

Published: 10/01/68
 (Supersedes: BJ.1.04, 11/03/67,
 BJ.1.04, 03/15/67)

Identification

The APT Hash Table

Robert L. Rappaport, Michael J. Spier, A. Evans

Purpose

Communication between processes is done on the basis of (symbolic) process identification, but the communication itself requires knowing the location of the target-process' APT entry. To make looking up the APT, given a process id, efficient, a hash-table of process ids is maintained by the Traffic Controller. APTSH is a typical hash table associating process ids and relative pointers into the APT.

Description

The APTSH is declared as follows:

```
declare 1 apthsh(n) based(aht_ptr),
        2 process_id bit(36),
        2 pointer bit(18);
```

An entry in the APTSH can be one of the following three types depending upon its value:

- a. empty entry (process_id="0"b) & (pointer="0"b)
- b. "skip" entry (process_id="0"b) & (pointer="0"b)
- c. valid entry (process_id="0"b) & (pointer="0"b)

The Traffic Controller has three subroutines dedicated to the manipulation of the APTSH.

A call to enter (process_id, pointer), where both arguments are input arguments, will allocate an APTSH entry to process_id and associate it with pointer.

A call to lookup (process_id, pointer), where the first is an input argument, will lookup the hash table for process_id; if found it returns the pointer associated with process_id, else it returns a null-pointer.

A call to delete (process_id) looks up the table for an entry containing process_id. When found, it zeroes out the entry's process_id thus making it into a skip entry. If the immediately following entry is a valid one, the subroutine returns, else it scans forward until it encounters the first non-skip entry, then backs-up again and goes backwards resetting all the skip entries to zero until it encounters a non-skip entry. Both subroutines lookup and enter treat skip entries as if they were non-valid entries. Subroutine lookup abandons the search as soon as it encounters an empty entry.

The mapping of process-ids into the hash table is done by simply using the process-id's low-order bits as index into the hash table; this is sufficient because process-ids are unit-distant from one another (they are generated by simply incrementing some system counter.)

APTHSH and its subroutines are internal to the Traffic Controller and cannot be reached externally.