

'83 in fall

Lime

Limited' Expressin
of
KTTU Computer Club

No. 5

metugi

- Page 1 COMPUTER部員が COMPUTERに
無関係なバイトをすることは無意味か
take
- Page 3 MULTIPLE , MICRO
PROCESSER S. Takeoka
- Page 17. 初冬 深夜の座談会 Editor
- Page 19 DATA SHOW 83 物見遊山
竹田
- Page 21 キーボード?? Takeshi
Kishida
- Page 22 @のページ @
- Page 26 部長のおこぼし M

はみだし もんくたれ

はたして

「COMPUTER 部員が、COMPUTER に
無関係なバイトをすることは無意味」
か?

Take

- ▶ Tom が上記を前号の Lime に書いてあって、「どう
かなかな問題じゃ」と思っていると、色んな女から
「これ、ほんまでっか?」と尋かれるので、一言文句を
たれとく。俺の結論がいうと、「全く逆じゃ」。
- ▶ Computer をやる女は色々知っていると、正しく
System 分析がでけらんから、あちこちに首を
つっこんで、様々な雑学を身に付けた方がええ。
(例えば、建築の見積りとか、縫い物のデザインとか???)
- ▶ しかし、自分が何者で、何のために知識を求めているか
を、いつでも持っている心、仕事自身に押し流されて、
たおの金もろけたけに、終ることになるだろう。(それは
それでも良いのだが) ▶ また、せめて仕入れた知識が
ぼろぼろでも、単なる器用なだけすぎない。学ばば
驚かねばならない。(おもえば学ば) ▶ 恐れるべきは
単なる器用貧乏で、もし豆頭の中で、各々の知識が一つの
系をなすべく、常に努力するなら、なるべく計算機以外の
仕事をねがが良い。誌上にて反論を待つ。(Jun/25/1983)

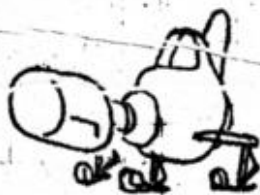
はみだし もんたれ

▷ 新入生の中では、木村様語を知ってる女も居るし全然
 知らぬ人も居るようだ。▷ 木村様語知らぬ君、興の起
 持つのも良いが、ハッハハ、木村様語本人かやめとしておなはれ。
 ▷ もともと BASIC みたいな IP- は、木村様語といっしょやし、人なれ
 やつて、ええとない。それより BASIC を骨までやぶる方が、
 よほど、バンキューになる。▷ 物事万事、極めるということが
 大切だバ。もし、木村様語をやつてりゃったなら、LISP か
 PROLOG やんやさい。PASCAL や C や ADA は、精神構造に
 革命を起こしにくいから、勧めぬしな。

1980年代、人類は
 宇宙から侵略のキョーイに
 ささかれていた。



UFO



SHADO

ぼくは、この番組を、小学校4年のときに
 み、感動はしました。ほんでもて、コンピュータを
 ずえたい、自分のものにするおっ、と決ま
 したのでした。ときに1970年でした。MicroProcessorは
 まだ、この世には、ありませんでした。

MULTIPLE
MICRO
PROCESOR

20/10/2020

実践的マルチプロセッサ・システム

- ④) 新しくは、データ・FD-マシンやRISCに代表される様な、全く新しいアーキテクチャに基づき、様々な形態のマルチプロセッサ・システムが考案されている。しかし、ここでは、COMPUTERの誕生以来、比較的よく行われて来た、昔ながらの、マルチプロセッサ・システムを、具体的にどう作り上げるかを、解説する。多少、散文記取有るかも知れないが、色々な角度から分析する。
- ここに、ある実例の一部は、System House 各社の開店日は、Takeo, Koma 各社に、KMLの坂下氏、田中氏で、飯を食いながらしゃべった時は、田中氏より、うかがった話しである。(1981年の秋?)
- 当時、マスター・田中は、Z-80 が2つ載ったボードを10枚近く、制作され、ラックに全部入りきらないので困ると、おっしゃっていた。正確にそのSystemは、RAM16K、2つのZ-80で共有し、そのボード4つを別々のZ-80、1台で管理するということだったと記憶している。しかし、Supervis Z-80のMonitoring Programは、どこにあるのだろうか???

1) Multi Processor を巡る 諸 問題

- ▶ 普通、Micro Processor による Multi Processor では、限定された用途に投入されることが多いので、Hardware 上の問題は片づけばほとんど仕事が終わったといえる。しかしながら、我々アマチュアが汎用マシンとして手がけた場合、どのような使い方をすればそのマシンが生きるかは、大きな問題である。例えば、1つの CPU を BASIC の処理系に割り当てる。しかし、Single User の System ならば、残りの CPU は遊んでしまふ。それならもう一つの CPU は、Printing Spooler にすれば、良いと思うだろうが、Micro Processor と、いう状況から判断すれば、Spooler には、専用の Computer を別に作り、Main の Computer とは、そ結合する方がコストパフォーマンスが良い。
- ▶ FM-8 の事を、Multi Processor だと思っ
ている奴が多いのだが、あれは、完全に分散処理を目指して作られているので、普通は、マルチプロセッサとは、いかない。ただし、コミュニケーション用の Buff. Area に 128B の共有空間があるし、こちらから、CRT 用の CPU に、任意のプログラムを転送して実行させられる。そのため、マルチプロセッサ風のことできる。
- ▶ それでは、どんな物を Multi-Processor と呼ぶかという、これは、難しい。厳密な定義は、全く無い。しかしながら、お互いが、重きながら参照できる共有メモリ空間が、4KB を越えるあたりから、マルチプロセッサシステムと拮抗なく呼ばれるようだ。

- ▶ ちがいに Koma 先生などは、64KB 完全に共有でよいとおもわない。D.T. のたまわぬこともあるが。(拙者も今では、そう思うこともあるが) 実際問題としては、Working エリアは各 CPU 別々に持つ方が、より有利であると思われる。端的な例をおげれば、BASIC のインタプリタを、数個のプロセッサで共有すれば、各 USER の BASIC テキスト スクラッチパッドはプロセッサの残りの空間全部となり、マルチユーザ-BASIC システムが、簡単にできあがる。(この場合、I/O は、1 CPU に、1 組づつ、ちゃんと分けねばならない!!) こうすれば、REENTRANT とか、セマフォとかは、一切無関係。
- ▶ FM-8 風、分散処理もコストパフォーマンスから行くと、すぐれて 113L、PC-8001+PC-8031 にも、それは見られる。また、処理するプロセッサ(変数の)をたくさんを結合して、NET-WORK アーキテクチャ風、プロセッサの数を増やすのも、それなりにコストパフォーマンス良く効果は、上がる。
- ▶ しかしながら、マニアとしては、やはり密結合タイプの魅力には、たれも月替わらないだろう。私めなどは、高校の時に、6800 が簡単に Multi Processor できるのを知った時(実は、まだ中学の弟に教えてもらった)から、ずっと使い方と、管理の仕方を考えてきたのでありますが、1 user 1 task のものを Multi Processor で動かすのは、容易では無く、例え BASIC のインタプリタを、マルチプロセッサで処理するとか、マルチプロセッサ用のオブジェクトを出力する、BASIC コンパイラなどを高校の頃から考えては、おるのですが、一向に進展をみせぬ。上に書いたマルチユーザ-BASIC などは、高校の 2 年の時に考えた、秀作です。

- ▶ マスター田中は「何やってる人ですか」と尋ねた時も「まだ
 出てきてRAM TEST」と答へられて「おあや」と思った。
 「しかし、お口の木探素でもやらせようか」と思って作った
 人で村。森田氏(ASCII誌で有名)によると、お口の木探素は
 5Levelくらいだと十分らしい。それを各プロセッサに割り当てると
 ちょうどいいくらいみたいだから」という続き。うーん、なる程
 Koma先生とは、大分ちがって、ちがって、わかたらしめる。▶しかし、
 ながら、私しめや、Koma先生のは、本質的に問題はちがう。
 東大や京大のセンターでは、汎用のMain Frameが、2台密結合
 して、それが2組を結合するとき、血がさわぐ人です。
 ▶最近、どうもMain Frameも、つまり、こと、か、いて、
 というのは、解決するのだが、それでも、三つの魂百まで
 とか、言って、今、これ、マルチプロセッサを1チップに適用する方法を
 考える。▶ぐちみたいになつてきたが、やはり世間の主流は、
 Graphicや、自然Simulation専用のSystemを作ることにし、
 ▶また、PrologやObject Oriented Languageは、少し
 違った意味で、Multi Processor向きなのだね。(拙者の
 「Multi Processの本質的参照」)。Prologなどは、Back Track
 しなければ、非常に単純なのだが、また、副作用も、問題は
 ある。SHARP-UPのMeta-Logicなどは、論外である。
 ▶この5月の連休に、秋葉原で買ったKoma先生のBO9Eは
 Prolog Machineに在るという。うかさだたが、MMUの導入によつて
 例によつて発表されたまじらしい。有らして、Multi Processorに
 メモリのダイナミックプロセッサが加わると、問題は非常に

Page 8

おれ又なるので、Koma 先生が、どなたもよく楽しんで(苦んで)いるか、よくわかる。

▶ 22で、もし BASIC コンプライを CPU 2 個向きに作るとしたら、

$$A = (B+C) * (D-E) \quad \text{など、() の中を、2 個の CPU で}$$

別々にやらせてその答えを、乗算したい。もし 2 つの CPU の実行が同時に終われば、問題は無い。しかし、通常は結果が出そろって同期能とらぬ気がするので、オプティミケ率が悪い。また、

$$10. A = B * C$$

$$20. B = A * C$$

など、10 行と 20 行を、単に、別々

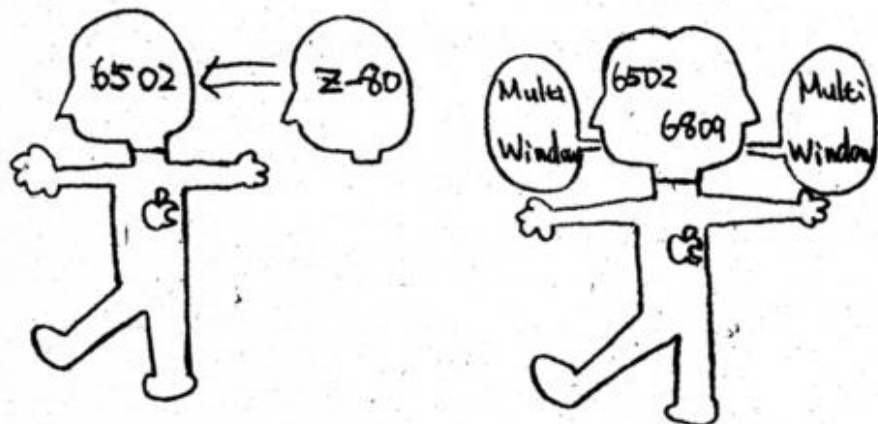
Processor に割り当てるだけでは、だめだから、難しいね。

▶ また、これは、そのうち Multi Processor のお勉強が、やるつもりだから、ぼちぼち考えとておくね。

▶ そろそろ、退屈しはじめたので、実践へ。と、つぎは、

▷ おまけ

▶ Hu-ENGINE は、Z-80 が完全に、L-3 上の 6809 と同時に重くか、単に I/O Port でしか、つながっていない。Multi Processor じゃない。▶ APPLE の Z-CARD や TOMATO も、6502 を止めちまうので Multi Process でさえない。▶ APPLE の MILL は、6502 と 6809E が、同一 BUS 上で、ちゃんと重くという、驚異の Multi Processor System。



2) 68系 BUS 上での Multi Processor

- ▶ 68の BUS で Multi Processor やるのは、マイン少年の間では、常識というもんだ。それという時、68系は $\phi 2$ (or E) という、Clock 同期に外部とやりとり。うらがえせば $\phi 2$ では BUS は 空いている。そして、 $\phi 2$ というのは、 $\phi 2$ と同じ期間だけあるだけ ($\phi 2$ は T_{cycle} 50%)。つまり、 $\phi 2$ に、もう一つ 68 の CPU の $\phi 2$ できる時間がある。ということは、一つの CPU に $\phi 2$ を入れた時、もう一つの CPU に $\phi 2$ を入れれば、それでできあがりである。(実際は、 $\phi 1$ と $\phi 2$ を入れ替えて入れ、また Bus Buff のために若干の回路が必要である)
- ▶ このことは、十数年も前から、あたり前である。

田中氏: 「昔、友達に 6800 * 20 の Multi Processor をした奴が、いましてね...」

田中氏: 「昔、友達に 6800 * 20 の Multi Processor をした奴が、いましてね...」

Takeo: 「ひょっとして、逆相 CLOCK とか...」

田中氏: 「そう、そのとおりですよ」

Takeo: 「しかし一般に、マルチプロセッサの同期をとるために必要なセマフォ書き換えなどの非可分手続きは、どうやる...」

田中氏: 「ところが、プロセッサが、特に 2 つだけのときは、簡単なんですよ。」

Takeo: 「もしかして、HALT だとか...」

田中氏: 「そうです」

Takeo: 「なる程」

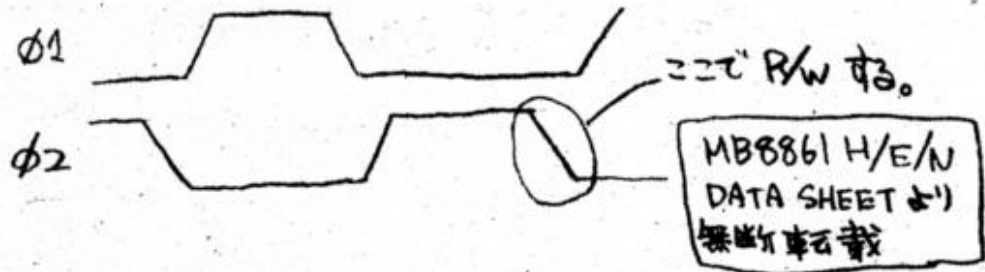
一同: 笑い。

▶ Takeoの、互に逆相の意味は、「2つのCPUが逆相CLOCKだとプロセス間の排他処理が、インストラクション単位では、不可能なのでどうやっているのか」と、きいているのである。その問いに対する田中氏の答えは、「プロセスが2つしか無い時は、片方を完全に止めしきっても問題は起きないのだよ」と言っているのである。それに対するTakeoの言葉は、「さては、排他処理時はCPUを、完全に止めしきっている人ですね」ということである。68のHALTは、8080のHOLD(2-800BUSRD)で、CPUが完全に止ってしき、Hard Wareの端子である。最後の一向:笑いは、ごく普通のSystemを、ごく普通に作るための安心感である。

▶ と、いう風に、68系のBusだと、それがどこで作っても、ほぼ同じものが、ごくあたり前々でさ上がるから、少しかべるだけで、そのSystemの全ほうがわかってしまうというわけですね。

▶ また、KOMA先生の実験則によると、VMAは、めったに出ないので、同相でも、かなりの交り率で6809は、動くそうです。

▶ しかし、6502にはVMAがないのに、なぜMILLは動くのか? CLOCKの裏はAPPLEの土場台、V-RAMだから、糸対無理だと、思うんだけどね。ひょっとして、6809のVMA有った時だけ6502を動かせるのかね。



3) 最近の Micro Processor と、再び諸問題。

▷ Multi Bus (IEEE-796) は、16~32 bit Processor に
対応して、マルチプロセッサのための制御線もある。
大昔(もう5~6年になるかな)から有名だが、やはり、

Main Memory を共有する程の密結合では、全然 Timing が
間に合わない。2つのプロセッサのバッファリングエリP. 程度を
共有する程度でないと、プロセッサはほとんどとまっている時間の
方が多くなってしまふ。

▷ 2-8K にも、リソース(共有資源)管理用の信号線が
あるが、あれも、あくまで、I/O や、めったにアクセスしない、バッファ用
メモリなど、低速でも許せる場合のリソース要求にしか
使えない。もし、我々が考える様な、メインメモリの大部分を
共有する System には、あんなまじろこいものは使えない。

▷ 68K なんかは、8085 をごっかくしたよーなもんだから、
マルチプロセッサなんか、頭にない。メモリ管理さえ頭にな
なかったのだから何を言わん。モトローラのP-キテク物は
いっつも一世代前のものなんだね。(メモリ管理のないう分、
Single User Single Task の System では有効にはなっているが、
これは明らかな、怪我の功名というやつだね。)

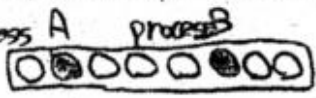
▷ メモリメモリと、なぜメモリの大規模な共有にこだわることかとい
うと、それは、現在の Machine ではメモリの使用が一番多い
からだ。それに、I/O をメモリ上の Image にとらえるのが一番
効率良いからだ。

▷ 例えば、Super Personal Computer の Bit Map Display があるが、
今で言うところの Video RAM なのに、Main Memory 上にあることか

望ましい。(と、いうよ) Memory空間上に自ければ"実用的な速さで Handlingできる)。次に LISP のガバージコレクション。LISP系の Processorは CELLを使いねにねのて、別なプロセッサ(ガバージコレクタ)が、CELLをリフレッシュしてやうねねねねね。それで、2つのプロセッサが同時に、同じメモ空間を共有したいのね。

DBit Map Displayを、2つのプロセッサが同時に Handlingできると、實際、画期的なねねねねね。単に直線を引くねねねねね、2つのプロセッサがねねねねねの早さねねねねね。例えば、DOMAINも、Displayの Handlingは、しどかていた。J-STARも、ちよとて感じねねね。今年の新産 XEROX-SIP (Dolphin)も、えらいとてねねね。

昔、MB-6890 (BASIC MASTER Level-3)を、6809+6809Eで、マルチプロセッサにするという話か。Kama, Takeの間にあてねねね。是非とも、画面を、2つの Processorで Handlingしたいよとてねねね。希望があてねねね。この時、"1 Pixel せつするのね、いちいちセマフォアックや、I/Oアクねねね、できねねねのね、メモアクねねねの Handで 解決するの"という方針の上で、おてねねねのねねねねねねねねねねね。

例ねねね、2つの Processorか。下のように、1つの Byteの 2つの Pixelを
process A process B
 せつする時、いねねねのように、Aが、その Byteを READしたおて、Aが Pixelをせつして、その
プロセス WRITEする前に Bが READすると、仕事をしじると
いう話。

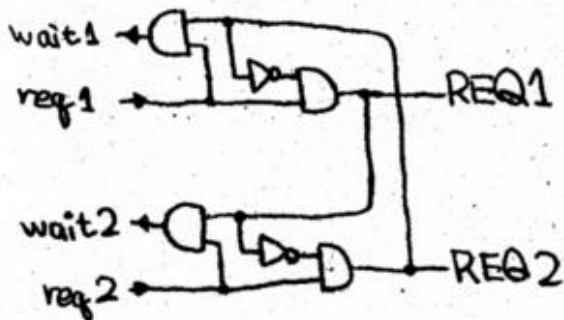
▷その解決法は、どれかのProcessorが、あるByteを読み出した後、そのProcessorが、もう一度メモリをアクセスするまで、そのByteの読み出しを禁止するということだ。これは、Hardware的にも簡単で、非常にエレガントなものだと思っております。

▷まあ、Bit Map DisplayのHandlingは、Windowの重なりがある際、もっと上位のレベルでの判断が必要なのですが。

▷最近のMicro Processorでも、メモリの共有は、やっぱり昔のProcessorと同じだけ、手間をかける必要で、結局は、1/10のアクセスは、O.S.がさみでやるんで、結局は、昔の6800やZ-80でも、同じようなものですね。それなら、安い昔の8bit Processorをたくさん投入した方がおもしろいSystemが組めるんですね。

▷APX432は、GDP 57まで1Systemに組めるように設計されているが、昔風のマルチプロセッサ方式は、はなからないので、また別な機会に、またお話しします。

4) Z-80 2. やってやるよ.



© copyright 1983 by M. MATSUDA

と、いう回路でよい。ただし、2つのCPUのCLOCKは、逆相になるようにして req が全く同時に起こさなくする。REQ で Bus Buff. の Tri-State や、Multiplexer のコントロールを行なう。REQ はそのまま REQ に使う。req は MEMREQ と IOREQ の OR でよいはずだ。ただし、WAIT が inactive にならないうちの CPU のデータの取り込みや出力が問題になる。特に、M₁ のインストラクションの取り込みは WAIT がはずれた直後に起こるので、wait が inactive にならないうち Data Bus 上にインストラクションが確定するまでの Set up Time の間だけ、WAIT を active にする様な DELAY 回路が必要になるだろう。通常のデータの READ, WRITE は 1CLOCK の余裕があるので、普通の Static RAM を使えばいいだろう。DRAM だとアクセスがめんどくさい。IOREQ は確認していいが、速いメモリアクセスなら M₀ 以外は、この回路のまじりゃいいので、あとは Buff. をつけるだけで、カンパキ!! M. MATSUDA 君にけーぶくはー。と、いうこと。Z-80 はこれで Multi-Processor できる。(ただし、2コだけ)

5) おわりに

これももて、昔の、ひとは、 $\langle 3 \rangle$ してマルチプロセッサ
して右のでした。しかし、これ、Micro Processorが出現
してから手軽にできる、考える人も増えたのでした。

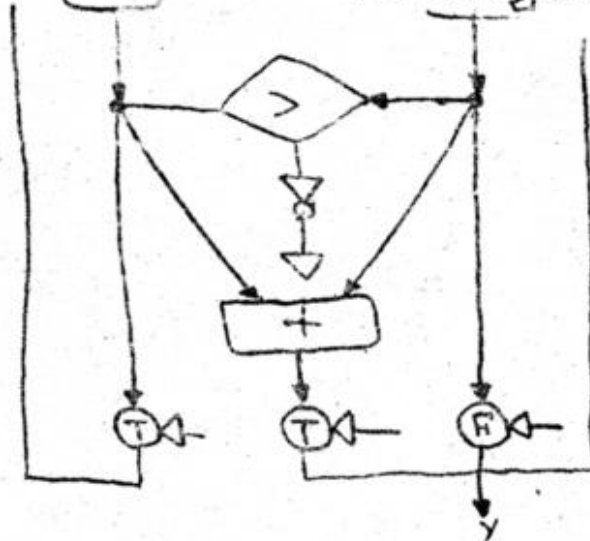
けれど、Multi Processorには、Micro Processorが、
向いてないものもたしか有ります。なせ、メカ、おすめ回路
以外は、なかなか難しくなるのでからね。

Memoryの大量な共有は、今風VLI Micro Processor
でも、考えられていないことは、前述のとおり。

まあ、君達 Hardware Hackerも、なまバグテくらえ。

(完、19/Nov/83 Takeshi)

まっ、まるちる、3世すのおばんきょーの
げんこーがおちぞーじゃ。
『ごーた・ふるー、ましん』なんてきらいさ!



project M:IC
もてたか、
DATA FLOW
MACHINEの
ちが、Object
Oriented ちが。

Take

ぼくのお勧め Machine

DEC の Professional-350

これ、すごい

- MEMORY が 4MByte まで 11ヶ所。
- MEMORY が 512KByte 標準実装。
- 10MByte HardDisk 標準実装。
- 5 1/4 inch FLOPPY Disk 2基標準実装。
- Bit Map Display で、漢字も出る。
- P/OS は、RSX-11 コンパチ。
- Tool-Kit は、C compiler もついで、RSX-11 と完全コンパチにする。
- Bit Map Display も 64色もできるようにする。COLOR は オプション (漢字もOK)
- 日本語エディタは「JEDI」というのだ。
- 日本語ワープロのセッサは「LEIA」だぜ。
- たった 220万円 で、上のもんが買える。
Tool-Kit は 30万円。 やっやすい。

☆ ぼくは、DEC の、まわしもんではないです。

LANGUAGE (Prolog) ←
(L.(I.(S.(P))))

'83 初冬 深夜の座談会

討論会 08. 吉田 あつし

4回 田中 Tom

3回 Editor

見物者 3回 丸本先生

序論

実は座談会の書式で文章化しようとしたが、録音していなかったの、まとめで御基弁して下さい。

本文 (突然 はじまる)

今、アメリカでは IBM の PC-5550 が、APPLE の Lisa を追い越しているようです。(と話を進む)
 とこで、日本語の処理には文字コードに依存しないものが必要なのだろうか? 中国語の処理システムが完成すれば、日本語の処理なんかは簡単に済ませられるのではないかと。しかし中国語の発音は同音の異なる発音の組み合わせが多岐にわたる。 (ここで話の地理へ)

Page 18

一年に平均2°C世界の気温が上昇すると、百万都市下。残るのは中国のチョン（くらいなあたり）うか？（氷びたしになつてけう）中国は人口が牛ほといふ（非常に多いる）の下 植物の品種改良も人海戦術下 とうもろこしと麦の混ぜたものを作つたりしている。ホムトというホトとトマトを混ぜたものがあるが、生産性も高くなくあんまりもうからない。もうからない話としてもう一つ、ある学者が父親の建てたビルの日照権の目録を解消するため、二人の子アデアを出して実行してみた。

ビルの屋上下集光して、その光をファイバー室内に引き込んだ。しかし被害をこうむる方は、この設備にかかる費用よりも、もっと安い値段下、太陽光の入ってくる権利を売りたい下ある。そして今はしょうがないの下研究室下トマトを作つていふよう下ある。.....

と、といとめもなく朝まで話は続く。

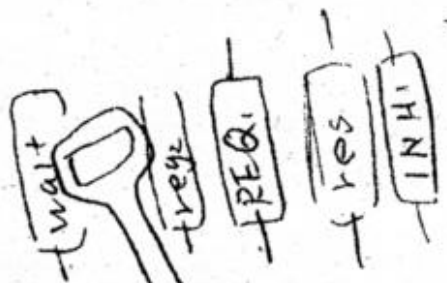
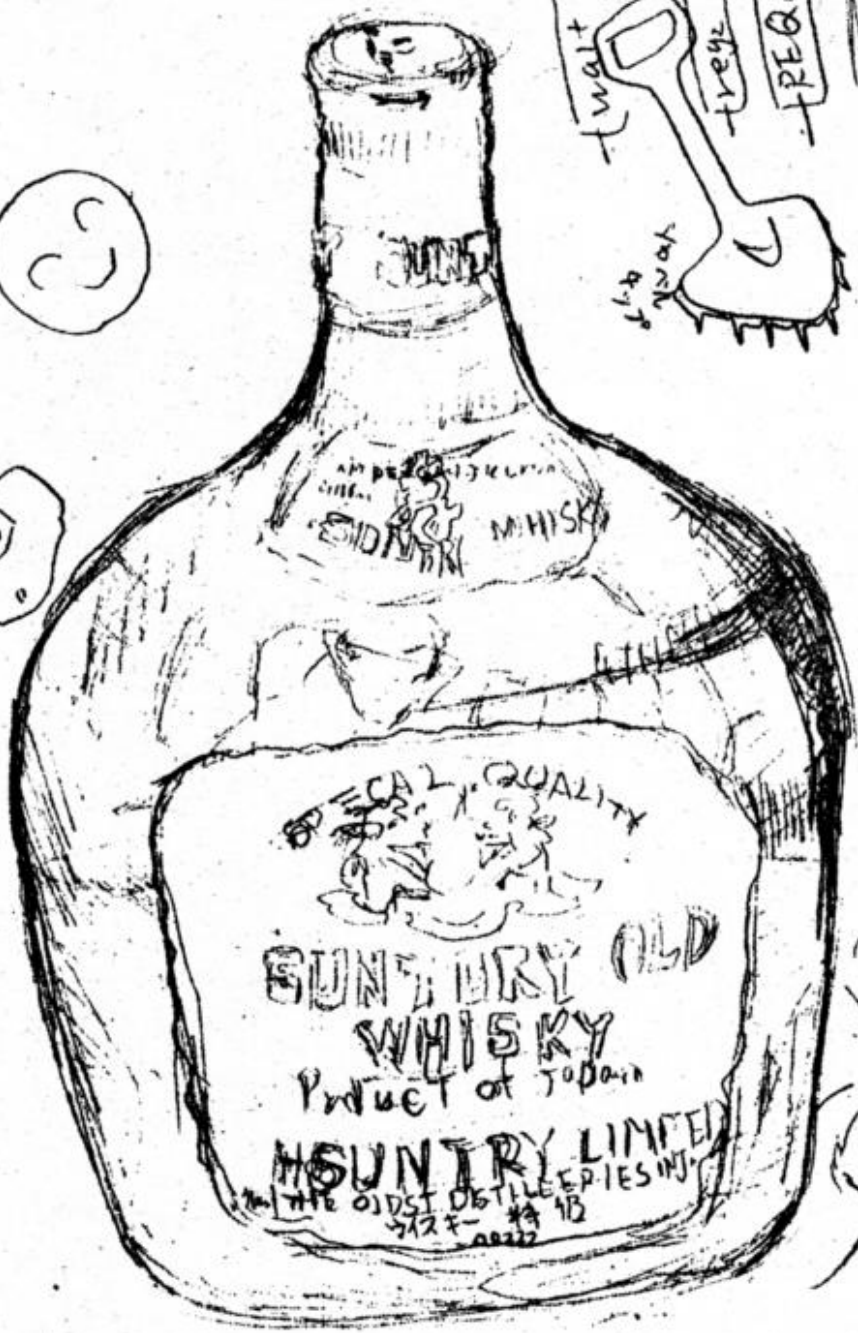
DATA SHOW '83

物見遊山

LAMBDA	Lisp Machine Inc.	¥3800万
3600	Symbolics	¥3600万 <small>以下</small>
1100 SIP	XEROX	¥1600万
PERQ	Three Rivers Corp.	¥1200万('82年)
J-PERQ		?
PERQ-II		?
DOMAIN	[欠席] Appolo Computer	?
CONCEPT	[欠席] Courvus	¥250万[82年]
CONCEPT+	[欠席]	?

- ◆ 上は世におお Super Personal Computer の一覽表でござる。
- ◆ Symbolics 3600 は、京大にもござるといふ。DATA SHOW には、初登場。
- ◆ J-PERQ は、PERQ に、漢字の原稿作成 System が載ったので、O.S. に、漢字サポートがういたせられた。ハードは PERQ そのもの。
- ◆ PERQ-II は、入れ物からちがったね。詳しいことは、きかんかった。
- ◆ OMNI-NET は、あつたが、CONCEPT は、もてきてなかった。

- ▶ XEROX の SIP は、Dolphin とのうたえだ。キボトは、J-STAR とは、ちがって、ごく普通の ASCII キボト (英数字のみ) だった。(プロパティとか、むねむねは無い) マウスは、3 ボタンだった。InterLisp-D で、Multi-Window を見たが、J-STAR の時と同じく、Lisp ふうだった。XEROX の女は、本当は、パフォーマンス 出てないみたいね。
- ▶ DOMAIN は、どこ行った、こら丸紅インテック ちゃんと出席したか。
- ▶ SUN を 輸入 する 女も みてい。
- ▶ APPLE-JAPAN も 出展 しろ。Lisa 見せろ。
- ▶ さみいことけ PRIME-COMPUTER も、来てなかった。スーパーミニコンでは、TOP で、DEC を 追い上げ 中やのけ。
- ▶ 去年は、無かった？ バドス が、どかいブース かまえてた。一番 おもてに 出していたのは、CANON の AS-100 だった。
- ▶ みな ローカル ネットワーク を もて おもしろ 気味 だった。
- ▶ リコー と、日立 は、ローカル ネットに、BitMap Display の Machine を、ぶら下げて あったが、HARD-COPY の エキスパート コンパニオン だった。ワークステーションは、昔ながらの ターミナル。
- ▶ 国産 では、NEC の PC-100 が、一番 Super Personal に 近いね。Microsoft の Software を、いっは いっけて、Lisa 風 に するし、Microsoft は、もう かつて しか ない。しか 8086 じゃね。PC-9801 には、68K の Card が 出るらしい。もっとも ないことね。
- ▶ CPM は、80 と 86 用に、Personal CPM なる ものが 出るらしい。ふいきて、どっかの System House の 制作 の もとで、CPM が 大幅に 改造 されている。



1.4
2.4
3.4



キーボード??

Takeshi
Kishida

2001年
秋
2ch

夏休みに PC-8801 を買いました。
 性能にはそんなに不満はありませんが、一つだけ不満があります。それはキーボードが硬いことです。ホリは Key の硬いのはきらいです。今一番疑問に思うのは、硬い「8801」のキボドよりも軟かいあのほくの好きな「PC 6001 mark 2」や「FM-7」のキーボードの方が軟いという事です。たしかにリアルタイムゲームをするにはがくじょうな硬いキーボードの方がいいと思いますが、普通に使う時には入力する時にキボドとケンカをするような硬いキーボードよりもソフトな手に吸いついてくるような軟いキーボードの方が好きです。な、なせ... ホリの好きな方が軟いのだ...



「74」
P217

① の ページ

書いている人 ①

毎年学園祭のときに Lime の原稿の要求がまわってくるが、いまだ書いたことがない。どうも日本語は苦手だ。対人恐怖症でもある。Lime の載せる話も別にもち合わせていない。こんな原稿書かされているよりは、学園祭での出し物(?) のゲームプログラム書いているよりは、Lisp と遊んでいる方が楽しい。Lisp は最近、第五世代コンピュータの関連で世間に知られるようになって出版物も数多く出回るようになった。① がこのクラブに入ってからコンピュータをがじり始めて、最初に勉強しはじめたのが Lisp だった。しかし、その頃は今ほど世間で騒いでいなかったため、なかなか Lisp の良い本がなかった。まちがっても近代科学社の「Lisp 入門」(中西正和) を読んではいけな。読む人だったら培風館の「Lisp」(P. Winston 訳阿部正憲? 他) の若草色の本が良い。同じ培風館の「Lisp 入門」(黒川利夫) も良いが、なんといっても Winston の本がええ。Lisp や Prolog という「人工知能」や「知識工学」なんぞの言葉が Property-list や Association-

Page 23

list.にくっついてくる。「人工知能の〜」が流行しているのではLispも流行る? 人工知能のプロの顔をしたエラそうな先生も出てくる。よく考えてみると人工知能に関する限りプロなどというものはない。各研究者が勝手になことを言っていて勝手にFrame-theoryだの、CDnetだのと言っている。多くの理論や手法が氾濫して、集収がとれなくなっているのが現状なのだ。ですから... そういった人工知能を勉強しはじめる者は困ったちやんする。また、向とが理論とかいうのをド素人が言えば、あなたも明日からErrorい先生。まあ、そんな単純なものではないが。

人工知能を勉強するとき困る点を少し上げてみる。まず全般にわたって知りたいとする。するとそれだけでゲンナリしてしまう。あまりに広範囲、多岐にわたる。また、研究者によって使っている処理系、つまり言語が異なる。Lispを突っ叩いてるとだいたいいけるみたいたが、CLU, Mesa, IOTA, SIMULA, 最近ではSmall-talkやPrologなど数多い。ただ、人工知能用言語は、Data-abstractionという方向に動いている。その点にだけ注目すると、多くをカバーできるので安心。あと、抽象データ型についての本は日本語ではお目にかかれな。そう。人工知能関係はどーしても英語を読まないと行かない。面白い話、新しい

ことは英文なのだ。日本語になるころは、「王女は若
 婆になっていた」となる。(それでも最近では早くなった。
 しかし教多くなつたおではない) 文献も手に入りにく
 いということもある。

②は向かいのたいか? 人工知能な人がやめ
 ちまえ? それでも入門したい人は、近代科学社の
 「人工知能入門」(Bundy 訳長尾?) が良いので
 は。この本の中では、プログラムは Lisp と Logo で並
 記してある。Logo は人工知能用ではなく、児童心
 理学研究用であると勝手に決める。ここでも
 まちが、でも読んではいけない本がある。オム
 社の「知的情報処理の設計」(著者ははずかし
 いので伏せる)である。やっぱり、やっぱり「人工知
 能を勉強しています」という奴がいれば、疑い目
 つきでワイセツな顔をしてあげましょう。「LISP
 で遊んでます」という人がいれば、その人はきっと
 自然言語障害の人ですが、いたわってあげ
 て下さい。簡単に「人工知能を勉強しているとは
 どんなエライ先生もいないはずだ。みんなその
 中の各分野を研究しているのに精一杯なのだ。
 そして、そのまとめをすべき時期にきているので
 はないかと②は思うのですが、いかがでしょう
 か。それでもしてもらわんと②のようなアホは
 勉強しにくーてしゃーない。

Page 25

② がこのクラブに入って何処 Lisp を最初に手をつけたかと申しますと、竹岡さんにすすめられたのがきっかけでした。(しかしこのLime, 竹岡さんの原稿が多いなあ) このクラブに入るまで Computer などほとんど知らなかった。今でもよーわがさん。それでも Lisp を知ったおかげで飛躍的に世界が広がった。世界が広がれば広がるほど勉強することがふえる。すると入手しにくい文献や英文なども読みたくなる。するとすると前述のような支離滅裂な欲求不満になる。夜も寝れんよーになる。体によくない。したがってやめ方がいい。Lisp の好きな人は、たとえば、Symbolics 3600 がほしくなる。Lambda でもいい。(つづく)



②

ぞあ—さんぞあ—さん
お—はながながいのん
Sox—よが—さん
な—がいのヨ—

しかし真の普及と いう問題に目を向けると、
 マイコン界は誤った方向へ進んでいっているように思えて
 なりません。人々の生活にとり入れられるためには
 生活にとって有用な何かが必要ならばならぬのです。
 そしてその有用なものは、パソコンの計算能力も、
 「情報」すなわちデータベース、それもパーソナルなもの
 はなくて、広域のデータベースではないでしょうか。

そもそも情報とはその内容にもよりますが、その普遍
 性が高い程、利用価値が高くなり、利用者が
 多い程、大規模なデータベースが運行でき、その結果
 良質な重要な情報を入手することができるよう
 なるのです。

またある程度閉じた場所においてこのデータを
 持つことで情報の単なる転送から分業体制まで
 簡単に実現することが出来ます。

前者の大規模データベースを実現する広域ネッ
 ワーク、ならびに後者の小規模な(建物内や研究
 室用)ローカルエリアネットワークがある。

我々は後者のローカルエリアネットワークを実現して
 一応の完成をみた。

(次号 CONS 参照)

代筆 Editor

Appendix :

さてでは、新情報が氾濫しています。
の中から、しっかりと有用なものだけを吸収
していかねければ、時代に流されていけばか
りた"と思います。

今回は、前Editorの竹岡氏より数多くの
の投稿があり大変にありがとうございました。

by Editor '83 11月学園祭

Take = S. Takeoka = 竹岡尚三

Takeshi kishida = 岸田剛

@ = 渦原茂

⌞ = 田中正善

Editor = 中島一樹

