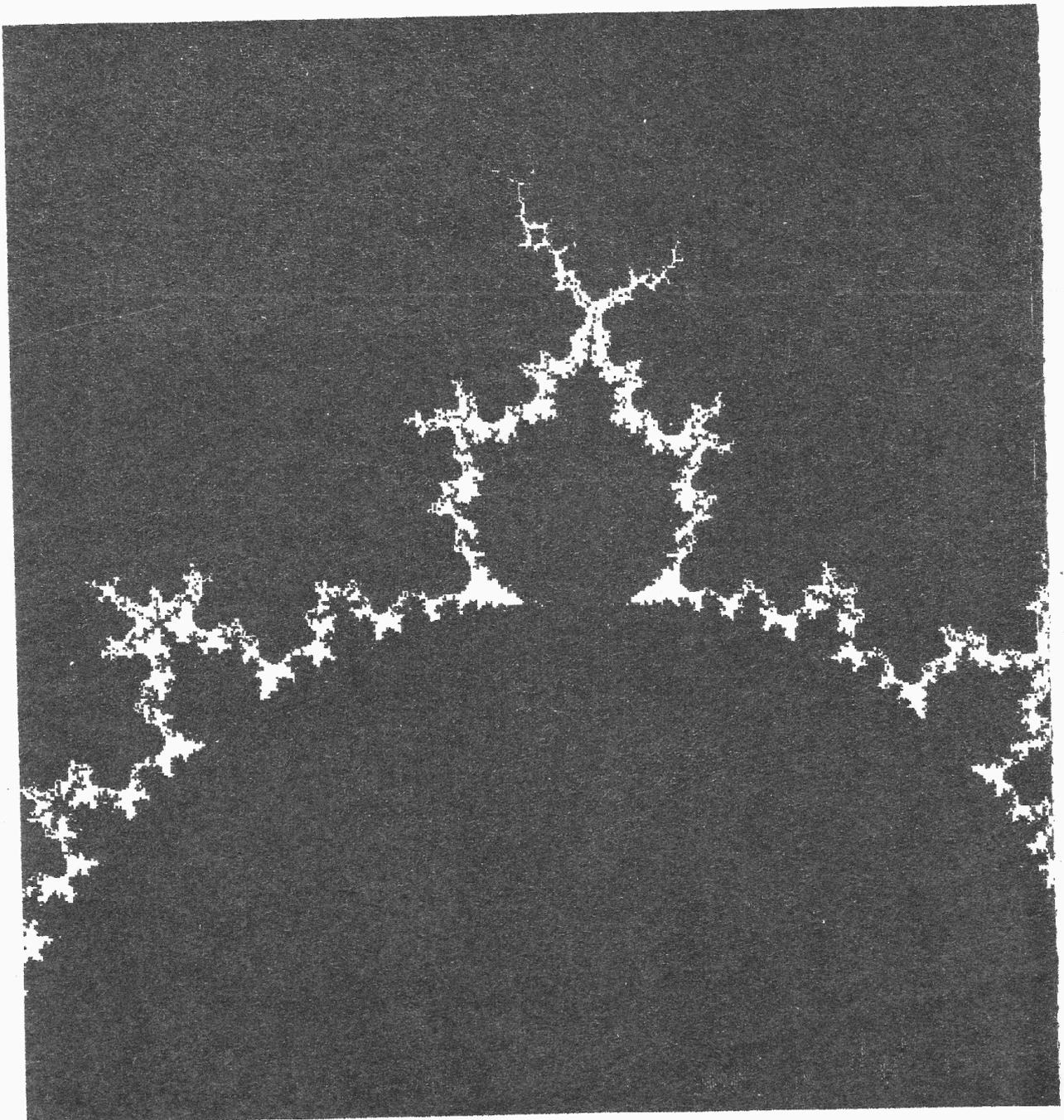


百万石

Micro computer Making Association

930411



Contents

ハードからソフトへ

村上 敏之

1

X Window System について

uncover

2

ネットワークの話

今井 潔

3

Floppy Bootable X server

楯岡 孝道

7

CRYPT とパスワードの (だれもしらない) ひみつ

清水 了

10

hterm の 30 行化計画

rikachan

12

試験に出る MMA 語録

横瀬 泰洋 編

17

ハードからソフトへ

— 部長の言葉に代えて —

MMA 部長 3C 村上敏之

(Email: e11163@ced.cas.uec.ac.jp)

MMA は "Micro-computer Making Association" の略で、日本語に訳せば、「マイクロコンピュータを作る部」となります。

よって、普段の活動は読んで字の如し、といたい所ですが、実際はちょっと違ってきています。

もともと MMA は『まだワンボード・コンピュータが全盛で、市販のコンピュータが高価であったころ、もっと安価で使いやすいものを作ろうということで誕生した』(*1)とされています。要するに、各自が使いやすいコンピュータを作るという夢を持ち、追ったわけです。

伝え聞く所によれば、数年前までは確かに MMA が設計・制作したマシンのほうが学校の計算機室のマシンよりも速く、環境も良かったという事です。

ですが、ワークステーションやパーソナルコンピュータ等が安価(昔と比べてですが)、高性能、複雑化した今、私達アマチュアがそれに対抗できるマシンを作るのはほとんど不可能になりました。

例えば、DEC 社が世界最速と自称する Alpha という CPU のシリーズには、200MHz で駆動し 30W もの電力を食うのがありますが、こいつはプロでも実装に手間取ったりします。ベンチマークが速いのは確かなんですが……

そういう事情のため、MMA の活動のプラットフォームは自作のハード+ OS/9 から、市販品の IBM PC/AT 互換機+ UNIX に移り、それに伴い、ハードウェアからソフトウェアへと活動の中心が移りました。それでも、理想的なコンピュータの像を追い求めるのに違いはありません。(*2)

ところで、入部に、特別な知識は必要ありません。私がみなさんに求めるのは、ただ一つ。コンピュータに夢や興味を持っている事。それだけです。

例えば、

俺のプログラムで皆の環境改善をしてやる。

もっと使い良いコンピュータ(OS)が出来るはずだ。等々。

当部に興味を持った方は、サークル棟2階のMMA 部室の扉を叩いて下さい。別にとって食ったりはしませんので、お気軽においで下さい。

脚注

*1 『まだワンボード……誕生した』
1990年11月発行 「Super 百万石」 冒頭文より引用

*2 理想的……ありません。
単に、見果てぬ夢を見ているだけかもしれないが。

(BGM: 老犬トレイ)

ネットワークの話

今井 潔

平成6年 4月 7日

1 はじめに

この文章は、現在多くのコンピュータネットワーク構築に使われている(もちろん、MMAの部室でも)イーサネット(Ethernet)について、主にハードウェア側から書いたものです。

2 イーサネットのこと

ゼロックス社のロバート=メトカフ博士が1973年にイーサネットを発明しました。信号の衝突を検出することによって1本の同軸ケーブルに多くのコンピュータを接続できるのがその特徴です。¹現在よく使われているイーサネットは、次の3種類です。

名称	ケーブル	LAN 構成距離	データ転送速度
10BASE 5	太い同軸ケーブル	1 セグメント 500m	10Mbps
10BASE 2	細い同軸ケーブル	1 セグメント 185m	10Mbps
10BASE-T	より対線	HUB-マシン間 100m	10Mbps

このなかで10BASE-Tだけは、ネットワークの形態や衝突検出のやり方が異なっています。

2.1 CSMA/CD

イーサネットでは、ケーブルを複数のコンピュータが共有することによってネットワークを構築します。そこで問題になるのが信号の衝突です。ケーブルを複数のコンピュータで共有するというのは素晴らしい考えですが、あるコンピュータが通信しているときに他のコンピュータが通信しようとするとう通信の衝突が起こってしまいます。この衝突を回避するには時間に依存して

¹CSMA/CD-Carrier Sense Multiple Access with Collision Detection-搬送波感知多重アクセス/衝突検出と呼ばれる。

通信を割り振れば可能ですが、それでは柔軟性に欠けます。時分割にすると深夜ネットワークががらあきなのにもかかわらず、いつもと同じ決められた速度でしか通信できません。またコンピュータが増えれば増える程、通信速度が落ちていきます。イーサネットは衝突は*あるもの*と考えることで成功しました。イーサネットの通信手順は次の通りです。

1. 誰かが、話していないか聞き耳をたてる。
2. 話していないなら相手に話しかける。
3. このとき不幸にして他の人の呼びかけと*衝突*してしまったら、ぶつかったとみんなに知らせてからだまる。
4. 適当な時間(これは乱数による)沈黙したのち、最初の手順に戻る。
5. また、イーサネットを使用するソフトもやりとりする情報をいくつかに分ける²ことによって、回線の占有率を減らしています。

3 10BASE 5 のこと

10BASE 5 は古くてなおかつ研究室が入っている学内の建物でよく見られます。その様な建物の廊下の天井付近を這っている黄色いケーブルが10BASE 5 の同軸ケーブルです。

3.1 ネットワーク形態

ネットワーク形態はバス型と呼ばれているものに分類されます。バス型は一本のケーブルを複数のコンピュータが共有することによってネットワークを構築します。

3.2 衝突の検出

イーサケーブルの中を信号は0-2Vの電圧で伝わっていきます。信号が衝突を起こすと、ケーブル内の電圧が通常ではありえない4Vまで上昇し、それとわかります。

3.3 ネットワークの接続

10BASE 5 では1セグメント中100個のトランシーバの取り付けが可能です。このトランシーバを2.5mおきに付けられているマーキングにあわせ

²この分けられた情報のかたまりをバケットという。

てケーブルに取り付け、そこからトランシーバ・ケーブル³で、コンピュータのインタフェースにつなぐだけです。

4 10BASE 2 のこと

10BASE 2 は細い同軸ケーブル⁴をつかったネットワークで、経済的です。MMA の部室ではこの 10BASE 2 が使われています。

4.1 ネットワークの形態

10BASE 5 と同じバス型です。

4.2 衝突の検出

これも 10BASE 5 と同じです。

4.3 ネットワークの接続

10BASE 2 の場合、トランシーバがインターフェース上に乗っているため 10BASE 5 のようにケーブル上にトランシーバやタップを付けることはしません。T 字コネクタと呼ばれるものによってインターフェースと接続します。その結果、ケーブル-T 字コネクタ-ケーブルという形になります。⁵

5 10BASE-T のこと

アメリカの電話線を使ったネットワークです。故障、不具合を見つけやすく故障個所の切りはなしも容易なことから、現在最もはやっているネットワークです。

5.1 ネットワークの形態

こちらは、他の 2 つと違ってスター型の形態をとります。スター型とは中心となる機械 (HUB という) に、それぞれのコンピュータがつながります。コンピュータはその HUB 以外とはつながりません。そのため、あるコンピュータが故障したときはそのコンピュータをネットワークから切りはなすだけですみます。

³DSUB15 ピン、AUI ケーブルともいう

⁴インピーダンスは異なるが、外見はテレビのアンテナケーブルと同じ。

⁵詳しくは、部室のマシンの裏を見よ。

5.2 衝突の検出

10BASE-T では送信と受信にそれぞれ 2 本の線を使います。送信しているとき、受信部に信号が入ると衝突したと判断します。これは 10BASE 5/2 のように電圧によっているわけではないので、ケーブルの途中で光ケーブルをいれたとしても問題はありません。10Mbps に追いつける変換コネクタがあればですが。

5.3 ネットワークの接続

先に述べたように、HUB とコンピュータを、より対線でつなぐだけです。

Floppy Bootable X server

楯岡 孝道 (tate@cs.uec.ac.jp)

平成6年 4月 1日

1 概要

Floppy Bootable X server (以下 FBX) は、386 以上の CPU を実装した IBM-PC/AT (以下 PC) を X 端末化するプログラムの集合である。

最大の特徴は、ハードディスクを一切使わない事であり、新しい OS のインストールなどを行っている途中でハードディスクが使用不可能な場合でも、フロッピーディスクが使用可能な PC を X 端末とすることができる。

2 背景

現在、MMA では 386 以上の CPU を使った PC を各種 OS の実験プラットフォームとして利用している。しかし、OS 実験中には実験参加者以外はその PC を使う事ができず、端末不足も手伝って実験を行う事に抵抗があった。

そのような状況で、MS-DOS の DOS-expander で動作する商用 Xserver (Xappeal) の評価バージョンが制限はあるものの利用できるようになった。また、フロッピーディスクから起動可能でフリーな UNIX システムも現れた。このような UNIX 上では一般的なワークステーションと比較しても遜色無い Xserver を動作させる事ができる。

3 フロッピーディスクで動作させる事の問題点と解決法

Xappeal も UNIX 上の Xserver も、普通に動作させるにはハードディスクを利用するのが前提になっている。これは、これらが動作するのに必要な巨大なプログラムとフォントなどのデータを保持し、必要なときに即時利用する為である。また、主記憶が不足したときにハードディスクを利用した仮想記憶によって動作する事もある。

しかしながら、低速で小容量のフロッピーディスクでは巨大なデータの保持や、仮想記憶での利用は現実的ではない。そこで、これらの要求をハードディスクを使わずに満たす工夫を行った。

巨大なプログラムの格納

これは利用するシステムによって異なるが、次のような手法が考えられる。

- diet 等を用いて、実行時自己展開型圧縮ファイルにする。
- 不必要な機能を省いた状態でコンパイルし、小さな実行形式を作成する。
- NFS を用いてファイルサーバ上のファイルを利用する。

巨大なデータの格納

巨大なデータのほとんどはフォントデータである。そこで、X11 Release 5 から導入された font server の機能を用いてデータを

取り寄せるようにする。こうする事により、サーバマシン上に用意したフォントデータを、ネットワークを通じて、必要に応じて主記憶に読み込むことができる。

仮想記憶の実現

これも、NFS を用いてファイルサーバ上のディスクに行く。ただし、用いている OS によっては、この機能はない。

4 試作したシステムの詳細

今回は 2 種類の異なるシステムを試作した。1 つは OS に MS-DOS 5.0 を利用し、Xserver として Xapeal を用いた。

もう 1 つは OS に BSD 系の UNIX の 1 つである NetBSD 0.9 を利用し、Xserver として XFree86 2.0 を用いた。

以後、前者を MS-DOS 版、後者を NetBSD 版と呼ぶ。

4.1 MS-DOS 版

MS-DOS 版では、diet による圧縮と、font server の利用を行った。MS-DOS 用 NFS は利用せず、全て 1 枚のフロッピーディスクに収めた。

4.2 NetBSD 版

NetBSD 版では、必要な機能に限定してファイルサイズを小さくしたカーネルと、一部の設定ファイルをフロッピーディスク上に用意し、他のファイルは NFS マウントしたディスク上に用意した。

5 効果

FBX を起動すると、UNIX マシンの XDM に XDMCP を発行し、XDM の画面となる。そ

れから先は、普通に login 可能である。

フォントは、font server を利用しているので font server が供給しているフォントは全て使えた。しかし、フォントの読み込みにネットワークを使うので、時間がかかる。

6 それぞれの利点と問題点

FBX には次のような利点と問題点がある。

6.1 MS-DOS 版

6.1.1 利点

- 付属の SVGA driver は豊富な機種に対応している。
- フロッピーディスク一枚で起動し、接続するサーバを選ばない。
- 設定が用意である。

6.1.2 欠点

- フォントの読み込みが低速である。
- ハードウェア依存部分が少ないらしく、描画が低速である。
- 評価バージョンのため、物理解像度が 640x480dot 固定である。
- 評価バージョンのため、X プロトコルのコネクションを同時に 4 コネクションしか張れない。さらに、font server でコネクションを 1 つ消費するので、同時に 3 コネクションしか使えない。

6.2 NetBSD 版

6.2.1 利点

- ハードウェア的に描画可能な多くの解像度を利用可能である。

- ハードウェア毎に最適化が行なわれていて高速に動作する。
- X サーバとしての制限はほとんどない。

6.2.2 欠点

- NFS サーバ上に X server などのファイルを用意する必要がある。
- 仮想記憶に用いるディスクがないので、仮想記憶が使えない。
- ハードウェア毎に設定が必要なので、設定が難しい。

7 今後の課題

どちらのシステムも仮想記憶が使えないので、普通の使用では問題無いが、大量のフォントなどを読み込んだ時などに問題の起きる可能性がある。

そこで、より小さく、かつ NFS の先にスワップディスクを取れる Linux などの OS を用いてシステムを構築すると、より実用的になると考えられる。

CRYPT とパスワードの (だれもしらない) ひみつ

清水 了

1994/4/8

crypt はパスワードの暗号化ルーチンです。これは、NBS Data Encryption Standard(NBS のデータ暗号規格)に基づいており、キー検索についての、DES のハードウェアインプリメンテーションの使用をせずにするように(その他にもいろいろと)意図された変更が加えられています。

crypt の暗号化アルゴリズムは、いわゆる不可逆変化というもので、encrypt された文字列からオリジナルの文字列を得ることは不可能です。ただ、crypt のアルゴリズム自体は公開されているので、encrypt された文字列が、あるオリジナルを変換したものかどうかを評価することは可能です。UNIX 等におけるユーザ認証は、これにより root でさえもユーザー個人のパスワードを知られずにユーザ認証することが可能です。

しかし、いくら不可逆変化と言っても、暗号化のアルゴリズムが公開されているというこは、たかだが 64bit の入力(パスワードに関しては 96 の 8 乗通り)である為、総当りに試せばパスワードを見つけることが可能です。それなのにどうしてその保守性が保たれるのか? その答えは crypt のアルゴリズムにあります。crypt のアルゴリズムは非常に時間のかかるアルゴリズムを採用しています。一般的な速度のマシンを使い、総当りに調べていっても数ヶ月~数年かかると言われています。当然数ヶ月の間にそのユーザがパスワードを変更する確率はかなり高いと思われるので、これによりある程度の保守性が保たれているのです。

しかし、パスワードに使われるキャラクタコード、そして文字列(単語)にはある程度の傾向があります。もし、パスワードに使われているキャラクタがアルファベットの小文字だけで構成されているとするならば、総当りに調べるとしても、大幅に計算量を減らすことが可能です。更に言うなら、もし、パスワードに使われている文字列が、一般的な英単語、あるいは日本語のありがちな言葉をアルファベット化した物とするならば、たったの数千~数万語で収まってしまうことでしょう。数万という数字は、高速なプロセッサに

とっては取るに足らない量です。これくらいの計算ならば、経験上、数分～数十分で終わってしまうことでしょう。

つまり、いくら crypt が遅いからと言って、アルファベットだけのパスワードにするとか、大好きなアイドルやアニメのキャラクタの名前にするなんてことは絶対にしてはいけないということです。

それでは、私のいままでの経験に基づく、悪い password の例を挙げてみましょう。

まず、アカウント名と同じパスワード。もう論外です。あなたが誰かに hack して欲しいなどという理由がない限り、悪いことは言いません、一刻も早くパスワードを変えるべきです。

あなたの本名 (ファーストネーム、ファミリーネーム)。/etc/passwd には、大抵あなたの本名も書かれています。これで検索されたらおしまいです。

あなたが頻繁に使うホスト名、およびマシンの名前。これもすぐに見つけられることでしょう。

アニメのタイトル、アニメのキャラクターの名前。アイドルの名前。そして、ありがちな女の子の名前。こんな物はその手の辞書を使って調べれば一発で引っかかります。alphonse とか totoro とか、madoka、miyuki、ayumi、chisato、などというパスワードには絶対にしてはいけません。hack されるのは当然。更に、それをネタに大笑いされることは必須です。

車の名前。これも結構あるようです。pajero とか skyline とか bluebird とか……。私は車の種類に関してはよく知りませんが、その手のデータベースがあれば簡単に見つかってしまうことでしょう。

... 等といろいろ書いてきましたが、当然、上に挙げた物の、すべて大文字にした物や、一文字目だけ大文字にした物、前後に記号を付け加えた物もあぶないと思って良いでしょう。一番良いのは、半年～一年に一度くらいは、サイコロをふって 0x20～0x7e までのキャラクタが一様に出てくるようなパスワードを作ることです。覚えるのが大変かも知れませんが、年に 1～2 回の苦勞で、安心が買えるのですから安いものです。

password: のプロンプトを見た時には、あらゆる可能性を考えて神経質になりましょう。どこに視線があるか、どこにトラップがしかけられているか、わかったものではありませんから...

1. htermとは

htermとは平野 聡氏が作成されたIBM-PC/AX/PC9801/PC98XA/J3100用通信ソフトである。軽快な動作、MS-DOS上で動作するソフトウェアには少ないVT220エスケープ・シーケンスのサポート、X/YMODEMのサポートなど多くの特徴を持ち、最新版が1990年11月に発表されたという比較的古いソフトウェアであるにもかかわらず現在でも十分実用に耐えるソフトウェアである。

2. 30行計画とは

30行計画という言葉はPC-9801ユーザーなら一度は聞いたことがあるであろう。その名の通り、通常では25表示が限界のPC-9801ノーマルモードでそれ以上の表示を可能とするソフトウェアの総称である。現在では初期の30行計画をさらに発展させたTT*¹や30行BIOS*²が発表されている。

3. htermを30行対応にする

筆者は普段BBSに接続する場合はKmTermX*³を使用している。しかし、KmTermXはVTエスケープシーケンスのサポートが不完全であり、tinが立ち上がらなかつたりviの画面が乱れたりして実用に堪えなかったので、viなどを利用する場面ではhtermを利用していた。

htermの最終更新日の時点で30行計画はまだ存在しなかったのでhtermは当然対応していない。普段の筆者の画面行数はTTによる拡張で31行になっているのだが、31行に馴染んだ者にとってhtermの25行というのは非常にストレスが溜まる。たかが6行であるが、その6行は大きいのだ。

こうして、htermを30行計画に対応させることとなったのである。

— note —

*1 30n行環境 [TT.com] Copyright (C) by 「早紀」

*2 30行BIOS (C) lucifer

*3 KmTermX Copyright (C) kim, DAIJU

4. 30行対応以前の問題

筆者の利用しているコンパイラはBorland C++である。このコンパイラにhtermは正式対応していない。事実、htermのソースをそのままコンパイルしてもエラーを吐いてコンパイルは止まってしまう。まずはコンパイルを正常に通るようにhtermのソースを書き換える必要がある。

また、htermはPC-9801の拡張グラフィックスモード（16色モード）に対応していない。この点についても書き換える必要がある。

4.1 コンパイラを通す

そのまま1から書き直すのはいかにも面倒なので、1992年11月に発表されたKenji Rikitake <kenji@rd.macrofield.or.jp>氏によるB98-01及びMASM 5.1、Microsoft C 6.0に対応させるパッチキットを利用する。筆者のPC-9801DSはシステムクロックが8MHz系統なので9600BPS以上にRS-232Cの転送レートを設定できないが、B98-01に対応させることにより筆者の環境でも19200、38400BPSに転送レートが設定でき、SportSter 14400モデムの高速性を活かすことができる。

しかし、このパッチキットを適用しただけではBorland C++を通すことはできない。MASMとTASMの仕様の違いなどにより再びコンパイラが止まってしまう。

以下においてその対処法を述べていこう。

4.1.1 関数プロトタイプの問題

file.cのsearchPath()において

```
>char *getenv(), *strcat();
```

と関数をプロトタイプ宣言している。しかし、これ以前にstdlib.h, string.hをincludeしているのでこのプロトタイプは必要ないし、関数の2重宣言となってBorland C++はコンパイルを中止してしまう。

これは、この部分をコメントアウトすることで対処した。

4.1.2 MASM, TASMの仕様の違い

上の変更でコンパイル自体は全て通るのだが、リンク時にエラーが出て止まってしまう。これはMASMとTASMの仕様の違いによるものである。

エラーの出たserdrv98.asmを見てみると、

```
> .data
>
>;
>; C interface
>;
> extrn cMask:byte ; character mask
> extrn maskFlag:byte ; 1=XOFFed
```

```

>     extrn  downloading:byte      ; 1=down loading
>     extrn  xonXoff:byte         ; 1=XON/XOFF enable, 2=RTS/CTS enable
>     extrn  portAddress:word     ; port address
...
>     extrn  mouseEvent: proc      ; mouse handler C part procedure

```

というようにデータセグメント定義の後でexternal procedure宣言がされている。TASMはこの点をMASMよりも厳密に見ているのでリンク時にエラーが出てしまうのである。

これは、.dataの位置をexternal宣言の後に移動することで対処した。

4.2 拡張グラフィックスモードに対応させる

PC-9801VM以降の機種では従来の標準グラフィックスモード（8色中8色表示可能）に加えて拡張グラフィックスモード（4096色中16色表示可能）がサポートされたが、htermでは標準グラフィックスモードしかサポートしていない。これが原因でバックグラウンドカラーを表示している場合などに画面が乱れる場合がある。大阪府立大学版グラフィックスを使用する場合はglio98.cを16色モード対応に書き換える必要があるが、とりあえずこれを使用しないことにすると変更点はpc98.cのsetCRTMode()で8色モードであると決め打ちしてパレットレジスタにデータを書き出している部分のみである。

この対応部分をインプリメントするために機種にPC98VMを追加し、#ifdef文で条件コンパイルすることにした。

4.3 その他の変更

Borland C++に対応したということで、コンパイラタイプにBorland C++を追加した。

5. 30行に対応させる

これからがいよいよ30行対応の本番である。まずはヘッダファイルから変更を加えてゆくことにする。

まず、30行パッチを条件コンパイルによって適用する・しないを設定するためにTTというオプションを設定した。

また、各種環境を取得するための各種ルーチンをtt.cというファイルにまとめることにする。このファイルに記述される関数については以下で随時説明する。

5.1 25行決め打ちへの対応

htermで設定可能な行数はどの機種でも25行までである。これは内部で最大行数が25行であると決め打ちしている部分があるからである。

まず、default.hで行モードを次のように決め打ちしている。

```

>#define LINE_MODE_24  0
>#define LINE_MODE_25  1
>#define LINE_MODE_19  2

```

```
>#define LINE_MODE_20 3
```

とりあえずこの設定に加えて

```
>#define LINE_MODE_29 4
>#define LINE_MODE_30 5
```

をつけ足す。また、TTが#defineされている場合はMAX_LINE_MODE, DEFAULT_LINE_MODEもそれなりに変更するようにした。

hterm.hでは各行モードにおいて利用可能な最大行数を記述している。30行環境では利用可能な行数は不定なのでとりあえず、この#defineを利用している部分は変数を利用するように書き換えることにして、ここでは適当な行数を設定しておく。今回は以下の様になった。

```
>#define MAX_LINE_29_MODE 24 /* 30mode with FEP */
>#define MAX_LINE_30_MODE 35 /* may not with status line */
>#define BOTTOM_LINE_29_MODE (MAX_LINE_29_MODE-1)
>#define BOTTOM_LINE_30_MODE (MAX_LINE_30_MODE-1)
```

この定義の直後で

```
>#define MAX_CHAR (MAX_COLUMN * MAX_LINE_25_MODE)
```

と表示可能な最大文字数を決め打ちしている。このMAX_CHARは各所でバッファの確保、画面の退避などのために利用されている。

とりあえず、ここは

```
>#ifndef TT
>#define MAX_CHAR (MAX_COLUMN * MAX_LINE_25_MODE)
>#else
>#define MAX_CHAR (MAX_COLUMN * MAX_LINE_30_MODE)
>#endif /* TT */
```

とすることで対処した。この対処法ではMAX_LINE_30_MODE以上の行数は利用できないことになるが、これはその部分の記述を書き換えればすぐに対処可能なのであまり神経質になることはないだろう。

さて、これで決め打ちをしている部分はなくなったが、30行環境に対応するには環境によって変わる部分を取得しなければならない。

そこで、グローバル変数を定義しているヘッダファイルであるglobal.hに変数extMaxLine, extBottomLine, existTTというグローバル変数を用意した。変数の意味はそれぞれextMaxLineが利用可能な最大行数、extBottomLineが最下行の行番号、existTTはTTが常駐しているかどうかのフラグである。これらの変数はtt.cのTTInit()で設定することにして、hterm.cのinitialize()関数から呼出すことにした。

5.2 30行関係のコードを埋め込む

さて、以上の変更でほとんど変更点はなくなったに等しい。もともとhtermは20行、25行モードに対応するために行モードに依存した部分はまとめてあるので、その部分を変更すれば30行対応は終わりである。

まずは、console.cのsetLineMode()である。この部分で各モードに応じた変数を設定しているので、ここにLINE_MODE_29、LINE_MODE_30に対応したコードを埋め込めば終わりである。

次は、pc98.cのsetCRTLineMode()である。この関数を呼出す部分は全て

```
>setCRTLineMode(realMaxLine < 21 ? 0 : 1);
```

のように20行モードなら0、それ以外なら1を引数に呼出している。呼出されたsetCRTLineMode()ではその行数に応じてCRT BIOSを呼出している。そこで、上の記述は

```
>if(realMaxLine < 21)
>  setCRTLineMode(0); /* 20 line mode */
>else if(realMaxLine < 26)
>  setCRTLineMode(1); /* 25 line mode */
>else
>  setCRTLineMode(2); /* 30 line mode */
```

として30行モードなら2を引数に呼出すことにした。setCRTLineMode()では引数が2ならTTを拡張モードにするようにTT APIを呼ぶことにした。このために、tt.cにTTMode()という関数を作成した。

最後はhterm Setupに29行、30行モードを記述すれば基本的な30行対応は終わりである。

6. おわりに

htermはかなりモジュール化が進んでいるので比較的簡単な変更で30行対応が完了した。もう少し手をつけないと完璧ではないのだが、一応簡易的なパッチであるということで御勘弁いただきたい。

最後に、htermの作者である平野氏と、B98-01パッチに対する改造を許可して頂いた力武氏に感謝してこの文章を終わりたい。

試験に出るMMA語録

横瀬 泰洋 編

インタラプト

コンピュータのハードウェア割り込みのこと。
転じて仕事の途中で命令が入ること。

例 ちょっとまってインタラプトが入った。

落ちる、飛ぶ、こけた

正常動作しなくなること。

例 今井が落ちてる。

ガベージ コレクト

小さい余り領域を集めて、大きいメモリ領域を作ること。
転じてゴミをどかして場所を作ること。マロックしようとしたときに頻繁に生じる。

コアダンプ

実行中にエラーを起したプログラムがそのメモリエイメージをファイルに書きだすこと。
転じて、酒を飲んだ人間が P-----。

例 松崎がコアダンプした。

コンフリクト

イーサネット上でのパケットの衝突のこと。
転じて同時に複数の人が話し始めてしまうこと。

例 ちょっとコンフリクトしたよ。

サーバー クライアント

仕事請け負い人(サーバー)と仕事依頼人(クライアント)のこと。

ソート

順番を並びかえて整列すること。
転じて麻雀の理牌のこと。

ツリー

別名 未構造とも呼ばれるデータ構造のこと。
それとは別にMMAにtreeさんと呼ばれる人物が存在するがこれは入学時の心理テストでひっかかったことに由来する

ツルツル

障壁がない状態のこと。

デフォルト

初期設定のこと
転じて一般常識のこと

とほほ、しゃれにならん、だめか、へぼ
MMAの今井の一人言で、ポピュラーなもの

パケット

データの塊のこと。
転じて会話の単位のこと。

例 ごめん、今パケット落としたからもう一回言って

まじでくやしーよな、なんでなんだ
MMAの松崎氏があるとき口にした言葉。

MALLOC(マロック)

データ領域を動的に確保するためのCの関数。
転じて場所を確保すること。

NMI

Non-Maslacable-Interrupのことで、拒否できない割り込みのこと。

例 先輩の"ねえねえ"がこれにあたる。(どきどき)

RS232C

有名なセントロニクス規格のシリアルインターフェース。
treeさんのうなじに端子があることで有名。

百万石

ISBN 10-9823-4750-91

940411 発行

発行者 MMA

装丁 清水了

編集 清水了

製作・著作 MMA

無断転載複写禁止