

MMCカードのアクセスのしかた

MMCカード（以後カードと呼びます）は現在SDカードの前進にあたる仕様です。よって現在のカードは、ほとんど網羅していると考えて間違いありません。

組込みシステムの場合メモリの扱いで大変苦労してしまいます。不揮発のメモリを選択する場合、容量と速度、共用性を考えた場合どうしても選択するメモリが無いのが

実情です。デバイスのメモリの場合EEPROMやフラッシュROMを考えがちです。しかし、こうした場合上位のシステムとのデータの共有を考えると上位とのリンク

インターフェースが必要です。つまりシリアルであったり、パラレルであったり、今流行のUSBであったりするわけですが、インターフェースが必要になります。

インターフェースを検討する時に上位システムとのリンクを図るために、通信やダウンロードといったことを考慮したファームウェアを考えなければなりません。

当然色々な意味で上位システムとのリンクという意味でインターフェースが必要です。ここでは大きなデータやパソコンとの簡単なリンクの方法としてマスメージという

メディアを考えてMMCを考えて見たいと思います。

【マスメージの種類】

現在ではIPODならぬ音楽メディア等でHDDが使われています。非常に大きなストレージです。HDDは機械的な動作部分を有しているため、耐久性や安全保障と

いった面で必ず優位なメディアではないと考えられています。ですから、各サーバマシンではサブのストレージを持つことによって安全を確保しています。

一方フラッシュROMの登場によってデバイスの容量が飛躍的に伸びています。デバイスの容量が増えている面、デバイスの製造歩留まりは上がりません。

そこで、この歩留まり改善に多少ボイドを持ったデバイスでも使える技術が考えられました。そういった技術がNOR型NAND型フラッシュの技術です。

最近携帯電話やデジタルカメラなどに使われているカードはこれらの技術を集めて集約されたものの推移だと考えます。

カードは事前にカードに組み込まれたコントローラによって、ボイドを取り除かれています。つまり、カードの使用者には不良部分を取り除かれたシステムになっています。

ます。この不良部分が無いといったメディアは使う側にとってはとっても厄介な部分を考慮しなくてもいいところですから、スマートな設計が期待できるわけです。

メディアに記載されている保障された範囲の中で使用するには、エラーフリーと考えられるわけですから、多少容量が減っても充分恩恵に浸れるわけです。

こういった意味からHDDより使いやすく、信頼性が高いと考えられます。また、組込みシステムではWEBシステムみたいな莫大なデータを扱うことは

少なく、比較的抽象化されたデータになっているため1TMB（テラ）までの必要性は少ないのではないかと思います。

【カードの容量】

カードの容量は最近飛躍的に伸びてきているのが実情です。1 G M B、4 G M B
使う側は口を開けて待っていれば、恩恵は受けられます。

カードの容量によって制御する方法も少しずつ変わってきますが、ある一定の使い方をすれば、4 Gまでは同じように使えます。

【カードの制御方法】

カードを最適に動かすには、カード独自の使い方をすれば高速で最大限のパフォーマンスを引き出すことができます。しかし、もうひとつの使い方としてS P Iという

インターフェースで使うこともできます。これは既存システムを使えるといったメリットもありますが、シリアル通信であるためパフォーマンスは1 / 1 0程度

落ちてしまいます。しかし汎用インターフェースですし、特別な考えをしなくても汎用I / Oのポートを持ってさえいればアクセスできる利点があります。

ここではS P Iを使用したアクセス方法を説明していきます。

【ファイルシステム】

一般にファイルシステムはメディアの各システムの共用化のために使用します。
ここではWINDOWSでも使われているF A Tシステムについて一緒に適用してい

きたいと思います。

【展望】

長くなりましたが、誰もが安価にシステムの共用性ができ、更なる飛躍ができるような環境が少なくなりました。技術というのは使い込んでこそ次なる技術が生まれるもの

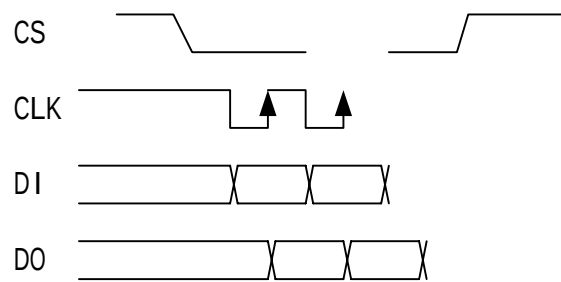
です。皆さんに使い込んで頂いて次なるものを作って頂ければ幸いに思います。

【SPIの動作】

SPIインターフェースはCSがアクティブ L になっています。つまりCSがLの間接続されているデバイスを選択されていることになります。

CLKの立ち上がりでデータの入力DIとデータの出力DOが確定されたり出力されたりします。

このDI, DOはデバイス側からの信号になりますから、DIは制御する側から見ると出力に接続します。DOも制御する側から見るとデバイスからの出力と見なしますので、入力に接続します。



SPIはバイトアクセスが基本になります。1回のアクセスはCLKが8個まとまった単位でSPIにアクセスします。

【カードのアクセス方法】

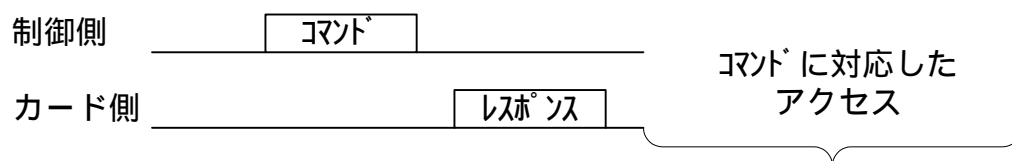
カードは固有カードモードとSPIモードと両方をサポートしています。電源を立ち上げてSPIモードに移行すると電源を落すまでモードを変更することができません。つまり1回SPIモードに移行した場合、他のモードに移行することができないため、システムとしては絶対条件として考えられるため、シンプルな制御が期待できます。

カードの基本的アクセスは制御側が最初にアクセスするコマンドに対してカードのレスポンスがペアーになっています。その後にコマンドに対する対応を制御側がすることになります。

カードに対するアクションは大きく分けて4つになると思います。

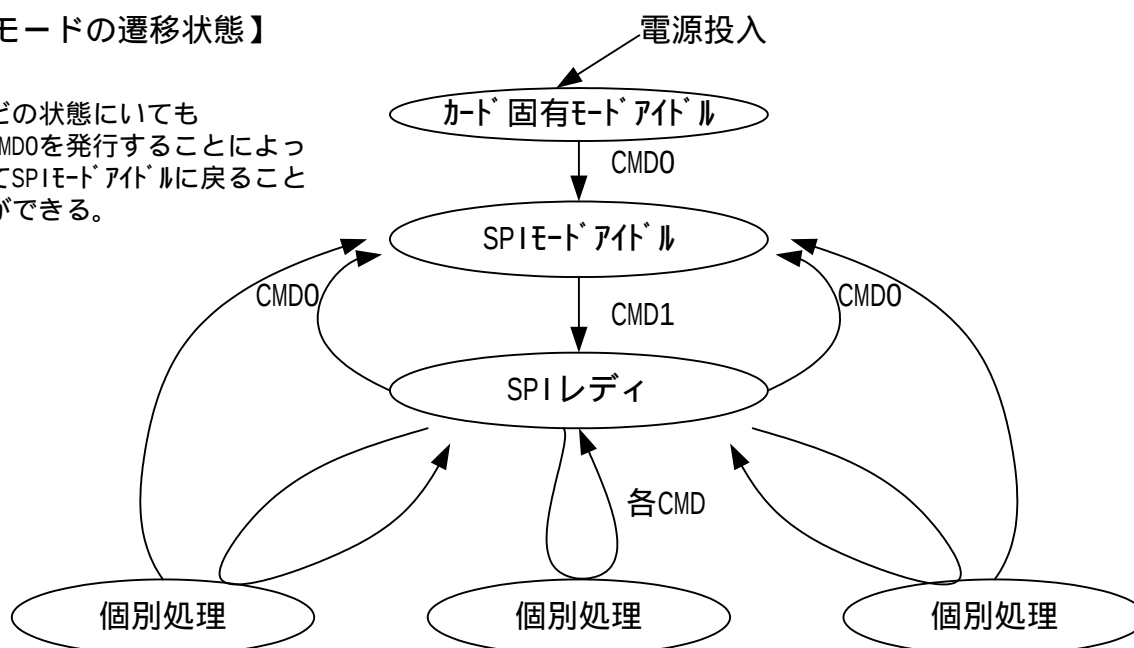
- 1、カードのモード設定とリセット
- 2、カード固有情報の読取
- 3、カードにデータの書込み
- 4、カードに書き込まれたデータの読込み

上記4つのタイプのアクションを使ってカードにデータの読み書きを行います。



【SPIモードの遷移状態】

どの状態にいてもCMD0を発行することによってSPIモードアイドルに戻ることができる。

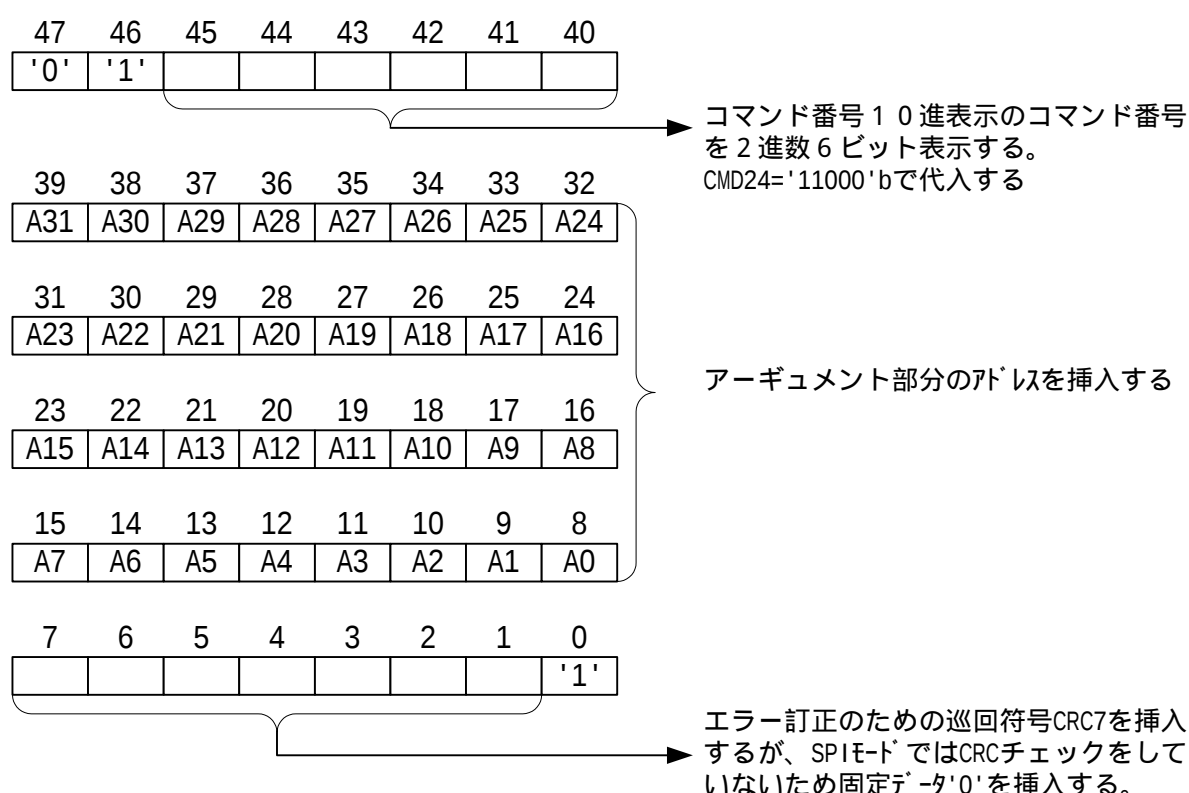


【コマンド】

カードのコマンドはカード固有モードとSPIモードともコマンド番号、意味合いは同じです。コマンドは種類も豊富でCMD0～CMD60まで用意されています。しかし、SPIモードで使用できるコマンドは限定されています。また、使用するコマンドもカードをストレージとしての扱いを基本的に行いますので限定的になります。以下に基本的なコマンドを記します。

コマンド	アーギュメント	レスポンス	コマンド機能	説明
CMD0	なし	R1	SPIモード IN	カードリセットとSPI IN
CMD1	なし	R1	カード初期化	SPIレディ状態
CMD9	なし	R1	CSDデータ読取	カード固有データを読取る
CMD10	なし	R1	CID読取り	カードID読取り
CMD17	[31..0]カードアドレス	R1	シングルBLK読取	アドレスのデータBLK読取り
CMD24	[31..0]カードアドレス	R1	シングルBLK書込	アドレスにデータBLK書込み

コマンドのビットイメージ



アーギュメントのアドレス指定について

カードのアドレスは512byteブロックで行います。カードは0番地より512byte単位でしか扱えません。書込みも読み込みも同じ方法です。例えば1番地のデータを読込む場合、0番地をスタート番地としたデータブロックのはじめから2番目のデータが1番地のデータとなります。

スタートアドレスの指定は必ず512バイトで割切れる値の指定でないとアドレスエラーがレスポンスで帰ってきてしまいます。

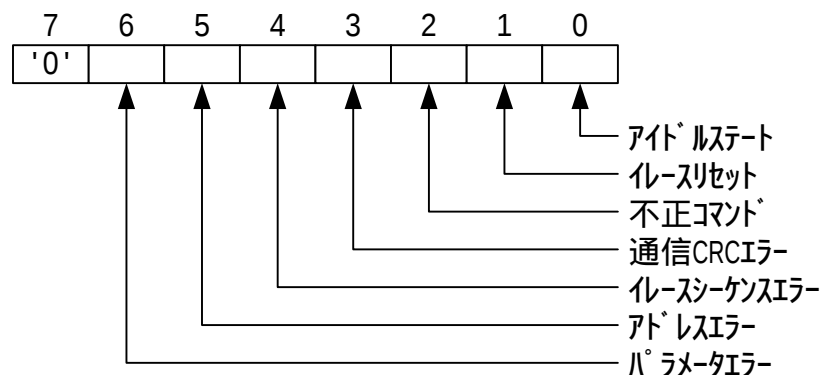
よって読みたいアドレスを512で割った商をスタートアドレスとし、読込んだ512バイトのデータを余りの値でスキップさせたデータの箇所が目的のデータです。

書込みの場合、限定箇所を変更する場合読取りと同様に目的のアドレスブロックを読取り、目的の箇所を変更して全512データを書込む作業が発生します。

【レスポンス】

カード に対しコマンド を送信するとカード はレスポンスを返します。
レスポンスの種類は 4 タイプ ほどありますが、ここではR1レスポンスを説明します。

レスポンスは必ず最上位ビット(MSB)より送出します。R1レスポンスは 1 バイトで構成されます。
最上位ビットは'0'となっているのでこれを送出スタートビットと認識できます。



【データトークン】

制御側に読み込み系のコマンド ではレスポンスに続いてカード からデータが送出されてきます。
このレスポンスの続きからデータの始まりを示すデータトークンを確認して所用のデータ長を読み込みます。
このデータトークンは1バイトで0xFEとなっています。

8ビット毎に読取って0xFEをトークンとして認識するのもいいですし、ただ単にD0を加減を送出しながらD0=0の次からデータブロックとして読取るのも1つの手かもしれません。
いずれにしても、最下位ビットの0の次からデータ郡が送出されることになります。

【データについて】

データはカード 標準は512バイトがデータの固まりです。この固まりをセクタといいます。
0x200の整数倍のスタートアドレスで指定し、読み込むデータは必ず512バイトです。(データ長の指定なき場合) 512バイト送出したらカード はデータのエラー訂正のためにCRC32の 2 バイトのデータを送出します。

制御側では必ずこの2バイトも読み込まなければなりません。
 $512+2=514$ バイトを読み込むような制御が必ず必要です。

これは書き込みにも同じく適用します。データトークンを送った後に512バイトのデータを送出しCRCの2バイトを送出します。CRCについてはSPIモード ではチェックしていないため0を2バイト分送出します。

【カードのデータ共有性】

これまでのカード制御の方法でカードに対し書込み読み込みができます。しかし、巨大デバイスとしてアドレス管理を行って使用するしかありません。そこで、パソコンとの共用や他システムとの共通化を考えた場合どうしても決められたフォーマットで読み書きする必要があります。

そこで、現在一般に使用されているマイクロソフトFAT32フォーマットを考えていきたいと思います。組込みシステムを考えた場合フォーマットを組み込み側で行うのではなく、フォーマットされたメディアとして扱うことを前提に考えます。

FAT32は上限4GBまでのメディアに対応し、ロングファイル名にも対応していますが、ここでは従来の名前8文字拡張子3文字に対応したものと限定します。

【FAT32の構造】

カードをパソコンなどでフォーマットを行うとカード上にエリアごとに区分され、パラメーターが随所に書込まれます。まずはこの意味を理解しないとフォーマットに則って読み書きができません。

FATには取り扱う単位としてセクタという単位とクラスターという単位があります。

・セクタ

セクタとはデータを一括で扱う最小の単位を言います。バイト/ビット毎にメディアにアクセスするのではなくある一定量のバイト数をためてアクセスする単位のことです。カードのブロックサイズに当る容量がそれにあたります。ですので、カードの1セクタ=512バイトになります。

・クラスター

クラスターは一定のセクタをまとめたデータの固まりになります。FATは基本的にはこのクラスターという単位で区切られています。1クラスター=Xセクタといった一定量ではなく、メディアの容量によって決められます。256MBのカードで4セクタ/クラスターになっています。また1GBでは8セクタ/クラスターに設定されているようです。

・領域

メディアの全範囲の中でFATは領域を持っています。先頭から予約領域、FAT領域（通常FATはバックアップを持っていますのでFATは2枚持っています。）データ格納領域と大きく3つの領域を持っています。

・パラメータエリア

メディアの領域情報やFAT情報を記述してある場所があります。カードをフォーマットに則ったアクセスをするためには2つのパラメータ領域を読む必要があります。

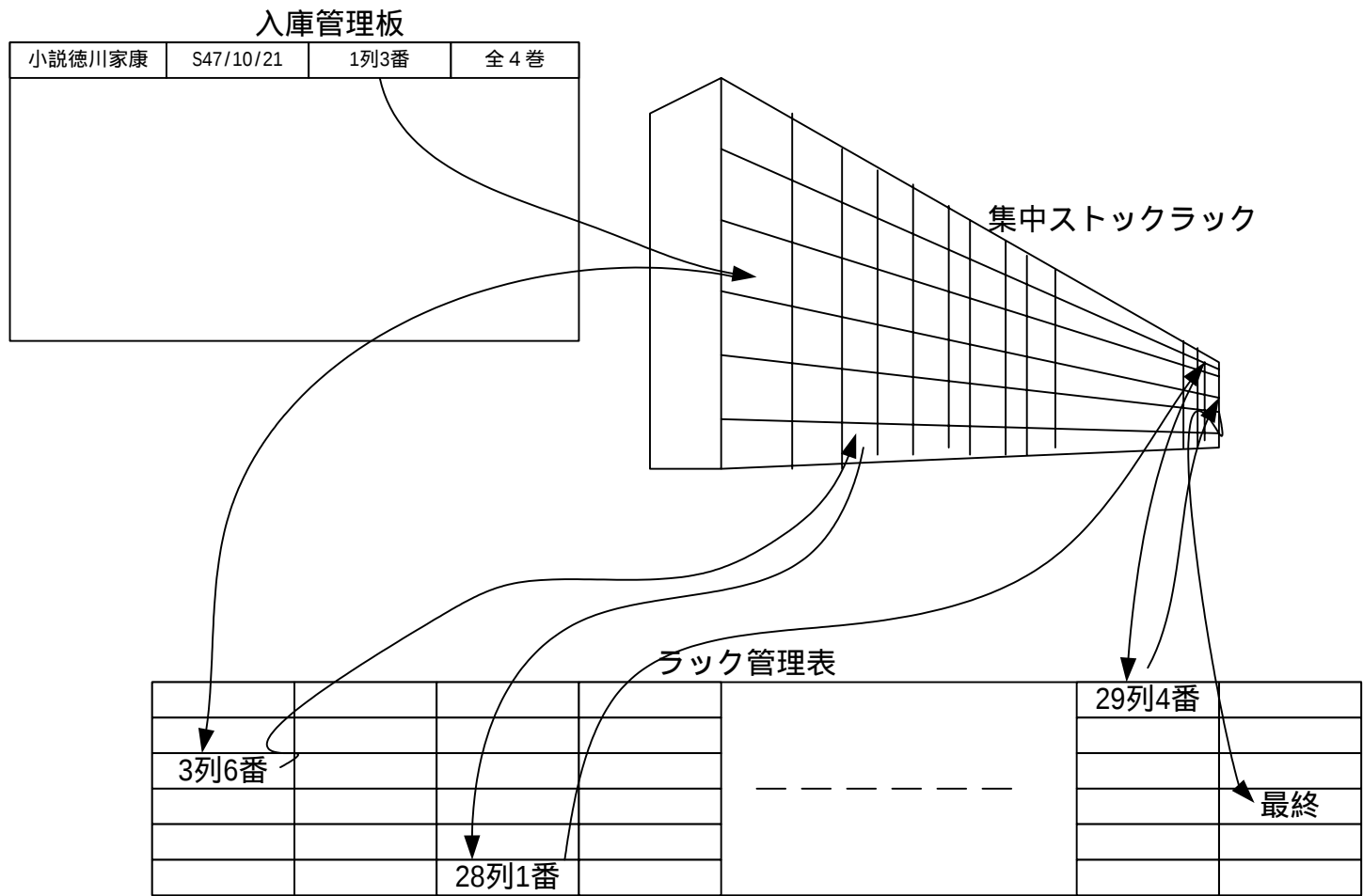
MBR（マスターブートレコード）カードの先頭に記述されている基本的情報エリア

BPB（ブートパラメータブロック）メディアの詳細フォーマット情報を記載されているエリア

・FAT（ファイルアロケーションテーブル）

メディア中のエリア同士のつながり状態、空き状態を把握するためのテーブル。

【FATの概念】

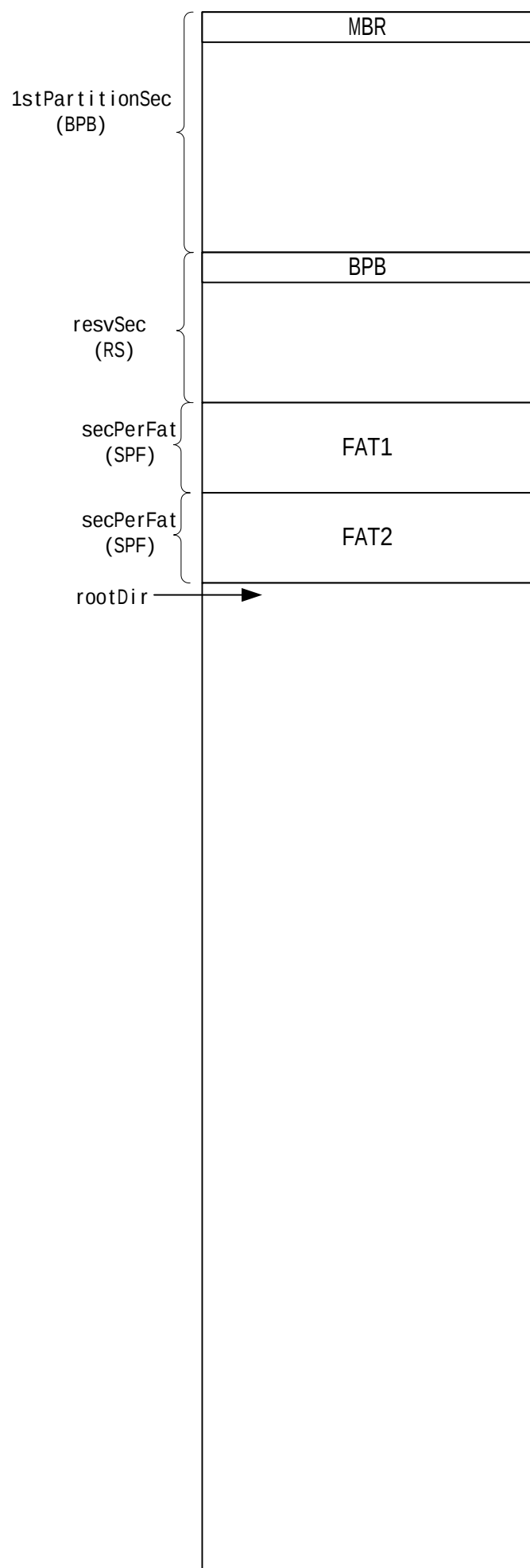


FATは管理された集中ラックで考えた場合、自分の欲しい物を入庫管理板でまず調べます。例えば図で説明すると、「小説徳川家康」は入庫管理板では1列3番に入っています。そして全4巻があることが解ります。まず最初は1列3番にあるのでその場所に入ってるものが最初の本です。次に4巻あるわけですから、これでは足りません。次にラック管理表から1列3番の場所を参照すると、3列6番に次の本があることが解ります。3列6番のラックから本をラック管理板から次というように、順次本を取ってきて最後の29列4番で全4巻を取ってきたことになります。最後のラック管理板はそのまま何も記述しないと空いたラックになってしまいますので、最終のマークを記する訳です。

新規に入庫する場合入庫管理板に記述する前に、ラック管理表から入庫したい分量の空いているラックを調べて空いていたら入庫管理表に記述し順番にラック管理表に続きを記入してきます。

このように入庫管理板に相当するのがルートディレクトリで、ラック管理板がFATになります。集中ストックラックが予約領域以外のエリアに相当します。

【FATの実際のマッピング】



カードの最初のセクタ(512バイト)にはMBRというシステムエリアとカードの属性、最初のパラメータが記述してあります。

【 MBR 】

ロケーション	バイト数	名称
[0..445]	446	SYSTEM
[446]	1	bootDescriptor
[447..449]	3	1stPartitionSec
[450]	1	fileSysDescrp
[451..453]	3	lastPartiSec
[454..457]	4	1stSecNum
[458..461]	4	numOfSec

MBR[454..457]の1stSecNumがBPBの位置を示しています。この値を読んでBPBのセクタのアドレスを得ます。

カードのデータはビッグエンディアンになっていますので、もし4バイトのデータであれば若い方のアドレスにあるデータが最下位バイトで大きいアドレスにあるデータが最上位バイトになります。これはFATで規定されている約束事項です。

【 BPB 】 (FAT32)

ロケーション	バイト数	名称
[0..2]	3	jmpOpeCode
[3..10]	8	OEMname
[11..12]	2	BytePerSec
[13]	1	SecPerClus
[14..15]	2	resvSec
[16]	1	numOfFat
[17..18]	2	rootEntry
[19..20]	2	totalSec
[21]	1	mediaDescrpt
[22..23]	2	secPerFat
[24..25]	2	secPerTrack
[26..27]	2	heads
[28..31]	4	hideSec
[32..35]	4	bigTotalSec
[36..39]	4	bigSecPerFat
[40..41]	2	extFlag
[42..43]	2	FS_Ver
[44..47]	4	rootDirStrtClus
[48..49]	2	FSinfoSec
[50..51]	2	bkupBootSec
[52..63]	12	reserved
[64]	1	driveNum
[65]	1	unused
[66]	1	extBootsig
[67..70]	4	serialNum
[71..81]	11	volName
[82..89]	8	fileSysType
[90..509]	420	loadProgCode
[501..511]	2	sig "55 AA"

【FATで重要なパラメータ】

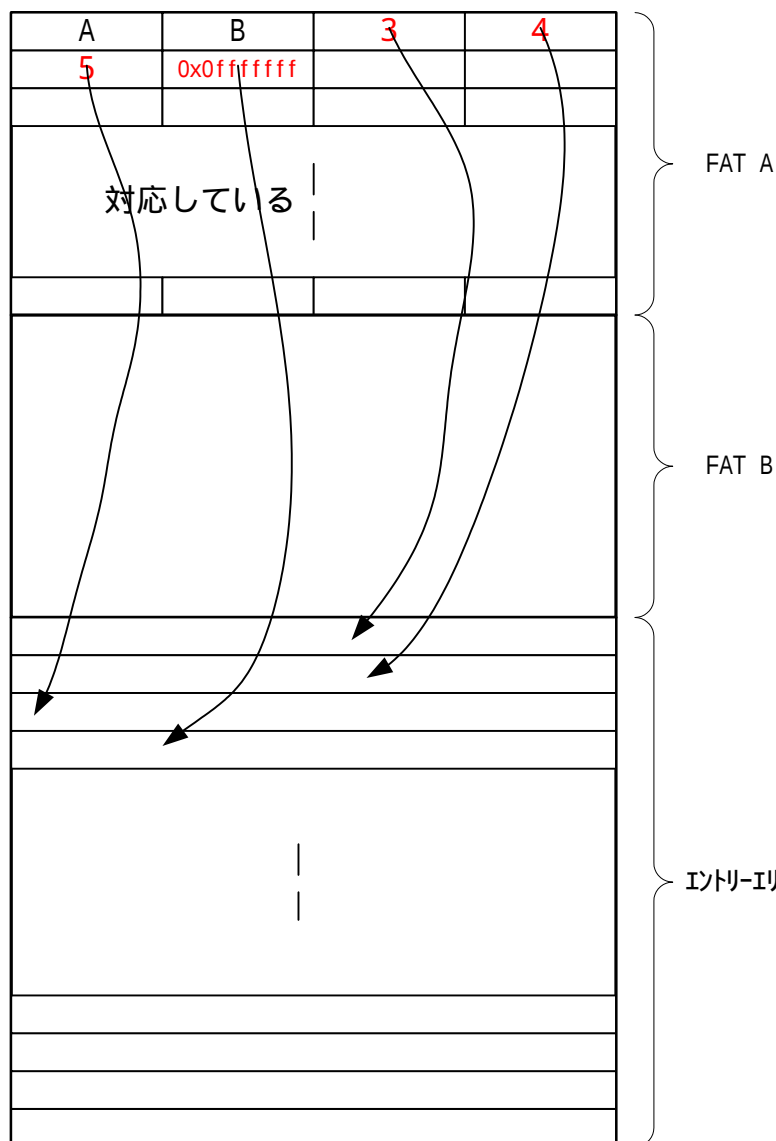
- BPB(1stSecNum)
MBRからBPB位置を抽出します。MBRからBPBまでの領域はシステムの領域でユーザには使用できません。
- RS(resvSec)
リザーブ領域ですBPBを含めたシステムの領域ですこれもユーザは使用することができません。
- SPF(bigSecPerFat)
secPerFat2バイトのFATのセクタ数が格納されています。これを参照してFATの領域を確認します。
- SPC(SecPerClus)
secPerClusで1セクタに幾つのセクタが割当てられるかを格納しています。
- 最初のFAT番号(rootDirStrtClus)
ユーザが使える最初のFAT番号です。通常2となっています。ルートディレクトリ先頭の番号です。

SPFで示される数がFATに割当てられるセクター数ですが、1セクタ512バイトです。FATは4バイトで1クラスタを示しますから、1セクタ128クラスタとなります。ここで示された値と128を積算すると総クラスタ数、総セクタ数、つまりは総バイト数が判別できます。

FATは2枚構成で2つとも同じデータになっています。ここでは先頭に近い方からFAT A、FAT Bと呼びます。メインに書き換えを行うのをFAT A側で行いFATの構成が確定したらFAT BにFAT Aからコピーするといった方法をとります。

【FAT】

FATは上記から何処にセクタがあるのかが解ります。BPB+RSがFAT Aの先頭セクタです。このFATは実際どのようなになっているか。FATはFAT32の仕様では4バイト単位がひとつの領域になります。この4バイトがひとつのクラスタを表しています。FATの先頭から4バイトずつFATの後に続くセクタと結びついているのです。



FATの先頭2つはシステムのためのFATです。ですからBPBで読取ったrootDirStrtClusが2からスタートしているのです。

FAT上の4バイト=1ロングワードは先頭のクラスタと同じ順番に並んでいます。そしてFAT情報によって次のデータは何処のクラスタ番号なのかをこのFATに記してるのです。

クラスタ番号の意味	
番号	意味
0x00000000	未使用クラスタ
0x00000001	予約クラスタ
0x00000002	有効なユーザデータクラスタ番号
~	
0x0fffffff6	
0x0fffffff7	不良クラスタ
0x0fffffff8	
~	
0x0fffffff	最終クラスタ

赤字は実際の例を記しています。クラスタ2に続くデータはクラスタ3でそれに続くデータがクラスタ4、次にクラスタ5、クラスタ6までが1つの連続したデータということがFATによって解ります。

では最初のクラスタの指示はというと後で説明するディレクトリ上のファイルエントリに書いてあります。

【ディレクトリエントリ】

BPB+RS+2(FAT)=ルートディレクトリであることは解りました。
ディレクトリなのでファイルに限らずサブディレクトリをエントリすることができます。エントリフォーマットは以下のようになっています。

名称	ロケーション	バイト数	備考
name	[0..7]	8	ファイル名
extension	[8..10]	3	拡張子
attribute	[11]	1	ファイル属性
reserved	[12]	1	
createTime_ms	[13]	1	
createTime	[14..15]	2	
createDate	[16..17]	2	
accessDate	[18..19]	2	
clusterHiword	[20..21]	2	上位ワード
updateTime	[22..23]	2	
updateDate	[24..25]	2	
cluster	[26..27]	2	下位ワード
filesize	[28..31]	4	ファイルの総バイト数

エントリエリアに上記のフォーマットでファイルまたはディレクトリをエントリしていきます。1セクタは512バイトなので1セクタあたり16エントリ数記述することができます。1クラス以上エントリできますからSPCの数によって1クラスにエントリできる数は可変します。しかし、FATに続きの番号を指定することによってファイル数は可変できます。

● name

MS-DOS形式の8.3(名前 8 文字 + ドット + 拡張子 3 文字)ファイル名に属する先頭の 8 文字にあたる場所です。以下の制約があります。

- ・ ASCII 英数字で大文字で記述する
- ・ [] . " / ¥ : ; | = を使うことができない
- ・ CON AUX COM1 ~ 4 LPT1 ~ 3 PRN NUL の名前の使用禁止

先頭の 1 文字目には下記の意味合いも含まれます。

- ・ 0x00 空エントリ
- ・ 0x05 削除されたエントリー（使用可能）
- ・ 0xE5 削除されたエントリー（使用可能）

8 文字以下の名前の場合空いたスペースには0x20（スペース）で埋めます。

● extension

拡張子にあたる 3 文字の領域です。適用はnameと同じです。但し先頭の意味合いはありません。