

Design of absolute loader

With an absolute loading scheme, the programmer and the assembler perform the task of allocation, relocation and linking. Therefore, it is only necessary for the loader to read cards of object deck and move the text on the cards into the absolute locations specified by the assembler.

There are two types of information that the object deck must communicate from the assembler to loader. First, it must convey the machine instructions that the assembler has created along with the assigned core locations. Second, it must convey the entry point of the program, which is where to transfer the control where all the instructions are loaded. Assuming that this information is transmitted on cards, a possible format is shown below:

Text Cards (for instructions and data):

Card Column	Contents
1	Card type = 0 (for text card identifiers)
2	Count of number of bytes (1 byte/column) of information on cards
3 - 5	Address at which data on cards is to be put
6 - 7	Empty (Could be used for validity checking)
8 - 72	Instructions and data to be loaded
73 - 80	Card Sequence number

Transfer cards (for entry point to the program):

Card Column	Contents
1	Card type = 1 (Transfer card identifier)
2	Count = 0
3 - 5	Address of Entry point
6 - 72	Empty
73 - 80	Card Sequence number

Algorithm:

The algorithm for an absolute loader is simple. The object deck for the loader consists of series of text cards terminated by transfer card. Therefore, the loader should read one card at a time, moving the text to the location specified on card, until the transfer card is reached. At this point, the assembled instructions are in core and it is only necessary to transfer the entry points specified on the transfer card. A flowchart now follows.

Flow Chart for absolute loader design

