

Naarmate men zijn "programmatheek" verder uitbreidt, zal steeds meer de behoefte ontstaan om delen uit verschillende BASIC-programma's te gebruiken om zelf een nieuw programma samen te stellen. Maar hoe doet men dat? Als antwoord daarop geven we hier een programma waarmee men deze mogelijkheid kan toevoegen aan een Junior computer met DOS. Bezitters van andere systemen kunnen het programma vrij gemakkelijk aanpassen, op voorwaarde dat de aanwezige DOS of BASIC is uitgerust met een I/O-besturing die het mogelijk maakt het geheugen te beschouwen als een randapparaat, zoals dat bij de Junior kan.

merge voor BASIC-files

een programma
voor het
samenvoegen
van twee
BASIC-files

De functie van dit merge-programma is het stap-voor-stap samenvoegen van verschillende BASIC-programma's. Niet alleen dat is interessant, maar ook de wijze waarop dit verwezenlijkt wordt. Hiervoor wordt name-lijk gebruik gemaakt van een interessante eigenschap van de BASIC en de DOS van de Junior computer, die ons in staat stelt om het geheugen als een I/O-device te gebruiken. Deze eigenschap is trouwens bij de meeste moderne computers aanwezig.

De I/O-besturing kan worden vergeleken met een schakelaar die, als hij op de juiste manier geprogrammeerd is, het werkgeheugen kan koppelen met de conventionele randapparaten (toetsenbord, beeldscherm, printer, enz.). Ook het geheugen kan echter als "randapparaat" worden gezien. Bij OS65D heeft het geheugen I/O-nummer 5. Bij andere systemen zal men even in de handleiding moeten kijken om te zien hoe de I/O-sturing daar gebeurt.

De I/O-besturing staat onder controle van het DOS-systeem, maar het is ook mogelijk

dit direkt vanuit BASIC te doen. Met behulp van de instructie LIST#5 wordt bijvoorbeeld een BASIC-file in het werkgeheugen (\$3A7E...), die daar in een kompakte (tokenized) vorm is opgeslagen, overgebracht naar adres \$8000 en verder. Op de nieuwe geheugenplaatsen staat het hele programma dan wel in ASCII-vorm, dus in dezelfde vorm als het op het scherm of op de printer zou verschijnen. Het beginadres \$8000 wordt automatisch door het DOS-systeem gekozen, maar de gebruiker kan zelf een ander beginadres kiezen.

Om het belang van deze manipulatie in te zien, moeten we goed weten in welke vorm een BASIC-programma in het geheugen wordt opgeslagen. Gewoonlijk wordt iedere BASIC-instructie in de vorm van een kode in het werkgeheugen opgeslagen, en niet als een stel ASCII-kodes die samen het woord van de instructie vormen. Na het geven van het kommando LIST#5 staat het programma wèl geheel in de vorm van ASCII-tekenen in het geheugen, dus letterlijk zoals we het ook

Tabel 1.

```
2000 FORX=1TO24:PRINT:NEXT
2010 PRINTTAB(10)"-----
2020 PRINTTAB(10)"--FILE MERGE UTILITY-
2030 PRINTTAB(10)"-----
2040 PRINT:PRINT:PRINTTAB(10)"written by A. Nachtmann
2050 PRINT:PRINTTAB(10)"feb. 19. 1984
2060 PRINT:PRINT:PRINT
2070 PRINT"Be sure that both files to be linked have different line numbers.
2080 PRINT"If both files have some common line numbers boot up your system
2090 PRINT"with the RSEQ utility to renumber the lines.
2100 PRINT:INPUT"In which drive are the files to be merged A/B/C/D";D#
2110 D#=LEFT$(D#,1):D=ASC(D#):IF D<ASC("A") OR D>ASC("D") THEN2000
2120 PRINT:INPUT"enter first file name ";F#
2130 INPUT"enter second file name";S#
2140 PRINT:INPUT"are you ready";I#
2150 IF LEFT$(I#,1)<>"Y" THEN2140
2160 REM---RESET MEMORY INPUT POINTER
2170 POKE9098,0:POKE9099,128
2180 DISK!"SE A":DISK!"CA E400=12,7": DISK!"SE "+D#:DISK!"GO E481"
2190 A1=8*16^3+11: A2=8*16^3+2*16+4
2200 REM---
2210 A=A1
2220 FORX=1 TO LEN(F#)
2230 POKE A,ASC(MID$(F#,X,1)):A=A+1
2240 NEXT
2250 REM---
2260 A=A2
2270 FOR X=1TO LEN(S#)
2280 POKEA,ASC(MID$(S#,X,1)):A=A+1
2290 NEXT
2300 POKE8993,16
```


Tabel 2.

```

HEXDUMP: E400,E4FF
  0 1 2 3 4 5 6 7 8 9 A B C D E F
E400: 50 4F 4B 45 38 39 39 33 2C 31 0D 0A 00 0D 0A 44 POKE8993,1....D
E410: 49 53 4B 21 22 4C 4F 20 20 20 20 20 20 22 3A ISK!"LO
E420: 4C 49 53 54 23 35 0D 0A 44 49 53 4B 21 22 4C 4F LIST#5..DISK!"LO
E430: 20 20 20 20 20 20 20 22 3A 4C 49 53 54 23 35 0D
E440: 0A 44 49 53 4B 21 22 47 4F 20 45 34 35 32 22 0D .DISK!"GO E452".
E450: 0A 00 AE 91 23 AD 92 23 8E 66 E4 8D 67 E4 A2 00
E460: BD 00 E4 F0 1B 8D FF FF EE 66 E4 D0 03 EE 67 E4
E470: AD 66 E4 8D 91 23 AD 67 E4 8D 92 23 E8 D0 E1 60
E480: 60 A2 00 A9 80 8E 66 E4 8D 67 E4 A2 0D D0 D1

```

merge voor BASIC-files
 elektuur juni 1984

Tabel 3.

```

HEXDUMP: 2200,2251
  0 1 2 3 4 5 6 7 8 9 A B C D E F
2200: A9 01 8D 5E 26 20 BC 26 A9 2A 85 FF 20 54 27 86
2210: FE 20 67 29 A0 BF 20 EC 22 F0 03 88 D0 F8 8C 00
2220: 23 20 AC 15 20 9E 0F 20 94 15 20 73 2D 0D 0A 2D
2230: 2D 20 44 49 53 4B 20 32 20 2D 2D 0D 0A 0A 00 4C
2240: E6 2A 20 20 20 20 20 20 20 20 20 20 20 20 20
2250: 20 51

```

zouden intypen. Van deze eigenschappen maakt het merge-programma nu juist gebruik.

BASIC + BASIC

Het "koppel"-programma bestaat uit een stukje machinetaal (tabel 2) en een kort BASIC-programma (tabel 1). Nadat het programma weet in welke "eenheid" de bewuste files zich bevinden (D\$) en wat de namen van de files zijn (F\$ en S\$), initialiseert de processor de pointer die het adres bepaalt waar begonnen wordt met het neerschrijven van de "verplaatste" file. Vervolgens worden een kort machinetaalprogramma en een look-up-table geladen vanaf \$E400 (van sektor 7 van track 12; er blijft nog plaats achter de directory). Het machinetaalprogramma wordt gestart door de instructie GO op regel 2180. In de buffer (\$8000...) wordt dan een serie instructies *zonder regelnummer* gezet (zie het gedeelte rechts in tabel 2). De regels 2190 tot 2290 van het BASIC-programma zorgen er voor dat de namen van de twee achter elkaar te koppelen files (F\$ en S\$) achter de twee instructies LO (van de directe instructie uit tabel 2) worden geplaatst. De instructie op regel 2300 programmeert de input-distributor zodanig dat het geheugen als input-device dienst doet. De BASIC-interpretter ontvangt nu de reeks instructies van de buffer, vanaf adres \$8000, alsof deze een voor een op het toetsenbord werden ingegeven... en hij voert ze een voor een uit. Dat betekent dat hij file F\$ laadt, overbrengt naar \$8000 en verder (LIST#5), vervolgens wordt file S\$ geladen en dan achter file F\$ in het geheugen gezet. Daarna wordt de instructie DISK! "GO E452" uitgevoerd, de laatste directe instructie die de interpreter ontvangt van het geheugen als input-device. Het machinetaalprogramma vanaf \$E452 zet achter de twee geladen files in de buffer \$8000 de instructie POKE 8993,1. Deze "direct mode"-instructie wordt niet vooraf-

gegaan door een regelnummer, waardoor ze direkt wordt uitgevoerd als de interpreter haar tegenkomt. Deze laatste instructie zorgt voor het "terugzetten" van de input-distributor naar de originele stand (het toetsenbord wordt dan weer input-device). Nu laadt de BASIC-editor in zijn werkgeheugen de twee files F\$ en S\$ vanuit de buffer \$8000... , voegt ze samen tot één geheel, vertaalt alle instructies weer in de betreffende kodes en list het geheel tevens op het scherm. Als hij de laatste regel van de tweede file heeft afgewerkt, ziet hij de instructie POKE 8993,1. Deze wordt uitgevoerd en dat betekent dat het toetsenbord weer input-device wordt.

Als men nu een LIST geeft, ziet men dat het geheugen netjes achter elkaar de files F\$ en S\$ bevat.

RSEQ

Voor een effectief gebruik van deze merge-faciliteit is het eigenlijk nodig dat men de programma's (of stukken daarvan) eenvoudig kan henummeren. Op disk 2 van de serie van vijf diskettes van de Ohio-DOS bevindt zich een programma RSEQ dat deze taak verricht. Tot nu toe hebben we nog niet beschreven hoe deze disk 2 kan worden aangepast voor de Junior computer. Voor de aanpassing dient tabel 3, waarna RSEQ beschikbaar is voor het henummeren van alle BASIC-programma's, speciaal diegene die men wil samenvoegen. Het aanpassen van disk 2 is niet moeilijk. Men begint met het maken van een kopie van de originele disk (voor het geval er iets mis mocht gaan!), vervolgens wordt track 0 van disk 2 geladen met behulp van de TRACK 0 R/W UTILITY (RA200) op \$A200 (of een andere plaats). Daarna wordt de inhoud van deze track veranderd volgens tabel 3 en dan kan men track 0 weer terugschrijven op de diskette (WA200/2200,8). Dat is alles!