

Published: 01/10/68

Identification

Shared use of GIOC in a 645 Pseudo-Process
G. S. Stoller, M. A. Padlipsky

Purpose

When it is necessary to share a single GIOC between GEGOS and a Multics pseudo-process (e.g., in a 645 configuration with only one GIOC, or when only one GIOC is currently available), the GECOS Supplement (see BE.7.12) must be informed of the situation so that it knows that a pseudo-process will be using the GIOC and so that it will dump the Multics portion of the GIOC mailbox and halt the Multics I/O when the pseudo-process terminates. Note that only those pseudo-processes which perform their own I/O and which must do so with only one GIOC in the system are involved here. Three segments described herein establish the communication between the pseudo-process and the Supplement through 635 escape coding.

Use

1. Load the following three segments, with the attributes listed:

<set_gioc_pseudo_mailbox>	master procedure, slave access
<data_gioc_pseudo_mailbox>	data, slave access, preferably unpagged
<gioc_pseudo_mailbox>	data, unpagged

2. Perform a call to

```
<set_gioc_pseudo_mailbox>|[set_gioc_pseudo_mailbox]
```

before any I/O is done; this is necessary so that the Supplement can set up for a shared GIOC. No arguments are given in this call.

When the pseudo-process is terminated, the Multics part of the GIOC mailbox is copied into <gioc_pseudo_mailbox> by the Supplement and all Multics I/O is immediately terminated.

Note that the Multics portion of the mailbox is copied directly into `<gioc_pseudo_mailbox>`. Therefore, when reading the dump, those control words in the mailbox which pointed to addresses within the mailbox itself must be mentally translated to point to the appropriate relative positions in the copy.

Implementation

`<gioc_pseudo_mailbox>` is a segment that provides space in core storage to which the supplement moves the Multics part of the GIOC mailbox when the pseudo-process is terminated.

`<set_gioc_pseudo_mailbox>` is a mastermode procedure which passes the absolute location of `<gioc_pseudo_mailbox>` to the Supplement. It must be called in the pseudo-process before any Multics I/O is done. When entered, this procedure gets the SDW of `<gioc_pseudo_mailbox>` and tests to see if it is paged. If paged, an error message is given and the pseudo-process is terminated. If not paged, the argument list for the Supplement is created in `<data_gioc_pseudo_mailbox>` and `<escape>` is called, which escapes to STPSMB, a 635 escape coding subroutine. This subroutine passes to the Supplement the argument list set up by `<set_gioc_pseudo_mailbox>`. (The argument list is described in BE.7.12, under "Location of Multics GIOC Mailbox".)

`<data_gioc_pseudo_mailbox>` is a data segment used by `<set_gioc_pseudo_mailbox>` to pass arguments to procedures that it calls. It is used only to avoid the necessity of copying data from `<set_gioc_pseudo_mailbox>` into `<stack>`. (`<set_gioc_pseudo_mailbox>`, being a master procedure, is not accessible as data to a slave procedure).