IEEE Standards Interpretation for IEEE Std 1003.1™-1990 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)

Interpretation Request #50
Topic: fcntl() locking **Relevant Sections:** 6.5.2.2 **Classification:** No change

POSIX.1-1990 Section 6.5.2.2 fcntl(): Should adjacent locks be coalesced when F_GETLK is used to check for existence of locks?

X/Open would propose that they should not be so: In fcntl(), page 107 line 161ff of the System Interfaces and headers document it says:

"There will be at most one type of lock for each byte in the file. Before successfull return from an F_SETLK or an F_SETLKW request when the calling process has previously existing locks on bytes in the region specified by the request, the previous lock type for each byte in the specified region will be replaced by the new lock type."

Meaning that if two locks for overlapping regions of a file are obtained by a single process, the region which overlaps will be owned by the new lock. Coalescing is not required.

Interpretation Response
IEEE Std 1003.1-1990 does not specify whether there may be distinct overlapping locks for the same process or multiple locks of the same type on the same byte for the same process. It is unspecified, when multiple F_SETLK or F_SETLKW requests for the same lock type have been made by the same process that address a common extent in the file, whether multiple requests with F_UNLCK are necessary in order to unlock that common extent.

Coalescing of locks is neither required nor prohibited. It would be conforming for an im-

plementation to treat adjacent or overlapping locks of the same type for the same process as if they were coalesced immediately on creation.

Rationale for Interpretation

The request exhibits some confusion between the concepts "one type of lock set for each byte" and "one lock set for each byte". POSIX.1 does not prohibit having more than one lock of the same type on the same byte of a file for the same process. The standard does not provide a handle by which an individual lock may be identified after it is set. This means that a process can identify a lock only by the values in the struct flock returned by an F_GETLK request. The standard does not require that the values returned by an F_GETLK request correspond exactly to an extent that was locked by a single F_SETLK or F_SETLKW request. The specification quoted in the request (that each byte in the file be affected by only one type of lock) can further obscure the identity of individual locks, since the creation of a new lock of a different type could cause an old lock to be truncated or to be divided into discontinuous extents. See B.6.5.2, page 271, lines 3661-3666.

It is unspecified whether a process can use F_GETLK to identify its own locks. A strictly conforming application must cause the F_GETLK request to be made from a different process.