NAME

setclasscontext, **setclasscpumask**, **setclassenvironment**, **setclassresources**, **setusercontext -** functions for using the login class capabilities database

LIBRARY

System Utilities Library (libutil, -lutil)

SYNOPSIS

#include <sys/types.h>
#include <login_cap.h>

int
setclasscontext(*const char *classname, unsigned int flags*);

void
setclasscpumask(*login_cap_t *lc*);

void
setclassenvironment(*login_cap_t *lc, const struct passwd *pwd, int paths*);

void
setclassresources(*login_cap_t *lc*);

int
setusercontext(*login_cap_t *lc, const struct passwd *pwd, uid_t uid, unsigned int flags*);

DESCRIPTION

These functions provide a higher level interface to the login class database than those documented in login_cap(3). These functions are used to set resource limits, environment and accounting settings for users on logging into the system and when selecting an appropriate set of environment and resource settings for system daemons based on login classes. These functions may only be called if the current process is running with root privileges. If the LOGIN_SETLOGIN flag is used this function calls setlogin(2), and due care must be taken as detailed in the manpage for that function and this affects all processes running in the same session and not just the current process.

The **setclasscontext**() function sets various class context values (resource limits, umask and process priorities) based on values for a specific named class.

The **setusercontext**() function sets class context values based on a given login_cap_t object and a specific passwd record (if login_cap_t is NULL), the current session's login, and the current process user and group ownership. Each of these actions is selectable via bit-flags passed in the *flags* parameter, which is comprised of one or more of the following:

LOGIN_SETLOGIN Set the login associated with the current session to the user specified in the passwd structure using setlogin(2). The *pwd* parameter must not be NULL if this option is used.

LOGIN_SETUSER Set ownership of the current process to the uid specified in the *uid* parameter using setuid(2).

LOGIN_SETGROUP Set group ownership of the current process to the group id specified in the passwd structure using setgid(2), and calls initgroups(3) to set up the group access list for the current process. The *pwd* parameter must not be NULL if this option is used.

LOGIN_SETRESOURCES Set resource limits for the current process based on values specified in the system login class database. Class capability tags used, with and without -cur (soft limit) or -max (hard limit) suffixes and the corresponding resource setting:

cputime	RLIMIT_CPU
filesize	RLIMIT_FSIZE
datasize	RLIMIT_DATA
stacksize	RLIMIT_STACK
coredumpsize	RLIMIT_CORE
memoryuse	RLIMIT_RSS
memorylocked	RLIMIT_MEMLOCK
maxproc	RLIMIT_NPROC
openfiles	RLIMIT_NOFILE
sbsize	RLIMIT_SBSIZE
vmemoryuse	RLIMIT_VMEM
pseudoterminals	RLIMIT_NPTS
swapuse	RLIMIT_SWAP
kqueues	RLIMIT_KQUEUES
umtxp	RLIMIT_UMTXP

LOGIN_SETPRIORITY Set the scheduling priority for the current process based on the value specified in the system login class database. Class capability tags used:

priority

LOGIN_SETUMASK Set the umask for the current process to a value in the user or system login class database. Class capability tags used:

umask

LOGIN_SETPATH Set the "path" and "manpath" environment variables based on values in the user or system login class database. Class capability tags used with the corresponding environment variables set:

path	PATH
manpath	MANPATH

LOGIN_SETENV Set various environment variables based on values in the user or system login class database. Class capability tags used with the corresponding environment variables set:

lang	LANG
charset	MM_CHARSET
timezone	TZ
term	TERM

Additional environment variables may be set using the list type capability "setenv=var1 val1,var2 val2..,varN valN".

LOGIN_SETMAC Set the MAC label for the current process to the label specified in system login class database.

LOGIN_SETCPUMASK Create a new cpuset(2) and set the cpu affinity to the specified mask. The string may contain a comma separated list of numbers and/or number ranges as handled by the cpuset(1) utility or the case-insensitive string ‘default’. If the string is ‘default’ no action will be taken.

LOGIN_SETLOGINCLASS Set the login class of the current process using setloginclass(2).

LOGIN_SETALL Enables all of the above settings.

Note that when setting environment variables and a valid passwd pointer is provided in the *pwd* parameter, the characters ‘~’ and ‘$’ are substituted for the user’s home directory and login name respectively.

The **setclasscpumask**(), **setclassresources**() and **setclassenvironment**() functions are subsets of the setcontext functions above, but may be useful in isolation.

RETURN VALUES

The **setclasscontext**() and **setusercontext**() functions return -1 if an error occurred, or 0 on success. If an error occurs when attempting to set the user, login, group or resources, a message is reported to syslog(3), with LOG_ERR priority and directed to the currently active facility.

SEE ALSO

cpuset(1), ps(1), cpuset(2), setgid(2), setlogin(2), setloginclass(2), setuid(2), getcap(3), initgroups(3), login_cap(3), mac_set_proc(3), login.conf(5), termcap(5)

HISTORY

The functions **setclasscontext**(), **setclasscpumask**(), **setclassenvironment**(), **setclassresources**() and **setusercontext**() first appeared in FreeBSD 2.1.5.