NAME

quotactl – manipulate disk quotas

SYNOPSIS

#include <sys/fs/ufs_quota.h>

int ioctl(int *fd,* **Q_QUOTACTL, struct quotactl** **qp***)**

DESCRIPTION

This **ioctl( )** call manipulates disk quotas. *fd* is the file descriptor returned by the **open( )** system call after opening the **quotas** file (located in the root directory of the filesystem running quotas.) **Q_QUOTACTL** is defined in **/usr/include/sys/fs/ufs_quota.h**. *qp* is the address of the **quotctl** structure which is defined as

```
struct quotctl {
        int op;
        uid_t uid;
        caddr_t addr;
};
```

op indicates an operation to be applied to the user ID *uid*. (See below.) *addr* is the address of an optional, command specific, data structure which is copied in or out of the system. The interpretation of *addr* is given with each value of *op* below.

Q_QUOTAON Turn on quotas for a file system. *addr* points to the full pathname of the **quotas** file. *uid* is ignored. It is recommended that *uid* have the value of **0**. This call is restricted to the super-user.

Q_QUOTAOFF Turn off quotas for a file system. *addr* and *uid* are ignored. It is recommended that *addr* have the value of **NULL** and *uid* have the value of **0**. This call is restricted to the super-user.

Q_GETQUOTA Get disk quota limits and current usage for user *uid*. *addr* is a pointer to a **dqblk** structure (defined in **<sys/fs/ufs_quota.h>**). Only the super-user may get the quotas of a user other than himself.

Q_SETQUOTA Set disk quota limits and current usage for user *uid*. *addr* is a pointer to a **dqblk** structure (defined in **sys/fs/ufs_quota.h**). This call is restricted to the super-user.

Q_SETQLIM Set disk quota limits for user *uid*. *addr* is a pointer to a **dqblk** structure (defined in **sys/fs/ufs_quota.h**). This call is restricted to the super-user.

Q_SYNC Update the on-disk copy of quota usages for this file system. *addr* and *uid* are ignored.

Q_ALLSYNC Update the on-disk copy of quota usages for all file systems with active quotas. *addr* and *uid* are ignored.

RETURN VALUES

This **ioctl( )** returns:

0 on success.

–1 on failure and sets **errno** to indicate the error.

ERRORS

EFAULT *addr* is invalid.

EINVAL The kernel has not been compiled with the **QUOTA** option.
op is invalid.

ENOENT The **quotas** file specified by *addr* does not exist.

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

ESRCH No disk quota is found for the indicated user.
Quotas have not been turned on for this file system.

EUSERS The quota table is full.

If *op* is **Q_QUOTAON**, **ioctl**() may set **errno** to:

EACCES The quota file pointed to by *addr* exists but is not a regular file.
The quota file pointed to by *addr* exists but is not on the file system pointed to by *special*.

EIO Internal I/O error while attempting to read the **quotas** file pointed to by *addr.*

FILES

/usr/include/sys/fs/ufs_quota.h
quota-related structure/function definitions and defines

SEE ALSO

quota(1M), **getrlimit**(2), **mount**(2), **quotacheck**(1M), **quotaon**(1M)

BUGS

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

This call is incompatible with Melbourne quotas.