Identification

Process Group creation and destruction
C. Marceau

Purpose

The two entries described in this section respectively create and destroy process groups. They are used by system procedures to create and destroy system process groups and by the overseer to create and destroy user process groups as users log in and out.

Usage

To create a process group

```plaintext
call process_group$create (name, projid, tag, status);
dcl (name, projid) char (24),
    tag char (2),
    status fixed bin (17);
```

To destroy a process group

```plaintext
call process_group$destroy (name, projid, tag, status);
```

Implementation

`process_group$create` calls `append_branch` to create a group directory with the name indicated by its caller and creates a segment "processes" in that directory. It then calls `mode$set` to permit the directory and the segment "processes" to the new process-group. The caller of `process_group$create` is responsible for creating processes in the new group.

If anything goes wrong, `process_group$create` returns as status the status code returned to it by `append_branch` or `mode$set`. A status of zero indicates that all is well.

`process_group$destroy` calls `delete_subtree` to destroy the group directory and all segments in that directory. Before doing this it checks the segment "processes". The segment has the format of a PL/I structure:
dcl 1 processes based (p),
2 n_proc fixed bin (17),
2 process_id (p->processes.n_proc);

That is, the segment is a list of the processes in the
group. It is updated by create_proc and destroy_proc
whenever a process is created or destroyed. Process_group$destroy
checks that there are no processes in the group before
destroying the group directory by a call to delete_entry.

If there are processes in the group process_group$destroy
returns status = -1. Otherwise it returns whatever status
delete_entry returned to it. A status of zero means that
all is well.