1990 11 創刊2号

Super 百万石

MMM

MMAとはMicro computer Making Associationすなわちマイクロコンピュータを作る会であります。今から十数年前まだワンボード・コンピュータが全盛で、市販のコンピュータが高価であった頃、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。

そして現在に至るまで数多くのソフトやハードが生み出され、より快適な計算機環境をめざして活動を続けてきたわけです。

しかし近年、巷では高速で安価なワークステーションなるものが出回りはじめ、それらのパフォーマンスに比べると我々が日夜使用しているメインシステムはおせじにも快適とは言い難いものになってきました。市販のワークステーションに対抗するようなコンピュータをアマチュアレベルで製作するには多くの技術的問題があり、かなり困難なことであると思えます。

「買えないから作る」は裏を返せば「買えるものは買ってしまおう」であり、適当なワークステーションを手にいれ、その上で活動を続けるのも１つの手段でしょう。しかし「買えるものは買ってしまおう」は「売ってないものを作る」ことでもあります。

MMAは今まさに大きな岐路（危機？）に立たされているといえます。

でもまあどうころんでもMMAはさらにより快適な環境をめざしてがちゃがちゃやっているでしょう。またこれまで受け継がれてきた手作りの良さは大切にしていきたいと思います。

なお、本誌「百万石」は季刊の予定でありましたが、この第２号が創刊号の発行からすでに１年経っていることについて深くおわび申し上げます。

μEMACS　Ver３．１０の移植　　橋谷貴光

今回私がした仕事は、最近パソコンなどでもときどき見掛けられるようになってきた高機能エディタのμEMACSバージョン３．１０というのものの移植です。

このエディタはＵＮＩＸなどの上で動くＥＭＡＣＳというエディタが元になっていて、そのＥＭＡＣＳのある程度の機能を抜き出してＣで作り直した物です。もとのＥＭＡＣＳはＬＩＳＰでかかれていて、またあまりに高機能なため、非常に大きく、コンピューターへの負担も重いものでした。

移植作業は難航しました。ターミナルへのＩ／Ｏのコントロールがわからなくて、結局先輩が昔移植したＶｅｒ３．９のＩ／Ｏ部分を参考にして（そのまま使ったという話もある）移植しました。

相手が自分の実力を越えて大きかったためプログラムがなにをやっているのかを理解するのも一苦労でしたが、いろいろ勉強になりました。

ＣＡＴ　　渡辺雄一

ＣＡＴというコマンドは、ＵＮＩＸを初め多くのＯＳ上で見られるコマンドなので、特に説明をすることは省くが、ＭＭＡのＡｒｅｓのＯＳ／９にのせられているＣＡＴは、Ｃ言語で組まれているものなので、それを高速化させるという目的のために筆者にはアッセンブリ言語で書かれたＣＡＴコマンドを作成するように課題が与えられた。

そして、それは同時に筆者に６８０００のアッセンブリ言語を学ぶ際の非常に役立つ良い手本を示すことにもなり、さらにＯＳ／９のシステム・コールについての知識や、実際的な利用法も分かるようになるという、まさに一石三鳥もの意義がある課題であった。

もっとも、それはあくまで筆者の立場から見た考えであり、他のＭＭＡ部員が課されたプロジェクトと比べると、かなり見劣りのするものであると思われる。とはいえ、以前このＭＭＡというサークルに入るまでの（入部後もしばらくそうであったが）Ｚ８０をＣＰＵとして持っている８ビットマシンのせまいメモリ空間上で、ちまちまとその比較的不整合なアッセンブリ言語の命令体系によるプログラムを作っていた筆者にとっては、非常にとまどわせることの多い課題でもあった。

だが、ここで先に感想を述べておくと、やはり俗に６８系と呼ばれるＣＰＵは、特に６８０００ともなると、非常に整った命令体系と周辺環境を備えていて、その設計思想の延長線上にあるＵＮＩＸ－－ＯＳ／９のすばらしさ、使い易さがよく理解することができて良かったと思う。

これを機会に、筆者もこれから６８０００のアッセンブリ言語に慣れ親しみ、より高度なプログラムを開発していけるよう、努力していくつもりである。

ｐｒｅｓｓとｍｔａｃｈｏ、そして愉快な仲間たち　　楯岡孝道

ｐｒｅｓｓは、名前の通りファイルを圧縮するプログラムです。ことの起こりはある日、某先輩が「ハフマン圧縮を書いてみない？」と持ちかけてきた事でした。その後、何回かの試行錯誤の末にプロトタイプができあがりました。内容としては、圧縮方法としてハフマン符号化を採用し、２パス方式を用いて、１パスとして符号化テーブルを作り、２パス目で実際に出力するというものです。

ところで私は知らなかったのですが、ＡＲＥＳには既にｕｎｉｘのコマンドであるｃｏｍｐａｃｔとｃｏｍｐｒｅｓｓがインストールされています。それぞれ、ハフマン符号化とｌｚｗ法を用いた圧縮プログラムですが、その存在を知って、さっそく性能比較を行ってみました。

結果は散々なものでした。圧縮率では、圧縮法の関係から勝負の見えているｃｏｍｐｒｅｓｓに負けたのはともかく、同じハフマン法を使っているはずのｃｏｍｐａｃｔにもわずかとはいえ負けてしまいました。さらに速度の面でも両方共に負けてしまい、ｐｒｅｓｓはその存在意義を全く無くしてしまったのです。

その後私は、「肺が潰れる」という病気にかかったりして、ｐｒｅｓｓの呪いをひしひしと感じたので（余談ですが、肺の圧縮率は高かったです）、いま、「ｃｏｍｐｒｅｓｓ」に勝てるプログラムとして発展させるべく拡張を繰り返しています。が、ハフマン圧縮さえ資料片手に試行錯誤してやっと組み上げた私が、いつ、新ｐｒｅｓｓを完成させるかは、誰にもわからないのです。

さて、ｍｔａｃｈｏは、ｕｎｉｘ上のｘｔａｃｈｏという計算機の負荷状況をタコメーターのようにグラフィック画面上に表示するプログラムを参考にａｒｅｓ用に作成したものです。１秒もしくは任意時間毎に１回起動し、負荷状況を表示します。

ａｒｅｓでは既に、ｐｅｒｆｍｏｎという負荷記録を表示するプログラムが存在するのですが、１種類しかないのも芸がないので作ってみました。またその他にも思いつくまま、いくつかのメーターを作ってみました。ｄｐｍ，ｂａｒなどがそれで、ｄｐｍは数字で表示し、ｂａｒはステレオなどのデジタルメーターのように棒の長さで表示します。

また、同じ関数を用いて、ｂｅｓｔという最も負荷の重いプロセスのｂｅｓｔ３を表示するプログラムも作成しました。これは、某空手男が「ｂｅｓｔでなくｗｏｒｓｔと名乗るべきだ。」と言った事からも想像できるように、「今重いのは、何のせいだ？」という素朴な疑問に答える為のものです。しかし、プロセス名とともに、ユーザー名及びプロセスｉｄの表示も行うので、部内の評価が、「これは便利だ」と「極悪だ」とに分かれています。

このタイプのプログラムは、自分自身が重くてはいけないので、１回の表示が早くなるように、いくつかの工夫がしてあります。そのひとつとして、前回の表示結果を記憶しておいて、同じものは表示しないようにしてあるのですが、このために実行中に画面を書き換えると正常な表示ができなくなるという問題が起きました。

この問題は現在では、ｐｒｉｅｓｔさんの提案によりプロセスにシグ

ナルを送ると画面を書き直すようにして解決しています。
それでも、シリーズすべてを同時に動かすとｃｐｕタイムの２０％近くを消費してしまうので、普段から動かしておくのには問題があるのは残念です。

プリンター・スプーラーの作成

飯田　卓

私が右も左も分からないＣ言語の世界に足を踏み入れてから、およそ一月が経過したころのことです。とある先輩が「ん、課題はだいたいできたようだね。じゃあ、今度はこれを作ってみて」と、ＵＮＩＸ上のコマンドである”ｌｐｒ”のリファレンスを差しだしました。「これには、まず”ｌｐｄ”というデーモンをつくって……」その後の話はきちんと覚えていませんが、当時の私が理解するには少々難しい内容だったことは確かです。それ故、このプログラムは半年ほどほっぽからしにされていました。
そして、調布祭がまじかにせまった十月。さきの先輩に「できた？」と聞かれて、ようやくこのプログラムの存在を思いだしました。（いくらなんでも調布祭までにできてないとまずいだろうなあ）と考え、あわててロジックを組み立てたのですが、実際にはそれほど難しいものではなかったようで、結構簡単にできてしまいました。
プリンター・スプーラーとは、プリンターに出力するデータを一時的に蓄積し、それをプリンターへ専門的に出力するプログラム（あるいは装置）のことと考えてください。コンピューターがプリンターに何かを出力している間、普通は人間がコンピューターを利用することができません。この様な不便を解決するために、スプーラーというものが必要とされるのです。
私の製作したスプーラーは、三本のプログラムによって構成されています。

１．ｌｐｄ……実際にプリンターへデータを出力するプログラム
２．ｌｐｒ……出力するデータをキューに登録するフログラム
３．ｌｐｑ……現在のキューの状況を表示するプログラム

”キュー”というのは、プリンターに出力するデータの順番を記憶しておく部分（ここではファイル）で、複数の人が同時に一台のプリンターを使用する場合には必ず必要となってくるものです。
このプログラム、確かにロジックは結構簡単なのですが、途中で「二段組み印字をするようにしよう」と考えたために複雑怪奇なものとなってしまいました。そこで若干悩んだだけで、あとはすんなりとできた様な気がします。

a t　&　c r o n

田口景介

atとは指定された時間になると、指定したファイルを実行する為のコマンドです。具体的には、

```
at time [date] filename argument ...
```

とすることで、filenameが実行されるわけです。dateは曜日の指定も可能で、毎週一回実行するといった指定も可能です。但し、現時点ではまだサポートされておらず、使用出来ません。（現在、時間の指定のみ使用可です。）以下の文章では、現在予定されている機能を含めて説明します。

atを使うためには、実際にファイルの実行等を行うcronというdaemonを走らせておく必要があります。普通はシステム立ち上げ時に実行されます。この時データファイルを読み込み、前日より持ち越しのデータを登録します。登録されたデータの中に実行時刻が過ぎている物があれば、この時実行します。

cronは次のデータを内部にもっています。

実行時刻・実行日
登録した人のユーザーID・実行ファイル・パラメータ

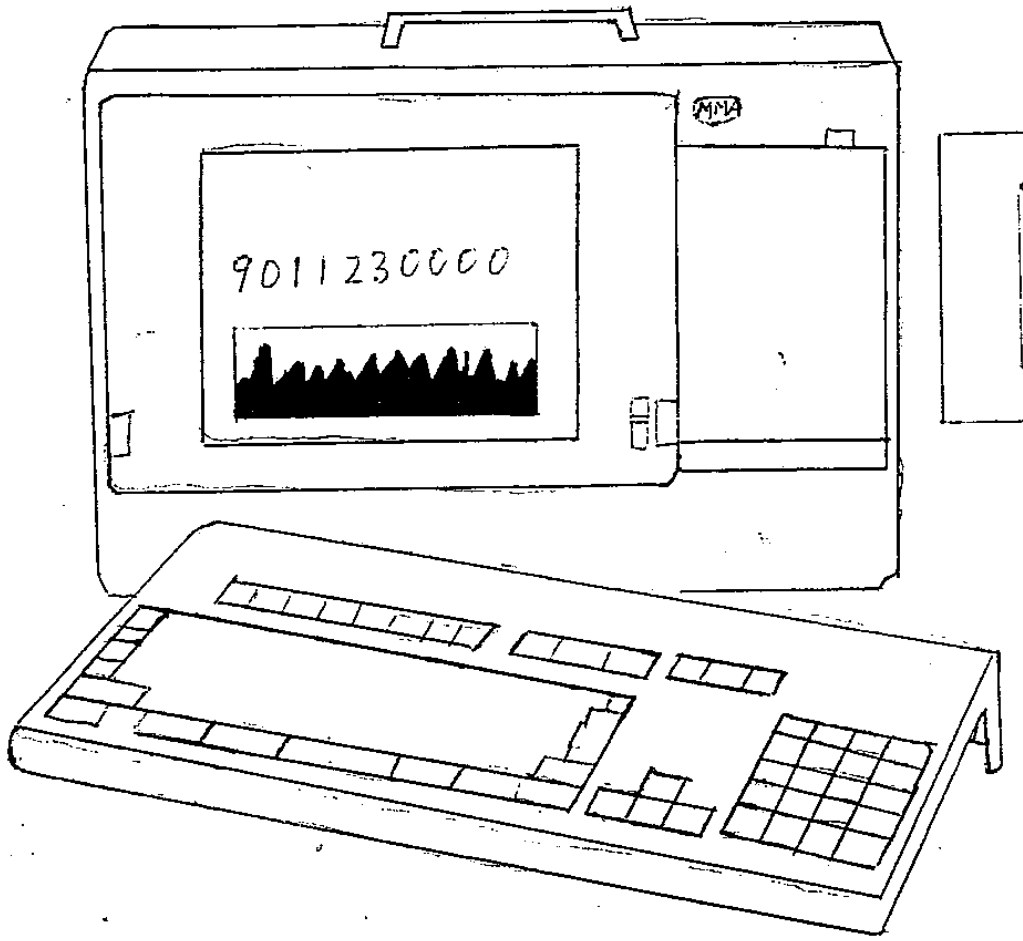
cronは登録されたファイルの中で、最も早く実行するべきファイルの実行時刻までsleepし、時間になったら実行ファイルをシェルを利用して実行します。この時問題になるのが、cronはrootで実行されているため、cronによって実行されるプロセスもrootで実行されてしまうということです。シグナルは同じグループIDを持つユーザーに送れるため、誰にでもatは使えてしまいます。これをun*xではどうしているのでしょうか。

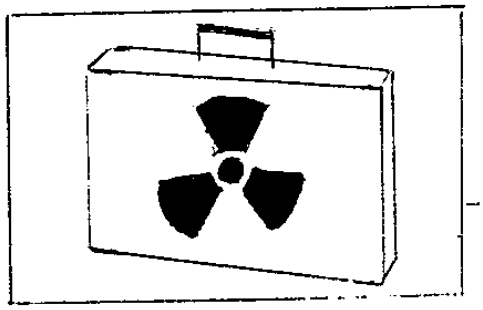
atは必要なデータをデータモジュールに書き込み、オプションによって異なったシグナルをcronにおくります（オプションは登録、削除、一覧表示）。データモジュールにアクセスする時はイベント処理をしないとまずいのでしょうが、今のところ何もしていません。イベントについては今後の課題です。シグナルを受け取ったcronは、シグナルによってそれぞれの処理行い、またsleepします。そしてシステムをダウンさせる時に、登録されたデータを次にたち上げる時に読み込むためにファイルにおとします。

このcronは、ある程度つくってからun*xのcronの仕様を知り、ずいぶんと違った物であることを知りました。そこで、よりun*xに近いものに作り替えようと思っています。もっとも、通常のun*xと違い、ARESでは毎日電源が落されますから、いくつか変更が必要ですが。

SuperHYAKUMANGOKU VOL.1 #1 NOVEMBER 1990 ISSUE 1

OS/2 省スペース・ステーション、誕生。





MMAパーソナルシステム
HANDY Ares

定価：0円（本体：0円、非売品）

雑誌68040-33