

the source listing, the address of the next section will be inserted into the instruction which precedes (on the same line) the argument LINK. (Remember, LINK is an illegal label, and in fact this is the only time where an argument will not have a corresponding label in the source listing.) The argument LINK follows all the rules for other arguments -- it may be used as many times as necessary. Every time LINK is encountered as an argument, the address of the next section to come will be inserted in that instruction.

Provided the various sections do not have to be in any particular order to run -- a possibility when you have certain initializing routines you use often that you would rather not turn into subroutines -- the sections may be linked without assigning a label to each one.

In the case where one section will be called vs. the next section depending on the outcome of a certain test or key press, a skip instruction could specify a jump (or a subroutine call, or an AXXX set "I" instruction) to either a particular labeled program section or the first instruction of the next section via the link. The first line of such a section may also be labeled with no effect on the link, providing a method of jumping around in memory (and a