kommandoen

MEMORY kommandoen oplyser om størrelsen af kildeteksten og antallet af ledige bytes i arbejdslageret:

```
Text: rrrr bytes (aaaa-bbbb)  
Free: ssss bytes (cccc-dddd)
```

hvor tallene i parentes er start- og slutadresserne (i hexnotation) på området.

5.4 ZAP kommandoen

ZAP kommandoen sletter den nuværende kildetekst. Som en sikkerhedsforanstaltning spørger kommandoen:

Are you sure (Y/N)?

Teksten bliver kun slettet, hvis der svares 'Y' eller 'y'.

5.5 QUIT kommandoen

QUIT kommandoen afslutter COMPAS og returnerer til CP/M. COMPAS kan senere varmstartes som beskrevet i afsnit 1.1.

Appendix A

Systemkonfigurering

COMPAS systemet er designet således, at det relativt nemt kan omstilles til brug på en anden skærm og/eller et andet tastatur end det oprindelige. Bemærk, at den kopi af COMPAS, der findes på distributionsdisketten, allerede er konfigureret til brug på et specielt mikrodatamatsystem (medmindre det er en "ukonfigureret" kopi). Detaljer herom findes i tillægget til brugermanualen.

Til re-konfigurering af COMPAS systemet medfølger der et program på distributionsdisketten, der hedder CONFIG.COM. Dette appendix beskriver de parametre der kan ændres med CONFIG, samt deres adresser i systemet, i tilfælde af, at man ønsker at foretage ændringerne "manuelt" med DDT programmet.

Der findes to konfigurationstabeller i COMPAS. Den første tabel indeholder informationer om systemets skærm, og er lagret fra \$10A til \$1DF.

- | | |
|-------------|---|
| \$10A-\$10A | Skærbredde. Denne byte angiver antallet af tegn pr. linie på skærmen. Værdien skal være den præcise bredde, og ydermere skal den være et multiplum af 8. |
| \$10B-\$10B | Skærmhøjde. Denne byte angiver antallet af linier på skærmen. Værdien skal være den præcise højde. |
| \$10C-\$10C | Auto-scroll flag. En nul i denne byte indikerer, at skærmen scroller af sig selv, hvis et tegn skrives i den yderste position på den nederste linie. Andre værdier indikerer, at dette ikke sker. |
| \$10D-\$115 | GOTOXY startsekvens. Disse bytes angiver den streng, der skal sendes før markørkoordinaterne, når markøren adresseres direkte. Den første byte indeholder længden af strengen (0-8), og de efterfølgende bytes indeholder tegnene. |
| \$116-\$11A | GOTOXY separatorsekvens. Disse bytes angiver den streng, der skal sendes mellem markørkoordinaterne, når markøren adresseres direkte. Normalt sendes der ingen tegn, men nogle standarder (f.eks. ANSI standarden) kræver en separator. Den første byte indeholder længden (0-4), og de efterfølgende bytes indeholder tegnene. |
| \$11B-\$11F | GOTOXY slutsekvens. Disse bytes angiver den streng, der skal sendes for at afslutte en direkte adressering af markøren. Normalt sendes der ingen tegn, men nogle standarder (f.eks. ANSI standarden) kræver en slutsekvens. Den første byte indeholder længden (0-4), og de efterfølgende bytes indeholder tegnene. |
| \$120-\$120 | Koordinatrækkefølge. Et nul i denne byte angiver, at Y koordinaten (linien) skal sendes før X koordinaten (positionen) når markøren adresseres direkte. Værdier forskellige fra nul angiver X før Y. |

- \$121-\$121 Koordnatoffset. Denne byte angiver det offset, der skal lægges til markørkoordinaterne, før de sendes ved en direkte adressering af markøren. Koordinaterne starter ved nul. Almindeligvis anvendes et offset på 32 (\$20), men dette kan variere.
- \$122-\$122 Koordnatformat. Et nul i denne byte angiver, at koordinaterne skal sendes som enkelte tegn i binært format (dvs. ASCII værdien af tegnene svarer til koordinaternes værdi plus koordnatoffsetet). Værdier forskellige fra nul angiver, at koordinaterne skal sendes som numeriske strenge (strenge, der er sammensat af tegnene '0' til '9'). Antallet af cifre (2 eller 3) afgøres af den byte, der lagres her.
- \$123-\$12B CLREOS sekvens. Disse bytes definerer den streng, der skal sendes for at slette alle tegn fra markøren til slutningen af skærmen. Den første byte indeholder længden (0-8), og de efterfølgende bytes indeholder tegnene.
- \$12C-\$134 CLREOL sekvens. Disse bytes definerer den streng, der skal sendes for at slette alle tegn fra markøren til slutningen af den nuværende linie. Den første byte indeholder længden (0-8), og de efterfølgende bytes indeholder tegnene.
- \$135-\$137 Hopvektor til GOTOXY rutine. Hvis den metode, der benyttes af skærmen til at adressere markøren direkte, ikke kan defineres via de ovenstående tabelindgange, kan der i disse bytes indlægges en hopvektor (\$C3 i den første byte efterfulgt af adressen i omvendt format) til en rutine, der udfører funktionen. Rutinen modtager Y koordinaten i D registeret og X koordinaten i E registeret, og den kan installeres i brugerområdet (se nedenfor). Rutinen skal returnere med carry-flaget sat (afslut den med en SCF instruktion umiddelbart før RET instruktionen), og den må ikke ændre på BC, DE, HL og IX. Bemærk, at hvis der ikke er indlagt en hopvektor, skal disse bytes indeholde værdierne \$00, \$B7, \$C9 (i den rækkefølge), svarende til instruktionerne NOP, OR A, RET.
- \$138-\$13A Hopvektor til CLREOS rutine. Hvis systemets skærm ikke har en CLREOS funktion, kan der i disse bytes indsættes en hopvektor til en rutine, der udfører funktionen. De regler, der gælder for GOTOXY hopvektoren (se ovenfor), gælder også her, bortset fra, at der ikke overføres nogen parametre til denne rutine.
- \$13B-\$13D Hopvektor til CLREOL rutine. Svarer til CLREOS hopvektoren, bortset fra, at denne hopvektor gælder for CLREOL funktionen.
- \$13E-\$13F Adresse på outputrutine. Dette felt er udelukkende et informationsfelt, og det bør således ikke modificeres. Feltet indeholder adressen (mindst betydende byte først) på en rutine, der udlæser tegnet i A registeret til skærmterminalen. Brugerdefinerede drivere bør anvende denne rutine til at udlæse tegn.

Bemærk, at CALL instruktionen ikke skal kalde adresse \$13E, men derimod den adresse, der er indeholdt i adresserne \$13E og \$13F. Udlæsningsrutinen ændrer ikke på BC, DE, HL og IX.

- \$140-\$1BF Brugeroråde. Dette område bruges ikke af systemet, men er i stedet reserveret for at give plads til brugerdefinerede driverrutiner.
- \$1C0-\$1DF Terminalnavn. Navnet på den terminal, der er defineret ved indholdet af konfigurationstabellen, bør indlægges i dette område (op til 32 tegn, med blanktegn i de bytes, der ikke anvendes). Navnet bruges af CONFIG programmet, men ikke af selve COMPAS systemet.

Den anden konfigurationstabel lagrer informationer der bruges af COMPAS editoren. Adressen på tabellen er ikke fast, men startadressen kan findes ved at aflæse adresse \$101 og \$102. \$101 indeholder den mindst betydende byte og \$102 den mest betydende. I oversigten nedenfor angives den relative adresse på hver tabelindgang, dvs. det tal, der skal lægges til tabellens startadresse for at finde den absolutte adresse.

- \$05-\$05 Backup flag. En værdi forskellig fra nul i denne byte indikerer, at SAVE kommandoen skal ændre typen, på filer af samme navn som den nye fil, til '.BAK', før den nye fil oprettes. Et nul indikerer, at filer af samme navn blot skal slettes.
- \$06-\$06 Auto-tabulator flag. En værdi forskellig fra nul i denne byte indikerer, at auto-tabulatoren skal være ON efter en koldstart af COMPAS. Et nul indikerer, at den skal være OFF.
- \$07-\$08 Keyboard pause. Disse to bytes definerer det antal gange, keyboard status rutinen skal kaldes, før statuslinien opdateres. Den mindst betydende byte af antallet lagres først. Eksperimenter med forskellige værdier, indtil pausen er "længere" en tastaturets repeteringshastighed (typiske værdier ligger mellem \$0080 og \$0400). En værdi på nul indikerer, at CP/M BIOS'ens statusrutine altid returnerer "klar", eller, at tastaturet ikke har en buffer.
- \$09-\$88 Editorens kommandotabel.
- \$89-\$91 Insert line sekvens. Ved at sende denne sekvens indsættes en blank linie på skærmen, og den nuværende linie, og alle derunder, rykkes en linie ned. Funktionen bruges af editoren når skærmen rulles nedad. Hvis den ikke findes på terminalen, bør længden af sekvensen sættes til 0.
- \$92-\$9A Delete line sekvens. Ved at sende denne sekvens slettes den nuværende linie, og alle linier derunder rykkes en linie op. Den nederste linie på skærmen blankes. Funktionen bruges af editoren når skærmen rulles opad (derved undgås, at statuslinien forsvinder). Hvis den ikke findes på terminalen, bør længden af sekvensen sættes til 0.

\$9B-\$A3	Reverse on sekvens. Når denne sekvens sendes, bliver de efterfølgende tegn skrevet i inverteret skrift (eller med forøget lysstyrke). Funktionen bruges af editoren til at fremhæve statuslinien. Hvis den ikke findes på terminalen, bør længden af sekvensen sættes til 0.
\$A4-\$AC	Reverse off sekvens. Når denne sekvens sendes, returneres der til normal skrift (den ovenstående funktion ophæves).

Editorens kommandotabel består af et antal strenge, der hver især definerer det tegn eller den tegnfølge, der bruges til at aktivere en bestemt editorkommando. Den første byte i hver streng indeholder længden og de efterfølgende bytes indeholder tegnene. Strengene lagres umiddelbart efter hinanden, dvs. længden af en streng følger umiddelbart efter det sidste tegn i den foregående streng. Den editorfunktion, der aktiveres af en given streng, bestemmes udfra den nedenstående tabel:

<MODE>	<TAB>	<BACKSP>	<RETURN>
<REMOVE>	<AUTO>	<LEFT>	<RIGHT>
<UP>	<DOWN>	<PREV>	<NEXT>
<BEGIN>	<END>	<FIND>	<CONT>
<REPLACE>	<RPLCON>	<MARK>	<ZAP>
<LIST>	<INSERT>	<QUIT>	

Tabellen skal forstås således, at <MODE> defineres af den første streng, <TAB> af den anden, <BACKSP> af den tredje, osv.

Når keyboardtabellen ændres, bør man altid sikre, at den ikke er flertydig. Antag, som et eksempel, at en eller flere af strengene starter med et ESC (\$1B) tegn. En streng, der kun består af et ESC tegn, ville da gøre tabellen flertydig, idet editoren, når den modtager et ESC tegn, ikke kan afgøre hvorvidt dette tegn kommer "alene" eller som en del af en tegnfølge.

○

○

○

○