

Published: 05/21/68
(Supersedes: BX.9.02, 02/18/68)

Identification

Print a Segment in ASCII
print
K. Martin, C. Garman

Purpose

The print command takes a segment composed of ASCII text and writes it in the user's output stream (usually on a console). The print command assumes that new line characters are embedded in the text appropriately and makes no provision for lines that may be too long to be printed.

Usage-as a command

```
print segname -lineno-
```

where segname is the path name of the segment to be printed. The path name may be relative to either the root directory (in which case the first character is >) or the user's current working directory. The optional argument lineno (typed in as ` ` when not wanted) is the line number of the first line to be printed.

Usage-from a program

```
call print$ptr(p, count, line);  
dcl p ptr, (count, line) fixed bin (17);
```

This call is provided for commands which themselves need to print out a segment, where the pointer is already known and where other constraints may apply (one of these is the mail command).

p is a pointer to the base of the segment to be printed, count is the number of characters to be printed from the segment, and line is the number of the first line to be printed (its value should be zero in most cases).

Implementation

The print command initiates the segment segname via the Segment Management Module. If no such segment exists or is of zero length, print comments to the user and returns. Print assumes that the bit count on the branch is correct (a safe assumption for ASCII text segments prepared using the context editor (BX.9.01) or other commands). Print could simply treat the entire segment (of length bit-count/9 characters) as a single character string and call write_out (BY.4.02) to place the contents in the output stream.

However, since `write_out` copies its argument into its own stack frame, one could easily run out of stack, thus `print` chews through the segment in chunks of `maxi` characters at a time (`maxi` is changeable, but 400 seems a reasonable, though arbitrary, number). It calls `write_out` with specifier (created by `cv_string---`BY.10.03) pointing at the current chunk (actually the chunk ends, if possible, at the last `new_line` character before the `maxi`th character from the last chunk). In case of errors, `print` comments to the user and returns. Before printing the segment, a short header is printed, consisting of the segment's name with blank lines surrounding it.

If `lineno` was present and non-null, its value is converted to binary, and printing does not start until the `lineno`'th line is found. If fewer than `lineno` lines exist in the segment, an error comment is printed, but nothing from the segment will have been printed. Existence of the `lineno` argument suppresses the printout of the header. Entry at `print$ptr` also suppresses the header printout.