

新入生歓迎の言葉

MMA 部長 shimiz98¹

2000年4月

MMA はコンピューター、特にパソコンで動く UNIX のサークルです。サークルとしてまとまって何かを作ったりはしていないので活動内容を一言で言うのは難しいのですが、C 言語で X Window System のプログラムを書いたり、VC++ で MS Windows のプログラムを書いたり、Perl で CGI を書いたり、XML で 日記を書いたり、Ruby や Emacs Lisp や Guile を使っている人がいます。

MMA は Microcomputer Making Assocation の頭文字をとった名前です。昔は CPU を買って来て 半田づけなどをして独自のコンピュータを作っていましたが、現在はソフトウェア関係に比重が移っています。もちろん AT 互換機ぐらいはみんなやりますし、Sparc というか Sun が好きな人もいます。

元部長の受け売りですが、**MMA は緑と環境を提供します。** という言葉が MMA を良く表していると思います。

emacs の使い方 基本編

君島 秀征¹

2000 年 4 月

本文書は、emacs に初めて触ってから一週間後くらいの人を対象としている。従って、emacs の入門にあるような、基本用語の説明、終了、起動、C-*, M-*の意味、M-x command, C-g については敢えて書かない。

私自身が「5 年前にこの文書があれば、もっと早く emacs が使えるようになっていたのに」と思えるような文書を目指して書いたつもりである。若い方々が emacs に慣れる一助となれば幸いである。

ヘルプ

emacs には強力なオンラインヘルプ機能がある。英語でしか提供されていないが、もし日本語で提供されれば日本の emacs 人口は増えるだろうにな。

C-h b	describe-bindings	定義されているキーバインドを全て表示
C-h k	describe-key	キーの説明を表示
C-h c	describe-key-briefly	キーに結び付けられている関数名をミニバッファに表示
C-h w	where-is	そのコマンドが結び付けられているキーを表示
C-h f	describe-function	関数の説明を表示
C-h v	describe-variable	変数の説明を表示
C-h a	command-apropos	コマンドの名前を正規表現で探す
C-h m	describe-mode	現在の mode のドキュメントを表示
C-h ?	help-for-help	上記のヘルプのヘルプ

C-h b で、使えるキーの一覧が表示される。C-h k の後に任意のキーを押すと、そのキーの説明が出てくる²。

以上で、本文書の趣旨の 8 割は終わった。残りはおまけのようなものである。

カーソル移動

emacs のゲテモノな部分の一つ。昔のハードウェアの制限に由来して、それを現在まで引きずっているのではないかと思われる。

C-b	backward-char	左
C-f	forward-char	右
C-p	previous-line	上
C-n	next-line	下
C-a	beginning-of-line	行頭
C-e	end-of-line	行末
M-<, home	beginning-of-buffer	文頭
M->, end	end-of-buffer	文末
M-v, PageUp	scroll-down	1 ページ上
C-v, PageDown	scroll-up	1 ページ下

カーソルキーも使えるが、emacs に慣れてきたら上記のキーも使ってみてほしい。使えばすぐに慣れるし、慣れれば意外と使いやすい。

バッファ操作

C-x k	kill-buffer	バッファを kill する
C-x b	switch-to-buffer	別なバッファに切り替える
C-x C-b	list-buffers	バッファ一覧を表示する

C-x b を実行するとバッファ名の入力を促される。ここでファイルの basename を入力するが、TAB 補完が使える。

C-x C-b でバッファ一覧が表示される。表示したいバッファの上で RET か o を押す。また、k で印を付けて x を押すと、選択したバッファを kill できる。

²本文書を書くのにも大いに役立った。

カット/コピー&ペースト操作

C-SPC	set-mark-command	範囲指定の始点(マーク)を設定
C-w	kill-region	選択範囲(リージョン)をカット
M-w	kill-ring-save	選択範囲(リージョン)をコピー
C-y	yank	ペースト
M-y	yank-pop	ペーストする文字列を選択する

C-SPC と C-w/M-w でカット/コピーし、C-y でペーストできる。過去にコピーした文字列は全て記憶されていて、C-y した後に M-y を実行すると、ペーストする文字列を過去に遡る。

ちなみに、(transient-mark-mode 1) を実行するとリージョンが反転表示されるようになる。早めに.emacs に書いておこう。

ウィンドウ操作

C-x 2	split-window-vertically	ウィンドウを上下に分割する
C-x 3	split-window-horizontally	ウィンドウを左右に分割する
C-x o	other-window	次のウィンドウに移る
C-x 1	delete-other-windows	カレントウィンドウ以外を消す
C-x 0	delete-window	カレントウィンドウを消す

アンドウ

C-x u	advertised-undo	アンドウ
-------	-----------------	------

C-x u を続けて実行することで、アンドウを遡ることができる。途中で C-g を押すと遡る向きが切り替わり、C-x u がリドゥになる。

整数の n 進数表記を求める—Ruby 編

君島 秀征¹

2000年3月23日

昨年(2000年)の5月27日「任意の整数の n 進数表記を求める」ことが話題になり、数名がそれぞれ得意な言語で実装し、誰が最初に作り上げるか競争したことがありました²。

以下が、Ruby による実装です。なるべく簡潔になるよう書いてみました。

```
class Integer
```

```
  # 整数の n 進数表記を求める。
```

```
  # 例) p 100.base(16)
```

```
  def base(n)
```

```
    str = (self % n).to_ss
```

```
    if (a = self / n) > 0 then a.base(n)+str; else str; end
```

```
  end
```

```
  protected
```

```
  def to_ss
```

```
    if self > 9 then (self-10+65).chr; else (self+48).chr; end
```

```
  end
```

```
end
```

使い方

例えば、`100.base(16)` とすることにより、100 の 16 進数表記の文字列を得られます。

説明

整数オブジェクトへの操作なので、Integer クラスのメソッドとしました。to_ss メソッドはサブルーチンのみで使っているだけですので、可視性を protected とし、Integer クラスの中からは呼べないようにしてあります。

²uirou さんが C で、fujita くんが Scheme で、私は Ruby でした。

実用的 lex&yacc 入門

fujita@mma.club.uec.ac.jp

2000 年 4 月

Abstract

昨今認知度が上がったフリーな UNIX ライク OS には、コンパイラやデバッガ、各種ライブラリなど様々なプログラミングツールが標準で付属しています。lex と yacc もそんなツールの一つです。しかし、lex と yacc についての文書、特に実際の使用法を解説したものはなかなかありません。大概是文字列一致のアルゴリズムとか有限状態機械、言語理論といった小難しい理屈が並んでいて一気にやる気を奪ってくれます¹。そこで、昨年 lex と yacc の泥沼にはまり、そのまま食物連鎖の最底辺になってしまった筆者が、おこがましくも lex と yacc の現実的な使い方を紹介したいと思います。

1 lex の概要

lex は LEXical analyzer の名の通り、任意の文字列を検出するプログラム (サブルーチン) を作ってくれます。文字列を特定するために正規表現²を使うことも出来ます。しかし、取り出したい文字列に対応する正規表現だけを書いても何も起こりません。lex を使うときにプログラマがしなければならないことは、自分がどんな文字列を検出したいかを考え、そしてその文字列が検出されたときどのような処理をしたいかを明確にすることです。

2 lex の実際

では実際の lex のコードで説明をしてみましょう。こういうことは実例を使った方が説明が楽です。

```
1: %{
2: #define yywrap() 1
3: %}
4: NUMBER [0-9]
5: %%
6: {NUMBER}+ {printf("input -> %s\n",yytext);}
7: "+" {printf("input -> \'+\'\\n");}
8: "-" {printf("input -> \'-\'\\n");}
9: "*" {printf("input -> \'*\'\\n");}
10: "/" {printf("input -> \'/\'\\n");}
11: "(" {printf("input -> \'(\'\\n");}
12: ")" {printf("input -> \')\'\\n");}
13: \n {;}
14: %%
15: int main(int argc,char **argv){
16:     yylex();
17:     exit(0);
18: }
```

まず 1 行目から 3 行目にかけて。この “%{” と “%}” の間には #include 文や #define 文、外部変数や関数のプロトタイプの宣言などを書くことが出来ます。それだけです。

4 行目は正規表現のかたまりを定義しています。ここで定義することによって、次のパターンの羅列での正規表現が見易くなります。

5 行目の “%%” 以降の 13 行目までが lex の肝の正規表現です。一つの正規表現に対しては一つのアクションを “{” と “}” の間で定義することが出来ます。このアクション部分は当然 C 言語で書きます。6 行目は特徴的です。このアクション部分で使用されている変数 “yytext” は、正規表現にマッチした文字列へのポインタです。また、さっき定義した “NUMBER” もここで使われています。7 行目から 13 行目までは単独の文字をマッチさせています。13 行目のように、改行文字は “\n” で、タブは “\t” で表すことが出来ます。マッチした文字列に対して何もアクションを定義しないのであれば、13 行目のように書きます。

14 行目の “%%” でパターンマッチング部分と 15 行目以降の C 言語の部分とを区切ります。特に関数などを書く必要が無ければ、14 行目以降はいりません。

¹NUTSHELL 本などは特に。

²正規表現の説明はしません。各自で調べてください。

15行目から18行目まではプログラムのmain関数です。16行目でmain関数は関数“yylex”を呼び出しています。この関数“yylex”がlexの作り出すパターンマッチングのための関数です。プログラマは関数“yylex”を呼ぶだけで、容易にパターンマッチングとマッチした文字列に対する処理を行なうことができます。

では実際にコンパイル・実行してみましょう。

```
% lex sample.l
% gcc lex.yy.c
% ./a.out
1+2-3*(4/3)
input -> 1
input -> '+'
input -> 2
input -> '-'
input -> 3
input -> '*'
input -> '('
input -> 4
input -> '/'
input -> 3
input -> ')'
%
```

lex コマンドは“-o” オプションで明示的にしない限り、“lex.yy.c” というファイルを生成します。このファイルには先ほどのmain関数も含まれているため、そのままコンパイルすることが出来ます。実行してみるとパターンマッチングと、それに対応するアクションが期待通りの動作をしていることがわかります。

3 yacc の概要

yacc は Yet Another Compiler Compiler、即ちコンパイラを作るためのコンパイラです。と言うと何だかスゴそうな感じに聞こえますが、そんなに大層なものではありません。yacc の作業は、lex によって検出された文字列がプログラマの指定通りの文法で並んでいるかを判断するだけです。ようするに、yacc は lex がいないと何も出来ないわけです³。lex と yacc が対で説明される理由はここにあります。

さて、yacc も lex と同様に、文法の記述をただ並べても何も起こりません。文法と、その文法に合致したときの処理をプログラマは考えなければいけません。

4 yacc の実際

今度も実際のコードを見てみましょう。まずは lex のコードです。

```
%{
#include "y.tab.c"
#define yywrap() 1
}%
NUMBER [0-9]
%%
{NUMBER}+ {return(NUM);}
"+" {return('+')}
"-" {return('-')}
"*" {return('*')}
"/" {return('/')}
"(" {return('(')}
```

³lex と同じ機能のコードを一から書くのなら、lex を使う必要はありません。が、そんな面倒なことをする人もいないでしょう。

```

)" {return('')};}
[\n \t]+ {return(SPACE);}
%%
int yyerror(char *s){
    fprintf(stderr,"ERROR -> %s\n",s);
}
int main(int argc,char **argv){
    yyparse();
    exit(0);
}

```

以前と色々違いますが、今注目するのはパターンマッチングしたときのアクションです。return 文で“NUM”とか“+”を返しています。ここで返した値は yacc の関数に戻り、どの文字列を検出したかを判断するのに使われます。この return 文によって lex と yacc が結合されるのです。

次に yacc のコードです。

```

1: %token NUM
2: %token SPACE
3: %%
4: state : expr {printf("result -> %d\n",$1);}
5:       | state expr {printf("result -> %d\n",$2);}
6:       ;
7: expr : prim {$$=$1;}
8:       | space '(' '+' space expr space expr space ')' space {$$=$5+$7;}
9:       | space '(' '-' space expr space expr space ')' space {$$=$5-$7;}
10:      | space '(' '*' space expr space expr space ')' space {$$=$5*$7;}
11:      | space '(' '/' space expr space expr space ')' space {$$=$5/$7;}
12:      ;
13: prim : NUM {$$=atoi(ytext);}
14: space : {;}
15:       | SPACE {;}
16:       ;

```

1、2行目の%token文によって、lexがパターンマッチ時にreturn文で返す値を記述します。実際には%token文で指定された“NUM”や“SPACE”はyaccコマンドによって#define文に直され、定数のマクロとして扱われます。しかしプログラマにとっての%token文は、単にlexのコードでreturnした値をyaccのコード中で使えるようにするための形式的な呪文でしかありません。

3行目の“%%”以降がyaccの核となる文法を記述した部分です。ここでは文法の定義は四つに分けてあります。4行目から6行目は“state”を定義しています。7行目から12行目は“expr”を、13行目は“prim”を、14行目から16行目は“space”を定義しています。

説明のために4行目から6行目を見てみましょう。ここでは“state”というものを定義しています。この“state”というのは筆者の趣味であり、プログラマが任意の名前をつけても一向に構いません。さて、“state”の次に“:”が来ています。これが文法定義の印です。即ち、4行目を簡単に言うと「“state”とは“expr”のことである」となります。その後が続いているprintf文は“state”が本当に“expr”だったときのアクションです。このprintf文では謎の変数\$1を表示しようとしています。変数\$1は“expr”をさらに細かく文法解析した結果の値です。この変数の意味は後になってから解ってきます。次に5行目では“|”があり、さらに定義が続くことを示しています。そして6行目でようやく“;”があり、“state”の定義が終わります。「“state”とは、“expr”であるか、または“state”と“expr”の並びである」というのが4行目から6行目までの定義の意味です。

ところで、5行目のアクションにはこれまた謎の変数\$2が登場します。先刻出て来た変数\$1との違いは、5行目における“expr”の位置です。5行目の“expr”は“state”の次にあります。即ち、“state”が一番目で“expr”が二番目です。今、表示したいのは“expr”をさらに解析した結果の値なので、それを明示するために\$2という変数を用います。もしこれが\$1であれば、それは“state”を解析した結果を表示することになってしまいます。この“\$数字”という変数の扱いは8行目から11行目でも同様です。\$5と\$7、五番目と七番目の項目、即ち“expr”を解析した結果を演算しています。この変数“\$数字”は、perlで“m/(a*)(b*)(c*)/”としたときに括弧内にマッチした文字列を変数\$1、\$2、\$3で参照出来るのと同じようなものと考えると理解しやすいかもしれません。

“\$数字”という変数で解析結果の値を参照出来ることは説明しました。今度はある値を文法解析の結果の値として返す方法についてです。7行目から12行目までの“expr”の定義を見てみましょう。アクション部分に謎の変数\$\$が再三再四出て来ます。この変数

\$\$は“expr”を解析した結果の値を格納する変数です。即ち、この変数\$\$が“state”のアクションで\$1や\$2として参照されていたものの実体なのです。“expr”の定義を見ると、\$\$は演算子の種類によって様々な値を取ることが分かるでしょう。ここで\$\$に代入された値が、“state”のprintf関数で表示されたり、またさらに演算に使われたりするのです。

13行目がこの文法の最も下の定義です。“NUM”はlexのコード中でreturnされるもので、数字の並びにマッチしたことを表わします。lexでマッチした文字列は変数“yytext”で参照出来るので、atoi関数を用いて文字列から数字に直したものを変数\$\$に代入して、“expr”の7行目のアクションで使えるようにしています。

14行目から16行目の“space”の定義は改行やタブ、空白文字があっても何もしないことを定めています。この定義がある理由は実行してみると分かるので、実際にコンパイル・実行してみましょう。

```
% yacc sample.y
yacc: 12 shift/reduce conflicts
% lex sample.l
% gcc lex.yy.c
% ./a.out
(+ (- 1 2)
  (* 3 (/ 4
        2
      )
  )
)
^D
result -> 5
%
```

yacc コマンドのエラーは文法が曖昧だったときに起こるものです⁴。ここではあまり気にする必要はありません。yacc コマンドも“-o” オプションを付けて明示しない限り、“y.tab.c”というファイルを生じます。このファイルをlexのコードの冒頭でincludeしているため、ファイル“lex.yy.c”をコンパイルするだけで実行ファイルを作ることが出来ます。

さて、話が随分前に戻りますが、lexのコードの関数定義の部分を見てください。関数“yyerror”は文法エラーを起こしたときに呼ばれる関数で、プログラマが任意で定義することが可能です。これが無ければコンパイラに「undefined reference to ‘yyerror’」と怒られるでしょう。また、main関数で呼ばれている関数“yyparse”はyaccコマンドが作り出す、文法解析をするための関数です。関数“yylex”と同様に、プログラマは関数“yyparse”を呼ぶだけで、容易に文法解析と解析に対する処理を行うことが出来ます⁵。

yaccの文法を見ると分かるように、サンプルとして提示していたのは単純なLISPチック電卓です⁶。yaccの文法でわざわざ空白文字や改行文字を間に狭んであっても良いようにしたのは、LISPでは改行や空白は意味を持たず、“(”と“)”の対応が全てであるからだ、ということが実行結果から分かっていただけのものでしょうか。

5 その他の TIPS

lexが文字列を読み込むのは通常は標準入力からですが、変数yyinにFILE構造体へのポインタを代入することで、そのファイルから文字列を読み込ませることが出来ます。出力は変数yyoutで同様に切り替えられます。

yaccの変数\$\$や変数\$1の型は通常はint型ですが、プログラマが任意で指定することも出来ます。ただし、この型はunionであるという制限が付きます。例えば次のようにします。

```
%union{
  char *str;
}
%token ...
```

こうすると\$\$strや\$1.strという形で値を代入したり参照することが出来ます。そして、このunionの型はyaccの外部でもYYSTYPEという型で使うことが可能です。

⁴“if-then-if-then-else”の“else”がどちらの“if”にかかるのか、という問題が有名ですね。

⁵まとめると、プログラマは自分のコードの任意の場所で関数“yyparse”を呼び、“yyparse”はパターンマッチングのために関数“yylex”を呼ぶ、という構成になっています。

⁶これもまた筆者の趣味です。:p

また、yacc の外部で大域変数 `yyval` に値を代入することによって、`$$` と同じ効果を得ることが出来ます。この変数を用いて先の例を次のように変えても、動作に変わりはありません。lex のコードは

```
{NUMBER}+ {yyval=atoi(yytext);return(NUM);}
```

に、yacc のコードは

```
prim : NUM {$$=$1;}
```

に変えます。変数 `yyval` の値は `$1` や `$2` といった変数で参照します。

yacc の `%token` 文での定義を他のコードで使いたくなるかもしれません。これらの定義はヘッダファイルとして取り出せ、yacc コマンドに `-d` オプションを付けてファイル `"y.tab.h"` を得ることが出来ます。ファイル `"y.tab.h"` には `%token` 文の定義が `#define` 文のマクロに直されて書かれています。

6 最後に

lex と yacc の初歩的な書き方を述べてきましたが、大抵のコードはこれ位の知識で書けるはずで、重要なのはエラーの無い文法を事前に作っておくことと、その文法を解析して何をさせたいかをプログラマ自身が把握しておくことです。

特に言語のインタプリタやコンパイラを作ろうとするのであれば、lex のパターンマッチングや yacc の文法解析はさほど大事ではありません。問題は、文法を解析してどのような解析木を作るか、その木を解釈して実行するルーチンをどのように作るか、あるいはどういうアセンブラに焼き直すかを考えることです。lex と yacc を使うとき、実にこの部分が一番苦労します。

では、最後に一言。

結局 perl には勝てない。

#ダメじゃん…。

参考文献

[1] FreeBSD 日本語マニュアルプロジェクト, 「lex(1)」, <http://www.jp.freebsd.org/man-jp/>

[2] 情報工学実験第二 P コースの配布資料

L^AT_EX で数式を表示しよう

omaeda-
2000 年 4 月

Abstract

数式を表示する例文集です。よかったらソースに組み込んで使ってください。

1 行列

行列を表示するには `array` 環境を使います。

```
\normalsize \[
A = \left( \begin{array}{rrr}
-1 & 2 & 4 \\
3 & 5 & 1
\end{array} \right), \quad
B = \left( \begin{array}{cc}
5 & -1 \\
8 & 7
\end{array} \right), \quad
C = \left( \begin{array}{lll}
1 & 5 & 0 \\
2 & -3 & 6 \\
4 & 0 & 3
\end{array} \right)
\]
```

出力結果

$$A = \begin{pmatrix} -1 & 2 & 4 \\ 3 & 5 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 5 & -1 \\ 8 & 7 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 5 & 0 \\ 2 & -3 & 6 \\ 4 & 0 & 3 \end{pmatrix}$$

`array` の次の中括弧の中身は `r` のほかに `c`、`l` があり、列数だけ書きます。
それぞれ 行列の数字を右づめ、中央、左づめにします これを応用して連立一次方程式を表示しましょう。

`\normalsize` 次の連立一次方程式を解け。

```
\[ (1)
\left\{ \begin{array}{l}
x+2y+3z=7 \\
-x-2y-z=-3 \\
x+2y+z=3
\end{array} \right.
\]
\[ (2)
\left\{ \begin{array}{l}
y-z+w=-4 \\
x+2y+z+w=-1 \\
2x+y+5z+6w=3 \\
x+y+2z+3w=1
\end{array} \right.
```

```
\end{array} \right.
\]
```

出力結果

次の連立一次方程式を解け。

$$(1) \begin{cases} x + 2y + 3z = 7 \\ -x - 2y - z = -3 \\ x + 2y + z = 3 \end{cases}$$
$$(2) \begin{cases} y - z + w = -4 \\ x + 2y + z + w = -1 \\ 2x + y + 5z + 6w = 3 \\ x + y + 2z + 3w = 1 \end{cases}$$

2 微分方程式

微分方程式とありますが要は分数で、`frac`を使います。

```
\ 長さ$2l$の糸(重さは無視できる)の中央部と先端に質量$m$の質点が付いてい \\
\ る2重振り子を考える。中央の質点の垂直からの振れ角$\theta_1$、先端の質点の垂 \\
\ 直からの振れ角$\theta_2$として、振れる範囲が小さいとすると運動方程式より \\
\[ \frac{d^2\theta_1}{dt^2} = -g(2\theta_1 - \theta_2) \]
\[ \frac{d^2\theta_2}{dt^2} = 2g(\theta_1 - \theta_2) \]
\ の関係式が得られる。ただし、$g$は重力の加速度である。 \\
\[ \frac{d^2\theta_1}{dt^2} = -g(2\theta_1 - \theta_2) \]
\[ \frac{d^2\theta_2}{dt^2} = 2g(\theta_1 - \theta_2) \]
\ の関係式が得られる。ただし、$g$は重力の加速度である。
```

出力結果

長さ $2l$ の糸 (重さは無視できる) の中央部と先端に質量 m の質点が付いている 2 重振り子を考える。中央の質点の垂直からの振れ角 θ_1 、先端の質点の垂直からの振れ角 θ_2 として、振れる範囲が小さいとすると運動方程式より

$$l \frac{d^2\theta_1}{dt^2} = -g(2\theta_1 - \theta_2)$$
$$l \frac{d^2\theta_2}{dt^2} = 2g(\theta_1 - \theta_2)$$

の関係式が得られる。ただし、 g は重力の加速度である。

`frac` の前半の中括弧は分子を、後半の中括弧は分母を表します。

3 積分

積分には `int` を使います。

```
\ 積分公式
\begin{displaymath}
\int_{-\infty}^{\infty} \exp(-x^2) dx = \sqrt{2} \sqrt{\pi}
\end{displaymath}
```

出力結果

積分公式

$$\int_{-\infty}^{\infty} \exp(-x^2) dx = \sqrt{2} \sqrt{\pi}$$

最初の中括弧は積分の下限を、次の中括弧は上限を表します。
infty は無限大を表します。

4 ベクトル

ベクトルには `vec` を使います。

```
\ ベクトルの内積と外積 \\
\begin{math}
\vec{a}\cdot\vec{b}=|\vec{a}||\vec{b}|\cos(\theta) \\
\vec{a}\times\vec{b}=|\vec{a}||\vec{b}|\sin(\theta)\vec{e} \\
\  ただし、\vec{e}は\vec{a}、\vec{b}の両方に垂直な単位ベクトル。 \\
\end{math}
```

出力結果

ベクトルの内積と外積

$$\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos(\theta)$$

$$\vec{a} \times \vec{b} = |\vec{a}| |\vec{b}| \sin(\theta) \vec{e}$$

ただし、 \vec{e} は \vec{a} 、 \vec{b} の両方に垂直な単位ベクトル。

`\cdot` は内積の記号、`\times` は外積の記号です。

5 表

表には `tabular` 環境を使います。

```
\begin{table}[h]
\ 4族までの元素周期表の作成例 \\
\begin{tabular}{|c|c|c|c|c|c|c|c|c|} \hline
族/周期 & 1 & 2 & 13 & 14 & 15 & 16 & 17 & 18 \\ \hline
1 & H & & & & & & & He \\ \hline
2 & Li & Be & B & C & N & O & F & Ne \\ \hline
3 & Na & Mg & Al & Si & P & S & Cl & Ar \\ \hline
4 & K & Ca & Ga & Ge & As & Se & Br & Kr \\ \hline
\end{tabular}
\end{table}
```

`table` 環境の隣の大括弧の中身は領域の確保を意味し、`h`、`t`、`b`、`p` の四種類があります。`h` は `table` の書かれた位置に、`t` はページの上端、`b` はページの下端に、`p` は独立したページに領域を確保します。`hline` で表の横幅いっぱいには水平の罫線をひきます。`tabular` の中身の中括弧の中身は `r`、`c`、`l` の三種類あってそれぞれ欄の中身を右寄せ、センタリング、左寄せします。さらに `c` と `c` の間の棒線は列間に縦罫線を引くことを意味します。また、`caption{}` で表題をつけることもできます。

出力結果

4族までの元素周期表の作成例

族/周期	1	2	13	14	15	16	17	18
1	H							He
2	Li	Be	B	C	N	O	F	Ne
3	Na	Mg	Al	Si	P	S	Cl	Ar
4	K	Ca	Ga	Ge	As	Se	Br	Kr

F 学科 2 年 3 学期 傾向と対策

omaeda-
2000 年 4 月

Abstract

2 年 F 科の試験の傾向と対策を考えます。なお、年度が改まり同じ授業でも教官や使用する教科書、試験方式が変わる可能性があるのでご注意ください。

1 関数論 (伊東)

試験の実施状況：期末試験のみ

レポート：授業中に出す演習問題を毎週

備考：前年度の試験問題&解答を試験期間近く教官が配布 その他：試験問題は全部で 8 題。そのうちの 5 問を解答する。具体的には
1. 代数方程式 (n 乗根など) 2. 複素数の級数 3. 複素積分 4. コーシーリーマンの関係式 5. ローラン級数展開 6. 実積分の計算 I(留数定理) 7. 実積分の計算 II(留数定理) 8. 実積分の計算 III(留数定理) が概観である。
いずれも教科書の章末問題と似ているものが出題されている。

2 応用数理解析第一 (橋本)

試験の実施状況:期末試験のみ

レポート:中間試験と同等の扱いをするレポート 1 回あり

備考：授業中に演習問題を配布。

その他：レポートの中心はベクトル解析、期末試験の中心はフーリエ級数。

いずれも教科書の章末問題、例題と似たものが出題された。

3 電気電子回路学第一 (高木)

試験の実施状況：小テスト 2、3 回と期末試験

レポート：なし

備考：講義で扱わなかった Y - δ 変換が出題された。

その他：試験は概ね講義で扱った事柄が出題されるが、上記のようなこともあるので注意。

4 基礎物理化学 (鈴木)

試験の実施状況：4 回

レポート：なし

備考：毎回出席を取り、3 回欠席をすると履修削除される。また、授業中 学生たちに質問し、挙手をして答えると点数がもらえる。出席カードに質問を書くと次の回の冒頭で答えてくれる。

その他：授業の 2 回目で最初の試験をした気がする。内容は「物質とエネルギー」の復習テストのようなものだった。試験は多いがその分範囲が狭く、出題は講義ノート中心であった。

セルオートマトンのおはなし

ottan

2000年4月

1 まえふり

そそるネタを思いつかなかったので、ToDo list にあることをそのまま原稿にしてみる。

2 セルオートマトンってなに？

セルオートマトン (cellular automaton 略して CA) っていうのは、ある種の理論であり、シミュレーションもしくは数値計算の手法なんだけど、感じとしては、ライフゲームを思い浮かべてもらうと一番分かりやすい。

ライフゲームを一般化して自然現象とかにも応用しようとして出来たモノってぐらいに考えればいい。ほんとは逆なんだけどね。あと、“オートマトン”とは名前についてるけど、状態遷移図かいたり 正規表現解析したりする有限オートマトンとは見た目あんま関係ない。理論的背景ではあるけど。

言葉の意味は、“次の状態がセル (cell) ¹単位で独立して自動的に生成される” ぐらいかな。

3 なりたち

元ネタを考えたのは、あのノイマンさん。

ENIAC の真空管の故障による機能停止²の多さに辟易して、人間の脳のように、部分的にどっかの細胞に寿命がきても別の細胞がその役割を代行している間に新しい細胞がその役割を担うという自己修復機能をつけたかったのがキッカケらしい。あと、人間の脳をモデル化するのに有限オートマトン理論を発展させてなんとかしちやろうってのもあったようだ。

で、ウラムとかいうおっちゃんから均一なセルを敷き詰めたセル空間というアイデアを得て、CA 理論を考えついたようだ。

しかし、ノイマンさんも忙しい人だったようで考察だけで終わってしまい実装するところまではいけず、50年代、60年代は一部の人間によって細々と研究されるにとどまってしまう。

そして、70年代のライフゲームの流行³を契機にやっとメインストリームにでてくる。

その後、80年代にはウルフラム⁴の自然現象への応用と、流体なんかへも応用できるような理論の拡張⁵を行ったりして、現在にいたるわけだ。

4 しくみ

基本原理は単純で、

- 同じ大きさの均一なセルにのしきつめられた空間を想定
- 各セルは k 種類の状態をとる
- 次の時間のセルの状態は現在の自身と近く (近傍) のセルの状態だけの局所的な規則できまる
- 出来上がるパターンはセルの初期条件と適用する規則によってきまる

の4つだけ。こんな単純なルールだけなのだけど、扱う対象によっては自然現象をあらわした微分方程式⁶をちゃんと解いたときと同等な結果が得られ、またそれぞれのセルは付近のセルのデータさえあれば計算できるので簡単に並列化できる。

¹この文章では、セルという言葉はコンピューターサイエンス的な単位元というか、自然現象を再現するモデルの最小単位というような意味で使っています。で、生物学的な cell を細胞と表記しています。

²きつと箱庭で災害が起こる確率に似たようなものでしょう (^^; 個々の発生率は低くても、システム全体やある程度以上のスパンで見て、ベキ乗してくると結構な確率になる。

³この頃のハッカー達のよいなぐさみものだったらしい。ネタ本 [?] にはライフゲームがハッカー文化の発展とコンピューターグラフィックスの発達をうながし、ゲームソフト開発の契機となったとまで書いてある...

⁴Mathematica の作者でもあるらしい

⁵格子ガスオートマトン法とか、こっちは各ステップの対象が個々のセルではなく格子の交差点 (オセロから囲碁へ、とえばわかってもらえるかな) で、その格子の上でのみ粒子を動かすというモノ

⁶微分方程式は連立1次方程式を解くこと、つまり行列の計算に帰結できるのだけれど、行列の計算では各々の要素の独立性は低いので、並列化するのは一苦勞のようだ

5 ライフゲーム

で、ライフゲームの場合は、

- 各セルは生と死の 2 種類の状態をとる
- 次のステップのセルの状態は隣接する 6 つのセル⁷の状態から以下のような規則によって決まる
 1. 隣接する 6 つのセルで生きてるセルの合計を sum とする
 2. $2 \leq sum \leq 3$ のとき、生きてるセルは生きつづける
 3. $sum \geq 4$ || $sum \leq 1$ のとき、セルは死ぬ
 4. $sum = 3$ のとき、死んでるセルは生き返る

というような感じになる。

たった、これだけのルールで、ライフゲームの複雑な動きが作り出される。

これは、周囲にほどよい数があるとき生存できて、過疎や過密のとき死滅し、最適な条件のとき生まれるという、動物やバクテリアの環境を単純化したモデルだったりする。

6 ししま

ライフゲームだとゲームとしての印象が強く自然現象の再現という部分ではちょっと薄いので、もうすこしそれっぽいモデルを取り上げてみよう。

で、動物の体表面の紋様(シマウマのしましまとか)の生成をシミュレーションしてみる。

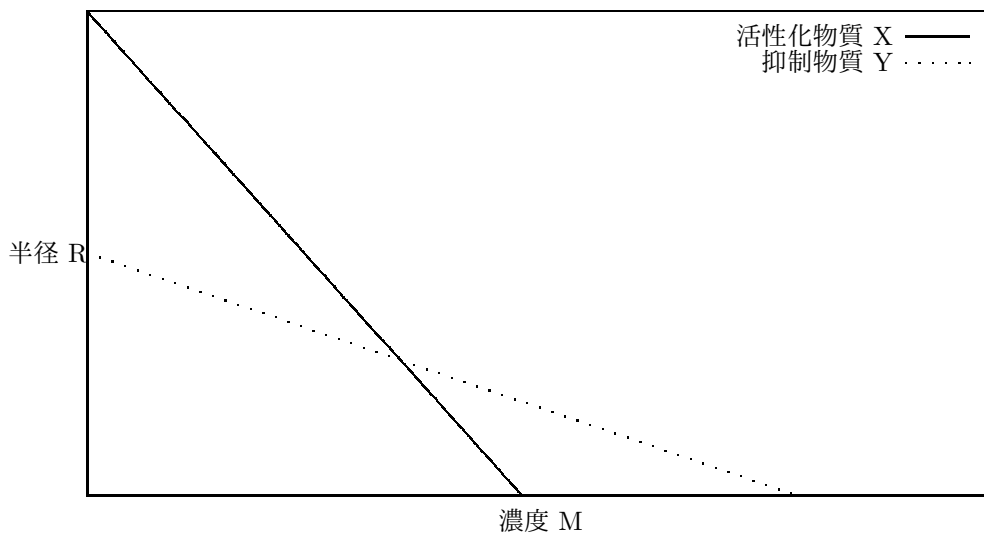


図 1: 微分方程式なグラフ

シマシマやブチってのは、メラニン色素の発色した細胞が集まったところだ。

で、発色した細胞は、狭い範囲で強い影響を与える周囲の細胞を活性化して発色を促す物質と、薄く広く影響を与える周囲の細胞を抑制して発色を抑える物質の両方をだす。発色してない細胞は周囲に何も影響を与えない。

ようするに、発色した細胞は近所同士で活性化を促しあい、ある程度遠いところには活性化を抑制しようとするので、発色してる部分はある程度固まり、シマシマやブチになるわけだ。

発色した細胞からの影響をグラフにしてみると図??のようになるが、このままだと、フツーに微分方程式を特のとかな湾内ので、図??のように離散化して処理する。

これを基本原理のパターンに沿ってまとめてみると、以下のような感じになる。

- 各細胞は活性化して発色した状態と活性化しておらず発色してない状態の 2 種類の状態をとる

⁷上下左右斜めの 6 つをムーア近傍という、ちなみに上下左右の 4 つはノイマン近傍という

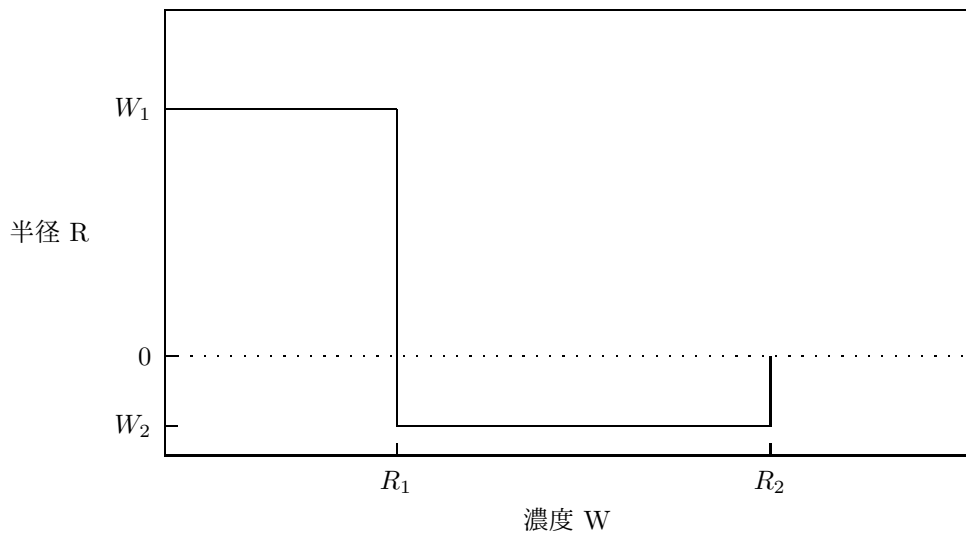


図 2: デジタルなグラフ

● 次のステップの細胞の状態は

1. 活性化させられる半径内に存在する発色した細胞の総数を x とする
2. 活性化させられる半径以上 抑制される半径以内 に存在する発色した細胞の総数を y とする
3. 活性化させる影響力を係数 W_1 とし、抑制する影響力を係数 W_2 とする
4. $W_1 * x + W_2 * y > 0$ なら活性化し発色する
(活性化した細胞は活性化したまま)
5. $W_1 * x + W_2 * y == 0$ ならそのまま
6. $W_1 * x + W_2 * y < 0$ なら抑制され発色していれば消える
(活性化していない細胞は活性化しないまま)

こいつを繰り返してみると、5 ステップ程度で平衡に達して安定なパターンが得られる。

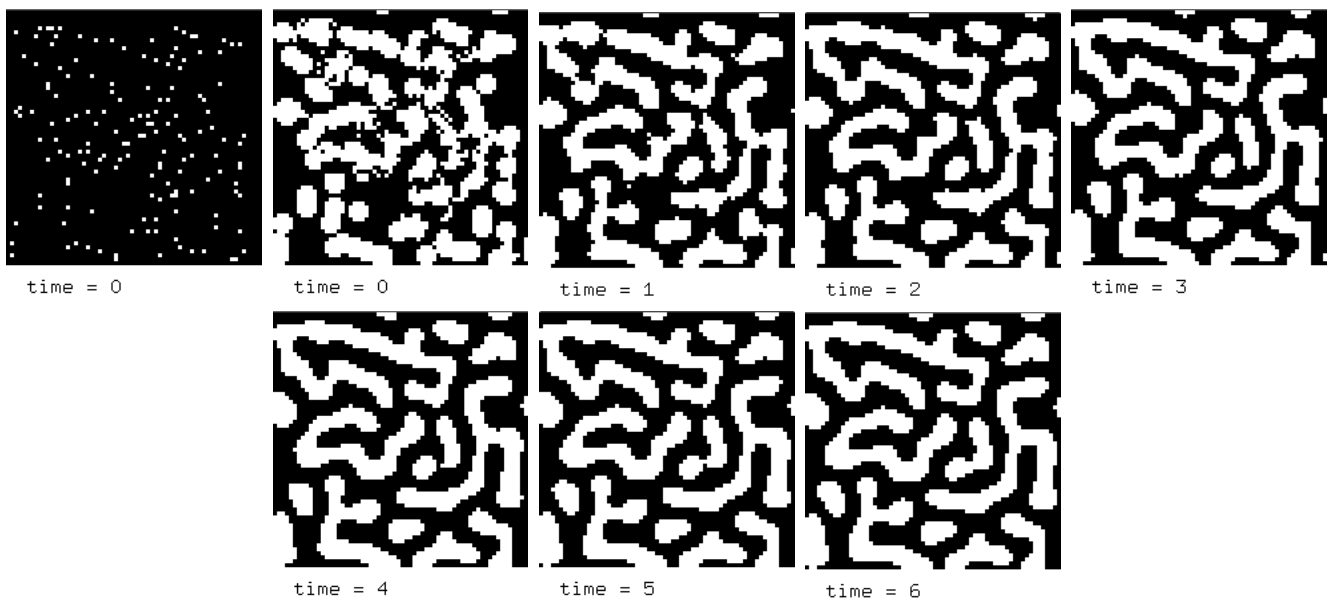


図 3: しましまの図

ちゃんと、それっぽく見えるよね。こんな単純な規則からこんな自然現象に近い複雑な結果がでてくるなんて、かっこいいとおもいませんか？

7 まとめ

ここでは図??のアナログで連続な微分方程式から図??のデジタルで離散的なオートマトン法モデルへの展開をさらっと流してしまっていますが、実際に自分でモデルを組んでシミュレーションするときはこの辺が味噌になってくると思います。

ここに書いてある内容はネタ本 [?] の縮小再生産なだけのような気もするので、ちゃんと知りたい人はそっちもあたってください。あと、実際の描画には CA 用統合環境って感じのソフトである [?] を使っています。

参考文献

- [1] 加藤恭義 光成友孝 築山洋 共著, セルオートマトン法, 森北出版
- [2] dana@rucs.faculty.cs.runet.edu,
The Cellular Automata Simulation System,
<http://www.cs.runet.edu/dana/ca/cellular.html>

AKIBA junk map 2000 Spring

近藤 毅 (kondoh)

2000年03月24日

まえがき

去年入学した kondoh です。個人的にハードウェアや秋葉散策に興味を持ち日々探求をしています。特に秋葉に行った場合は必ず勝利すべく 出物プライスに目を光らせています。その怪しいジャンク徘徊によって得た経験や知識を拙い文章ですがここに記します。以下の文は私の一方的な思い込みですので 必ずしも客観的な意見ではありません、参考程度に読んで頂けると幸いです。

8 目的別店回り

1 HDD

このごろ音を立てて価格が乱れている HDD ですが 漢ならやはり SCSI ですよね RAID ですよね。

その1 まっとうな道を行く HDD SCSI/IDE

CPU メモリと並んで価格変動が激しいのが HDD です。秋葉を一回りしているうちに値段が変わってしまうのではないかとというくらい変動しているようです。しかし最低価格は固定した店が閉めているように思います。

安い方から順に紹介します。

PCin 秋葉原

通販も受け付ける店。場所はアキバの一等地だが ビルの上。値段が妙に安く一時期はオコムと間違えられたが 本当は 三井物産系列のまともなお店。互換機関係の品揃えが良いが 値段は相場くらい。グラフィック系に特化した CGin も同ビルにある。この店は特にストレージ系の価格が安い。たいていはアキバ最安値をマークしていると思う。CGin の価格についてはコメントを差し控させていただきます。

店員の質はまあまあなのだが 在庫が少々不安でもある。売れるからかも知れないが 必ずしも在庫しているわけではないらしい。また 通販と店頭販売価格に開きがあり しかも 通販の方が安かったりとかするお店。

水曜定休のため 平日のアキバまわりには使いにくいかもしれない。逆にアキバに遠い人にとっては 通販が利用でき市価も安いのでお勧め。

ZOA 秋葉原本店

駅からの距離が少々有るが コンスタントに安い店。HDD はたいてい在庫しており 好感度高し。しかも ポイントカードによる3%がしかし 保証期間が短い。HDD の場合初期不良 10 日だったりする。HDD を酷使し 半年も持たないような人たちは 保証がしっかりしたお店で選んだ方がいいかもしれない。もっとも そんな人種が壊れた HDD を わざわざ店に原状のまま運ぶ また 保証証を 1 年も保管しておくかは はなはだ疑問ではあるが。

最近店内を改装したりしているが 総合互換機ショップだが いまだに手薄な商品があったりもする。店員は豊富にいる。教育もとても行き届いているが 知識経験的には 疑問符。遠くてもかならず 足を運びたい。

USER'S SIDE

まっとうな互換機屋のイメージがある (あった)。HDD の値段はそこそこなので紹介しておきます。最近どんどん変な方向に走っているようなきもするが扱っている商品移動も偏りがあるように思うが 店からの提案的な商品もありおもしろい(かった)。一人非常に知識のある店員が居るが 平日に来店すると 買わせたいオーラが漂っている商売熱心なお店。

SCSI HDD は ガラスショーケースの中 EIDE HDD は店内の棚にストックされている。

ここでは HDD がらみの商品が豊富で ヒートシンク機能付き 5 インチベイマウンタや HDD 消音機能付きマウンタなどもそろえている。私もしっかり見たわけではないが 消音マウンタはちょっと廃熱の面で難があるように思う。今売られている商品は二代目目かなり考慮されているようだが...

そういえば この店は ポロっと IDE-SCSI 変換コネクタを入荷し 問い合わせが殺到した。AEC-77?? という名前のこの変換アダプタ センチュリーのものよりもパフォーマンスに優れており高性能 IDE HDD との組み合わせでシーケンシャルを測るとかなりのもの。今となってはかなりの多くの店で見かけることができる。最安値で 7k 円くらい。

最近バラクーダを5台 RAID した 狂マシンの展示があったりもした... また RAID カードを注文扱いが売っているようだ全然安くないけど。

このようなショップが新品でまっとうは HDD を買うのならお勧めです。もしかしたら 他の店の方が安かったりするかも知れませんがやはり 足で稼げ ですね。

その2 サーバ用 SCSI ケースを入手しなくては行けない場合

つい出来心で 鯖筐体を買ってしまい どうしても専用ホットスワップ SCSI ケースを入手したいと思う場合。そういう思いを人間なら一度はすると思う。そんなとき メーカーの純正パーツの値段を見ると愕然とするし そういうパーツは一般の人には売ってくれない。それでも手に入れなくてはならない場合 アキバを歩くしかないわけですね。それなりに見かけることができる。特に Compaq などは専門の店があるから安心である。

闇雲にアキバを徘徊してもなかなか見つからないと思うので 私の経験からちょっとしたアドバイスをすると、

Compaq 超級電腦に行けばだいたいあるのではないだろうか。

HP 筐体は非常によく見かけるが SCSI ケースはなかなか見かけない。いつそのこともう一台筐体を買ってしまう⁸というのはどうだろうか。PC-net で安く見かけるし マップで確実に売っている。

DEC 最近やっと見かけるようになった。M&J や PC-net で見かけた。PC-net でもう 1 台筐体ごと買うのもありだと思ふ。また DEC の SCSI タワーがマックラフトで安価に売られていた。この SCSI タワーかなりかっこよく危うく買ってしまいそうになりました⁹。

NEC こいつも PC-Net で筐体ごと買ってしまう方向で。

Fujitsu 上に同じ

こういった特殊パーツは 何を扱っているかわかっている店でもそれほどぼった値段は付かないことが多い。中古で見掛けた時は DEC の 2G バラクーダ付きで 5k 円だった。まだ 鯖マシンを持っていなくても とりあえず将来のための投資をするのも なかなか漢じゃないかと思ひます。ホットスワップケースを買ったからそれに合うマシンを買うというのも 良いものですよ。

その3 PC-98 で使いたいんだ!

PC-98 で HDD を増設するのは一般的に SCSI だと言われています。そもそも 98 では内蔵 HDDIF は E-IDE 擬きで 認識出来る最大容量は 4.3/8.4GB の物が多い。この制限がかなりタチが悪くそれ以上の容量の HDD をごと認識しないのです。つまり使えない。また 適合する容量の HDD 用意したとしても 98 では最後まで PIO mode2/3¹⁰ ととてつもなく遅い IF なのです。それを補強するために SCSI が 98 での HDD 増設の基本とされています。また MMX 化改造を行った多くのマシン (Triton2 ではないチップセットでかつメルコや IO などのゲタを使わない場合) も内蔵 IDE-IF が使用不能となります。また 高速 EIDE ドライブを使いたいという要求に答える解として EIDE-IF ボードなども存在するが 貴重な PCI スロットにはもっぱら汎用性の高い SCSIIF が挿されてきました。

ということで SCSI なら 今でも手に入るが どうしても標準 IF に増設したい場合にはどうしたらよいか。

- 本体が認識できる最大容量を知る (メーカーは公表していない)
- マウント金具等の問題を知る
- 実際に物を探す

このような手順を踏む必要があります。まず本体の認識できる容量のは経験によって知るしか無く 結局他のユーザの情報を頼ることになります。Internet 上には 98 コミュニティーが存在し そこには非常にたくさんの情報が蓄積されているので FAQ として知ることが出来ます。有名な壁は 500MB, 4.2GB, 8.4GB。500MB の壁はちょうど An が壁にあたり (BIOS を Update することで対応)、8.4GB まで認識できるのは 3 桁 ValueStar の一部とのことです。手持ちで大容量ドライブがあるのならまずはテストしてみるのが良いと思ひます。

次にマウント金具。乗せ換えなら問題がないが 増設となると しかるべき金具が必要になります。Logitec や IO-DATA などが売り出しているのでカタログを参照。マウント金具を BitINN や特定のショップを経由して補修部品として入手する手段も存在しています

⁸新たに買った筐体に挿すホットスワップケースはどうしましょうねえ

⁹いやホントかっこよかったですよ細長いタワー型で前後両面から開く扉があり HDD はすべてホットスワップケースによる装着、電源までも同形状のスロットインタイプついでに IF までスロットイン!

¹⁰涙が出るくらい遅いです。どんなドライブを使ってもシーケンシャルで 2MB/s 超えないような気がするし CPU 占有率も高い

が、最低条件として型番を知ることに加え BitINN は最近補修部品を個人相手に売ってくれません。困難な道であります。結局アキバを回るのは最終段階となりますが今となつては新品での小容量 HDD を探すのは困難で結局中古を当たることになります。新品でも希に売っていますので危険な中古に手を出す前にこまめに回りたい物です。しかしまともな中古の HDD を単体で置いてある店が意外と少なく 私の知る限りでは超級電脳や PC-Net などですがコンスタントに置いてあるわけではありません、こればかりは当たり外れもあるので 十分足で稼がなくてはならないです。まずは 十分な情報を仕入れてそれから 秋葉を回しましょう。幸運を祈ります。

HDD を選ぶ時のためにアドバイス

買ってはいけないドライブが存在するため注意買っても良いドライブは IBM Quantum Seagate(MedalistPRO 以上)。買っては行けない物は 上記以外。ノート用やサーバ用などは別ですが 民生用の 3.5 インチ HDD は 非常に低品質な物が売られているので注意する必要があります。

ちなみ それぞれのメーカーの特性は以下です

IBM

私的に一押しメーカー。音が静かなのが取り柄 値段も手頃。しかし リリースがちよつと遅めで 性能も並ランク。IDE 製品はそれなりにトップの性能を誇るが SCSI は技術導入が少し遅め。具体的には面密度が低い。守りのメーカーだと思う。HDD の名前がアルファベット 4 文字で まだ覚えやすい。10000 回転のドライブを作ったりキャッシュが 4MB だったり このところ SCSI 部門でもかなり健闘していると思うので 私は同社の HDD ばかり買っています。

Quantum

昔よりも良くなってきた。かつてのファイアボールで泣いた人も多いかも。Fireball シリーズも CR あたりから特に攻めのドライブで 面密度も高く シーケンシャルに強い。が 反面 Seagate ほどでないが 高音が激しい。また HDD 内部のヘッド(サーボ)と制御基板を結ぶフィルムがむき出しになっているため 私はどうしても信頼できない。一度このフィルムを半田付けしたことがある。Viking4.5GB SCA 2 回立て続けに成功。1mm の間に 4 本ジャンパ線を半田付けする必要があるので あまりお勧めできませんが...

このメーカー Fireball で売っているが 実は SCSI 部門にも 昔から力を入れてきたようです。ATLAS とか Viking などがハイエンド向けとして売られている。かつて Fireball にも SCSI モデルが存在したが 性能がクソクソで どうしようもなく¹¹ST¹²を最後に SCSI モデルが無くなったと記憶しています。その後 ATLAS シリーズと Viking シリーズは合併したらしい。Viking のハードウェア部分と ATLAS のソフト部分が一緒になったらしい。そして 今の ATLAS10k ATLAS VI に繋がっているらしい。初代 ATLAS や Viking を使ったことがあるが 発熱騒音 なかなかすごいです。

そう言えば BigFOOT という HDD を作っていたのもこのメーカー。5 インチの E-IDE HDD、当時としては多少コストパフォーマンスに優れていたが時代に逆行していたようにも見え あまり売れなかった。シークが遅く国内ではもう見られない。海外ではそれなりに売れたらしいです。

Seagate

Cheetah は昔からのあこがれのドライブでブルジョアなベンチマーカは持っていました。最近になってメダリストシリーズで 低価格部門でも頑張っている会社ですが そもそもハイエンド指向の HDD を作っていたメーカー。Cheetah、バラクーダは速いが熱い。また リードとライトのスピードが違うのも 同社の特徴¹³。このメーカー MedalistPRO で低価格部門で名を確立したのだが SCSI モデルの最初のロットはかなりの発熱だったらしい。また 同社の最低価格ラインの HDD は買っては行けないらしい¹⁴。

このメーカーの HDD は中古市場で春先によく見かけるらしい。鯖などから 外された物が市場に流れるようだが 値段につられて買ってしまふと 熱と騒音に悩まされることになる。また SCA-IF だったら変換が必要だし FastWide というのも良くある。中古はきちんと物を見て買った方がいい。型番が数字の羅列でわかりにくいことでもまた有名だったりする。

最近発売されたバラクーダ ATA¹⁵は SCSI 変換アダプタや UIDE-98 との相性が劣悪でパフォーマンスが出ないらしい、買うのなら互換機のみで使うつもりでどうぞ。

WDC

とんでもない不良率を 故障率と 私的に思っている。PC-98 では同社のドライブを非常に多く採用しているのだが私の手元で次々と崩壊していくことから全く信頼できないと思っている。つい最近もリコール騒ぎがあり 信頼もがた落ちと言ったところだろうか。

¹¹同じモデルでインターフェイスが変わるだけで性能が 2/3 未満になる

¹²不幸なことに Alpha に入っている

¹³ライトの方が遅い

¹⁴HDD が黒い振動防止材で覆われた HDD は品質が低いという話を聞きました

¹⁵シーケンシャル RW で 30 数 MB/s 出るし安い

利益に出ない E-IDE HDD 部門を辞めるのではないかと思うのだが いまだに製品をリリースして... なかなかしぶといですねえ。最近また SCSI ドライブをリリースしているようですし。

JTS

最近聞かなくなりましたね、インド製 HDD。品質は低いらしい もう売っていないから 被害に会う人も居ないだろうけど。

コナー

かつてはコナー方式という物が存在したらしい。PC-98 にも同社の HDD が採用されていることもある。個人的にあまり信頼していない (良く壊れた物を目にする) が あたりを引くとそれなりに長生きするようです。昔は 『IDE の増設は難しい WDC 方式と コナー方式があつてね』 と 語られたのだが 今は昔。同社は Seagate に買収されました。

Fujitsu

PC-98 時代メルコ製の外付け HDD の中には Fujitsu 製の SCSI ドライブが内蔵されていました。そんなに不良率も高くなかったと思うが 自分で使ったことはないので今ひとつ判りません。そんなに静かではなかったと記憶しています。今でも 鯖用と思われる同社の SCSI HDD をたまに見かけるが 今となっては価格的なメリットも少なく 発熱 騒音的に 不幸な点ばかり見られる。

HITACHI

12000rpm の HDD を売り出したように記憶しているが 3.5 インチドライブは一般ルートにあまり見られないので その全容は判らない。2.5 インチは結構見かけます。

MAXTOR

品質が低いらしい。DiamondMAX7200rpm モデルはやたら熱かったらしい。かつては高級 HDD メーカーだったと聞いたが 今は低価格高性能を売りにして EIDE ドライブを売っている。その筐体はアルミダイキャストで異常に安っぽい。カタログ上ではそれほど軽いわけではないのだが 手に取ってみると異常に軽く感じる。スピンドルモータへの電源供給ラインがエナメル線と言うのはどういう神経をしているのだろうか。とにかく安っぽいメーカーだった。

2 ビデオカード

こここのところビデオカードも 異常な性能向上をしていますがそのような まったうなビデオカードではなく 一世代前の名機を探す旅に出るといのはいかがな物でしょうか。

当然それらを見つけるのは中古売場。運任せになります。とりあえず 自分も運を最大限に引き出せる道を 示します。

ここでの目的はやはり matrox #9 S3(昔の) の製品。ついでに OpenGL カード。特に Millennium シリーズはこここのところ潤沢に出回っているのだから安く入手するかが手腕になります。もしかしたら この機を逃すと非常に入手しづらくなるのではないかと思います。ということで まずは それぞれのカードの市場価格を示します。

Matrox

Millennium 2MB 4000

もっとも手に入れやすいのではないだろうか。がしかし VRAM が少なすぎです。

Millennium 4MB 6800

なかなか見つからない。

Millennium 8MB 11000

滅多に目にしない。見たら買うベシ¹⁶。

Millennium 増設メモリ 2MB 2000

将来に備えて備蓄すべし。いまだに在庫店有り。

¹⁶ボード上に 8MB 載っているモデルは存在せず 4+4 or 2+6 という形で 8MB になります。個人的に後者のほうがかっこいいと思います

MillenniumII 4MB 6800

最近よく見かける。店によって PCI/AGP 値段が違う。普通は PCI の方が高い。

MillenniumII 8MB 8800

あまり見かけない。安く見つけたら買うベシ¹⁷。

MillenniumII 増設メモリ 12MB 10000

プラットホームが在庫している。価格が崩れたら買うベシ。

もっとも増設メモリの入手がしやすい MATROX でも 早々手に入れることは出来ないです。そのため 出来る限り 自分の必要とする容量の付いたカードを買うべきですね。

#9¹⁸

Motion771(Vision968) 4MB 1980

なかなか悪くないカードだと思う。IBM RAMDAC だしなんとプリント基板上の配線が一般的な 45 度の 2 回曲がりではなくカーブになっていると言う珍しい板。

Imagin128S2 4MB 5000

超級が在庫していたが 最近無くなった。4MB では 64bit 動作になってしまうのが痛い。

RevolutionIV¹⁹ 32MB 9800?

32MB でも そんなに高くなかった覚え有り。

RevolutionIV 16MB 8800?

増設メモリはなかなか手に入らない。と言うか、見たことないです。

RevolutionIV WRAM 8MB 5800

見逃してしまったが 買うべきだった。

S3

Vison968 搭載カード ~1980

いろんなメーカーから出ていた。みな PCI だし VRAM4MB 以上なら買うべきかと。IBM RAMDAC が付いているのなら即買いかも。テストマシンとかサーバマシンとか用途はあると思うし。なお廃人にとっては ViRGE VX 8MB も人気があるようです。

Savage4PRO+ 32MB 8800

あまりお勧めできないが PCI 版の流通量も多少有る。

だいたい以上のような価格になっています。特に VRAM 容量は重要になるので きちんとチェックしてみてください。

なお Millennium 初代は RAMDAC の種類によって 3 種類有る。OEM 版の 175MHz, 通常パッケージの 220MHz, PowerDok の 250MHz. 175 をよく見かけるが 出来る限り 220 を買うように、また 250 を見たら 即買うベシ。MillenniumII にも RAMDAC220 と 250 が有る。出来れば 250 の方がいいが VRAM の容量と相談すべし。なお VRAM は 8MB 以上有っても 2D では意味がない まさか 3D なんて MillenniumII でやらないだろうし。

巡回路

今まで中古ビデオカードを扱っていたところ以外で 最近 Millennium シリーズの放出が相次いでいる。それらは突発的にやってくるので全てをカバーできない。このあたりはアキバ PC ホットラインに出てくるので要チェック。一般的な互換機ショップが放出することが多いようみたいです。

¹⁷VRAM が 8MB 以上になっても現実的にうれしいことはないのもっとも実用的です。

¹⁸倒産してから一気に市場から中古も消えました。ほしい人はがんばって探してください。

ソフマップトレジャー館 6

ここは 1F 奥にプラスチック箱に入ったカード類をたくさん扱っている。値段も判ってないことが多く チェックすべし。

また PC-UNIX を低価格中古マシンやサーバマシンの 2F にも あまりやすすくないがコンスタントに Millenium や FireGL1000²⁰などが売られている。

ソフマップトレジャー館 5号店 USED PC PRO SHOP

改装されて 3F にあったビデオカードは 1F に移動した。移動によって魅力的な商品が減ってしまったような気もする。どちらかという 箱に入った中古を見るのが多く 状態は良いのかも。箱が欲しい人は要チェック。

MAXUS

ここにはちよくちよく MilleniumII が売られる。6800 円という破格で売られていることも多く 必ずチェックしたい。なお同店には レインボーランナー (Mille 初代用 MPEG1 キャプチャ)25k 円なども売られている。更に Adaptec の AAA が異常に安く売られていた過去もある。金さえ有ればとんでもない環境を構築することが出来る。また一時期 MillenniumG200 の増設メモリを扱ったりコンスタントに Millennium(/II/G200/G400) が入荷したり それなりに高級で珍しい物が多く入ってくる店。秋葉でもなかなか見かけない時期に RageFURY MAXX を売っていたり。

プラットホーム

もう入荷しないと何度も言っている割には ガンガン Millennium シリーズを入荷する したたかなお店。ここは唯一 MilleniumII 増設メモリを扱っていた。以前は 4/8/12 とそろっていたのだが今では 12MB が 10000 円で売られているのみ。これの値段が崩れたら買うべきだと思う。これさえ有れば最強の MilleniumII を手に入れることができれば輝きを失った MillenniumII 4MB が宝石に変わります。また 同店では Millenium の 8MB(2+6) が 11800 円で売られていた。これは買うべきだったのだが逃してしまった... 6MB 増設は遠くなってしまった。今でも MillenniumIIAGP 版や Millennium4MB(2+2) を売っている。G200PCI 15k 円で売るのはよして欲しい、買いたいから。

超級電脳

鯖からの外し品として レパートリーの広い商品が売られている。FireGL3000/4000 や OXIGEN なども売られている。ここで Imagin128S2 4MB が 5000 円程度で売られていたが 買わなかった²¹... 増設 VRAM が売られていることもあるし 妖しいブツがとにかく多い店。お店のレイアウト変更や昔は営業していた 3F 部分をたたんだり、営業成績は芳しくないのかも知れない。最近 IBM 製品 (Taken や TP) の取扱が増えてきたように思う。

小学校付近の妖しい店 (紙風船の向かい)

ちょっと説明しにくい場所に立地している店だが ここには Mille2MB 増設が 2000 円で売られている。将来のためにも これはストックしておくべき。かなりの長い間売っているのだからそれなりの数の在庫を持っているようである。とりあえず買おう。その他明らかに DEC Alpha 用のヒートシンク (ファンがはずされています) が置いてあったりして怪しさ爆発。

じゃんばら

最近もつとも Millenium との遭遇率が高い店。値段も結構安く コンスタントに物があるのでお勧め。TwinTurbo や FireGL1000Pro などたくさんのカードを買った。以前一度 MilleII 増設 4MB メモリが売られていた。とにかく毎週見ておく必要があると思う、しかも早い時間に。最近 PenIIODP が複数個出て騒然となった。MGA-II も売られていたこともあるし。

ビデオカードとの出会いは一期一会 見たら買うがジャンク屋の勤めですね。二桁の Millennium をストックしましょう。

3 ネットワークカード

こここのところ異常な値下がり記録している NIC だが やはり名の通ったものを買おうとすると それなりの値段になります。Intel とか パッケージで買うと 1 万円近いし... ということで 中古やバルクでも良いから 良い物を安く買う旅へ...

ターゲットはやはり Intel DEC 3COM 安い方の市場価格だと それぞれ 4000 3000 3000 あたり。これを切ればまあ安いと思います。逆にこれを切る値段は普通じゃない売り方になります。例えば中古だったり 廃棄マシンからの取り外し品だったり。こういうの

²⁰FireGL1000PRO ではないです

²¹64bit 駆動になっちゃうんじや魅力半減ですよ

はコンスタントに出ているわけではないのでなかなか手に入れることは出来ません。こまめにアキバを回る必要があります。わたしの買った最安値は Intel が 500 円 (少量) 1500 円 (多量に在庫有り) 位でしょうか。コンスタントに安い店 出物のある店などを下に記します。なお 貴重になりつつある C バス NIC を扱っている店も併記します。

ファーストポイントラジ館 1F

C バス NIC を道沿いで コンスタントに 2500 円で売られている、いつも売っているので安心して買える。必ず置いてあるというのは非常に貴重なことだと思う。なお 7F のファーストポイントには H-98 用の NIC が置いてあるので 欲しい人は急行。

湘南通商

ここでは特に NIC に限定しないで 様々なカードがなかなかの安価で売られている。コンパックの OEM 品なども有るようで 見ても楽しい。以前ここでは nest 用 AHA-2940UW を 16800 円で購入したりした。IntelNIC は 4000 円位だろうか。

千石の隣のお店

なにげに NIC が置いてあったりして それが DEC チップだったり Intel チップだったりするお店。よく見ていないと見逃してしまいそうだったりもする。

OA リサイクル 1 号店

contec や Xircom の PC カード NIC が 1500 円でかなりの長期間売っていた。今はなくなってしまったがまた復活するのではないかと 思って楽しみにしている。いや、実際復活して現状でも売っている。10BASE とはいえ 1500 円は魅力的なので 予備の予備も考えてしっかり買うベシ。私は 3 枚くらいしか買いませんでしたが。また 98NOTE 用 拡張バス用 NICBox も売られていたりする。欲しい人は欲しいのだろうが 私はさすがに使うアテはない。<500 円 C バス NIC は昔は安かったのだが最近同店では 2500 円位で売られている。

ソフマップ トレジャー館 6 号店

同店の前のジャンク箱の中に C バス NIC2000 円位 H-98 用 NIC が売られている。店員が適宜追加するので 注意してチェックする必要がある。

若松 LAN2 プラザ

同店のジャンク箱の中には MCA の IntelNIC が売られている。とりあえず IntelNIC なので IntelNIC が欲しい人は買うベシ。買ってから対応機種を探しましょう。

超級電腦

コンスタントに 3COM DEC Intel の NIC が置いてあるように思う。値段は安めの相場位。希に IntelNIC が安かったりすると思う。また ジャンク扱いで売っている場合もあるのでチェックする必要がある。この店は 鯖からの外し品なんかも扱っているらしい。以前 IntelNIC を 500 円で買った覚え有り。その他値段の付け方が謎なヅツもあるのでしっかり見るベシ。あと Quad な IntelNIC が 20k 位で売っています、取りあえず買いたいですね。

メカ風船

同店では以前から C バス NIC が適価で売られていたがしばらくの間 IntelNIC が 1500 円で売られていた。かなり買い込んだ人もいるようです。こういうチャンスもあるので 常に金を下ろせる状態にして 買い占め体制を敷くかないとダメですね、1500 円の IntelNIC なら転売しても赤字にはならないかも、どうせ使うんだし 有り金はたいて買しましょう。shimiz98 さんに 4 枚買わせたり、背中押しも重要。

ストレートの前の露天

最近になって沸いて出た露天だが C バス NIC が 500 円となかなか挑戦的。中身も 108 ボードだったりして貴重。手元に二桁のストックを作るくらい買うベシ。また DEC チップの PCI も 2500 円くらいで手に入る。

PCIの100Base NICなら腐ることもないし とりあえず十分なストックが欲しいですね。ふとマシンを買い込んでも対応できるよ。もちろん家にはラックマウントハブ。また PC-98用は希少になる可能性もありますので 十分十分に収集したいものです。

9 厳選ジャンク屋

今の秋葉 回るとしたらこの店は外せないというジャンク屋を列挙します。英文字は Impress の秋葉 MAP を見てください。店の詳細な位置はそちらでどうぞ。

B02

1 日米商事 2号店

弁当屋の地下のお店、営業時間が曜日によってまちまち。扱っている商品は レーザー光源 (ポインタなどではない) や ガスマスクから NEWS や SPARC Alpha NEC PC-98 NeXT 中古ソフトなど様々。常にストックしているわけではなく 行くタイミングによって物が違う。常軌を逸した値段で売っていることで有名。また 発送は基本的に出来ないと思った方がよく 車で乗り付ける必要があるかも。店番のおばちゃんは それほどやる気がない、営業時間だが曜日によってまちまちで ひどい日には3時からという日もあったように思う。入り口の戸に張り紙がしてあるので見ると判る。以前ここで どうしても買いたいブツを 半歩先に店に入った他人に買われてしまった非常に苦い経験があるため 出来る限り開店直後に行きたいと思っているのは秘密。あのときは悔しかった... 相手がブツを見て 買うまで約20分見ていたのだが... 結局買われてしまった。Xa13/w 11k 円... 商品には CPU クロックとメモリ HDD 容量などが書かれている。大容量 HDD が内蔵されていても 筐体がでかかったり ブツが判らなかつたりするとかなり安い値段で売られている。UPS もそれなりに安いので車で行きたいですね。ただやっぱり以前よりは魅力が減ってしまったようです。

2 日米商事 1号店

2号店とは また 毛色が違う。基本的にジャンク ハードウェア 電子より 電気と言った感じの店。外し品と思われる 液晶パネルやトランス 妖しいブツが売られている。基本的に値段は高めかと思われる。PC-98の486モデルが多数売られているがそれほど安くない。パチンコ台外し品の液晶モニター単品また NTSC 信号を受け付けるような改造を施した物も売られている。ノートの壊れたマザーなど 崩壊したパーツを売っている。巨大なコンデンサや抵抗などのパーツや コネクタなども売られている。

3 JC-WORLD

ここは2階でノートのアップグレードをしてくれるらしい。しかし わたしは未だ足を踏み入れたことはない... が一階は 道に所狭しと並べられた 古めのパーツや オリジナル超低価格マシンなどが存在する。実はこの店 ゲタで有名。PowerLEAP 社と太いパイプで繋がっているらしく 同社のゲタを良く取り扱う。噂の PCIにつける セレロンが乗ったアップグレードボードやソケ8用セレロンを使ったアップグレード CPU を取り扱う。下駄 極みシリーズの取扱もあり。その他 互換機パーツをメインに CPU HDDなどを扱う普通の互換機パーツの値段は高め。路上販売のブツの値段は 結構安いこともある。中古も ぼろぼろと扱っている。最近店舗が移動したらしい。

ラジオ会館

昔は良かった... のかな? 最近は どんどん変になっているようです。変な本屋とか プラモ屋とか... 残ったのは 高級オーディオ系 ショップと 若松通商 ファーストポイントくらいでしょうか。ここのオーディオは私には手が出ないので...

4 若松通商

ラジオ会館5階。基本は互換機ショップだがケースやベアボーンなどにも力を入れている。知識レベルの高い店員が居る。また CPUにも力を入れていて PentiumPRO や WinChip なども扱ってくれていた 優良店かと。VRM もたまに売っているし。価格は 妥当なラインかと。ここでは 細かいパーツと WinChip 位しか 買ってないが。ちょっと前までは PenPRO の VRM も売っていた。キーボードや筐体なども力を入れているが少々高いと思う。

5 ファーストポイント 6F

ラジオ会館 6階。NEC に最大の力を入れた店。NEC マシンが取扱商品のほとんど全て。しかし 中身をよく見ると 展示品処分が多く 各店で 展示された NEC 機はファーストポイントに集まるのだろうかと思うくらい多い。そのため まともな NX(そんな物存在しない)の新古が欲しくなったら 行くと良いかも。PC-98 系列も 根強く取り扱っている。が 値段が高め。マシンをばらしてパーツ売りをしているので それは一部安いかも。NEC 純正や サードパーティーのパッケージ品なども多数取りそろえている。最近はかなり NX のシェアが高くなってきた。C バスのライザーカードがコンスタントに扱われている。また Ra もタイミングによっては出てきたり Rs20 があつたり物の流れは思った以上に大きいので頻度にチェックしたい、もっと低い階にあつたら楽に回れるのだが。

6 ファーストポイント 1F

6階と同様の店。扱う商品もほとんど同じだが 展示が違うから ちょっと違う印象を受けるかも。6階とは あまり商品の移動をしていないようなので 6階にあつても 1階に無い商品があるから 両方回る必要がある。最近このタワー型 98 は みんな St15 Xt13 で面白みがない。また こちらでも マシンをばらしてパーツを売っている。NEC 純正パーツもたまに売られている。C バス NIC が 2k 円でコンスタントに有る。C バスのライザーカードもコンスタントに扱われている。本体は最近並な物ばかりになってしまったが一時期はなかなかすごい物が置いてあつたりもしたのでやはりチェック外せないと思う。HK6-MD350-N3 が 6800 円だったので買った。基本的にメーカーの新品は安くはないはずなのだが希に投げ売りをするのでしっかりと情報を入れたい。同時に出てくる数が多いので情報を聞いてからでも間に合うケースもある。ここで NetFine を買った。24k 円 (P55C-166 消費電力 30W 最小ケース)

NEC のショールーム bit-inn と ファーストポイントとは 何かしら繋がりがあつたらしいという話を聞かないでもないのだが詳細は不明。

B03

7 ソフマップ 8号店 中古パソコン専門館

広大なフロア面積を持ち まともなマシンの中古から Mac の中古 パーツの中古さらに PC-98 も扱っている。基本的に 物の価値が判っていない店なので 急に安いものが出ていたり相場から著しく高かつたりする。ここの売値は 買い取り価格次第ということらしい。また HDD メモリの中古も売つたりする。出物を求めて こまめにチェックする必要がある。値札の付け間違いをしていることも有つた... メモリはある程度規格が決められていて値段が決まっているのだが、ソレ以外のメモリはそれぞれ値段が付いているのでしっかりと見る必要がある。中古 HDD はお勧めできないと思う。また コンパック製の 100base リピータハブが破格で売られていたり EPSON 製過去の高級スキャナが安かつたりもする。カード売場は在庫が豊富なためしっかりと見る。ショーケースの中も希に掘り出し物有り。また新品も扱つておりそれなりに安く HDD などを手に入れることが出来るかも知れない。新品と中古を混ぜて扱うのはマップの相変わらず。

B04

8 ロビンマイコンショップ

かなり昔からの店の様だが 立地的に弱いので 最近はあまり行っていない。休日の路上ジャンクは確かに魅力的。ケースなど互換機パーツが割安。店自体はそれほど広くはないが 扱う点数は多い。SCSI ケースや NLX ケースに凝つてみたくなつたら 行つてみると良いかも。また 休日の路上カートには それなりに魅力的な物がある。不完全な状態であることが多いが...

9 マグラボ

一時期はアキバの CDR の隠れた最安値をマークしていたようだ。本来は小売りはしないはずのお店らしいのだが最近結構積極的に小売りをしているように思う。最近の値段はそれほど魅力的では無いかも知れない。

10 ジャンパラ秋葉原 4号店

新規に開店したジャンパラ。店内にもある程度商品はあるのだが 開店当初は 紙に商品名と価格が書かれていてその中から選ぶようになっていた。店内にはケースや妖しいマザーなどが売られている。ショーケースの中には CPU やメモリ カードなども売られている。ショーケースの中の商品はすぐに見せてくれるので メモリにチップを確認したかつたり よく見てみたいカードなどがあつたら遠慮しないで声をかけてみるべし。じゃんばらはやはりガラスショーケース内が重要だと思う。

C01

11 オリックスレンテック

リース屋オリックスのリース落ち販売店。結構状態の良い98が売られている。しかし最近ちょっと高めかと思われる。また店員に声をかけるとメモリを売ってくれる。メモリはSIMMなどもあるため重宝する。さすがにリース落ちで付属品等はしっかりしている。黄色いオリックスの箱を見たことがあるはず<付属品箱です。オリックスレンテックの月刊のパンフのような物 今月のリース落ち商品が書かれた本があるのだがこれを丹念に見ていると良いこともあるらしい。しかし、最近では価格のアドバンテージがないですね。

12 の二階

なおこの店の二階には腐海おいが有るため覗く必要がある。二階も人によっては魅力的に映ることだろう。最近では営業時間を午後からにしているらしい。

13 浜田電気

ここは土日に路地でキーボードやCDROMドライブを売っている。全て新品だが値段はぼちぼちというところ。NICなども売っている。CDROMなどもそれなりに安かった覚え有り。

14 MAXUS(マクサス)

ここの4階は要チェック。本来の価格を完全に無視した売り方をしていたりもする。AAA-134CA? が3万円程度で売られていたりメーカー製のパッケージ商品も半中古で異常な値段で売られている。またショーケースの中にはCPUやメモリが売られてる。ここでMilleniumシリーズを見かけることも多く値段も安い。また何故かFM-TOWNS MSX?も売られている。とにかく変だとおもうし外せない巡回路。

15 湘南通商

超絶電脳に似てコンパクト製品の扱が多い。ショーケースの中を覗くと高級OpenGLカードやPenIIXeonやPenPROなども売っている。PenIIXeonがかなり安かったりDIMMがそれなりの値段だったりする。その他カード類はそれなりの品揃えで良い感じに売られている。SCSIホストやNICビデオカードなど値段も相場よりちょっと安くくらいでよい感じ、とりあえず金を持って来店し買い占めましょう。開店時間がちょっと遅めですが見逃さないようにしたい。

16 ソフマップ5号店

本来はメーカー製の中古などを売っている店だが、ここの互換機中古は非常に重要かも。HDDを初めメモリCPUなどを売っている。値段は基本的に高いが何故か安い商品もあるというか同じ商品でも値段が違ったりする謎の店。安いものをきちんと見分けて買うべし。

17 ソフマップトレジャー館6

ここは1F,2F共に見所がある。1Fはメーカー製デスクトップマシンの中古及びカード類。このカード類は毎週必ずチェックすべき。中には値段の付け方を判っていない物も存在する。また中古マシンもPC-UNIXを勉強するには非常に便利かと、新たにマシンを組むより確実に安く環境を手に入れることが出来る。p54c-120で1万円くらいで何とかなるとおもう。3万円有れば十分なマシンを手に入れることが出来る。2FはPC-UNIX及び鯖。sgiのインディゴやHP,DECの鯖がずっしりと置かれている。特に鯖は異常な値段で売られているため間違って買ってしまいそうになる。PenPRO/PenII Dualを楽しみたい人はよく見ておきたい。Alphaマシンだっけ置いてあったりする。最近では1Fは厳しいチェックにされされているため開店直後でない掘り出し物に会えない。まだ2Fに対するチェックは緩い方なので丹念に見たい。Netfinity7000が2回入荷したりなかなか仕入れは太いらしい。

D03

18 T-ZONE アウトレット館

ここは遠くてあまり行く気がしないが、メーカー製マシンのアウトレットはそれなりの数展示販売されているので同じスペックで値段をかなり抑えることが出来る。しかしソレ以外の取扱商品は相場を判っていないのか異常に高かったりすることもある。

D02

19 B1 PS/PLAZA WAKAMATSU

互換機が主力にみえそうだが 同店はモバイルに力を入れている。最後まで 200LX を扱っていたり 修理改造サービスを行っていたりするお店。値段は高めかと思いきや 何故か安い新品互換機パーツなどもあり侮れない。

20 1F 若松通商

土日になると 路上で互換機グッズを売っている。その中に注意してみると 98 用グッズもあつたりして非常に興味をそそる。しっかりと見ておきたい。以前 WinChip200+ゲタ の ViperMULTI が 4000 円で出ていた。言うまでもなく全部買った。

21 3F ぶらっとほーむ

安くはないのだが この店にしかない物もあつたりして 重要なお店。Millennium シリーズを どこからか仕入れてきて『今回は最後次は無い』と言いつつはや半年 いまだに在庫を持って販売しているようである。が かなり値段は崩れているようで 同店のもくろみは崩れたのだろうか? とりあえず MillenniumII 用 12MB 増設メモリの値段がいつ崩れるのか目が離せない<現在 10000 円オリジナル製品にも積極的に キーボードやコンソール切替機などもオリジナルのものが有る。NLX 筐体や 工業用マザーなどもあつて 非常に興味をそそられる。Logitech のマウスがいつも置いてあるのでいつでも買える。<安くないけど HHK の 2 台目実際に使ったけどやっぱりそれほどタイプ感がよいわけではないと思う、ファンクションキーも必須なのでチト辛い。

22 7F LAN2 PLAZA 若松通商

OS/2 LAN 専門店を謳っているが 最近互換機パーツが主力になっている。ここでは 商品の回転がちよつと遅いのか暮れも押し迫ったときにもフェーズ 1 の DVDROM を在庫していた。また 同店のジャンク箱が異色を放っていたりする。MCA の NIC が有つたりするので PCI と間違えて買ってしまいそう。店員はそれなりに知識を持っているようで 時間もあるようなので ある程度まとめて買う用事があるのなら頼ってみるのもおもしろいかもしれない。品揃えが悪いということはないと思う。

23 USER'S SIDE

多少知識のある店員が居ると思う。ただ見ているだけでもかなり親切なので 買ってくれると判るとさらに親切だとおもう。また 店内にハイエンドパーツを展示販売しているのでおもしろいかも。最近 HDD 用消音きつなども売っているし、SCSI ケーブルでも最高級な物を売っている。ケースもとんでもないヤツを置いているが少々高いかも。とりあえず チェックをする必要がある店だと思う。ただ HDD のセクションにも書いたように多少変質してきた同店今後に一抔ではない不安がある。

24 超級電腦

昔からなのだが、カナリ妖しい店。Compaq の鯖外し品や補修部品などを手広く扱っている店。ThinkPad がらみの商品も置いてある。取扱商品が手広く NIC、ハブから本当にサーバまで。価格が異常である店が特筆。また ジャンク HDD として きちんと動く HDD を安く売っているのが魅力的。場合によるが 2GB で 3000 円くらい。ビデオカードやプロセッサなども有る。OXIGEN や GLORIA なども売っている。後は WS 用のメモリも有るので 必要な人は覗いてみるのも良いかも。扱っている商品がかなり偏っているため ある程度の知識を積んでいくとかなりおもしろいと思う。

25 otto

2F のお店。メモリからメーカー製の補修部品 メーカー製中古まで扱う。妖しい物ではやはりコンパック鯖の補修部品のばら売りだろうか。PenII + VRM がかなり手頃な値段で置いてある、VRM 目当てに買うにも十分だと思うおい。中古としては PC-98 やノートスキャナなど。ひっそりと Apple も扱っているようだ。全般的に値段が安く 必要な物があるのなら しっかりと買っておきたい。つい最近 NEC 用のフロッピドライブ 1200 円であった。かなりの在庫量だったのでいまだにあるかも。メモリは値段の反映が速いため 買いたいときに買っておきたいとおもう。SIMM を扱っている重要なお店。

26 otto UNIX 専門店

5 階に新たにオープンした otto。何故だ と思うような値段の付け方をしたりもしている。そらりす なども売っているようだ。また IBM の inteliStation も安価で売られていた。YAMAHA RT-100I が 18000 円というのはちよつとやりすぎかとも思ったが... コンパッ

クの補修部品の扱いが一部 2F から移行しているようにも思った。ここでスイッチングハブを買ったことがあるがはずれだったが... が引き取ってくれたので好感度かなり高い。今後商品構成が変わることが予想されるので 頻度に足を運びたい。

27 あきばお〜 2号店

初号店から分離した CDR/互換機ショップ。CDR の値段はコンスタントい。また UPS 付 AT 電源もいまだに扱っている。その他にも CD 内の mp3 ファイルを自動で再生してくれるアダプタや CDR の自動焼き込みアダプタなど そっち系のアイテムが多数。FreshMusic や Rio など扱う。またゲームラボのバックナンバーが豊富。激安互換機パーツを求めるのなら足を運ぶといいかも。コンスタントに商品があるわけではないがその分妖しく安い。ある程度商品は入れ替わるのだが どうしても偏っているので 飽きが来るかもしれないが。

D01

28 ストレート

アキバでもっとも妖しいと 一時期は言われていた同店。休日には必ずガレージ販売をやっているようだ。ガレージ販売ははじめてみるとときかなり妖しい物が多く衝撃的で刺激的なのだが あまり商品が入れ替わることがないため少々飽きてしまったかもしれない。最近ではガレージセールと平行してその更に道側で 別の系列の店が露天を開いている。扱っている商品はプロセッサやボードの中古。とり外し品と思われる。Millenium 初代や WS 用メモリ NIC などが市場価格よりは安く売られている。C バス NIC(108) が 500 円で売られているのも特筆に値する。が 店番がかなりやる気が無く 商品に関する知識も持ち合わせていない。質問してはいけないジャンク屋の典型のようなもの。値段の付け方を悪い方に間違えている商品もあるので注意。なお 2F の本来のストレートは 相変わらずに妖しいです。吹き溜まりの様に、、、。

29 マッククラフト (A'zTECH の上)

ウィークエンドには露天販売もしているショップ。名前から Apple 系列かと思えば中ではディスプレイの中古から SUN や偏った互換機なども売られている。私はここで DEC の PersonalWorkstaion200i を買った。そんなに安くなかったが 血が騒いだ... 露天ではそれほど商品を扱っているわけではないので 是非ビルを上ることをお勧めする。店内には サイン入りのマッキントッシュ筐体や SUN やコンパックの SCSI ケースなど突っ走った商品が多数有る。SUN の SCSI ケーブルが安かったように思う。お約束的に取り外し品も多数有るので こまめに回った方がいいかもしれない。

E01

30 メカ風船

以前は 1F に紙風船 (ゲーム関係) が有ったが 現在は 2F のメカ風船が 1F にやってきた。商品の構成も以前よりもグレードアップしており ディスプレやマシンの中古が点数共に増えた。が 残念なことに PC-98 系は減ってしまったように思う。ここでは ボード類の特価販売が行われており 特に C バスカードや NIC などが安く売られている。以前ここで IntelNIC が 1500 円でしばらく売られていた。MDC-926Rs も 500 円を出ていたようだ。はずれは基本的に少ないが 926Rs ははずれが混じっていたようである。以前ははずれなど無かったのだが... なお ディスプレや妖しいマシンのみを扱っている。階段となりの店も 実はメカ風船だったりする。こちらの店主は古典的アキバ系だろうか...

E02

31 じゃんぱら

非常に重要な店。中古ビデオカードなどかなり多数取りそろえている。そもそも互換機系のショップなので 中古のマザーなども多数有るが 私の目にはやはりショーケース内のプロセッサやメモリボード類が魅力的である。Millenium や #9 など 様々なボードがかなり速いスピードで回転するため 目が離せない。Adaptec の SCSI ホストなども安価にある場合が多い。とにかく きちんとチェックをすべき店だと思う。早い時間に巡回しないと良い物がなくなってしまうのは掘り出し物が多い店の常。

F01

32 ZOA 秋葉原本店

まともな互換機パーツを買うなら一押しのお店。駅から遠いが値段はかなり安いと思う。店員も在庫も豊富でおもしろい。またポイントカードによる3%還元があるため繰り返し使うには良いお店。また平日でも安いため土日にしか全力を出さない店とは一線、また高額商品なら値引きが効くかも... 社員は非常によい感じですよ(知識はさておき)。また互換機ケースは特筆的に安いのでお勧め。nestの構成パーツの大部分やCRTなどを買った。

33 HOPPER

NECの流れ品があることで有名かも、そもそも妖しい店で超小型マシンなどの展示販売などもある。NECの補修部品が売られているのは昔から有名。PCカードスロットやCバス音源、ビデオなどが現行で売られている。また入荷する商品が妖しくAlphaのマザーCPU電源のセットなども出てきた、壊れていたらしいけど。最近注目なのはMGA-II 4MBやMATE-NX用増設CPUだろうか。また昔から増設用VRAMを扱っておりMGA-II用のチップ3k円Millenium用4MB4k円など一部の人たちの間では結構有名だった。その他店内には凶暴なウサギが飼われ居るようでたまにケージの中に入らないのだがどこに居るのだろうか。とにかく要チェックな店。

34 大漁エレクトロ

妙に妖しい店、公園を望む位置に建っているのだが店の前の路上にマシンを積んで売っているし店内にもアーキテクチャがあふれている。SUN有りsgi有り互換機98有りと言った感じで何でも扱おうという店主の男気が感じられるが値段が高いように思う。店主はたまにアジア系のバイヤーらしき人とお話ししているのでやっぱりその道の人なのだろうか、失われし98の独自仕様SIMMやメモリライザなどの取扱もある。Cバスボードはどれも高い。メーカー製マシンの中古はそこそこの値段。友人が言うにはこの店では値切ってなんぼとのこと、が私は今お店でお買い物をしたことがないので何とも言えませんが... 隣には改造した互換機に値札をつけて売ること有名なPC夢工房がある。そのついでに大漁によるのもおもしろいかもしれない。

35 パソコン工房 秋葉原3号店

昔はパソコン工房全店でICMの流れ品特価セールをかなりの長期に渡って行っていたのだが今ではもうやっていない。IF-1771ET 2000円 IF-1769 1500円など魅力的だった。あのとき買いだめしておいて良かったと今思う。

36 SMC 商会

ちょっと前まで98のマザーをばら売りしていたことでその筋では有名。が最近扱っている種類が減ってきたと思っていたら最近あまり扱っていないようだ。98のマザーにも色々な型があり同じ型番のマシンに使われるマザーも必ずしも一つではなかった。そのためG8????という型番を凝視する人たちの群が見られたのだが... そもそも結構妖しいグッツを扱う同店なので相変わらず売れなさそうな物を扱っているのだろうか、店内のショーケースにはL2キャッシュや小容量メモリなどが陳列されているがいわゆるコモンなので買う人は少ない。そもそも値段も魅力的ではない。ライセンス証書のみNTが売られていたような気がしないでもないが... ちなみに98のマザー(PCI付き)は3000円で売られていた。なお同店は休日には近くのガレージに出店を出しているのそちらの方が有名かもしれない。

37 あーなんとか

平日はなんか陰気な店をしているが休日になるとその店の前の道を借りて中古の露天販売をやっている。その値段がまたすごくちょっと常識を疑ってしまう。ライセンスがうやむやなCDを売っていたり(ライセンス証を単品で売っていたり)Allidtelesisの24ポートハブ(マネジメント機能付き)を2400円で売っていたりこの店も休日に行ったのなら必ずチェックしたい店。マッキントッシュがらみの商品やマック本体も格安で売られていることが多いのでそっち系の人にもお勧め。

G02

38 PC-NET パシフィックネット

アキバでもかなりはずれの位置にあるお店。そのため足を運ぶ人が少ないかもしれないが売っている物はかなりのもの。特にジャンクコーナーはかなり強力な引きがある。サーバやWSなど部品とりだけでも十分に元が取れる商品がコンスタントに並べられてい

る。そもそもまともに売っている商品の値段は普通で品揃えは狭いが数が多いという形。結局ジャンクコーナー目当てに来店してしまう。CPUやNICなどのジャンクもあるがビデオカードのジャンクがないのが惜しいところ。がしかしこのジャンクコーナーは必ず見なくてはならない。そういうお店である。最近改装し多少ジャンクコーナーが狭くなったと思う。その分ばらさずそのままジャンク扱いになって居る本体が増え価格も下がった。相変わらず魅力的。

おわりに

今回原稿ということで書きましたが 実際には まだまだ書き足りないことだらけです。まだまだ 私が足を踏み入れている店も多くあるし。足りない部分は あなた自身が回って 発見してください。結果報告を頂けると 非常に嬉しかったりします。さらなる高みを目指し 買い漁りたいものです。

ポケステのプログラミングをしてみよう

ソフトウェア編

takkun@mma.club.uec.ac.jp

2000年4月

さあはじめよう！

さて、今回はポケステになにやら書き込んだりして、自分でコンパイルしたプログラムを動かす所まで紹介しました。今回は前回の記事で試しにコンパイルして実行してみる実験に使用したナンバーズの source code¹ を元に、いろいろポケステの内部構造について迫ってみたいと思います。

おや？ここに、「ぼけつとすてーしょんひみつだいひゃっか」なるものがあります。あ、S●EIとか、書いてありますね。ふむ。では、とりあえずこれを参照しながら読みといて行く事にしましょう。²

とりあえずファイルリスト

さて、ナンバーズに入っている source code は、こんな物が入っていました。

Makefile

まあ、Makefile です。TARGET に ARM gcc をインストールした場所を入れないといけないということ、前回説明した gcc へつけるオプション `-mcpu=arm7tdmi -mthumb-interwork -mapcs-32 -msoft-float` を付け加える程度で、後は問題ありません普通の Makefile です。

ldscript

ld(1) に対して `-T` オプションの引数として与えられています。なんだかメモリマップの配置を指定しているようです。これは、..... ます。

locore.S

頭の所から、なにやら定数がいろいろ書き込まれています。"SC"とか、"MCX0"とかは、K-Communications さん³ の所に置いてある、ファイルフォーマット (以下、`pkformat.txt`)⁴ の物と酷似しています。おそらく、そのような事が書かれているのでしょう。ところで、`locode.S` という名前は、何か意味があるのでしょうか。そういえば、NetBSD とかで、`src/sys/arch/i386/i386/locore.s` なんて file もありますし、なんか意味がありそうですけれど...

number.c

なにやら長たらしい source です。一応 `main()` とかいう関数が定義されているので、ここが頭のようなようです。

print.c

小さい source です。なにやら、`font[][]` とかいう global 変数が用意されています。print.c という名前といい、文字関連の事をなにやらしていそうです。

rand.c

とても小さな source です。というか、みれば解りますが、`rand()` という関数を定義しています。まあ、ゲームには最低限必要な疑似乱数って奴ですね。

system.c

これも、なにやら長たらしい source です。`init()` という関数が定義されていたりするし、名前も `system.c` なわけで、それっぽい事をしていそうです。

¹<http://www.geocities.co.jp/SiliconValley-PaloAlto/6226/>

²と、いうことで、守秘義務契約により情報は適当に修正されてしまうでしょうねえ...

³<http://www3.airnet.ne.jp/kcomm/psdev/pokedev.html>

⁴<http://www3.airnet.ne.jp/kcomm/psdev/pkformat.txt>

bitmap/*

なにやら XBM 形式のビットマップデータが沢山あります。でも、実際の source ではどれからも include されていないので、そのディレクトリにある b2w.c でもって適当な word(32bit です)の形式に変換して使っているのでしょう。

と、ということで、大体 locode.S, number.c, system.c がキモのようです。

それでは、読んでみよう

てなわけで適当に source code を読んでみる事にします。さっきざっと眺めた所では、どうも、件の pkformat.txt によれば、locode.S の中の、_progstart という所がプログラムの開始位置らしいです。ここでやっていることは…

```
_progstart:  
b main
```

だ、そうです。main に飛んできますね。あれま。いきなり number.c の main() に飛ばされています。ふむ。なるほど。そいでは main.c:main() では何をやっているのでしょうか。main() はこんな感じです。

```
int  
main(argc, argv)  
int argc;  
char *argv[];  
  
init();  
srand(sys_gettime());  
memcpy(&save_data, SaveDataBuf, sizeof (struct save_data));  
  
sleep_th = 32 * 30;  
func = title;  
  
for (;;)   
check_connect();  
vsync_wait();  
lcd_write(lcd);  
pad_check(0);  
  
(func)();
```

なにやら init() なりなんなりを呼んで、その後ループしていますね。func とかいう関数へのポインタな奴を状態変数として使っている感じがです。まあ、とりあえず、init() が何をやっているのかみてみましょうか。

init() のやっているコト

ということで、init() は system.c で定義されています。この中ではなにやら、

```
*(volatile int *)0x200 = 0;
(volatile char *)0x200 = 'B';
(volatile char *)0x201 = 'N';
```

と、いった、変な access がいろいろあります。これは、なんなんですよ。とりあえず、それぞれの前に、なんだかそれらしいコメントがあります。ちょっとそれぞれについて読み解いてみます…

for bianero

bianeroってなんなんですよ？うーん…どうも、PlayStation 側のソフトの名前のようです。こいつを使うと PlayStation にポケステを差しておけば、PlayStation 側のコントローラでいじれるようです。へえ。なんか、面白そうですね。でも、今回の場合は本質とは関係なさそうなのでとりあえず無視しておきましょう。

system clock

なんだか `sys_clock()` とかいう物と呼んでいますね。こいつは…`locode.S` 内で定義されている何やら `swi` を呼び出すだけの代物らしいですね。うーん、とりあえず `swi` についての考察はあとですることしましょう。

sound volume

音のボリュームを設定しているらしいです。これでも、`sys_status()` というなにやら `locode.S` 内で定義されている `swi` を使った何かを使っていますね…後は、`volume` という global 変数に値をセットしています。そういえば、ポケステでは音量の設定が出来ましたよね。なにやらそのあたりの関係の匂いがします…

mask interrupt

なんなんですよ。これ。意味不明な 16 進数の値を `volatile unsigned int *` だとかいって、その中身に値を入れています。ふむ。0xa00000c とか、0xa000008 とかいうアドレスには何か秘密がありそうです。という、…………… わけですね。なんとなく解って来ました。

regist handler

またまた `sys_ふんちゃら()` です。うーん、とりあえず後回しにしましょう。あとは、`IntTable[]` という配列になにやら `timer*_handler()` を登録しているようです。timer_handler と言うのだから、おそらく、タイマ割り込みとかの時に動いて欲しい物なんだと思います。

timer0

これも、何やら怪しげなアドレスに怪しげな演算を行ったりしています。ふむ。これは、…………… ということのようにです。

enable interrupt

これも、怪しげなアドレスへの演算です。この場合は、…………… で、その後……………

LCD on 32hz

これもまた怪しげなアドレスに何やら書き込んでいます。ええと、これは…………… LCDControl ですね。適当に LCD の周波数を 32hz に変更しています。

と、いうことで、`init()` の動作はほとんど理解できました。確かに、`init()` と言うだけあって、それっぽい事をしていますね。というか、どうも、これくらいなら K-Communications さんの所に置いてある、ポケステメモリマップー覧 (以下、`pkmemmap.txt`)⁵ にそれっぽい事が載っているみたいです。参考になるかもしれません。……………

swi ってなんじやらほい

さて、さっき眺めた `init()` の中では、`locore.S` の中で定義されている、`sys_*` とかいうものと呼んでいました。これって、だいたい、

⁵<http://www3.airnet.ne.jp/kcomm/psdev/pkmemmap.txt>

-
- swi 7
「カーネルのステータス設定?」だそうです。詳細は不明だそうです。まあ、重要なもんでも無いんでしょう。
- swi 8
「アプリケーションの FlashROM ブロック番号取得と設定」だそうです。今走っているプログラムの FlashROM のブロック番号を取得したり、後で走らせるプログラムのブロック番号を設定したりするらしいです。
- swi 9
「swi 8 で設定したアプリケーションの起動」だそうです。気分的には exec(3) ですね。どうもこれで他のプログラムを走らせたりするみたいです。
- swi 10
「シリアルナンバーの取得」だそうです。そういえばポケステにはシリアルナンバーが割り振られていたんですけど。どうも、それが手に入るらしいです。
- swi 11
「カーネルのステータス設定?」だそうです。ええと…と、いうのは、.....
.....
.....
と、いうことらしいです。
- swi 12
「リアルタイムクロックの設定」だそうです。内部に持っている時計を設定できるようですね。
- swi 13
「日付読み」だそうです。リアルタイムクロックから日付を読む事が出来るらしいですね。どうも、yyyymmdd といった BCD で帰って来るそうです。
- swi 14
「曜日と時間読みだし」だそうです。なるほど、曜日と時間は別に読み出すんですね。Owhhmmss の BCD で帰って来るそうです。
- swi 15
「?」だそうです。なんだかループに入ってしまうらしいです。いったい何なんでしょう。うーん、これは、.....
.....
- swi 16
「FlashROM への書きこみ(\$08 版)」らしいです。「swi 3 の\$08000000 版」って事らしいです。そういえば swi 3 では 0x02000000 からでしたっけ。
- swi 17
「通信割り込み設定、解除」らしいです。PlayStation に刺さっていないと設定しようとしても設定できないとか書いてありますね。
- swi 18
「指定したブロック番号のアプリケーション情報?」だそうです。なんだか説明を読むと、.....
..... ということが出来るらしいです。
- swi 19
「ステータスのアドレス?」だそうです。どうも固定の値が帰って来るんだそうです。で、その値がステータスのアドレスと一緒にだと言いたいようです。はあ、なるほど。
- swi 20
「swi テーブルが格納されているアドレス」だそうです。
- swi 21
「ステータスのアドレス 2 ?」だそうです。こいつも swi 19 と同じで固定の数値が帰って来るようですね。

swi 22

「現ブロック番号の読み取り」だそうです。うーん、そのままなのでしょう。

swi 23

「ステータスのアドレス3？」だそうです。こいつも固定の数値が帰って来るようですね。

swi 24

なんだかよくわからないものをチェックして返すようだとかなんとか書いてありますね…ええと、実際には、

1.

2.

..... ようです。

うーん、なんだか凄いですね。いろいろ swi で出来ちゃうようです。なんだかポケステの内部事情がいろいろ解って来た気がして来ます…ちょっとわくわくしますね。

まだまだmain()

てなわけで、init() の理解は終わりました。では、次に呼び出しているのは何でしょう。ええと、srand() です。まあ、これは、rand.c 的に乱数の種を指定しています。sys_gettime() で得られた物を指定しているようです。sys_gettime() は swi 14 を呼んでいるだけなので、これは現在の時間を得て、その時間を乱数の種に使っているんですね。なるほどです。次は…なんだかmemcpy() していますね。なんだか SaveDataBuf という locore.S で定義された 128bytes の領域を save_data という global な構造体に copy しています。ははーん。なんだか、自分の中にデータをセーブしておいて、後から読み出すとか、そういう事をしているようですね。あとは、sleep_th とかいう global 変数に 32*30 を設定しています。これって、なんでしょ。んー、なんだか system.c の中の pad_check() とかいうので使っていますけれど…まあ、とりあえずこれは置いておきましょう。次は…func という関数へのポインタに、title() を入れています。ははーん。やっぱり、func は状態変数として使われていそうですね。

メインのループはどーなってるの？

はいでは、次に main() 内でのループの中身について解析して行きましょう。ループの中では何をやっているんでしょう。ええと、

- check_connect();
- vsync_wait();
- lcd_write(lcd);
- pad_check(0);
- (func)();

と、いうことをしているようです。まあ、とりあえず上からみて行きますか。

check_connect()

これは、system.c の中で定義されていますね。ふむ。読んでみるとなにやら怪しげなアドレスに対していろいろやっているらしいです。pkmemmap.txt によれば、最初にチェックしている 0xd8000c ての bit4 は、PlayStation に刺さっているかどうかをチェックする物のようです。ははーん。どうも、PlayStation に刺さっているかどうかをチェックして、画面を 180 度反転させたりなんかするのかな？あ、どうもそれっぽいコメント文も見受けられます。なるほど。てことは、まあ、こいつは個人で楽しむソフト作りにはほとんど関係ないとか切り捨てちゃってもよさそうですね。.....

vsync_wait()

これも、また、system.cで定義されて…います…けれど…中身が…変ですね。

```
void
vsync_wait()

register int v = VsyncCount;

while (v == VsyncCount)
;
```

てのは、一体どういう事でしょう。vにはVsyncCountを入れていて、それが==の場合はループする…ということは、なんだか無限ループのようです。はて。なんじゃらほい？これは、変です。おかしいです。はてはて。って、あ、なんか、他の所でVsyncCountを使っている所がありますよ。timer0_handler()です。この関数は単純にVsyncCountを++して帰っています。どうも、これが呼ばれれば行けそうですね。って、timer0_handlerって、init()した時にIntTable[7]にセットされていましたよね。ははーん、なんだか今までとりあえずほったらかしていた事がなんとなくつながって来ました。ええと、要するに？init()でIRQハンドラをIntIRQ()に設定して、そのIntIRQ()はirq_handler()を呼び出して、その中ではなにやら怪しげなメモリ領域をいじって……………何やらstatusの中のn番目のbitで、IntTable[n]が定義されている場合、その関数を呼び出しています。てことは、さっきのtimer0_handler()が呼ばれるわけで、それが呼ばれてVsyncCountがなにやら++される…そうするとvsync_wait()の無限ループから抜ける…と。てことは、timer0_handler()は、その名の通り、タイマ割り込みを処理する奴みたいですね。てことは、多分、IntTable[7]ての、7てのは、タイマ割り込みの事なんでしょう。ふむ。なるほど。

lcd_write(lcd)

で、こいつですが、これもまた、system.c内部で定義してありますね。これは単純に0xd000100に対して指定されたポインタから32*sizeof(unsigned int)bytesだけmemcpy()しています。ええと、pkmemmap.txtによれば、0xd000100~0xd00017Fてのは、「LCDのVRAM」という事らしいので、そのままVRAMにデータをコピーしているわけです。てことは、unsigned int lcd[32]てのは、画面表示するためのバッファって事ですね。へーへー、なるほどー。

pad_check(0)

で、次はこいつです。ええと、こいつも、system.cですねえ。ええと、これは、なにやら0xa000004から読み出して、ごによごによやっています。

```
opad = raw_pad;
raw_pad = *(volatile u_int *)0xa000004 & 0x1f;
raw_pad |= *(volatile u_char *)0x203; /* from bianero */
(volatile u_char *)0x203 = 0; /* clear */
pad = raw_pad & opad;
```

ここです。0xa000004てのは、pkmemmap.txtによれば、割り込みステータスらしくて、割り込みを禁止にしても立つらしいとかなんとか書いてあるので、どうやらそういうことらしいです。で、その中の、0x1fを読みだしている、と。これは、なんなんですよ。うーん、pkmemmap.txtの同じ所を書いてある情報から行くと、上下左右のボタンと決定ボタンらしいです。ははーん。これはpad_check()というだけあって、まんまパッドのチェックをするみたいですね。へーへー。なるほどー。pad = raw_pad & opadっていうのがミソ臭いですね。これをやることによって、padには、ボタンを押し続けている間や、押され無いのが続けられている間は、padの値は0のまま、押された瞬間だけ1になるようになっています。で、padや、raw_padはglobal変数なので、後で参照できるわけです。もちろん、pad_check()が呼ばれるまでその値は変化しないので、変なpad情報を掴まされる事ありません。ふ

むふむー。良い感じですね。

んでは、その後の所では何をやっているんでしょう。ええと、`raw_pad == 0` だった時でのと、そうでない時でなにやら `global` 変数をいじっています。`sleep_count` とかいうのをいじっている所とかから考えると、キー入力がしばらく無いと自分から寝るようにしたりするみたいな気がします。なんだか実際 `sleep()` とかいうそのものずぼりっぽい奴を呼んでいますし。あ、なんだか `sleep_th` とかいう、`main()` の頭で設定した奴をカウンタの値に使っていますね。ふむ。なんだか、この値をいじると寝るまでの時間が変わったりしそうです。

`(func)()`

さて、最後のこいつです。うーん、これは、`func` という関数へのポインタが入る変数なわけで、最初は `title()` が入ってます。ふむ。で、とりあえず `title()` が呼ばれる、と。

func の動作

で、です。この `func` がかなり、キモです。要するに、`func` に現在の動作すべき状態 (実際には関数ですね) が入っていて、それが `main()` 内のループで毎回呼ばれるわけです。(実際には、`main()` 内でのループは `vsync_wait()` の時間で適当な時間区切りで回って来るようになるようになっていきますね) ほいでは、`title()` から読んでみましょう。

`title()` を読むと、なにやら `lcd[]` に変なバイナリを `memcpy()` して、その後、`print.c` で定義されている `printnumber()` を呼んでいます。なんだか、`save_data.highscore` てのを表示させてるみたいです。ああ、なるほど。これはタイトル画面のようです。って、`title()` なんだか、当然なのか。おお。なるほど。で? `func` に `button_wait()` を設定して、同じように `funcnext` に `gamestart()` を設定する…と。で、`button_wait()` てのは…

```
if (pad & PAD_BUTTON)
    func = funcnext;
```

てな事が書いてあるので、ボタンが押された場合は、`func` が `funcnext` で上書きされる…って、要するに `button_wait()` って名前そのものの動作をするってわけですね。

そうすると次は…`gamestart()` ですか。こいつは、なにやら `global` 変数に適当な値をセットして、`lcd[]` に、0 を設定していますね。これは、画面をクリアしているわけですね。で、次は `game()` ですか。

`game()` てのを読んでみると、どうも、これがナンバーズのメインループみたいです。さっき解析した `global` 変数の `pad` から、パッドの内容を良く読んで、適当な処理をして、画面に反映させるような動作をしたり、しているようです。で、`numbers_x` が、28 より大きくなったら、`save_data.highscore` をなんとなく更新して、`gameover` に移行する、と。は一は一。なんだか、ナンバーズの秘密を理解したような感じです。

まとめ

一応、ナンバーズの `source code` をざーっと読んでみました。なにやらポケステの内部構造について理解が深まったような気がします。今なら、K-Communications さんの所の解析情報を読んで自分なりのポケステプログラムが書けると思います。実際、きれいにまとめられていてとても使いやすいので、この情報は必須でしょう。といっても、いきなり一から書くのは大変なので、なんとなく理解したナンバーズの `source code` を改造して動かしてみるってのが面白いんじゃないかと思います。この記事を読んでポケステのプログラム開発に興味を持って頂けたら幸いです。

have fun!

Copyright ©IIMURA Takuji 1999