Linker Loader 2015

An algorithm for linking, then loading, with relocation:

Pass One: Create the **External Symbol Table**

BEGIN:

Get the program load address, ProgAddr from the Operating System

Set the subprogram address, SubAddr, to the value of the ProgAddr

While not the end of the input

    Read the H-Header record

    Assign the Subprogram Length value to SubLgth

    Add the Subprogram Name to the External Symbol Table, assign the value from SubAddr

    While not E-End record

        For each D-Define record

            Add each Symbol to the External Symbol Table, and assign the value from SubAddr plus the Value from the Define record

        End (For each)

    End (while not E-Record)

    Add the SubLgth value to the SubAddr value (= address at the start of next subprogram)

End (while not end of input)

The result of this Pass One is an **External Symbol Table** with a list of each of the external symbols and the symbol's assigned address.

Pass Two: Load and relocate the text content into memory

BEGIN:

Get the program load address, ProgAddr from the Operating System

Set the subprogram address, SubAddr, to the value of the ProgAddr

While not the end of the input

    Read the H-Header record

    Assign the Subprogram Length value to SubLgth

    While not E-End record

        For each T-Text record

            Copy the object code text into memory at the indicated Address plus the value of the SubAddr.

        End (For each)

        For each R-Relocate record (or M-modify record)

            Modify the object code text in memory at the indicated Address with the (+/-) value of the indicated External Symbol. The value is from the External Symbol Table.

        End (For each)

    End (while not E-record)

    Add the SubLgth value to the SubAddr value (= address at the start of next subprogram)

End (while not end of input)

Adapted from L. Beck, System Software, 1997