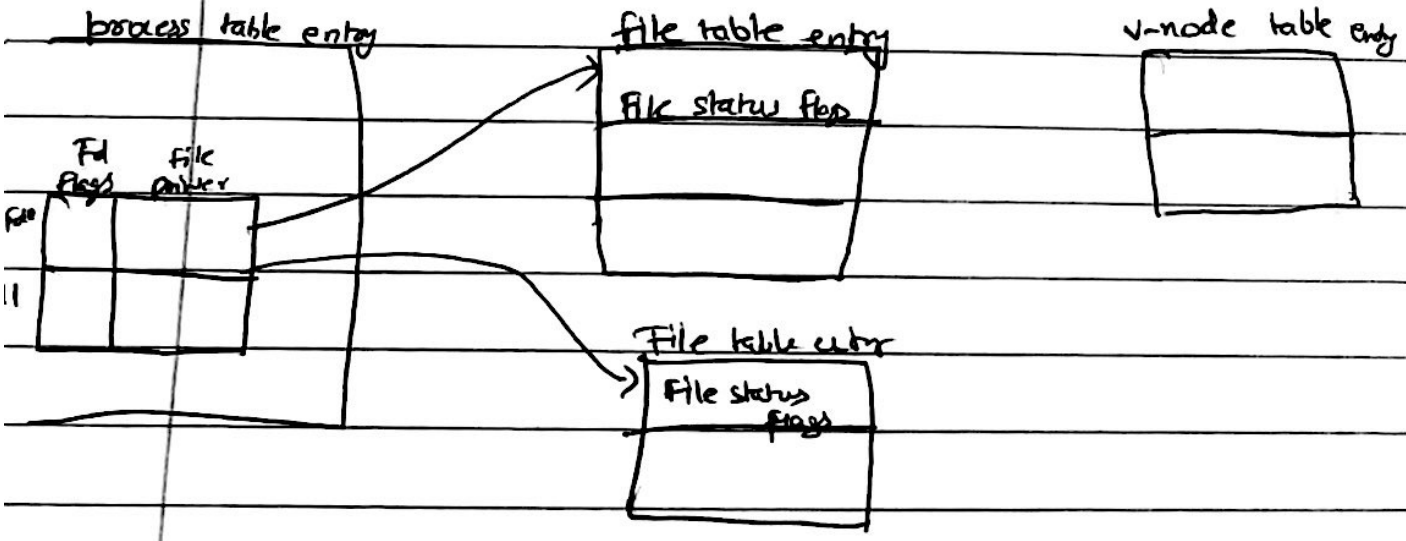
Advanced programming in Unix Environment

⇒ FILE SHARING

Data structure used by kernel for all I/O
Basically these are the three different tables
process table entry
file table entry
V-nodes table entry

it looks like:



In case of two independent processes have the same file open then we can have something like where two processes sharing the same V-node entry but having different file table entry with their own specification in file table entries like offset of the processes.

File descriptor flags & file status flags scopes.

- only to a single descriptor in a single process
- apply to all descriptors in any process that point to the given file table entry.

Everything works fine as long as we are reading the same file by multiple processes, but in case of writing we need to look out for strange errors.

In general, the term atomic operation refers to an operation that might be composed of multiple steps. If the operation is performed atomically, either all the steps are performed or none are performed.

FILES & DIRECTORIES

→ stat, fstat, fstatat, and lstat

stat → returns a structure of information about the named file.

fstat → obtains information about the file that is already open on the descriptor fd.

lstat → is similar to stat, but when the named file is a symbolic link.

fstatat function provides a way to return the file statistics for a pathname relative to an open directory represented by the fd argument.

File types → Regular file.
Directory file
block special file → disk drives devices
character special file → file providing unbuffered I/O access in variable-sized units to devices.
FIFO → type of file for communication between processes.
Socket → type of file used for network communications between processes
Symbolic link → file that points to another file.

file type is encoded in st_mode of stat structure.

S_ISREG()
S_ISDIR()
S_ISCHR()
S_ISBLK()
S_ISFIFO()
S_ISLNK()
S_ISSOCK()
→ file type macros in <sys/stat.h>

Set-User-Id & Set-Group-Id

→ Every process has six or more IDs associated with it.

real userID
real groupID
effective user ID
effective group ID
supplementary group IDs
saved set-user-ID
saved set-group-ID