NAME

quotactl - manipulate file system quotas

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/types.h>
#include <ufs/ufs/quota.h>

int
quotactl(*const char *path, int cmd, int id, void *addr*);

DESCRIPTION

The **quotactl**() system call enables, disables and manipulates file system quotas. A quota control command given by *cmd* operates on the given filename *path* for the given user or group *id*. (NOTE: One should use the QCMD macro defined in <*ufs/ufs/quota.h*> to formulate the value for *cmd*.) The address of an optional command specific data structure, *addr*, may be given; its interpretation is discussed below with each command.

For commands that use the *id* identifier, it must be either -1 or any positive value. The value of -1 indicates that the current UID or GID should be used. Any other negative value will return an error.

Currently quotas are supported only for the "ufs" file system. For "ufs", a command is composed of a primary command (see below) and a command type used to interpret the *id*. Types are supported for interpretation of user identifiers (USRQUOTA) and group identifiers (GRPQUOTA). The "ufs" specific commands are:

Q_QUOTAON — Enable disk quotas for the file system specified by *path*. The command type specifies the type of the quotas being enabled. The *addr* argument specifies a file from which to take the quotas. The quota file must exist; it is normally created with the quotacheck(8) program. The *id* argument is unused. Only the super-user may turn quotas on.

Q_QUOTAOFF — Disable disk quotas for the file system specified by *path*. The command type specifies the type of the quotas being disabled. The *addr* and *id* arguments are unused. Only the super-user may turn quotas off.

Q_GETQUOTASIZE — Get the wordsize used to represent the quotas for the user or group (as determined by the command type). Possible values are 32 for the old-style

quota file and 64 for the new-style quota file. The *addr* argument is a pointer to an integer into which the size is stored. The identifier *id* is not used.

Q_GETQUOTA Get disk quota limits and current usage for the user or group (as determined by the command type) with identifier *id*. The *addr* argument is a pointer to a *struct dqblk* structure (defined in <*ufs/ufs/quota.h*>).

Q_SETQUOTA Set disk quota limits for the user or group (as determined by the command type) with identifier *id*. The *addr* argument is a pointer to a *struct dqblk* structure (defined in <*ufs/ufs/quota.h*>). The usage fields of the *dqblk* structure are ignored. This system call is restricted to the super-user.

Q_SETUSE Set disk usage limits for the user or group (as determined by the command type) with identifier *id*. The *addr* argument is a pointer to a *struct dqblk* structure (defined in <*ufs/ufs/quota.h*>). Only the usage fields are used. This system call is restricted to the super-user.

Q_SYNC Update the on-disk copy of quota usages. The command type specifies which type of quotas are to be updated. The *id* and *addr* arguments are ignored.

RETURN VALUES

The **quotactl**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **quotactl**() system call will fail if:

[EOPNOTSUPP] The kernel has not been compiled with the QUOTA option.

[EUSERS] The quota table cannot be expanded.

[EINVAL] The *cmd* argument or the command type is invalid. In Q_GETQUOTASIZE, Q_GETQUOTA, Q_SETQUOTA, and Q_SETUSE, quotas are not currently enabled for this file system.

The *id* argument to Q_GETQUOTA, Q_SETQUOTA or Q_SETUSE is a negative value.

[EACCES] In Q_QUOTAON, the quota file is not a plain file.

[EACCES] Search permission is denied for a component of a path prefix.

[ENOTDIR] A component of a path prefix was not a directory.

[ENAMETOOLONG]
A component of either pathname exceeded 255 characters, or the entire length of either path name exceeded 1023 characters.

[ENOENT] A filename does not exist.

[ELOOP] Too many symbolic links were encountered in translating a pathname.

[EROFS] In Q_QUOTAON, either the file system on which quotas are to be enabled is mounted read-only or the quota file resides on a read-only file system.

[EIO] An I/O error occurred while reading from or writing to a file containing quotas.

[EINTEGRITY] Corrupted data was detected while reading from the file system.

[EFAULT] An invalid *addr* was supplied; the associated structure could not be copied in or out of the kernel.

[EFAULT] The *path* argument points outside the process's allocated address space.

[EPERM] The call was privileged and the caller was not the super-user.

SEE ALSO

quota(1), fstab(5), edquota(8), quotacheck(8), quotaon(8), repquota(8)

HISTORY

The **quotactl**() system call appeared in 4.3BSD-Reno.

BUGS

There should be some way to integrate this call with the resource limit interface provided by setrlimit(2) and getrlimit(2).