

As a programmer's skills grow there is more and more temptation to use scraps from different programs to make a new one. This is an interesting idea but it is not immediately obvious how it could be put into practice. The program given here, however, was written to do just this. It is a utility designed for the Junior Computer with DOS that can be adapted for other systems as long as the DOS (or BASIC) used has an input/output distributor that allows the *memory to be considered as a peripheral device*, as the Junior does.

# merging BASIC programs

a utility program to merge two distinct basic programs and making use of utility software from the OS disk 2

The purpose of the program given here is to merge different BASIC programs or to place them one after the other. This alone makes it interesting and it is doubly so as it uses an interesting property of the Junior Computer's DOS and BASIC; namely that the memory can be used as an input/output device. This is a characteristic that the Junior shares with the majority of modern personal computers.

The distributor is a *software switch* which, when programmed accordingly, allows the workspace memory to be equated to the conventional peripheral devices (keyboard, VDU, parallel or serial printer, etc.) and also to the main memory, and this is the interesting point as far as we are concerned. In the OS65D DOS system the number of the memory as an input/output device is 5. For any system other than the JC's it will be necessary to refer to the user's manual to find the information needed to modify the program.

The distributor is managed by the DOS but it can be used directly in BASIC. The LIST 5 instruction, for example, causes the BASIC file to be transferred from the workspace (\$3A7E...), where it is in com-

pact (tokenized) form, to \$8000 and from this address on it is found in integral ASCII format so that it can appear as easily on a VDU screen as on a printer. Address \$8000 is set by the DOS but this can easily be changed by the user if he so desires.

To understand this operation it is important to know that the file is compacted in the interpreter's workspace. The BASIC instructions appear there in shortened form as indicators (tokens) or markers rather than as a series of ASCII codes corresponding to the letters making up the reserved words of the instructions. In the normal memory, on the other hand, we find the file in the familiar form after the LIST 5 instruction has been executed. The I/O distributor allows the memory to be used as an input device just as the keyboard is. The merging program makes abundant use of the possibilities this opens up.

## BASIC and merging

The program given here consists of a machine-code section (table 2) and a BASIC part, which is where we will now

```
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=8X16^3+11: A2=8X16^3+2X16+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
```

Table 1. Unlike most of our recent software offerings this program is written in BASIC, or at least one part of it is. This makes the job of adapting it for systems other than the Junior Computer that much easier.

```

HEXDUMP: E400,E4FF
 0 1 2 3 4 5 6 7 8 9 A B C D E F
E400: 50 4F 48 45 38 39 39 33 2C 31 0D 0A 00 0D 0A 44 POKE8993,1....D
E410: 49 53 48 21 22 4C 4F 20 20 20 20 20 20 20 20 20 22 3A ISK!"LO "
E420: 4C 49 53 54 23 35 0D 0A 44 49 53 48 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 " :LIST#5.
E440: 0A 44 49 53 48 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 FF
E490: 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
E4A0: 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
E4B0: 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 84 8F
E4C0: 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
E4D0: 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
E4E0: 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
E4F0: 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF EE 21
E500:

```

Table 2. The second part of the MERGE utility is listed in this hexdump. This complements the BASIC program given in table 1.

```

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 48 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

```

Table 3. Diskette 2 from the set of 5 supplied with the Ohio Scientific DOS contains a utility, RSEQ, that can be used to renumber lines in a file. The hexdump given here lists the modifications needed to adapt this for the Junior.

begin. As soon as it knows the unit where the files can be found (D\$) and their names (F\$ and S\$ are two arbitrary names that must be in the directory of the unit designated by D\$ — lines 2000 . . . 2160) the processor initializes the pointer indicating the start address where the file transferred to memory can be found. It then loads a machine code program and a look-up table at \$E400 (from sector 7 of track 12; this is part of the space after the directory!). The machine language program is started by the GO instruction at line 2180. This loads the series of instructions found in the right side of table 2 in direct mode (i.e. without line numbers) into the area from \$8000 on. From line 2190 to line 2290 the BASIC program places the names of the files that are to be merged (F\$ and S\$) in direct mode after the two LO instructions that have just been loaded. The instruction at line 2300 programs the distributor to make the memory the input device. The BASIC editor then receives the sequence of instructions starting at \$8000 as if they were input one-by-one via the keyboard and it then executes them one after the other. What this means is that it loads file F\$, transfers it to \$8000 (LIST 5), and then loads file S\$ and transfers it, in turn, to the space after F\$. It then executes the DISK! "GO E452" instruction which is the last it receives in direct mode from the memory as an input device.

The machine-code program at \$E452 places a POKE 8993,1 instruction in direct mode after the two files loaded at address \$8000 and as this instruction has no line number it will be executed as soon as the interpreter meets it. The purpose of this last command is to reestablish the input

distributor in its original form where the keyboard is the input device. Now the BASIC editor loads files F\$ and S\$ into its workspace to form a single new file which it compacts and lists as it goes along. When it arrives at the last numbered line in the second file it finds the POKE 8993,1 instruction which it executes in direct mode thus making the keyboard again the active input device. If a LIST instruction is now given the display on the screen will show that the workspace does, in fact, contain files F\$ and S\$.

### RSEQ

In order to be able to effectively merge existing files it is essential to be able to easily manipulate the numbering of the lines in both files and then later of the single file resulting from the merger. On disk 2 of the 5 supplied with the Ohio Scientific DOS is a utility program called RSEQ that could be used to perform this task. Until now none of the myriad articles on the various aspects of the Junior Computer have dealt with adapting disk 2 for the Junior. The hexdump given in table 3 does just that, enabling JC users to easily change the line numbering of BASIC files, especially those that are to be merged. The adaptation procedure is quite simple. First copy the master diskette (this is always advisable as a safeguard) and then load track 0 of disk 2 by means of the TRACK 0 R/W UTILITY (RA200) at address \$A200 (or elsewhere). The contents of this track must then be changed according to the hexdump in table 3 and the modified first page of track 0 is then reloaded to the diskette (WA200/2200,1). And that's all, folks!