NAME

auditsvc – write audit log to specified file descriptor

SYNOPSIS

cc [*flag* ...] *file* ... **–lbsm –lsocket –lnsl –lintl** [*library* ...]

#include <sys/param.h>
#include <bsm/audit.h>

int auditsvc(int *fd,* **int** *limit***);**

AVAILABILITY

The functionality described in this man page is available only if the Basic Security Module (BSM) has been enabled. See **bsmconv**(1M) for more information.

DESCRIPTION

The **auditsvc( )** system call specifies the audit log file to the kernel. The kernel writes audit records to this file until an exceptional condition occurs and then the call returns. The parameter *fd* is a file descriptor that identifies the audit file. Programs should open this file for writing before calling **auditsvc( )**. The parameter *limit* specifies the number of free blocks that must be available in the audit file system, and causes **auditsvc( )** to return when the free disk space on the audit filesystem drops below this limit. Thus, the invoking program can take action to avoid running out of disk space. The **auditsvc( )** system call does not return until one of the following conditions occurs:

- The process receives a signal that is not blocked or ignored.
- An error is encountered writing to the audit log file.
- The minimum free space (as specified by *limit*), has been reached.

Only processes with an effective user ID of super-user may execute this call successfully.

RETURN VALUES

auditsvc( ) returns only on an error.

ERRORS

EAGAIN The descriptor referred to a *stream*, was marked for System V-style non-blocking I/O, and no data could be written immediately.

EBADF *fd* is not a valid descriptor open for writing.

EBUSY A second process attempted to perform this call.

A second process attempted to perform this call.

ENOSPC The user's quota of disk blocks on the file system containing the file has been exhausted.

Audit filesystem space is below the specified limit.

EFBIG An attempt was made to write a file that exceeds the process's file size limit or the maximum file size.

EINTR The call is forced to terminate prematurely due to the arrival of a signal whose **SV_INTERRUPT** bit in **sv_flags** is set (see **sigvec**(3B)). **signal**(3C), sets this bit for any signal it catches.

EINVAL Auditing is disabled (see **auditon**(2)).

fd does not refer to a file of an appropriate type. Regular files are always appropriate.

EIO An I/O error occurred while reading from or writing to the file system.

ENOSPC There is no free space remaining on the file system containing the file.

ENXIO A hangup occurred on the *stream* being written to.

EPERM The process's effective user ID is not super-user.

EWOULDBLOCK
The file was marked for 4.2BSD-style non-blocking I/O, and no data could be written immediately.

SEE ALSO

auditd(1M), **audit**(2), **auditon**(2), **sigvec**(3B), **audit.log**(4)