NAME

priocntlset – generalized process scheduler control

SYNOPSIS

#include <sys/types.h>
#include <sys/procset.h>
#include <sys/priocntl.h>
#include <sys/rtpriocntl.h>
#include <sys/tspriocntl.h>

long priocntlset(procset_t **psp*, **int** *cmd*, **/*** *arg* ***/ . . .);**

DESCRIPTION

priocntlset( ) changes the scheduling properties of running processes. **priocntlset( )** has the same functions as the **priocntl( )** function, but a more general way of specifying the set of processes whose scheduling properties are to be changed.

cmd specifies the function to be performed. *arg* is a pointer to a structure whose type depends on *cmd*. See **priocntl**(2) for the valid values of *cmd* and the corresponding *arg* structures.

psp is a pointer to a **procset** structure, which **priocntlset( )** uses to specify the set of processes whose scheduling properties are to be changed. The **procset** structure contains the following members:

```
idop_t      p_op;       /* operator connecting left/right sets */
idtype_t    p_lidtype;  /* left set ID type */
id_t        p_lid;      /* left set ID */
idtype_t    p_ridtype;  /* right set ID type */
id_t        p_rid;      /* right set ID */
```

p_lidtype and **p_lid** specify the ID type and ID of one ("left") set of processes; **p_ridtype** and **p_rid** specify the ID type and ID of a second ("right") set of processes. ID types and IDs are specified just as for the **priocntl( )** function. **p_op** specifies the operation to be performed on the two sets of processes to get the set of processes the function is to apply to. The valid values for **p_op** and the processes they specify are:

POP_DIFF set difference: processes in left set and not in right set

POP_AND set intersection: processes in both left and right sets

POP_OR set union: processes in either left or right sets or both

POP_XOR set exclusive-or: processes in left or right set but not in both

The following macro, which is defined in **procset.h**, offers a convenient way to initialize a **procset** structure:

```
#define setprocset(psp, op, ltype, lid, rtype, rid) \
(psp)→p_op      = (op), \
(psp)→p_lidtype  = (ltype), \
(psp)→p_lid     = (lid), \
(psp)→p_ridtype  = (rtype), \
(psp)→p_rid     = (rid),
```

RETURN VALUES

Unless otherwise noted above, **priocntlset( )** returns a value of 0 on success. **priocntlset( )** returns –1 on failure and sets **errno** to indicate the error.

ERRORS

priocntlset( ) fails if one or more of the following are true :

EAGAIN An attempt to change the class of a process failed because of insufficient resources other than memory (for example, class-specific kernel data structures).

EFAULT One of the arguments points to an illegal address.

EINVAL The argument *cmd* was invalid, an invalid or unconfigured class was specified, or one of the parameters specified was invalid.

ENOMEM An attempt to change the class of a process failed because of insufficient memory.

EPERM The effective user of the calling process is not super-user.

ERANGE The requested time quantum is out of range.

ESRCH None of the specified processes exist.

SEE ALSO

priocntl(1), **priocntl**(2)