

ファイルシステムサーバプロセスの本体(fsMain())とシステムコール入り口のスタブルーチン

オープンファイルを管理するクラス。C++ではFile構造体と関数群で多重度がnのクラスを表現する。

システムコールの入口: dirAccessクラスを使用してfileを探索する。fileをopenした時はFileインスタンスを作る。fileの長さが変化した時はdirを書き換える。

file

- isDir : boolean

- mode : int

- lenH : int

- lenL : int

- clst : int

- dirClst : int

- dirEnt : int

- posH : int

- posL : int

- posClst : int

- posSct : int

- fileBuf : char[]

- isEob : boolean

- flelsDirty : boolean

dirClst,dirEntはそのファイルが記録されているDIRエントリ

-1ならバッファが空

BLKSIZのバッファ

~ putByteFile(f : file, ch : char) : int

~ getByteFile(f : file) : int

~ seekFile(f : file, posH : int, posL : int) : void

~ flushBuf() : void

クラスタ番号、クラスタ内セクタ番号で操作するファイルシステム下位クラス、クラスタの割付け、解放を行いFATを書き換える。

クラスタ番号0はルートDIRを意味する。blkFileはルートDIRもクラスタ番号でアクセス可能にし、他のファイルの違いを吸収する。

BLK blk = {clst, sct};

nextBlkは、blkPosを更新しtrueを返す。EOFならblkPosを更新しないでfalseを返す。

allocClstはclstの次に新しいクラスタを確保しFATを更新する。新しいクラスタを返す。

fs

+ fsLink : int

- fatFsSysCall(op : int, prm1 : int, prm2 : int, prm3 : int, pcb : PCB) : int

+ fsMain() : void

+ open(path : char[], mode : int) : int

+ close(fd : int) : int

+ read(fd : int, buf : void[], len : int) : int

+ write(fd : int, buf : void[], len : int) : int

+ creat(path : char[]) : int

+ remove(path : char[]) : int

+ seek(fd : int, ptrh : int, ptrl : int) : int

+ mkDir(path : char[]) : int

+ rmDir(path : char[]) : int

fsDef.h

+ BLKSIZ : int = 512

+ FILEMAX : int = 20

+ P\_OPEN\_MAX : int = 5

+ UNASSIGN : int = 0x0000

+ ROOTDIR : int = 0x0001

+ READ\_MODE : int = 0x0000

+ WRITE\_MODE : int = 0x0001

+ APPEND\_MODE : int = 0x0002

ファイルシステム全体で共通の定義。mmcspli以外の全てにインクルードされる。

fatSys

- files : file[]

- srchDir(fName : char[]) : boolean

- srchPath(path : char[], int n : int) : boolean

~ sysOpen(path : char[], flags : int) : int

~ sysRead(fd : int, buf : char[], len : int) : int

~ sysWrite(fd : int, buf : char[], len : int) : int

~ sysSeek(fd : int, posH : int, posL : int) : int

~ sysClose(fd : int) : int

~ sysCreat(path : char[]) : int

~ sysRemove(path : char[]) : int

~ sysMkDir(path : char[]) : int

~ sysRmdir(path : char[]) : int

DirEnt

~ dirClst : int

~ dirEnt : int

~ name : char[]

~ attr : int

~ clst : int

~ lenH : int

~ lenL : int

- curEnt

dirAccess

- dirCache : char[]

- dirCurBlk : Blk

- dirIsDirty : boolean

- entToBlk(clst : int, ent : int, blk : Blk) : void

- setCache(dirEnt : DirEnt) : void

~ flushDirEnt() : void

~ readDirEnt(dir : DirEnt) : int

~ writeDirEnt(dir : DirEnt) : void

~ initDirClst(clst : int) : int

~ nextDirEnt(dir : DirEnt, boolean flag : int) : boolean

~ newDirEnt(dir : DirEnt) : boolean

LBAでセクタを指定してマルチメディアカードを読み書きするデバイスドライバ。

mmcspli

+ restart() : interrupt

~ initMmcSpi() : void

~ writeSct(h : int, l : int, buf : char[]) : void

~ readSct(h : int, l : int, buf : char[]) : void

現在のDIRエントリを表現するデータ構造。場所、名前、長さ、属性を持つ。sysOpen, sysCreate, sysRemove, sysMkdir, syrRmdir, sysWriteの操作で、ディレクトリエントリをメタデータとして読み書きするときに使用する。

DIRエントリのアドレス

file名("abcdeftxt"形式)

ファイルの属性

ファイルの開始クラスタ

ファイルのバイト単位の長さ

DIR fileの先頭クラスタ番号、エントリ番号でDIRエントリを操作するためのクラス

DIR fileの1セクタをキャッシュする配列

キャッシュしているDIRエントリが格納されていたクラスタ番号とセクタ番号

DIR fileの先頭クラスタ番号とエントリ番号でDIRを操作する。readDirは有効なエントリで1、これ以上有効なエントリがなければ0、無効なエントリなら-1を返す。

dirのclst,entを更新してtrueを返す。flagがfalseかつEOFならfalseを返す。flagがtrueならばクラスタを拡張する

DIRから未使用エントリを探す。なかったらDIRファイルを拡大する。新しいクラスタを割り付けることもある。dirのエントリ番号を更新する。領域不足の時はfalseを返す。