Linux Directory Notification

Stephen Rothwell <sfr@canb.auug.org.au>

The intention of directory notification is to allow user applications to be notified when a directory, or any of the files in it, are changed. The basic mechanism involves the application registering for notification on a directory using a fcntl(2) call and the notifications themselves being delivered using signals.

The application decides which "events" it wants to be notified about. The currently defined events are:

DN_ACCESS	A file in the directory was accessed (read)
DN_MODIFY	A file in the directory was modified (write,truncate)
DN_CREATE	A file was created in the directory
DN_DELETE	A file was unlinked from directory
DN_RENAME	A file in the directory was renamed
DN_ATTRIB	A file in the directory had its attributes changed (chmod,chown)

Usually, the application must reregister after each notification, but if DN_MULTISHOT is or'ed with the event mask, then the registration will remain until explicitly removed (by registering for no events).

By default, SIGIO will be delivered to the process and no other useful information. However, if the F_SETSIG fcntl(2) call is used to let the kernel know which signal to deliver, a siginfo structure will be passed to the signal handler and the si_fd member of that structure will contain the file descriptor associated with the directory in which the event occurred.

Preferably the application will choose one of the real time signals (SIGRTMIN + <n>) so that the notifications may be queued. This is especially important if DN_MULTISHOT is specified. Note that SIGRTMIN is often blocked, so it is better to use (at least) SIGRTMIN + 1.

Implementation expectations (features and bugs :-))

The notification should work for any local access to files even if the actual file system is on a remote server. This implies that remote access to files served by local user mode servers should be notified. Also, remote accesses to files served by a local kernel NFS server should be notified.

In order to make the impact on the file system code as small as possible, the problem of hard links to files has been ignored. So if a file (x) exists in two directories (a and b) then a change to the file using the name "a/x" should be notified to a program expecting notifications on directory "a", but will not be notified to one expecting notifications on directory "b".

Also, files that are unlinked, will still cause notifications in the last directory that they were linked to.

Configuration

Dnotify is controlled via the CONFIG_DNOTIFY configuration option. When disabled, fcntl(fd, F_NOTIFY, ...) will return -EINVAL.

Example

See tools/testing/selftests/filesystems/dnotify_test.c for an example.

NOTE

Beginning with Linux 2.6.13, dnotify has been replaced by inotify. See Documentation/filesystems/inotify.rst for more information on it.