１９９１創刊３号

ＩＮＤＥＸ

／：

／ｐｒｏｊｅｃｔ：

／ｅｓｓｅｙ：

／ｍｉｓｃ：

部長のお言葉

橋谷　貴光

「MMAとはMicro-computer Making Associationすなわちマイクロコンピューターを作る会であります。今から十数年前、まだワンボード・コンピューターが全盛で市販のコンピューターが高価であった頃、もっと安価で使いやすいものを作ろうということでMMAは誕生したと伝えられています。」

このように、去年の本誌の序文にあります。

しかし、今現在市販されているコンピューター（ワークステーションはおろかパソコンもポケコンも）は年々コストパフォーマンスが増加し、自作ではそれらにかなうものは作れなくなってきています。

このような状況の中では、我々のようなサークルの行方はきわめて不透明であるといわざるをえません。なかなか「やれることが見つからない」という部内の声も聞きます。ハードを作るような人間も年々少なくなってきているのが現状です。

しかし私はまだ我々の仕事が終わったとは思っていません。確かに、今まで我々が歩んできたパワーへの道は前途がほとんど閉ざされているかもしれません。ワークステーションのランクにいたってはたとえ金がいくらあっても自作は不可能ですし．．．。だけれども、「メーカーが作らないような変なもの／意表をついたもの」はまだ作れると思うのです。

たとえば、こんど私はＰＣＭのボードを作ってみたいと思っています。小さくまとめて携帯用にして鳴らして遊ぶ．．．いまいちですね．．．。

あえてこの世の中でひねくれてみる勇気と精神の持ち主はいませんか？

ワークステーションやパソコンでやれるようなことでなく、なにかアイデアのきいたものを作れないか、考える私です．．．。

ｌｏａｄｓｅｒｖｅｒ

楯岡　孝道

loadserverとは、ＡＲＥＳの負荷平均（load-average）などの情報を他のプロセスに知らせるサーバーです。

なぜこのような物が必要になったかというと、プロセステーブルの内容を定期的に参照するプログラムの増加にともない、テーブル参照に使うＣＰＵ時間が無視できなくなったためです。

問題となるプログラムのほとんどは、X-windowにおけるxloadのようなシステムに常駐してユーザーにその統計情報を示すもので、それらのプロセス自身が負荷になっては本末転倒になってしまう性格を持っています。

しかしながら、プロセステーブルの参照はＯＳ９／６８ＫにおいてもＵＮＩＸ程ではないにしろかなりのＣＰＵ時間を使用します。このテーブルの参照部分を一つのサーバーに任せれば、全体としての負荷が軽くなるのではないかという考えに基づいてこのプログラムは作られました。

現在のloadserverは次のような情報を約１秒毎に提供しています。

- プロセス毎の情報：
  - プロセスＩＤ
  - ユーザーＩＤ
  - モジュール名
  - プロセス生成時からの動作tick数（システム内、ユーザー）
  - 前回からの動作tick数（システム内＋ユーザー）
  - 前回の情報に含まれていたプロセスかどうか
- 全体としての情報：
  - 前回からの経過時間をtick数で表した物
  - 全プロセスの前回からの動作tick数の和（システム内、ユーザー）
- loadserver自身の情報：
  - データを作った時間
  - データモジュールのバージョン
  - サーバーによる書き込みフラグ
  - クライアントのアクセスカウント

クライアントがこれらの情報を利用するには２つの方法があります。１つめはクライアントプログラム作成時にヘッダファイルloadserver.hをインクルードし、ヘッダファイルに記述されている構造体を用いてデータモジュールを直接アクセスする方法です。２つめはヘッダファイルloadlib.h及びライブラリloadlib.oによって、すでに用意されているマクロや関数を用いる方法です。

１つめの方法ではデータへの自由なアクセスができますが、クライアントプログラムが複雑になる上、データモジュールの保護が不完全なため最悪時クライアント側からデータモジュールを破壊してしまうことも有り得ます。

２つめの方法を用いればクライアントは比較的安全にデータにアクセスできますし、プログラムも簡単になるのでほとんどのクライアントはこの方法を用いています。問題があるとすれば、余分なルー

チンまで含むために実行形式が大きくなってしまうことと、loadlib にバグがあったときに全てのクライアントがその影響を受けてしまうことぐらいでしょう。（これは実話だったりします。:-)）

クライアントによるデータへのアクセスは次のように行います。

・データモジュールへのリンク。（起動時のみで可）
・サーバーによる書き込み中かどうかのチェック。
・データモジュールのアクセスカウントのインクリメント。
・データの読みだし。
・データモジュールのアクセスカウントのデクリメント。
・データモジュールのアンリンク。（リンクと１対１に対応）

サーバーがデータを書き込むときには、モジュールの書き込み中フラグを立てた後アクセスカウントをチェックし、クライアントによりカウントが０以上にされている場合にはアクセスカウントが０になるのを待ちます。この簡単な機構によってクライアントは、サーバーが変更中の不完全なデータを読まずにすむようになっています。しかしながら、この仕様には不注意なクライアントによってサーバーを完全に止めることができるという問題点があります。

このようなサーバーのロックに対処するために、サーバーにシグナルを送ってデータモジュールのリセットを行うことができます。悪意あるプログラムに対してはこの程度の対策では対処できませんが、メモリプロテクションの無いＯＳ９／６８Ｋではセキュリティはあって無いようなものなので、あえてこれ以上のことはしてありません。

実際のクライアントの代表的なものは、次のようなものです。

・ｂｅｓｔ　　負荷の重いプロセスの上位５つを表示する。
・ｐｅｒｆｃ　　負荷状況をグラフ表示する。
・ｄｐｍ　　負荷状況をデジタル表示する。
・ａｃｃｔ　　プロセスのＣＰＵ使用時間等の記録をとる。

これらのクライアントのほとんどはloadserverができる以前からあったもので、プロセステーブル参照部分を書き換えたものです。実際に各プログラムの使用ＣＰＵ時間を比較したところ、ほぼプロセステーブル参照の分だけ少なくなっていました。

さらにテーブル参照アルゴリズムの変更やデバッグ等はloadserverに対してのみ行えばよくなり、一連のプログラムの保守性が格段に上昇しました。この結果、サーバー、クライアント共に改良が進に、今ではbestとperfcの２つのクライアントを動かした状態でも全ＣＰＵ時間の３％程度の消費ですんでいます。（私は個人的にこの負荷のことをＡＲＥＳの消費税と呼んでいます。）ちなみに初期バージョンのbestは５０％ものＣＰＵ時間を独占していたのですから、どれほど改良が進んだかが判ると思います。

しかしながら、loadserverにもいくつか問題があります。１つはすでに出てきたサーバーのロックです。これが起こるとすべてのクライアントが止まってしまいます。２つめはloadserverのバージョンアップによってloadserver.hに記述されているデータモジュールの構造体に変更があった場合に、クライアントをリコンパイルしなければならないことです。この問題はloadserverが完全に完成した場合には自然に解決されるものですが、今のところはクライアントがデータモジュールにリンクする

ときにモジュールバージョンのチェックを行い、異なる場合にはエラーになるようにしているだけです。

今後のloadserverの課題としては、統計情報の多様化とネットワーク対応があります。ＭＭＡのネットワークはまだ構想段階にすぎませんが、現在制作中の６８０４０マシンが量産できればそれらをイーサーネットでつなぐ計画があります。そのような環境においては各マシンの状態を知る要求は大きいでしょうから、loadserverのネットワーク対応は意味のあることだと思われます。もっともネットワークに現在のような情報を流すのは問題があるので、現在のloadserverとは似てもにつかない物になるかもしれません。

（ただの）端末

中村　仁一

ほんとは、もっとすごいものができあがるはずだったのが、いつのまにかただの端末になってしまったばかりか、まだ端末としての動作すらしていないＣＰＵボードについて以下簡単にのべます。（というほどのものでもないけど）

ＭＭＡでいろいろやっているうちに、やっぱり自分でもなにか作りたい、ＲＯＭライタも壊れている、入出力が複合されたＣＰＵのチップが出回っている、というような条件が重なって、インテリジェントなＲＯＭライタでも作ろうかというのがことのはじまりです。大まかな構成は次のとおりです。ＣＰＵには６８０００に入出力やタイマ、デコーダなどを複合した６８３０１というチップを利用し、書き込むＲＯＭは直接プログラムのアドレス空間に展開して（ここが一番のお気に入りで、ぜったい市販品ではやらないだろうなぁなんて思っている）、内臓のタイマ、ＰＡＬで書き込みのシーケンスを作ってあげます。このマシンの一番大名である点は、なんと言ってもメインシステムより早い１６ＭＨｚの６８０００をつかっている点でしょう。このような構成になったのはなるべく簡単にしてハード・ソフト両方でめんどうなところをなくす、メインシステムと同じＣＰＵをつかうことで開発期間を短くするなどの、まじめな理由のほかに、ＭＭＡらしい普通でないものを作るという大事な目的があったのです。

これがなぜただの端末になってしまったかというと、そのあまったＣＰＵパワーをなにかに使うためにとりあえずＣＲＴディスプレイでもつなぐことにしようかなどと言っているうちに、はやく作り上げるために機能を絞ろうということになってしまったのです。それでも特徴である、端末にしては速いという点を残したかったので、ＣＰＵとＣＲＴコントローラのアービトレーションを工夫して、１つのメモリサイクルで２文字書き込むことと、ＣＰＵのウェイトを最小限にすることを実現しています。具体的には、表示をアスキーキャラクタ（１バイト）のみに限って、そしてなんと８×８のフォントをつかって５０行、ビデオラムは１６ビット幅（ラムとキャラクタジェネレータは２つ）、ひとつのＣＲＴコントローラサイクルに表示と描画の両方をいれる、といった構成になりました。なぜタイトルが「ただの・・・」であるかわかってもらえたと思います。（ＭＭＡではこんなものも作っているということも）

この端末の制作を通して、プログラマブル・ロジック・デバイスの利用法がなんとなく身についたというのが最大の収穫だったように思います。これをもっと大きいシステムの制作につなげていきたいと思います。

編駐：大名とは．．．非現実的な理想を追い求める態度のこと。本誌創刊号により詳しい定義がある。

ＭＳ－ＤＯＳフォーマットディスクへのアクセス

飯田　卓

現在我がサークルでは、ＡＲＥＳと呼ばれる自作システムとその上で稼働している０Ｓ－９と呼ばれる０Ｓをメインに、コンピュータ環境の改善をモットーに日夜活動を続けております。

さて、既にご存知のことと思いますが、現在家庭にはびこっているコンピュータに乗っている０Ｓは、ＭＳ－ＤＯＳと呼ばれるものがほとんどです。当然のことながら、このマシンでフォーマットされたフロッピーはＡＲＥＳでは読めません。しかし、自分の持っているシステムで書いたプログラムを利用したい、と思うのが人情というもの。

事実、前にも同じようなことを考えた方がいたようで、現在ＡＲＥＳにはmrというコマンドが存在しております。これは、ＭＳ－ＤＯＳでフォーマットされたディスク上のファイルをＡＲＥＳに読み込もうというものです。しかしこのコマンドは、階層化ディレクトリをサポートしておらず、また読み出したファイルは標準出力へしか書き出せないというものなのです。

そこで、これらの欠点をカバーし、なおかつディスクへの書き込みもサポートするコマンドを作成するに至りました。現在、このコマンドはmwなる仮称で呼ばれております。

このコマンドは大きく分けて３つのモジュールによって構成されています。

１．intel 形式 12bit FAT を展開・再構成するもの

２．ディレクトリ・エントリを生成するもの

３．実際にデバイスとやりとりするもの

の３つです。このうち、３．についてはmrで使用していたものをそのまま流用させていただきました。残りの２つについては大きな困難もなく作成することができた…はずだったのですが、いざテストという段階になってＡＲＥＳのＦＤＤが突然死んでしまったのです。そのような訳で、現在このコマンドは十分なデバッグをすることができないでいます。

原理的にいえば、mwは次のような動作をするはすです。

１．ブートセクタからそのディスクのメディア情報を読み込む

２．扱いにくい intel 形式 12bit FAT を 16bit に展開する

３．与えられたファイルから malloc したバッファにデータを読み込む

４．フロッピーにセクタ単位で書き込む

５．ディレクトリ・エントリを生成する

６．FAT を再構成し、ディスクに書き込む

現在ＡＲＥＳは解体・大掃除・ＦＤＤの交換などの大手術を受けておりますので、mwが完動する日もそう遠くはないでしょう。

将来はftpライクなコマンド体系を持つmft(Ms-dos File Transfer)なるコマンドにしたいと密かに考えております。

ｊｓｅｒｖｅｒマシン構想

楯岡　孝道

ｗｎｎという漢字変換システムをご存知な方も多いと思います、このシステムはほかに例を見ないサーバークライアント方式による漢字変換を行います。これによって、辞書を一元管理できる、クライアントをアプリケーションに組み込むことができるので漢字を利用し易くなる、パワーのあるマシンでサーバーを動かすことができるので遅いマシンでも快適な漢字変換ができる等のメリットが生まれています。

しかしｗｎｎはＵＮＩＸ用に開発されたこともあり、システム資源の消費が並大抵ではなく、特にサーバーにおいては起動時に数メガバイトもある辞書をメモリに読み込むなどのメモリ消費や、Ｓｕｎ３／６０でも遅いといわれるほどの処理の複雑さから、手軽に利用するには問題があるといえます。

ソースの状態で利用できる漢字変換システムとしては、ｗｎｎの他にＮＥｍａｃｓというエディタ上で動作するＳＫＫといるシステムがあります。こちらはＮＥｍａｃｓさえ動かせばほとんど問題なく動くようですが、文節変換ができない等機能的に難があります。さらにＮＥｍａｃｓ自体も巨大なプログラムであり、簡単には動作させることはできません。

ひるがえってＭＭＡのマシンの状況を見ると、老体に鞭打って動かしている感のあるＡＲＥＳ（ＯＳはOS9/68K）に一応動作する漢字変換システムがあるものの、動作が不安定であり、改良しようにもソースもない状態なので手も足もでません。また、次期メインシステムはＯＳにＭａｃｈを利用する予定ですが、この上でｗｎｎを動かした場合メモリ不足からサーバープロセスのスワップアウトが起こり、速度低下を引き起こして、最近のわがままな部員の不評を買うことが予想されます。またＳＫＫにいたっては「パソコンより弱いなど却下である」と言われてしまうでしょう。

そこでｗｎｎのサーバーであるｊｓｅｒｖｅｒを動かすためのマシン、すなわちｊｓｅｒｖｅｒ専用機が考えられました。専用機であれば余分なハードウェア、ソフトウェアが必要なくなり開発コストも低くなりますし、スワップアウトの心配もないため迅速な反応が期待できます。また、クライアントだけならメインシステムを利用しているユーザーが負荷を感じることもないと思われます。

具体的なスペックはまだ決定されていません。ｊｓｅｒｖｅｒのプログラムが要求しているシステム資源をよく調べてみないといけませんが、今のところはＯＳとしてＯＳ９／６８０２０またはＭａｃｈを考えています。場合によっては自作してもおもしろいかもしれません。ＣＰＵは68020から68040あたりが有力候補です。そのほかには４～８メガ程度のＲＡＭとイーサーネットとのインターフェースが必要と思われます。ハードディスクはあれば辞書などをアクセスし易くなりますが、ＮＦＳかｔｆｔｐを使えばそれほど必要ないでしょうし、シリアルポートなどもデバッグ時に欲しくなる程度と考えられますから、必要性は低いでしょう。

このプランはまだ構想段階にすぎませんが、十分実現可能なものだと考えています。次期メインシステム上でアプリケーションプログラムが作成できる環境が構築された後に、実際にプロジェクトとして発動するでしょう。

６８０４０マシンへの分散処理オペレーティングシステムＭａｃｈの移植

石橋　尚史

ＭＭＡでは次期主力マシンとなる６８０４０マシン（制作中）にＯＳとしてＭａｃｈオペレーティングシステムを移植する予定である。　ＭａｃｈはＣＭＵ（カーネギー・メロン大学）で開発されたコンパクトなカーネルを中心にＵＮＩＸ互換のシステムコールやネットワーク機能などをサーバーとして備えているオペレーティングシステムである。

Ｍａｃｈのカーネルはタスク・スレッド管理やメモリ管理、タスク（またはスレッド）間通信などを行う。

タスク・スレッド管理ではプログラムの生成・実行の管理を行う。プログラムは資源の割当単位であるタスクと、実行単位であるスレッドとで構成される。実行単位と資源割当単位とを分離したことにより、プログラムのコンテキストスイッチング時に入れ換える情報量が少なくなりオーバーヘッドが少なくなっている。１個のタスクには少なくとも１個のスレッドが実行されるが、複数のスレッドを実行することも可能である。

メモリ管理ではメモリ資源の割当てを管理する。ページングによる仮想記憶が可能である。仮想記憶機構はＣＰＵのアーキティクチャーに依存しない部分（独立部）と依存する部分（依存部）の２つで構成されている。独立部と依存部の間のインターフェースが規定されているので移植性が高い。また、メモリマップドファイルという機能がある。これはメモリ空間にファイルを貼り付けて（マップして）、メモリ空間にアクセスするとファイルにアクセスできるという機能である。これによってファイルポインタの移動を気にすることなくファイルの任意の位置にアクセスできる。

タスク間通信はポートとメッセージを使って行われる。ポートはタスクがカーネルや他のタスクと通信する唯一の窓口である。タスクが生成されるとポートが１つ割当てられる。このポートを通って他のタスクへデータを伝達するのがメッセージである。メッセージを他のタスクに送ると受け側のタスクのポートにメッセージが到達する。１度に複数のタスクからのメッセージを受け取ることができる。

ＵＮＩＸ互換システムコールやネットワーク機能はサーバータスクとしてカーネル外で実行される。このサーバーによって、ＵＮＩＸのコマンドやツール群はほとんどそのまま動作する。したがってＵＮＩＸ互換システムコールサーバー（以後ＵＮＩＸサーバー）が動作していると見かけ上はＵＮＩＸが稼働しているように見える。

ＣＭＵから配布されているＭａｃｈ３．０にはＵＮＩＸサーバーは含まれていないので、ＭＭＡでは独自に設計・開発中のＵＮＩＸサーバーを採用する予定である。このＵＮＩＸサーバーとアプリケーションタスクとのやりとりはメッセージ交換によって行われる。しかしアプリケーションタスクはメッセージ交換を意識する必要がなく、ＵＮＩＸの標準ライブラリと同様の呼出インターフェースを持つスタブライブラリを結合するだけでシステムコールが行える。また、ネットワークサーバーが稼働していれば、ネットワークを介して他のマシンのＵＮＩＸサーバーへのシステムコールが可能である。

このように強力なＭａｃｈオペレーティングシステムを主力ＯＳに位置づけて、より快適な開発環境をめざしたい。

小は大を兼ねる　　深澤　真男

これと言って書くことがないのでポケコンのお話をば……。そもそもポケコンとはポケットコンピュータの略であり……、なんて事はどうでもいい。要するにシャープ系ポケコン、特にＰＣ－Ｅ５００とその互換機種についての話がしたいんだ！

思い返せば中学二年のとき、父が会社から貰ってきた１２６１がポケコンとの出会いであった。そしてあれは高２の冬（日付まで覚えてる。あれはまぎれもない１月６日だった）、私はとうとうＥ５００を買った。それ以来、どこかへ行く場合、ほぼ確実にＥ５００をもち歩いた。その年の９月には、気が付いたら覚えたばかりのアセンブラで妙なプログラムを作っていた。ちなみにそのプログラムは多少の改良の後、しばらくして（１２月２６日）にＰＪへと投稿された。載ったのはかなり後だったけれど。

前置きはそれくらいにして、とりあえず話を進める条件として、Ｅ５００のＲＯＭバージョンの識別法を知っておいて欲しい。それは簡単で、＆ＦＦＦＦ０をＰＥＥＫすることによって返される数値が、その機体のＲＯＭバージョンである。現在市販されているＥ５００／Ｅ５５０／１４８０Ｕ／１４９０Ｕ／１４９０ＵⅡにおけるＲＯＭバージョンは、１～７である。このうち、Ｖ１／Ｖ２がそれぞれＥ５００、Ｖ４／Ｖ５が１４８０Ｕ固有として与えられているが、Ｖ３／Ｖ６及びＶ７はＥ５００／Ｅ５５０／１４９０Ｕ／１４９０ＵⅡで共用されている。よって、出荷の古いＥ５００／１４８０Ｕのみが、ＲＯＭバージョンによる機種の特定が出来ることになる。

また、知ってのとおり、Ｅ５００／Ｅ５５０はエンジニアソフトを搭載しており、ＩＯＣＳも一部が追加されている（というよりもこちらが正しいＩＯＣＳである）。対して、大学生協でのみ販売されている１４８０Ｕ／１４９０Ｕ／１４９０ＵⅡは、エンジニアソフト無しであり、ＩＯＣＳが減った代わりに関数命令がいくつか追加され、怪しげなタイマーと、１４８０Ｕ以外には使いにくいグラフ機能と怪しげな伝言板ソフトがついている。

細かい違いを出していくと、Ｖ１からＶ７迄のうち、Ｖ２はＳ３：のＦＩＬＥＳが取れない。なぜかは分からないがＳ３：に不可視属性が付けてあった。Ｖ４／Ｖ５はＳ３：を持たないので蚊帳の外。そして、Ｖ３／Ｖ６／Ｖ７は、迷惑なことにＩＯＣＳでＢＲＥＡＫキーを読むことが出来ない！　他のコマンドを使えば読めるかもしれないが、マトリクスコードが読めないほどだから多分駄目だろう。私はこれでひどい目にあった（というほどでもないか……）。機種の違いをカバーするべきＩＯＣＳが変更されているのはプログラムを組む側からいって非常に迷惑だが、こんなことはよくある事だろう（特に某社では）。

さらにハードの方の違いを出すと、こちらは大したことがない。Ｖ１／Ｖ２／Ｖ４／Ｖ５では余計なジャンパはなく、Ｖ３／Ｖ６ではＧＮＤらしきジャンパが２本、ＲＡＭのＷＥに謎のセラミックコンデンサーがくっついている。さほど効果が無いように思えるが。Ｖ７は手元に適当な実験台がないのでよく分からないが、おもにＥ５５０／１４９０ＵⅡの６４ＫＲＡＭ機らしい。これの中身は、なんと本来６２２５６が有るべきところに、６２２５６が２枚とＴＴＬのちいさいのが載っているフィルム基盤が、あたかも１枚のＲＡＭチップかのように強引にはんだ付けされている。正直にいって、部品さえ手にいれれば素人でもできるお粗末な改造である。コストの面を考えればこんなものかもし

れないが。同じ基盤を使っている以上、市場からＥ５００が消える日は当分こないであろう。ということは、どうせ６４ＫＲＡＭだったら、Ｅ５００を買ってきて自分で改造したほうが安上がりということだ。１０００円ぐらいしか変わらないけれど。ジャンパは確かＶ３／Ｖ６と全く同じ。

おや、何か忘れてるな。なんだったかな？　そうか、Ｅ５００ＰＪというものがあった。あれは確かＶ３かＶ６だったはずだが……、たぶんＶ６だろう。前に調べたけどもう忘れてしまった。あれは中身はただのＥ５００で、ＨＥＡＶＹ－ＭＥＴＡＬ　ｍｉｎｉが最初からＬＯＡＤされている以外は、筐体がくすんだ水色なだけであった……。

もうそろそろネタがつきてきたので、これからの展望について。まずは、だいぶ前から手狭になっているＲＡＭを、なんとしてでも増設する事。現在は、３２Ｋ（Ｖ２のＥ５００）＋６４ＫＲＡＭで９６Ｋだが、今年中に内蔵用２５６ＫＲＡＭをつけ、近い内に２５６ＫＲＡＭカードを製作する。これでようやく５１２Ｋバイト、一息つけるといったところだろう。もっとも、これにはバックアップの面で問題が残るが、そんな事は大した事ではないので考えない事にする。後は特にしたい事はないので放っておくが、場合によっては、実験的に６４０ＫＲＡＭを実装したタイプや、謎の漢字対応機を作るかも知れない。できれば３．５ＦＤＤも作りたいところだが、残念ながらそこまでの知識がないので、当分は作れないだろう。ぐっすし……。

おすすめする本

植岡　孝道

THE CUCKOO'S EGG:Tracking a Spy through the Maze of Computer Espionage
邦題：カッコウはコンピュータに卵を生む　Clifford Stoll著／池央耿訳　草思社

本書は管理者及び善良なユーザーにはsendmailとfingerdとexpreserveとGNU-Emacsにかつてあった穴が自分のマシンで塞がれているかをチェックする必要性があることと、パスワードには英単語を使ってはいけないことを教えてくれる。さらにクラッカーにはユーザーguestやuucpやSYSTEMやFIELDやUSERの有用な使い方を教えてくれる。おまけにすべての人にクッキーの作り方まで教えてくれる。これなら一冊買っても損はしないという気にもなるというものだ。

前半のスピーディな展開に比べ、後半の役所に阻まれて動きがとれなくなってしまってからの部分が多少もどかしいが、読み物としても非常におもしろい。「現実は小説より奇なり」を地でいっている。この本は私のおすすめナンバー１である。

特に著者の同僚によるハッカーの人物像推定法には非常に興味深いものがある。「西海岸でコンピュータを扱う人間は誰だってバークレーＵＮＩＸだ。それが、このハッカーはいまだにＡＴ＆Ｔ／ＵＮＩＸでやっている」というセリフなどは我々をうならせるものがあるし、ハッカーが発行したpsコマンドのオプションを説明しての「残るはｆフラグだけだ。実は、バークレーＵＮＩＸにはこれはない。ｆはＡＴ＆Ｔ／ＵＮＩＸでここのプロセスのファイルを表示するコマンドだ。　…　ハッカーはバークレーＵＮＩＸを知らない、旧世代のＵＮＩＸ信者だ」や、シェルスクリプトを読んでの「ハッカーはキーボード・データの表示に<read>のコマンドを使っている。意識の進んだプログラマーなら<set>とやるところだ」などの具体的な言葉の持つ説得力には反論の余地さえないように思える。（どうでもいいが、この引用部の「表示に」は「入力に」の誤訳ではないだろうか？原文を読んでいないのでなんともいえない。）

メインになっている(?!)著者と政府の３文字機関とのやり取りのほかにRTM's wormについての記述があるのも無視できない。wormの増殖法、発生背景、退治法のほか、インターネットでの顛末などについて２０ページ以上を割いて述べてある。具体的な技術情報こそほとんどないが、日本ではごく僅かしか正確な情報源がなかったことを考えればこれは十分価値ある情報だといえるだろう。

この本は本学生協でも手にはいる（生協に置いてあるのを見た）ので、これを読んで興味を持ってくれた人は書籍部にいって内容を確認してほしい。私がうそを書いてはいないことが判ってもらえると思う。

註：私は草思社の回し者ではない。なにしろア○キーですらバイトをしていないのである。故に出版業界とは縁がないのである。

編集後記

ＩＮＤＥＸのページがｌｓコマンドの出力であることに気づく人はいるかも知れませんが、日付が欠けている理由に思い当たるのは、なにか悪いことをした人でしょう。「昨日から印刷所(プリンタ)止めて待ってるんですよ！」と何度言ったことか．．．。（今日は私の誕生日なのにぃ・ｔｒｅｅ）

「八神君の家庭の事情」　あるいは南極物語

「八神君の家庭の事情」のビデオを見た。途中からみたのでストーリーがいまいちよく分からなかったがなかなかすごかったぞ。

原作より過激になっているとは聞いていたがまさか八神君と五十里さんが行くところまでいってしまうとは思わなかった。

野美さんは野美さんでテレビで歌っていたかと思えばいきなり裸になったりしてこちらもサービス満点だ。そういえば「せぶんじーのおぺれーたー」とか言っていたがあれはいったいなんだったのだろうか？

クライマックスは四日市先生との激しい戦闘。しかし、仮にも教師ともあろうものが不倫したさに兵隊集めて教え子に実弾をぶっぱなすとはなんとも困ったものだ。対する八神君たちも武装バイクを盗んだりする凶悪ぶりだ。凶悪といえば一ヶ谷君たちはいつの間にあんなアブナイ不良になってしまったのだろう？一枝ちゃんなどすっかり太ってほとんどダンプ松本である。

挙げ句の果てに東洋人Ｘのムチで街中壊滅して終わりとは、最後まで気を抜けないとはこのことをいうのだろう。

それにしてもエンドクレジットに楠さんの名前がなかったのはいったいなぜだろうか？

（こいつが手を出した百万石はすべてバカになるといわれる男・ｚａｋｕ）

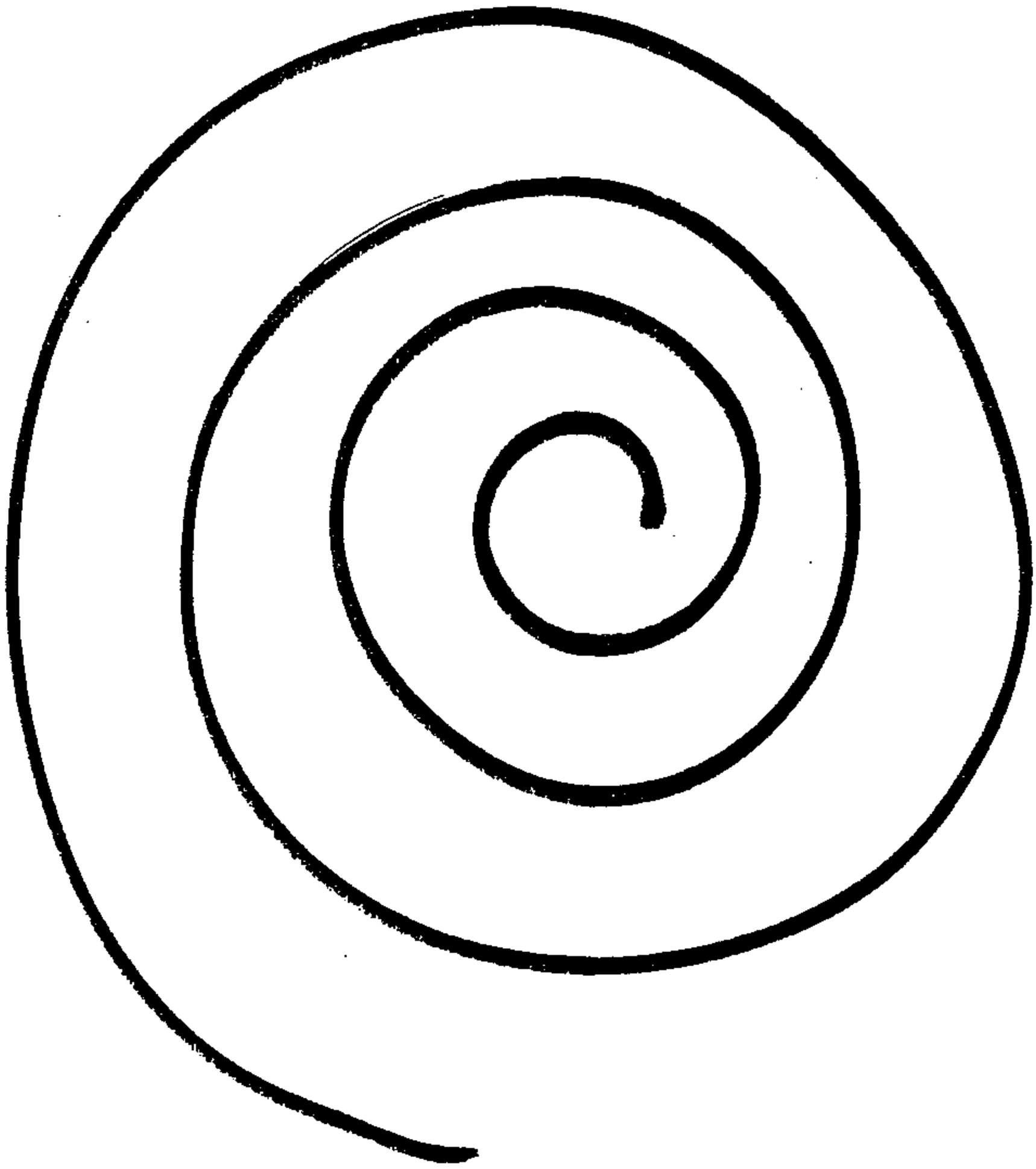
飯田さん飯田さん、編集後記です：ひゅみゃ、ねみゅい…。　あ、灰になってしまいましたね。

ふふふふふ、気がついたら表紙を書いていた。本来ならまだ表には書かないはずだったような気がするけど。まあいいや。それにしてもこう忙しいとゆっくり基盤を焼いている暇もないなぁ。ああっ、なんだこの基盤はっ！　焼きそこないではないかっ！　肝心なところが溶けないくせに配線が溶けてるではないかっ！　これでまた焼きなおしだ～！　というわけで私はこれから部室(あんしつ)にこもって基盤焼いてきます。（ＤｙｎａＢｏｏｋのキーボードが大きすぎると思うＰｍａｎ）

おくづけ

発　行：電気通信大学ＭＭＡ　　発行日：１９９１年１１月２３日
印　刷：バブルジェットプリンタ様　＆　サンチェーンのコピー機様　＆　学友会室の印刷機様
連絡先：調布市調布ヶ丘１－５－１　電気通信大学内　ＭＭＡ

乱丁本・落丁本は「当たり」ですので、もう一冊差し上げます。



すいこまれるぞ