Identification

format of "object" segments

C. Garman

Purpose

This document describes the format used to combine several intrinsically related segments into one "object" segment with the consequent savings in number of segments, segment overhead, and "breakage" of core/disk/drum pages (many linkage segments occupy less than 128(10) words, and few are longer than 256 words, thus the average "virgin" linkage section would tie up between 800 and 900 words that are not otherwise needed).

Criteria

The following criteria were used in developing the format:

1. Procedures should be directly executable without relocation; similar conditions apply to data segments referenced through the Multics linkage mechanism
2. The number of segments which may be combined should not be restricted (although restrictions may be placed on the sequence of the components.)
3. The technique for decoding the encoded segment into its original components should be fast and idiot-proof.

Criterion (1) was met by locating the text segment first, starting at location 0 of the segment. (Note that linkage segments are intrinsically self-relocating; while relocation bits exist for moving the text segment around, it is not done lightly).

Items 2 and 3 are met by the variable-length map (including a recognizable pattern) described later; the only condition is that the text, link and symbol segments occupy the first, second, and third positions in the segment, respectively or else have dummy place holders provided for them.

Conventions

Any number of "related" segments may be placed in one object segment, but in order for proper treatment to be accorded by the linker, binder, etc, certain conventions must be observed:

A. The order of the components shall be as follows (the list may be extended from time to time):

1. "text" segment (starting at location 0)
2. ".link" segment (must begin on even-word boundary) (ref ____)
3. ".symbol" segment (including relocation bits). (ref ____)

(For consistency, all components should start on even-word boundary)

B. Trailing components may be ommitted (e.g., the ".symbol" for a non-bindable data segment with inward references, but if an interior component is ommitted, at least one (two?) word(s) of "buffer" must be provided. (In the case of a null linkage section, for compatability with current practice, this should be 9 or 10 words, with "defs-in-link", and a null external-definition chain.)

Format

The object segment consists of the juxtaposition (subject to the restrictions and conventions above) of the component segments, followed by a map of

length 2n + 5 words, as follows:

Words 0-3: Binary pattern of alternate 1's and 0's, (starting with the first bit 1)

Word 4: fixed binary relative pointer to first component segment. (i.e., value is 0)

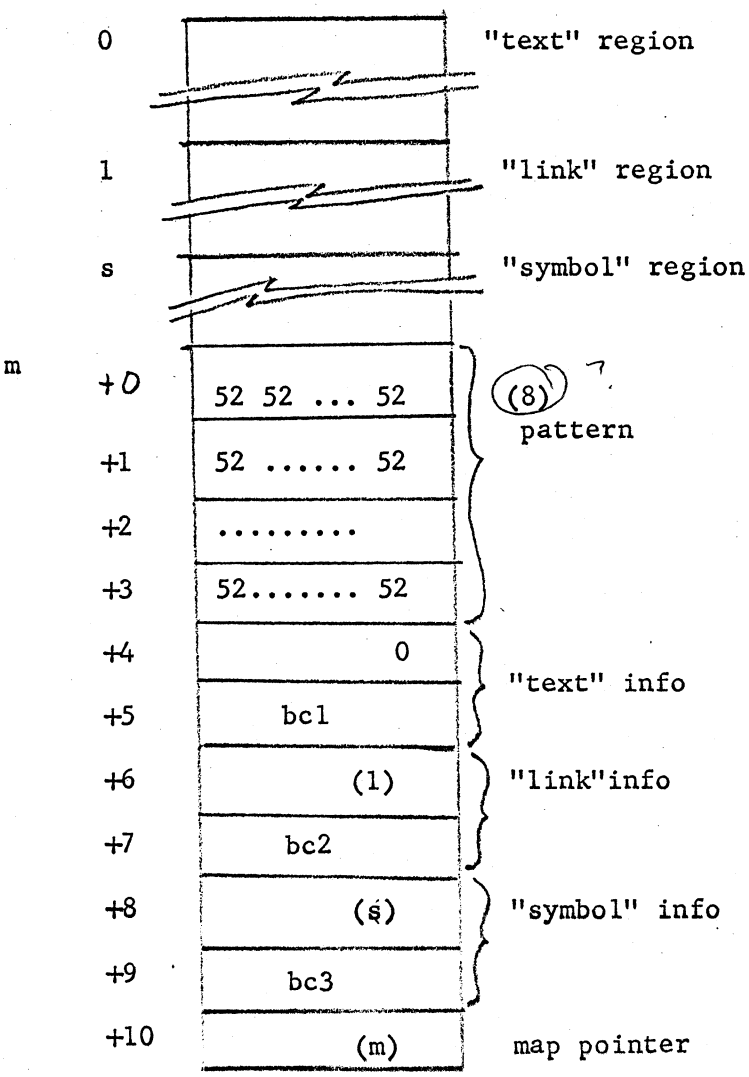
Word 5: fixed binary bit-count of first segment

Word 2i+2 fixed binary relative pointer to ith component

Word 2i+3 bit count for ith component.

Word 2n+4 fixed binary relative pointer to word 0 of the map

Figure 1. Example for text, link, symbol object segment



Decoding

The decoding process is quite simple, given a pointer to the base of the segment and the number of bits or words encompassed: (In fact, if the addrel function is used throughout the procedure, the pointer may be to an interior region of a segment, as in an "archive" segment.)

1. The length must be greater than some minimum: 9 words or 324 bits(2 words for 1 encoded segment, plus 7 words for the header)
2. Obtain the last word of the region, check this value for being greater than the length, or ≤ 0.
3. Form a pointer to the map by adding the value in (2) to the original pointer.
4. Check for the proper bit pattern in words 0-3
5. Extract the offset word for the current (first) component, checking that it be in a strict ascending sequence from the previous value.
6. Repeat step (5) until the map pointer is obtained or until the desired component has been decoded, (unless, of course, the check in step 5 fails).

Note that this is a sequential, but heuristic algorithm: any failure in any step will give an indication that the segment is not in object format, while the probability that a random piece of data that is not in object format successfully passing all the tests is so low as to be negligible.