NAME CALLING

Having looked in more detail at the way a BASIC program is stored, we can now extend the variable search program to include a facility to replace one variable name by another. Here we look at the BBC Micro and the Commodore 64 versions; in the next instalment we will develop the same program for the Spectrum.

Our variable replace program is a more demanding utility than the simple search for variable names that we developed on pages 664 and 700. For this reason we need to add a machine code program. The BBC Micro's 6502 CPU and the Commodore 64's 6510 CPU have the same Assembly language, so it is a good idea to look at them together.

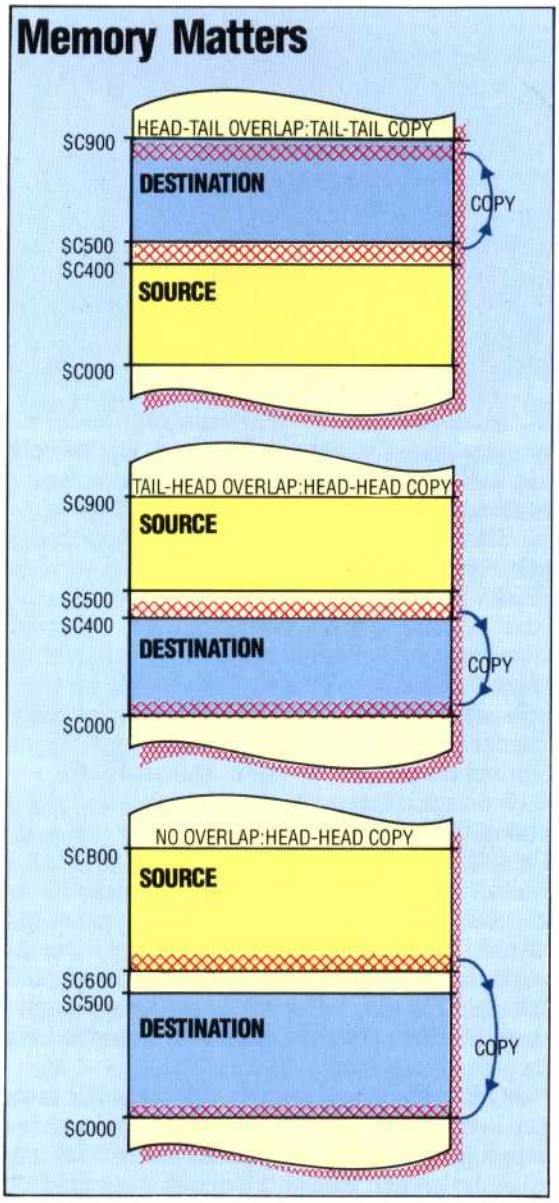
Our first task is to find a method of holding two separate programs in the computer at the same time. As we have already explained, we can do this on the BBC Micro by altering the built-in BASIC variables PAGE and HIMEM. On the Commodore 64, we need a machine code program to alter the various pointers in zero page memory. The first part of the Assembly language listing, beginning at the label SWITCH, will do this for us.

The routine SWITCH will enable us to accommodate two BASIC programs: one beginning at address 800 hex (the usual place for a BASIC program); and the other beginning at address 9000 hex. SWITCH begins by looking at the pointer TXTTAB to see which of the program areas is current, and then changes the pointer values to make the other program area current.

TXTTAB is changed to point to the start of the new program area, then FRETOP and MEMSIZ must point to the byte after the last byte of the new program area, while FRESPC points to the end of the new program area. The program then searches down the chain of link address pointers (see page 704) to find the end of the BASIC program, using VARTAB as the temporary pointer. When it finds the two zero link address bytes that mark the end of the program, it increments the previous pointer twice and copies the result into ARYTAB and STREND. In this way VARTAB, ARYTAB and STREND all point to the byte immediately after the BASIC program.

The main changes to the BASIC program that we need to make are the extra subroutines at lines 30500 and 30600. The first of these finds the end of the BASIC program, using the length of line bytes in the BBC version and the next line pointers in the Commodore 64 version.

The subroutine at line 30600 actually makes the change in the variable names. When the old and new variable names are the same length, the new name can simply be written over the old. Where the old and new variable names have different lengths, the procedure is a little more complicated. In this case, the program must either make extra space, or close up any unneeded space in the program it is changing, and make corresponding changes to the variables it uses to keep track of its position in the program being altered. It must also change the length of line byte in the BBC version and the next line pointers in the Commodore 64 version.



Intelligent Copy
The replace routine has to move large sections of the BASIC program up and down in memory when it inserts new variable names of different lengths. In this it encounters the four possible conditions of the source and destination addresses. If it always copies from the start of the source block then, when the head of the destination block overlaps the tail of the source block, the copying will overwrite some of the source data. An 'intelligent' copy routine will detect this case and avoid corruption by copying this source block from the tail first. A 'dumb' copy always copies from the head of the source block