TO: MSPM Distribution
FROM: D. L. Stone
SUBJECT: BF.20.03
DATE: 01/30/68

This revision mirrors the updating of BF.20.01 and adds the declaration of the data bases:

   cat_cst
   CCT
   DCT
   Status_segment

A new argument has been added to the assign call.
Identification

A Summary of GIM Calls and Data Bases
S. D. Dunten, T. P. Skinner, D. L. Stone

Purpose

This section provides a summary of all external calls to the GIOC Interface Module (GIM). EPL declarations for the parameters are given, as well as for Data Bases referenced.

The GIM Interface

All the calls specified below are accessible from ring 0 only. To call the GIM from other rings, the "hcs_@entry name" mechanism must be used.

Call giminit$assign(name, devx, event, type, rcode);

    dcl name char (32),  /* symbolic channel name */
    devx fixed bin(17),  /* device index returned by the GIM for this channel */
    event bit(70),       /* passed on to the DSTM for device signalling */
    type char(*),        /* passed on to IO assignment module */
    rcode fixed bin(17); /* error return, 0=ok */

Call giminit$fsassign(name, devx, rcode);

Call giminit$list_size(devx, listsz, rcode);

    dcl devx fixed bin(17), /* device index from assign */
         listsz fixed bin(12), /* size of list to be allocated */
         rcode fixed bin(17), /* error return */
Call gim$list_change(devx,dcwp,datap,listx,count,rcode);

dcl devx fixed bin(17),
    (dcwp, datap) ptr,
    (listx, count)fixed bin(17); /* pointer to "dcw_array" */
    /* pointer to "data_array" */
    /* index within DCW_list, (starting with 1) of first DCW to be changed */
    /* number of elements in "dcw_array" which are to be used */

where the data referred to by "dcwp" and "datap" is accessed as:

dcl dcw_array(count)bit(72)based(dcwp),
    /* a list of real DCW's */
    data_array(count)based(datap),
    /* pointer to user workspace
       if nth DCW is data DCW */
    p ptr,
    /* "10"b = readDDCW
       "01"b = writeDDCW */
    rw bit(2);

(Note: for privileged users, datap should be null).

Call gim$list_connect(devx,ciw,listx,rcode);

dcl devx fixed bin(17),
    ciw fixed bin(18), /* the CIW is treated as an 18 bit string which is the right half of a CIW */
    listx fixed bin(12),
    rcode fixed bin(17);

Call gim$get_cur_status(devx, listx, dcwt, rcode);

dcl devx fixed bin(17), /* current index of dcw being processed */
    listx fixed bin(12),
    dcwt fixed bin(12), /* current dcw tally */
    rcode fixed bin(17);

Call gim$get_status(devx,status_array_ptr,array_size,outsize,
    waiting,rcode);

dcl devx fixed bin(17),
    status_array_ptr ptr,
    (array_size, outsize)fixed bin(17), /* pointer to "status_array" */
    /* maximum number of status elements to be returned */
    /* actual number returned */
    waiting fixed bin(17), /* count of status words waiting but not returned in this call */
    rcode fixed bin(17);
The Channel Assignment and Channel Status Table

This segment is the major system-wide GIM data base. It is wired down and contains all per-user channel information and per-gioc information.

dcl 1 cat_cst based(p),
2 chan (0:260),  /* per-device-index */
    /* information accessed */
    /* by the "devx" */
    /* presented in the GIM */
    /* calls */
3 cctno bit (18),  /* segment number of the */
    /* CCT for this user */
    /* -only accessed by one */
    /* process */
3 dcw_re1_add bit (18),  /* offset of dcw list */
    /* within dcw segment */
    /* Zero is interpreted */
    /* as dcw-list not */
    /* yet allocated unless */
    /* "priv" is on */
3 dcw_list_len bit (12),  /* size of dcw list in */
    /* dcw's */
    /**/ 3 giocno bit (2),  /* which gioc is being */
    /* used for this channel */
    /**/
3 conno bit (2), /* which connect */
   /* channel are we using */
   ////
3 priv bit (1), /* on if user can */
   /* supply absolute */
   /* address in data DCw's */
3 dir_chan bit (1), /* on if direct channel */
   ////
3 channo bit (12), /* physical channel */
   /* number */
   ////
3 status_lost bit (1), /* ON if status lost */
   ////
3 padl bit (5), /* guess again */
   ////
2 giocl (2), /* per-gioc information */
   ////
3 mbx_base ptr, /* pointer to mailbox */
   /* area */
   ////
3 portno bit (3), /* which port is this */
   /* giocl on */
   ////
3 stat_base bit (3), /* offset for use in */
   /* avoiding bad status */
   /* channels */
   /* GIM uses only one */
   /* status channel */
   /* normally */
3 connect (0:2), /* connect channel info */
   ////
4 lock bit (36), /* uses locker */
4 ccw bit (36), /* ccw to be connected */
   /* is physically here */
4 cpw bit (72), /* fixed pointers to the */
   /* ccw's above */
   ////
3 status (0:1), /* status channel info */
   ////
4 lock bit (36), /* standard lock */
4 scwa bit (36), /* status channel word */
   /* A-refill */
4 oldest fixed bin (17), /* index to oldest */
   /* status */
4 basep ptr, /* ptr to top of status */
   /* queue */
4 intp fixed bin (17), /* index to last */
   /* interrupt processed */
4 thresh fixed bin (17), /* min. amount of free */
   /* stat. storage needed */
4 endindex fixed bin(17), /* size in status words */
The Channel Copy Table

This table is allocated as a separate segment for each user of the assign call and contains information about the DCW lists for each user.

\[
\begin{align*}
dcl & 1 \text{ cct based}(p), \quad \text{/* index of the last */} \\
& 2 \text{ copy_dcw fixed bin (12), \quad \text{/* dcw for which copying */} } \\
& 2 \text{ copy_word fixed bin (12), \quad \text{/* word for the "copy_dcw" */} } \\
& 2 \text{ nreads fixed bin (12), \quad \text{/* number of read */} } \\
& 2 \text{ nalloc fixed bin (12), \quad \text{/* number of buffers */} } \\
& 2 \text{ addr_list (4096), \quad \text{/* allocated to fit */} } \\
& 3 \text{ wksp ptr, \quad \text{/* size of wired down */} } \\
& 3 \text{ length_allocated fixed Bin (18); \quad \text{/* buffer allocated */} }
\end{align*}
\]

The Device Configuration Table

This per-system table describes all of the channels attached to all of the GIOC's which are administered by the GIM.

\[
\begin{align*}
dcl & 1 \text{ dct based}(p), \quad \text{/* device configuration table */} \\
& 2 \text{ ndev fixed bin(17), \quad \text{/* number of devices */} } \\
& 2 \text{ desc(300 /* dev_nam_max */ ), \quad \text{/* start of device description */} } \\
& 3 \text{ dev_nam char(32), \quad \text{/* device name */} } \\
& 3 \text{ phys_nam char(32), \quad \text{/* name of physical channel and GIOC */} } \\
& 3 \text{ devx bit(17), \quad \text{/* device index */} } \\
& 3 \text{ giocno fixed bin(17), \quad \text{/* GIOC number of this device */} } \\
& 3 \text{ phychn fixed bin(17), \quad \text{/* half the LPW channel number of this device */} } \\
& 3 \text{ conno fixed bin(17), \quad \text{/* symbolic connect channel number */} } \\
& 3 \text{ direct_chan bit(1); \quad \text{/* ON if direct channel */} }
\end{align*}
\]
The Status Segment

All hardware stored status is kept in a single wired down segment. Per GIOC, two status queues are kept - one for channel 0 and one for user status; associated with each status queue is a time queue administered by the interrupt handler. The base of the various status queues is determined from the cat_cst.gioc(N).status(M).basep entries in the cat_cst. Given the base of a queue, the queue can be accessed by:

```
dcl 1 status_q based (basep),
   2 status (0;endindex /*from cat_cst*/)bit(36),
      /*ModA GIOC*/
   2 pad bit(72),
   2 time (0;endindex),
   3 t bit(52),
   3 pad bit(2);
```

Buffer Areas

In addition to the above tables, one wired down contiguous segment is used to store all dcw lists and data buffers associated with dcw's. Each buffer is individually allocated within the segment as an area.