

Published: 07/17/68

Identification

Return a segment to CTSS from a 6.36 execution activity

write\_seg

J. M. Grochow

Purpose

It is often desirable to retrieve segments created during a 6.36 execution activity. write\_seg provides a simple interface between an EPL program and 6.36 pseudo-process output routines.

Usage

write\_seg has 3 entry points. Their usage is as follows:

```
call write_seg$init;
```

This call should only be made once and should precede any other calls to write\_seg.

```
call write_seg$write (segname, write_name, status_ptr);
```

where segname= character string representation of segment to be written

```
decl segname char(*);
```

write\_name=6 character (or less) name of file to be returned to CTSS

```
decl write_name char(*);
```

status\_ptr=pointer to seg\_util-like status array (see BY.2.12).

```
decl status_ptr ptr;
```

This call should be made once for each segment to be written (text, link, and symbol sections are written at each call).

```
call write_seg$final;
```

This call should be made only after all segments have been written by calls to write\_seg\$write.

In order for files to be returned to CTSS, it is necessary to specifically request their return by inclusion of the following card in the GECOS file used for that run:

FETCH write\_name TL

where write\_name is the same as above except that "\_" (underscore) has been changed to "-" (hyphen).

Segments thus written may also be punched by including the following card in the GECOS file:

DECK write\_name

### Implementation

write\_seg\$init and write\_seg\$final simply make appropriate calls to pseudo\_process\_io and pseudo\_process\_section\_output.

write\_seg\$write will first change any "\_" in write\_name to "-". It will then check the segment pointers in the status array pointed to by status\_ptr. Text, link and symbol segments will be written by calls to pseudo\_process\_section\_output if their corresponding pointers are non-null. A word count is determined by dividing the bit\_count in the status array by 36.