第2章

さまざまなPCカードを認識するための情報

PCカードCISフォーマット詳解

大中 邦彦／熊谷 あき

PCカードには，そのカードがどこのメーカの何のカードか，電源電圧やアクセス・タイミングはどうか，メモリやI/O，割り込みといったハードウェア・リソースをどれだけ必要とするか，そしてどのアドレスにマッピング可能かといった情報(CIS)がアトリビュート・メモリ領域に記録されている．このCISが規格化されているからこそ，PCカードではプラグ＆プレイが実現できる．ここではCISの解析手順を，実際のPCカードを例にとりながら順を追って解説する．（編集部）

1　PCカードとCIS

当初はメモリ・カードしかなかったPCカードですが，今では実にさまざまな種類のカードが存在します．1枚のカードで複数の機能をもつマルチファンクション・カードや，3.3Vで駆動できるカード，CPUにPentium Ⅱなどを搭載したノート・パソコンの普及でPCIバス並みの転送速度をもつCardBusカードも多く見かけるようになりました．

さらに最近では，16ビットPCカードとCardBusカード両方に対応するというPCカードまで登場しています．こんなにもいろいろな種類がありながら，同じPCカード・スロットに差し込んで使えるというのは驚きです．

PCカードは，このような柔軟なプラグ&プレイを実現するために，カード内部にCIS(Card Information Structure)という情報テーブルをもつように定められています．CISはその名のとおり，カードの情報を納めた構造体です．CISは，通常アトリビュート・メモリの先頭から格納されており，タプルと呼ばれる小さな構造体が集まって構成されています．

PCカードが挿入されるとシステムはまずCISを解析し，挿入されたPCカードがどのような種類のカードなのか見極めるわけです．システムに挿入されるPCカードの種類があらかじめ分かっている場合でも，予期せぬカードが挿入された際に誤った電圧を加えてしまってカードを壊したりしないように，きちんとCISの解析を行った方がよいでしょう．

CISの解析とは，つまりタプルの解析ということになります．タプルには非常に多くの種類が存在します．タプルは，カードの情報を柔軟に記述できるように規定されている反面，解析が非常に面倒な構造をしています．本章では，PCカードを使う上で特に重要なタプルについて，そのフォーマットを解説していきます．

● CISを16進数ダンプ表示する

何はともあれCISがどんなものなのか見てみましょう．ここでは一足早く(?)，第4章で設計している16ビットPCカードのCISを例にしてみます．**リスト1**(**a**)が16ビットPCカードのアトリビュート・メモリ空間を先頭からダンプ表示したものです．第1章で解説があるように，16ビットPCカードのアトリビュート・メモリ空間は，偶数アドレスしか使用されていません．これを見やすいように偶数アドレスのみ読み出してダンプ表示したものが**リスト1**(**b**)です．

しかし16進数ダンプ表示では，人間にとっては非常に分かりにくいことに変わりありません．そこで人間が理解しやすいように，タプル内容を解析して表示するツールというものが存在します．ここではPC UNIXの一つであるFreeBSD+PAOを紹介します．PAOとは，FreeBSDでPCカードをサポートするためのソフトウェア・パッケージです．この環境でPCカードをPCカード・スロットに挿入し，次のコマンドを実行してみましょう．

リスト1
CISのダンプ表示の様子

```
      +0 +1 +2 +3 +4 +5 +6 +7   +8 +9 +A +B +C +D +E +F | ---- ASCII -----
000h  01 FF 03 FF DC FF 00 FF - FF FF 17 FF 03 FF 1C FF | ................
010h  00 FF FF FF 21 FF 02 FF - 03 FF 00 FF 15 FF 1E FF | ....!...........
020h  04 FF 01 FF 4B FF 75 FF - 72 FF 75 FF 73 FF 75 FF | ....K.u.r.u.s.u.
030h  67 FF 61 FF 77 FF 61 FF - 2D FF 65 FF 6C FF 65 FF | g.a.w.a.-.e.l.e.
040h  2E FF 00 FF 50 FF 49 FF - 4F FF 20 FF 50 FF 43 FF | ....P.I.O. .P.C.
050h  43 FF 61 FF 72 FF 64 FF - 00 FF FF FF 1A FF 05 FF | C.a.r.d.........
060h  01 FF 04 FF 00 FF 02 FF - 03 FF 1B FF 0F FF C1 FF | ................
070h  01 FF 99 FF 09 FF 55 FF - 26 FF D0 FF 60 FF 00 FF | ......U.&...`...
080h  02 FF 07 FF 30 FF FF FF - FF FF 00 FF 1B FF 07 FF | ....0...........
090h  02 FF 08 FF D0 FF 60 FF - 80 FF 02 FF 07 FF 1B FF | ......`.........
0A0h  07 FF 03 FF 08 FF D0 FF - 60 FF 00 FF 03 FF 07 FF | ........`.......
0B0h  1B FF 07 FF 04 FF 08 FF - D0 FF 60 FF 80 FF 03 FF | ..........`.....
0C0h  07 FF 14 FF 00 FF FF FF - FF FF FF FF FF FF FF FF | ................
0D0h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
0E0h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
                        ～すべてFFh～
1E0h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
1F0h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
200h  40 FF 00 FF FF FF FF FF - FF FF FF FF FF FF FF FF | @...............
210h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
220h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
                        ～すべてFFh～
```

（**a**）アトリビュート・メモリ空間を読み出した場合

```
      +0 +1 +2 +3 +4 +5 +6 +7   +8 +9 +A +B +C +D +E +F | ---- ASCII -----
000h  01 03 DC 00 FF 17 03 1C - 00 FF 21 02 03 00 15 1E | ..........!.....
010h  04 01 4B 75 72 75 73 75 - 67 61 77 61 2D 65 6C 65 | ..Kurusugawa-ele
020h  2E 00 50 49 4F 20 50 43 - 43 61 72 64 00 FF 1A 05 | ..PIO PCCard....
030h  01 04 00 02 03 1B 0F C1 - 01 99 09 55 26 D0 60 00 | ...........U&.`.
040h  02 07 30 FF FF 00 1B 07 - 02 08 D0 60 80 02 07 1B | ..0........`....
050h  07 03 08 D0 60 00 03 07 - 1B 07 04 08 D0 60 80 03 | ....`........`..
060h  07 14 00 FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
070h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
080h  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF | ................
```

（**b**）偶数アドレスのみ読み出した場合

```
>pccardc dumpcis
```

実行すると，**リスト2**(**a**)のように何やらたくさんのメッセージが表示されます．このプログラムは，PCカードに記述されているタプルを順に読んでいき，人間に分かりやすいメッセージに解析して表示してくれます．Linuxなどのほかの PC UNIXでも同じようなツールが存在するはずです．

また，DOS版のツールとして第7章でもタプル解析表示ツールを作成しているので，そちらも参考になるでしょう〔**リスト2**(**b**)〕．

● タプルの基本フォーマット

一つのタプルは，**図1**に示すような構造をしています．先頭1バイトがタプルの種類を示すタプル・コードで，2バイト目が次のタプルへのアドレス・オフセット値(タプル・リンク)で，3バイト目からタプル本体(タプル・データ)となっています．次のタプルへのオフセットが1バイトしかないので，タプルのサイズは全体で256バイト以下と規定されています．

ここで注意すべき点があります．**リスト1**でも示したように16ビットPCカードのアトリビュート・メモリは，偶数番地にのみにデータが書かれています．つまり，実際にはアドレスは2バイトずつインクリメントしていくことになります．しかし，タプル・リンクの値には偶数番地のみを数えたオフセット・バイト数が記されているのです．

プログラムでタプルを扱うときは，タプルを連続したメイン・メモリにコピーして扱うことも多いと思われるので，その際は偶数番地のみをバイト単位で読み出し，メイン・メモリ上では連続したデータになるようにコピーすれば，そのままオフセット値で計算でき

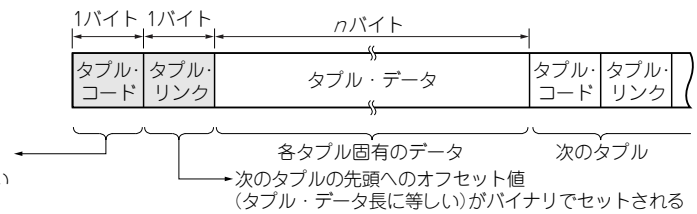

表1のタプル・コード
タプルの全長(n+2)は最大256バイト．
タプル・リンクが0FFhの場合，次のタプルは存在しない
(リンク・チェイン終了)

図1　タプル・フォーマット