

Published: 12/09/68
(Supersedes: BQ.6.07, 07/27/67)

Identification

The Device Signal Table Manager (DSTM)
Michael J. Spier

Purpose

In Multics, every I/O device is attached to a single process (which may however attach more than one device); when an I/O interrupt occurs, the strategy is to translate that interrupt into a software signal (wakeup) which wakes up the process which is responsible for the interrupting device.

A wired-down hardcore table, named the Device Signal Table (DST, see SECTION BJ.1.06) has an entry per I/O device and links each device to a process, providing the necessary link for the interrupt-to-wakeup conversion.

The Device Signal Table Manager (DSTM) is a collection of wired-down hardcore procedures to maintain the DSTM.

Introduction

DST entries are threaded into per-process device lists; the head of every such list is kept in the Active Process Table (APT) entry of the responsible process. The lists have a forward thread only; also, every DST entry points individually back to the responsible process' APT entry. Entries in the DST can be accessed by using the interrupting device's number as an index (device indexes are arbitrarily assigned to all I/O devices during system initialization). Thus, it is possible for the appropriate interface module (DIM, GIM) to find the responsible process by looking up the DST with the device index and finding there the backpointer to the process' APT entry. Likewise, the process can easily find the DST entries for its currently-attached devices.

Calls to the DSTM

```
call dstm$attach(devindex, mode, cname, code);  
declare (devindex, mode, code) fixed bin(71);
```

attaches the I/O device whose index is devindex to the calling process (which is implicitly known) by putting mode and cname into the appropriate DST entry and by threading that entry into this process' device list as described above.

The process interrogates its DST entries at intervals, and whenever it does it resets an interrupt count associated with that entry; mode is a flag which is stored in the DST entry and which specifies whether all incoming interrupts should be converted into wakeups and remembered in the interrupt count, or whether only the first incoming interrupt (after the interrupt count has been reset to zero) should be signalled and all subsequent ones ignored. The latter mode is the one which is normally used,

mode=0 -> count only the first interrupt after
interrupt-count has been reset.

mode=1 -> count all interrupts

event channel is the name of an event channel which belongs to this process and over which the process expects to receive the device-signal wakeup.

dstm\$attach returns an error code and does nothing in the following cases:

code=1 erroneous device index (out of DST boundaries)

code=2 device already attached to some process.

```
call dstm$detach(devindex)
```

detaches the DST entry pointed to by devindex from its present owner and resets the whole entry to zero.

```
call dstm$check_auth(devindex, processid);
```

```
declare processid bit(36)
```

looks up the DST entry corresponding to devindex and returns the ID of the process to which this device is currently attached (or a zero-value process ID if device is detached).

The above-mentioned conversion of interrupts to wakeups is done not by the DSTM but by the Traffic Controller itself in entry point `pxss$dst_wakeup`.