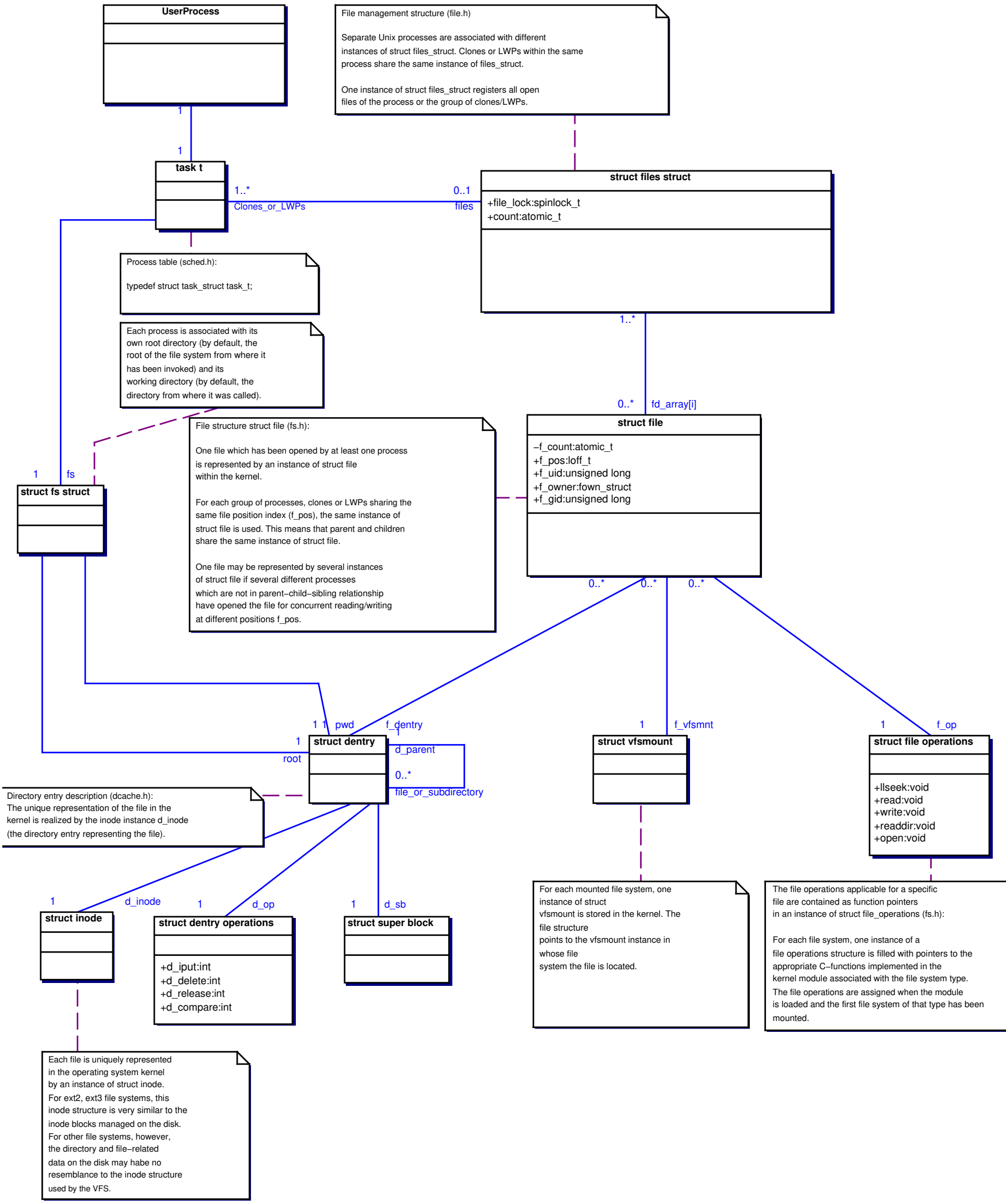
UserProcess
File management structure (file.h)
Separate Unix processes are associated with different instances of struct files_struct. Clones or LWPs within the same process share the same instance of files_struct.
One instance of struct files_struct registers all open files of the process or the group of clones/LWPs.
1
1
task t
1..*
Clones_or_LWPs
0..1
files
struct files struct
+file_lock:spinlock_t
+count:atomic_t
Process table (sched.h):
typedef struct task_struct task_t;
Each process is associated with its own root directory (by default, the root of the file system from where it has been invoked) and its working directory (by default, the directory from where it was called).
1..*
0..*
fd_array[i]
struct file
–f_count:atomic_t
+f_pos:loff_t
+f_uid:unsigned long
+f_owner:fown_struct
+f_gid:unsigned long
File structure struct file (fs.h):
One file which has been opened by at least one process is represented by an instance of struct file within the kernel.
For each group of processes, clones or LWPs sharing the same file position index (f_pos), the same instance of struct file is used. This means that parent and children share the same instance of struct file.
One file may be represented by several instances of struct file if several different processes which are not in parent–child–sibling relationship have opened the file for concurrent reading/writing at different positions f_pos.
1
fs
struct fs struct
0..*
0..*
0..*
1
1
pwd
f_dentry
1
struct dentry
root
d_parent
0..*
file_or_subdirectory
1
f_vfsmnt
struct vfsmount
1
f_op
struct file operations
+llseek:void
+read:void
+write:void
+readdir:void
+open:void
Directory entry description (dcache.h):
The unique representation of the file in the kernel is realized by the inode instance d_inode (the directory entry representing the file).
1
d_inode
1
d_op
1
d_sb
struct inode
struct dentry operations
+d_iput:int
+d_delete:int
+d_release:int
+d_compare:int
struct super block
For each mounted file system, one instance of struct vfsmount is stored in the kernel. The file structure points to the vfsmount instance in whose file system the file is located.
The file operations applicable for a specific file are contained as function pointers in an instance of struct file_operations (fs.h):
For each file system, one instance of a file operations structure is filled with pointers to the appropriate C–functions implemented in the kernel module associated with the file system type. The file operations are assigned when the module is loaded and the first file system of that type has been mounted.
Each file is uniquely represented in the operating system kernel by an instance of struct inode. For ext2, ext3 file systems, this inode structure is very similar to the inode blocks managed on the disk. For other file systems, however, the directory and file–related data on the disk may habe no resemblance to the inode structure used by the VFS.