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。

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

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

- isEob : boolean

~ flushBuf(): void

- fleIsDirty: boolean

~ getByteFile(f : file) : int

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

## fsDef.h

- + BLKSIZ : int = 512
- + FILEMAX : int = 20
- + P OPEN MAX : int = 5
- + UNASSIGN : int =  $0 \times 0000$

DIRエントリのアドレス

file名("abcdeftxt"形式)

ファイルの開始クラスタ

ファイルの属性

- + ROOTDIR : int =  $0 \times 0001$
- + READ MODE : int =  $0 \times 0000$ + WRITE MODE: int = 0x0001
- + APPEND MODE : int =  $0 \times 0002$

全体で共通の定 義。mmcspi以外の 全てにインクルー ドされる。

ファイルシステム

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。

### fatSvs

- files : file[]
- srchDir(fName : char[]) : boolean

+ mkDir(path : char[]) : int

+ rmDir(path : char[]) : int

- 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
- ~ svsClose(fd : int) : int
- ~ sysCreat(path : char[]) : int
- ~ sysRemove(path : char[]) : int
- ~ sysMkDir(path : char[]) : int
- ~ svsRmDir(path : char[]) : int

### blkFile

- fatCache : char[]
- fatLbaH : int
- fatLbaL : int
- fatIsDirty : boolean
- ~ IdBlk(dst : Blk, c : int, s : int) : void
- ~ stBlk(c: int, s: int, src: Blk): void
- readBPB(): void
- readMBR(): void
- readFat(clst : int) : int
- writeFat(clst: int, val: int): void
- ~ blkInit(): void
- ~ flushFat(): void
- b~ readBlk(blk : Blk, buf : char[]) : void
- o~ writeBlk(blk : Blk, buf : char[]) : void
- >~ nextBlk(blk : Blk) : boolean
- o~ allocClst(clst : int) : int
- ~ delClstChain(clst: int): void

# DirEnt

- ~ dirClst : int ~ dirEnt : int
- ~ name : char[]o-
- ~ attr : int

- curEnt.

- ~ clst : int
- ~ lenH : int
- ~ lenL: int
  - ファイルのバイト単位の長さい

# dirAccess

- dirCache : char[] - dirCurBlk : Blk
- dirlsDirty : 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。

## mmcspi

- + restart() : interrupt ~ initMmcSpi() : void
- ~ writeSct(h: int, I: int, buf: char[]): void
- ~ readSct(h : int, l : int, buf : char[]) : void

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、これ以上有効な工 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。