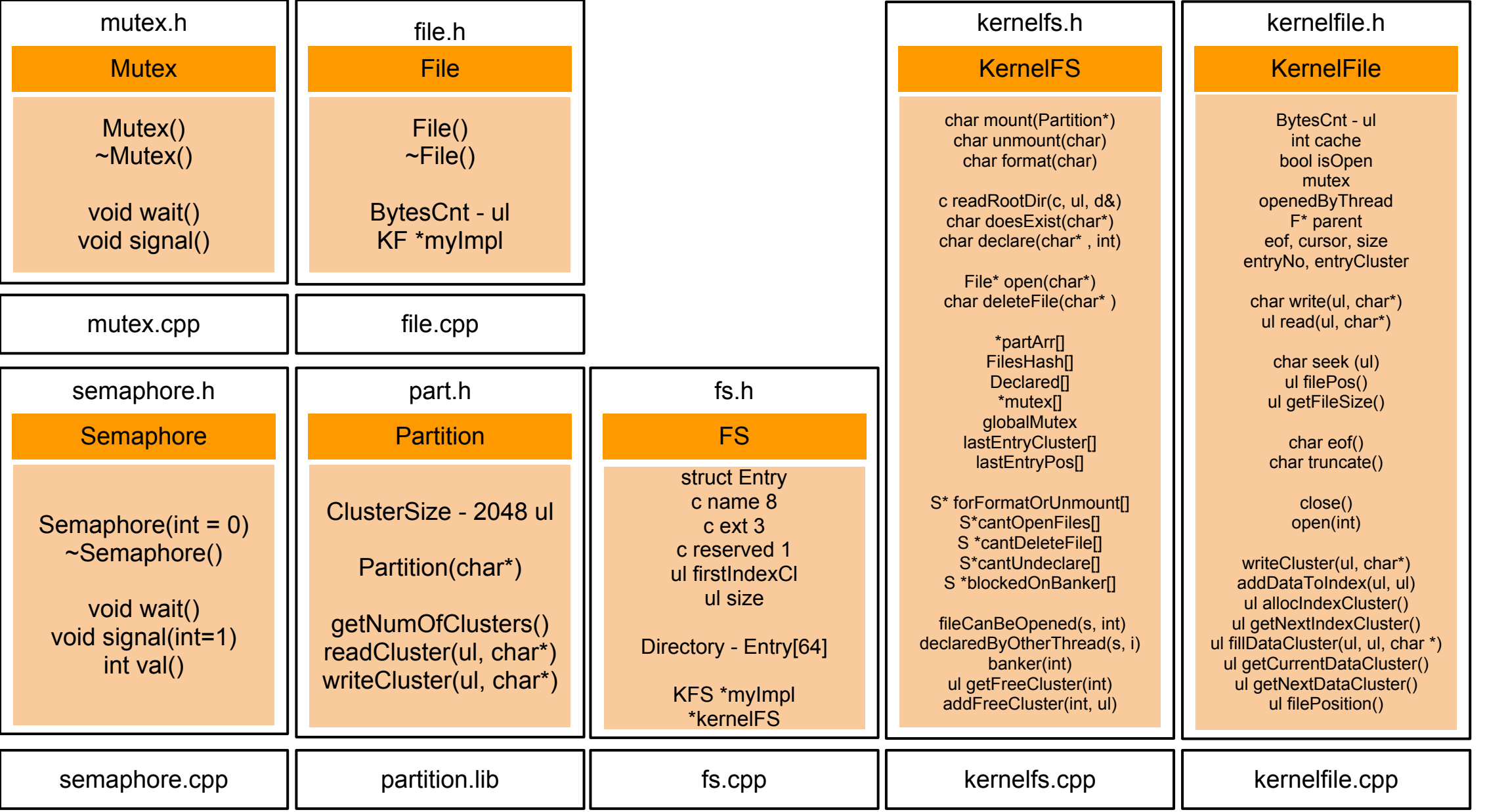
mutex.h
Mutex
Mutex()
~Mutex()
void wait()
void signal()
mutex.cpp
file.h
File
File()
~File()
BytesCnt - ul
KF *myImpl
file.cpp
kernelfs.h
KernelFS
char mount(Partition*)
char unmount(char)
char format(char)
c readRootDir(c, ul, d&)
char doesExist(char*)
char declare(char* , int)
File* open(char*)
char deleteFile(char*)
*partArr[]
FilesHash[]
Declared[]
*mutex[]
globalMutex
lastEntryCluster[]
lastEntryPos[]
S* forFormatOrUnmount[]
S*cantOpenFiles[]
S *cantDeleteFile[]
S*cantUndeclare[]
S *blockedOnBanker[]
fileCanBeOpened(s, int)
declaredByOtherThread(s, i)
banker(int)
ul getFreeCluster(int)
addFreeCluster(int, ul)
kernelfs.cpp
kernelfile.h
KernelFile
BytesCnt - ul
int cache
bool isOpen
mutex
openedByThread
F* parent
eof, cursor, size
entryNo, entryCluster
char write(ul, char*)
ul read(ul, char*)
char seek (ul)
ul filePos()
ul getFileSize()
char eof()
char truncate()
close()
open(int)
writeCluster(ul, char*)
addDataToIndex(ul, ul)
ul allocIndexCluster()
ul getNextIndexCluster()
ul fillDataCluster(ul, ul, char *)
ul getCurrentDataCluster()
ul getNextDataCluster()
ul filePosition()
kernelfile.cpp
semaphore.h
Semaphore
Semaphore(int = 0)
~Semaphore()
void wait()
void signal(int=1)
int val()
semaphore.cpp
part.h
Partition
ClusterSize - 2048 ul
Partition(char*)
getNumOfClusters()
readCluster(ul, char*)
writeCluster(ul, char*)
partition.lib
fs.h
FS
struct Entry
c name 8
c ext 3
c reserved 1
ul firstIndexCl
ul size
Directory - Entry[64]
KFS *myImpl
*kernelFS
fs.cpp