

Programme leichter laden

Dieses Programm vereinfacht das Laden von Programmen durch direkte Auswahl aus dem Directory.

Programme werden häufig nach dem Listen des Directorys durch Voranstellen eines »LOAD« und Ergänzung durch »,8:« oder »,8,1« geladen. Oftmals vergißt man dabei allerdings das Eintippen des Doppelpunktes oder man macht andere kleine Fehler, worauf sich der Computer mit einem »SYNTAX ERROR« meldet.

Mit »Directory-Lader« (Listing) kann das Directory einer Diskette nun so manipuliert werden, daß wahlweise ein »,8:« oder ein »,8,1« an den Filenamen angehängt wird. Das vereinfacht den Ladevorgang um einiges: Nur noch das Directory listen, mit dem Cursor in die gewünschte Zeile fahren, »LOAD« eingeben und schon geht's los.

So funktioniert der »Directory-Lader«

Jedes Programm auf einer Diskette ist in einem Directoryblock verzeichnet. Dort belegt der Name des Programmes 16 Byte. Ist der Name kürzer als 16 Byte, wird er durch ein »SHIFT Space« (CHR\$ 160) ergänzt. Dadurch wird auch das Ende eines Namens vom DOS erkannt.

Beim Listen des Directorys werden nun alle 16 Byte des Namens auf dem Bildschirm angezeigt. Das zu lösende Problem besteht also darin, den gewünschten Zusatz ebenfalls hinter dem Programm anzuzeigen.

Dazu wird zuerst der erste Block des Directorys eingelesen. Der Bufferpointer wird auf den ersten Namen des Blocks gesetzt und der Name ausgelesen. Hierbei werden alle 16 Byte des Namens verwendet. Durch Betätigen entsprechender Tasten wird nun der Name wie folgt bearbeitet:

- @ Löschen des Zusatzes
- ↑ Überspringen des Namens ohne Änderung
- M Schreiben des Zusatzes »,8,1«
- B Schreiben des Zusatzes »,8:«

Beim Schreiben des Zusatzes wird der Name des Programms von hinten durch den Zusatz ergänzt. Dabei werden Programmnamen, die länger als 11 Zeichen bei Maschinenprogrammen oder länger als 12 Zeichen bei Basic-Programmen sind, entsprechend verkürzt.

In dieser Weise werden sämtliche Namen eines Directoryblocks bearbeitet.

Da der Bufferpointer jedesmal weitergesetzt wird, kann es vorkommen, daß auch bereits gelöschte Programme bearbeitet werden. Das hat jedoch keinen weiteren Einfluß.

Nach der Bearbeitung eines Blocks wird dieser zurückgeschrieben, und das Programm fährt mit dem nächsten Block fort.

Tabelle 1 zeigt die verwendeten Variablen, Tabelle 2 die einzelnen Routinen des Programms.

Während der Bearbeitung wird ständig der nächste zu bearbeitende Block angezeigt. Erscheinen die Werte 0 und 255, so ist der letzte Block erreicht. Man hat jetzt eine Diskette mit modifiziertem Directory und damit einfach zu ladenden Programmen vorliegen (Bild 1).

(Holger Weihe/ev)

```
0 " test direktory " db 2a
8 "testdirektor",8: prg
8 "1234567890123456" prg
8 "loader" ,8: prg
8 "programm" ,8: prg
8 "basic" ,8: prg
8 "maschprog" ,8,1 prg
616 blocks free.
```

Bild 1. Beispiel eines modifizierten Directorys

O	Prüfsumme für Katalog-Routine
K	Ausgewählte Taste bei Menü
T, S	Track und Sektor für ersten Block
P, Q	Zwischenspeicher für Track, Sektor
Z	Position des ersten Namens eines Blocks
X\$	Warten auf Tastendruck
T\$, S\$	Track, Sektor des nächsten Blocks
N\$	Programmname
D\$	Modifizierter Programmname

Tabelle 1. Variablenliste zu »Directory-Lader«

Zeile	Funktion
100-260	Katalog-Routine
270-450	Titelbild
470-500	Auswertung der Eingabe
510-640	Anzeigen des Direktories
650-990	Einlesen des Blocks
	Aufbau des Untermenüs
	Auswertung der Eingabe
1000-1140	Schreiben des Zusatzes
	»,8:« bei Basic-Programmen
1150-1280	Schreiben des Zusatzes
	»,8,1« bei Maschinen-Programmen
1290-1390	Keine Änderung des Programmnamens
1400-1500	Löschen des Zusatzes
1510-1610	Rückschreiben des eingelesenen Blocks
1620-1740	Auswahl und Ende des Programmes

Tabelle 2. Die wichtigsten Routinen

```
10 REM ***** <119>
20 REM ** DIREKTORY LOADER <104>
30 REM ** <079>
40 REM ** BY HOLGER WEIHE <031>
50 REM ** AM SILBERBERG 1 <043>
60 REM ** 3344 BOERSSUM <050>
70 REM ** <119>
80 REM ** TEL. 05334/6589 <165>
90 REM ** <139>
100 REM ***** <209>
110 REM ↑ = PFEIL NACH OBEN <228>
120 REM @ = KLAMMERAFFE <068>
130 REM ***** <239>
140 REM ** DIREKTORY ROUTINE <217>
150 REM ***** <003>
160 REM <222>
170 REM <232>
180 DATA 169,36,133,251,169,251,133,187,16
9,0,133,188,169,1,133,183,169,8 <195>
190 DATA 133,186,169,96,133,185,32,213,243
,165,186,32,180,255,165,185,32,150 <216>
200 DATA 255,169,0,133,144,160,3,132,251,3
2,165,255,133,252,164,144,208,47 <072>
210 DATA 32,165,255,164,144,208,40,164,251
,136,208,233,166,252,32,205,189 <029>
220 DATA 169,32,32,210,255,32,165,255,166,
144,208,18,170,240,6,32,210,255 <200>
230 DATA 76,76,192,169,13,32,210,255,160,2
,208,198,32,66,246,96,234 <032>
240 FOR I=49152 TO 49152+105:READ D:POKE I
,D:D=0+D:NEXT <000>
250 IF 0<>15577 THEN POKE 199,1:PRINT"25P
ACE>FEHLER IN DATAS(2SPACE)":STOP <119>
```

```

260 REM <068>
270 REM <078>
280 REM ***** <017>
290 REM ** TITEL <056>
300 REM ***** <037>
310 REM <118>
320 REM <128>
330 POKE 53280,0:POKE 53281,0:PRINT CHR$(1 <013>
52) <010>
340 PRINT CHR$(14):PRINT CHR$(147) <010>
350 PRINT "{SSPACE}0000000000000000000000 <225>
00000" <154>
360 PRINT "{SSPACE}0{27SPACE}0" <072>
370 PRINT "{SSPACE}0{5SPACE}0DIREKTORY LOADE <174>
R{6SPACE}0" <072>
380 PRINT "{SSPACE}0{27SPACE}0" <174>
390 PRINT "{SSPACE}0000000000000000000000 <027>
00000" <248>
400 PRINT <248>
410 PRINT" <019>
420 PRINT "{SSPACE}";CHR$(18);" 0ITTE AUSWA <060>
EHLLEN ";CHR$(146):PRINT:PRINT <066>
430 PRINT "{8SPACE}";CHR$(18);"1";CHR$(146) <172>
;" {3SPACE}DIREKTORY ANZEIGEN":PRINT <136>
440 PRINT "{8SPACE}";CHR$(18);"2";CHR$(146) <190>
;" {3SPACE}DIREKTORY BEARBEITEN":PRINT <083>
450 PRINT "{8SPACE}";CHR$(18);"3";CHR$(146) <030>
;" {3SPACE}0UIT":PRINT <238>
470 GET X$:IF X$="" THEN 470 <062>
480 K=VAL(X$):IF K<1 OR K>3 THEN 470 <074>
490 ON K GOTO 510,660,1740 <160>
500 GOTO 330 <033>
510 REM <180>
520 REM <114>
530 REM ***** <124>
540 REM ** DIREKTORY ANZEIGEN <011>
550 REM ***** <217>
560 REM <039>
570 REM <015>
580 PRINT CHR$(147):PRINT:PRINT:POKE 199,1 <085>
:PRINT" 0ITTE 0ISK EINLEGEN UND 0ASTE <114>
DRUECKEN " <194>
590 GET X$:IF X$="" THEN 590 <204>
600 PRINT CHR$(147) :SYS 49152 <005>
610 PRINT:PRINT:POKE 199,1:PRINT" 0ASTE DR <205>
UECKEN " <025>
620 GET X$:IF X$="" THEN 620 <244>
630 GOTO 330 <254>
640 REM <160>
650 REM <124>
660 REM ***** <231>
670 REM *** KOMMA ACHTER <110>
680 REM ***** <194>
690 REM <094>
700 REM <221>
710 T=18:S=1:Z=5 <003>
720 OPEN 1,8,15 <029>
730 OPEN 2,8,2,"#" <091>
740 PRINT#1,"U1";2;0;T;S <200>
750 GET#2,T$,S$ <038>
760 P=T:Q=S <029>
770 IF T$="" THEN T=0:S=255:GOTO 800 <162>
780 T=ASC(T$):S=ASC(S$): <126>
790 PRINT#1,"B-P";2;1 <076>
800 PRINT#1,"B-P";2;Z: <020>
810 FOR I=1 TO 16 <091>
820 GET#2,N$ <200>
830 D$=D$+N$ <038>
840 NEXT I <029>
850 PRINT#1,"B-P";2;Z <162>
860 IF D$="" THEN 1510 <126>
870 PRINT CHR$(147):PRINT:PRINT:PRINT" {3SP <076>
ACE}NEXT 0IREKTORY 0LOCK : " <204>
880 PRINT:PRINT" {3SPACE}0RACK :";T;" 0SEKTO <228>
R :";S <228>
890 PRINT:PRINT:PRINT:PRINT:PRINT" {3SPACE} <217>
0ROGRAMM :";D$ <090>
900 PRINT:PRINT <090>
915 PRINT "{3SPACE}";CHR$(18);"0";CHR$(146) <021>
;"ASIC ODER ";CHR$(18);"1";CHR$(146);" <004>
ASCHINENPROGRAMM" <021>
920 PRINT <004>
930 PRINT:PRINT:PRINT:PRINT:POKE 199,1:PRI <020>
NT" {4SPACE}<0> 0EITER {3SPACE}<0> 0DESC <020>
HEN{6SPACE}" <020>

```

```

940 GET X$:IF X$="" THEN 940 <121>
950 IF X$="B" THEN 1070 <028>
960 IF X$="M" THEN 1160 <163>
970 IF X$="1" THEN 1300 <230>
980 IF X$="@" THEN 1420 <009>
990 GOTO 940 <028>
1000 REM <044>
1010 REM <054>
1020 REM ***** <042>
1030 REM ** BASICPROGRAMM <025>
1040 REM ***** <064>
1050 REM <096>
1060 REM <106>
1070 D$=LEFT$(D$,12) <059>
1080 NE$=D$+CHR$(160)+",8:" <021>
1100 PRINT#2,NE$ <053>
1110 NE$="";D$="";Z=Z+32 <198>
1120 IF Z>256 THEN Z=5:GOTO 1510 <074>
1130 GOTO 800 <098>
1140 REM <186>
1150 REM <196>
1160 REM ***** <184>
1170 REM ** MASCHPROGRAMM <010>
1180 REM ***** <204>
1190 REM <236>
1200 REM <246>
1210 D$=LEFT$(D$,11) <191>
1220 NE$=D$+CHR$(160)+",8,1" <102>
1240 PRINT#2,NE$ <193>
1250 NE$="";D$="";Z=Z+32 <082>
1260 IF Z>256 THEN Z=5:GOTO 1510 <214>
1270 GOTO 800 <238>
1280 REM <072>
1290 REM <082>
1300 REM ***** <021>
1310 REM ** KEINE AENDERUNG <178>
1320 REM ***** <041>
1330 REM <122>
1340 REM <132>
1350 PRINT#2,D$:D$="" <060>
1360 Z=Z+32 <055>
1370 IF Z>256 THEN Z=5:GOTO 1510 <070>
1380 GOTO 800 <094>
1390 REM <182>
1400 REM <192>
1410 REM ***** <131>
1420 REM ** LOESCHEN DES ANHANGS <247>
1430 REM ***** <151>
1440 REM <232>
1450 REM <242>
1460 D$=LEFT$(D$,12)+CHR$(160)+CHR$(160)+C <094>
HR$(160)+CHR$(160) <180>
1470 PRINT#2,D$:D$="" <175>
1480 Z=Z+32 <190>
1490 IF Z>256 THEN Z=5:GOTO 1510 <214>
1500 GOTO 800 <046>
1510 REM <056>
1520 REM <251>
1530 REM ***** <138>
1540 REM ** RUECKSCHREIBEN <017>
1550 REM ***** <098>
1560 REM <108>
1570 REM <248>
1580 IF T=0 AND S=255 THEN PRINT#1,"U2";2; <191>
0;P;Q:GOTO 1690 <149>
1590 PRINT#1,"U2";2;0;P,Q <148>
1600 P=T:Q=S:CLOSE 2:CLOSE 1:GOTO 720 <158>
1610 REM <168>
1620 REM <107>
1630 REM <081>
1640 REM ***** <127>
1650 REM ** ENDE <208>
1660 REM ***** <218>
1670 REM <087>
1680 REM <123>
1690 PRINT:PRINT:POKE 199,1:PRINT" {11SPACE <061>
}0ASTE 0RUECKEN{10SPACE}" <020>
1720 GET X$:IF X$="" THEN 1720 <020>
1730 CLOSE 2:CLOSE 1:GOTO 330 <020>
1740 SYS 64738 <020>

```

0 64'er
Listing »Directory-Lader«.
Bitte beachten Sie die Eingabehinweise auf Seite 6.