

TSG

Theoretical Science Group

理論科学グループ

部報 198 号
— '96 オリパンフ号 —

目 次

TSG のご紹介	〔部長 わたる〕	1
分科会のご紹介		2
C/C++言語分科会	〔わたる〕	3
Windows 分科会	〔はと〕	4
86 アセンブラ分科会	〔竹島, 高野〕	5
TOWNS 分科会	〔宮本〕	6
Java Applet 分科会	〔わたる〕	7
HTML 分科会へのお誘い	〔かもしま〕	8
テキスト処理分科会	〔Aleph-NUL〕	9
T _E X 分科会	〔Aleph-NUL〕	10
鉄拳分科会	〔ZERO〕	11
落ちモノ分科会	〔竹島〕	13

TSG のイベント	14
いぬ。BBS とは何ぞや	〔ったく (Tak@inubbs, SysOp)〕 16
3 年からの楽しみ：根津中央研究所	〔Face〕 18
305	〔Tellur〕 20
inter-talk	〔八重樫 剛史〕 23
一般記事	28
PCM への誘い	〔あじ〕 28
ごみ.JPG	〔TEA〕 35
初心者のための 8086 講座	〔高野〕 41
情報棟を活用しよう!	〔Aleph-NULL〕 50

TSG のご紹介

駒場の代表的なコンピュータ・サークル

部長 わたる

TSG の正式名称は「理論科学グループ Theoretical Science Group」です。『理論科学』と名乗っていますが、これは歴史的な経緯によるものであり、現在の活動は全てコンピュータに関係することです。

部室が駒場キャンパスの学生会館にあって、PC-9821Xa と PC-486GR、FM-TOWNS HR が置いてあります。毎日来る人もいれば月に 1、2 回くらいだけ来る人もいます。みんなで昼食を食べたり、プレステで遊んだり、パソコンやゲームの話をして盛り上がりたり、ただおしゃべりするだけというのも日常的な光景です。

ゲーム制作、ツール作り、フリーウェア集め、音楽、お絵描きなどいろいろな趣味の人がいます。ビギナーもパワーユーザーも分けへだてありません。

活動は分科会ごとに行います

サークルらしい活動は分科会で行います。分科会とは、何人が協力して一つのことをやる集まりです。部員はそれぞれの好きな分科会で、プログラムを作ったり勉強したりするわけです。具体的な分科会の紹介は後のページでします。複数の分科会に参加したり、全く参加しないのも自由です。新しい分科会を自分で作っても構いません。

個人を主体としたサークルです

この「分科会」はハッキリした活動単位ではありません。活動の主体は個人なのです。TSG は単にその環境の一つ、発表の場の一つです。しかし TSG には本当のエキスパートと呼んでよい人が何人も集まっていますから、決して退屈はしないでしょ。

TSG には他大の方や一般の方も参加できます。プログラミングができなくてもいっこうに問題ありません。TSG にはコンピュータを学ぶ環境が整っています。コンピュータを使って何かをやりたい方なら誰でも歓迎します。部費もありませんし、気軽に部室を見に来て下さい。私にご連絡いただければいつでもご案内いたします。

〒153 東京都目黒区駒場 3 - 8 - 1 東京大学 学生会館 305 号室
東京大学 理論科学グループ

代表 金子 済

電話 03-5454-4343 (学生会館 要呼び出し)

e-mail g540879@komaba.ecc.u-tokyo.ac.jp

分科会のご紹介

TSG 活動の中心となる分科会ですが、とりあえず現在では次のような分科会の開催が予定されています。詳しいことは各紹介文をみて頂くことにしましょう。

- プログラム系

C & C++ 分科会 C 言語と C++ 言語によるプログラミングについて

Windows 分科会 MS-Windows 上でのプログラミングについて

86 アセンブラ分科会 Intel 系の CPU でのアセンブラプログラミングについて

FM-Towns 分科会 FM-Towns 上でのプログラミングについて

Java Applet 分科会 Java 言語を用いた WWW 上のアプレットのプログラミングについて

- 実用系

HTML 分科会 WWW 上のホームページなどの作成法について

テキスト処理分科会 各種ツールを使った効率の良いテキストの操作法について

T_EX 分科会 フリーの組版ソフト T_EX (テフ) をもちいた文書整形法について

- 娯楽系

鉄拳分科会 Playstation の格闘ゲーム「鉄拳 II」の攻略

落ちモノ分科会 ぷよぷよなどいわゆる「落ちゲー」の追求

C/C++言語分科会

わたる

「便利なフリーウェアやゲームの作者に自分もなりたい。」と思っている人は結構多いのではないのでしょうか？けれどプログラムの作り方を知らなければ、どうしようもありませんね。それから、プログラミング言語を一つ知っているとは全然知らないとは、コンピュータの便利さに雲泥の違いが出るものなのです。また、本当に凄いホームページを作るためには、プログラムを作れる必要があることは知っていましたか？

そう言うわけで、プログラミングをちゃんと勉強したい人にお勧めなのがこの分科会です。C/C++分科会は、プログラミングの基本中の基本である C 言語と C++ 言語の勉強会です。

市販されている C 言語や C++ 言語の解説書のうち評判の良いものは、DOS や UNIX のある程度の理解を前提にして書かれている場合がほとんどです。またプログラムの例題が非常につまらなく、それでやる気がそがれてしまうことも少なくありません。当分科会ではプログラミングどころかコンピュータも初めてという人のために、とても易しいレベルから始める予定です。応用例の紹介や、パソコンと UNIX の基礎知識含めて説明します。C や C++ 言語を単なる知識に終わらせずに、実用的な能力として身に付けることを目指します。また、良質なプログラムを作るための作法であるオブジェクト指向の概念も理解できるよう工夫するつもりです。

分科会の開催時刻は参加者の都合によって決めますが、たぶん放課後になります。場所は情報教育南棟か部室です。この分科会に入らなくても、昼休みなどの空き時間に部室でたいていのことは教えてもらえますが、なるべく参加することをお勧めします。駒場祭の準備が始まる頃にはきっと立派なプログラマになれるでしょう。

Windows 分科会

はと

ども、Windows 分科会責任者のはとです。

この分科会では、Windows プログラミングを勉強します。

「教える」じゃなくて「勉強する」なんですね～^^; (僕も初心者なのだ)

内容は、Windows プログラムの基本をやってから、何かゲームでも作れたらいいなと思ってますが実際の活動は人が集まってから考えます。

高度な事は全然しません (できない^^;) が 98 の Win95 上で Visual C++ を使うそうなので、C 言語を知らないと何やってんだかわからないと思います。あ、C++ は別に分からなくてもいいです。(僕もよく知りませんから^^;)

これは勉強会ですから、集まった人達で Windows の研究をしていきたいと思います。

例えば、現段階では Visual C++ のみの環境ですが、Visual Basic や Delphi など他言語を使う人がいれば、そちらのほうでの開発もやってみたいと思っています。

```
//スペースが余ったのでおまけプログラム
```

```
#include<windows.h>
```

```
int PASCAL WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    return MessageBox(NULL, "Hello Windows World!",
                      "Windows 分科会", MB_OK);
}
```


86 アセンブラ分科会

竹島, 高野

リアルモードのパート

竹島

アセンブラの利点は、ハードや G-VRAM に直接アクセスした高速でコンパクトなプログラムを書くことができることです。その代わり、CPU の命令 1 つ 1 つを直接書くという性質上、いくつもの命令を組み合わせないとプログラムになりません。アセンブラでプログラムを作るのは大変な作業なのです。現在、プログラムを C 言語等で書くのも、プログラムを作る作業をすこしでも軽減するためです。しかし、例えば C 言語で PC98 シリーズの G-VRAM に直接書き込みを行う場合、ポインタにアドレスを代入し、1 データずつ転送しなければなりません。この部分にアセンブラを使えば、ストリング命令で高速に転送ができます。8086 系リアルモードや仮想 86 の場合、コンパイラによっては C 言語ではセグメントを 1 回ごとに切り替えるのでさらに遅くなってしまいます。このように高速化が必要な部分だけでもアセンブラで書けば、他の部分を C で書いても相当高速になります。また、アセンブラで書いたプログラムのサイズは、C 言語で書いたプログラムに比べて相当小さくなります。常駐プログラムやデバイスドライバはアセンブラでなければ書けません。(C 言語で書くと、サイズが大きくなり、しかも非常に遅くなります)

プロテクトモードのパート

高野

GNU アセンブラを使ってプロテクトモードのアセンブラをやります。プロテクトモードのアセンブラはリアルモードに比べてメモリなどの管理が非常に大変なので、直接アセンブラで書くと言うよりはむしろ djgpp 上でインラインアセンブラをやります。プロテクトモードの利点といったらやはりメモリの上限がないに等しいと言うことでしょう。(理論的には 4G バイトの上限はありますが...) C 言語上でプロテクトモードのプログラミングをすとしてもアセンブラの知識は必要です。たとえば割り込みのルーチンを書くにしても 8086 のセグメントに相当するセクターの概念とかの考え方が必要ですし、DPMI と VCPI の違いも知らなくてはなりません。プロテクトモードの仕組みを知ることで、Windows95 などの最新 OS の仕組みもわかるようになります。

TOWNS 分科会

宮本

初めまして、こんにちは、TOWNS ユーザー（ここが重要）の宮本です。ここを読む人は非常に限定される（すなわち TOWNS ユーザー）と思いますが、たくさんの人に読んでもらうことを期待して、TOWNS が、いかなるマシンであるかを少し書いてみたいと思います。ところで筆者は、非常に国語が弱かったので文章が異様になるかもしれませんが、そのところは、勘弁してやってください。

皆さんご存じの通り、今、CM で宣伝している FM-V を作った富士通が、7 年前頃に出したのがこの FM-TOWNS です。当初は FM-7 の実績もあったことから、68000 を使ったホビーパソコンということだったのだそうですけれど、噂によると、当時の課長の鶴の一声で一転。パブルの真っ只中にあったこともあり最新鋭の 80386 を使い、CD-ROM を標準装備という豪華な仕様になったのだそうです。その仕様を少し書いてみると…、SCSI 標準装備（初代機のみオプション）、メモリは SIMM、当たり前ですが CD は標準装備で CPU は全て 386 以上であるので、プロテクトモード（つまり 32 ビット OS）で動作するものがほとんどです。（それとあまり知られていないですが、純正品を使っているぶんには面倒な設定などがいらぬということもあります。つまり今風に言うと p&p ということです。）ようするに現在主流である仕様そのものなのです。（逆に言うと、これだけのパワーを持ってして今の様な状態というのは…（笑））

そこでみんなでこの TOWNS を盛り上げよう、というのが、この TOWNS 分科会の主旨なのですが、経験者（所有者）はもちろん未経験者も大歓迎です。具体的には、High-C によるプログラムを予定しています。（ちなみに筆者は TOWNS と心中する覚悟で開発を行っています。と言ってもべつに本当に死んでしまうという意味ではない（笑）。その証拠にこのような状態になってから、TOWNS に 10 万円投資してしまった。）ところで、こんな偉そうなことを言っていますが、実は筆者も 1 年前までは、TOWNS をまるっきりゲーム専用機にしていました。それに、このサークルの先輩方ときたら、我が町川越のちょらんま氏、SKY-DUAL や Oh! TOWNS 誌でお馴染みの Makken 氏、同じく Oh! TOWNS 誌の Malar 氏など、筆者のような技量ではとても後を継げないような、そうそうたるメンバーなのです。つまりところ TOWNS に興味があればだれでも OK！ということです。

話は変わりまして、近況というか今開発しているものを少し書かして頂きますと横スクロールのシューティングゲームを作ろうという壮大な計画を立てているのですが、運と体力があれば、これが読まれるころにはデモがそこで動いているでしょう（^o^；そしてその後で駒祭用のゲームでも誰かと共同開発できたらなあ…と思っています。

みなさん、一緒に TOWNS でゲームをつくりましょう!!!

と、ここで本当は終わりにしようと思っけていますが、最近になって、高校時代の友人が fMSX のソース（おおもとの作者は外国の方のようですが親切にもほとんど全て ANSI C で書いているので I/O 周りを書きかえればどの機種でもできるようになっています。事実そのソースには IBM PC/AT 版と UNIX 版がありました。他のメジャーな機種はほとんど全てが移植されているようですが、何故かそのドキュメントには TOWNS 版がありませんでした。）を持っていましたので、それを筆者と友人たちでせこせこ移植してゲームが動く（実際にパッドで遊べる）程度になりました。（ただし fm 音源周り、キーボード周りはまだ未完成。）おおもとの作者はフリーウェアを希望しているようなので、これを読んでいる TOWNS ユーザーの皆さん、ご希望ならば、無償で配付します。また、ほとんど目安ぐらいにしかならないとおもいますが、FM-TOWNS II HA (486DX2-66) でグラデウィスなどの MSX1 用ソフトがまともに動きますが、MSX2 用は、Pentium と言えどもまともなスピードを得るのは難しいでしょう、というような感じです。（なにせ汎用的な C でかいてあるもんで、もし他の移植を考えないならば画面関係や Z80 のエミュレートのところ、もっと速くできるでしょう。）

最後になりましたが、ここまで読んでいただき、ありがとうございました。それでは、さようなら。

Java Applet 分科会

わたる

Java、HotJava という言葉を新聞や書店で見かけたことはありませんか？ Java はインターネット上の異機種間分散環境を見据えた、未来志向のプログラミング言語です。簡単に言いますと、従来の言語によるネットワーク・プログラミングではデータのやりとりしかできなかったのに対し、Java はプログラムの実行コード本体を送ることが出来ます。開発キットは今年の 1 月の終わりにリリースされたばかりで、大変新しい言語です。

この分科会で取り扱う Applet とは Java のプログラム的一种であり、ホームページに張り付けて使います。これによって既存の HTML と CGI の環境とは一線を画した、animation や interactivity に富むホームページを作ることができるのです。

Java は version 1.0 がリリースされたとは言え、依然として発展途上の言語です。また Applet を使ったホームページも世界的に見てまだまだ多いとは言えません。そのために資料の不足や開発環境にきつい制限があり、Java の勉強をするのはなかなか大変なのが実状です。最近書店に増えてきた解説書も、開発キットの Tutorial を元ネタにした内容の薄いものばかりで、頼りにはなりません。また、Java と同じ目標をめざした他の言語が登場してきており、Java が標準的な地位を獲得するかどうかもわかりません。Java の

将来はまだ不透明なのです。

それでも敢えて報われないかも知れない困難に挑戦し、未開拓の言語を切り開いてみたいと言う方の参加を待っています。

HTML 分科会へのお誘い

かもしま

えーと、いきなりですが、あなたは HTML なるものをご存知ですか？ TSG に興味を持たれてこの部報を手にとっている新入生の方の中には結構ご存知の方もいらっしゃるかも知れませんね。HTML とは、HyperText Markup Language の略で、世間で「netsurf」やら「世界に向けて情報発信」やら騒いでいる(^_^; Internet の WWW (World Wide Web) というサービスで閲覧される文書を記述するのに用いられる言語のことをいいます。

HTML では、文書の中に整形や他の文書へのリンクといったコマンドを埋め込む形式をとっています。例えば `
` が強制改行を表したり、`` が `www.hoge.co.jp` というアドレスにあるコンピュータの `foo.html` という文書へのリンクを表すと言った具合です。こういう形式は、慣れてしまえば非常に効率よく文章を書くことができますが、いかんせん見たままの形で編集できなったり、あるいはコマンドを覚えるのが面倒だと言うために難しいと思われがちなようです。

そこで、HTML 分科会では、HTML の基本的な文法について、主に情報教育棟 (ここでは WWW を利用することができます) で実際にホームページをへろへろと作りながら学んでいこうと思います。まあ実は HTML はすごく簡単なので、そんなに長い期間に渡ってやるつもりはありませんし、WWW の上には HTML の書き方について書かれた文書は星の数ほど存在するので、もっと色々知りたいことがある人はそちらを参照することもできます。あるいは今話題の Java に興味があるならば、後に Java 分科会も開かれるらしいのでそちらに参加するというのも良いでしょう。

もともと HTML というのは人に書けと言われて書くものではありませんし、書きたいと思うならば人に教わらなくても自分で WWW を果てしなくさまよいながら気に入ったページの書き方を参考にすれば良いまでの話です。よって実際の分科会では、HTML の書き方そのものよりも情報教育棟の UNIX マシンの使い方も含めた WWW の楽しみ方の方に重点を置くつもりです。ですから当分ホームページを作る気はないという方にも多少は役に立つでしょう。お暇な方は是非ご参加ください。

テキスト処理分科会

Aleph-NULL

例えば何枚ものレシートを、買った金額毎に集計するとき、あなたはどのようにしますか？
あるいは、何万行ものデータの中から、「TSG」と「東京都」を含むデータのみをとり出すとき、どうしますか？

住所録の「郵便番号」と「住所」を全部入れ変えろといわれたら？

ちょっと昔の「03-123-4567」とかいう名簿を東京の3桁の人だけ「03-3123-4567」に直すには？

こんな処理を、いちいちワープロの検索機能に頼って手作業でやってたのでは話になりませんね。

こういった処理は、コンピュータに任せてしまいたい所ですが、ワープロ・エディタでは力不足ですし、かといってプログラムを組むには単純な処理の割にえらく面倒です。

こういう時は、sed, awk, grep, perl といったツールの出番です。上の例はそれぞれ、

```
% jgawk '{y[$1]+=$2} END{for (a in y) print a,y[a]}' data
```

```
% grep 'TSG' <data | grep '東京都'
```

```
% jgawk 'print $1,$3,$2,$4,$5' data.txt
```

```
% sed -e 's/03-\([0-9][0-9][0-9]\)/03-3\1-/g' <data
```

のように「たった1行で」処理できます。

しかし、これらのツールがはやらない理由には、「正規表現」とよばれる条件の設定が解りにくいことでしょう。

この分科会では、正規表現の使いこなしと、grep, sed, awk, perl といったツールを用いて、いろいろな処理を「いかに楽をして行かうか」を説明します。

ほとんど予備知識なくてもわかるような分科会にしようとおもっています。気軽に参加してください。

T_EX 分科会

Aleph-NULL

T_EX (テフ) というのは、端的に言えば文書整形ソフトです。この部報も T_EX で作っています。

元々は、アメリカの数学者 Donald E. Knuth が自分の数学本を書くに当たってつくったものです。

そんなわけで、T_EX は数学者の為のもので、一般には近寄りがたいと思われているようです。しかし、それは誤りです。とっつきにくいことは確かですが、(特に理系の人間には) 極めて使い易くできています。

特徴をあげると、

1. 欧文や数式に極めて強い
2. 見掛けにとらわれずに論理的に文書を作ることに集中できる
3. 使用環境に依存しない
4. フォントまで含めてフリーである

といったところがあげられます。

(1) はもともとアメリカ人が出版用に作ったものですから、欧文のプロポーショナルや数式の出力は極めて満足の行くものです。

“¿But aren't Kafka's *Schloß* and Æsop's *Œuvres* often naïve vis-à-vis the dæmonic phoenix's official rôle in fluffy soufflés?”

とか、

$$\int |Y_{lm}(\theta, \phi)|^2 \sin \theta \, d\theta \, d\phi = N_{lm}^2 \int_0^\pi \sin \theta \, d\theta |P_{lm}(\cos \theta)|^2 = 1$$

とかいったものを、いとも簡単に入力できます。この部報の後ろの方の記事にも何ヶ所か数式が出て来ますが、本文中にこんなのがごちゃごちゃでてきたら、普通のワープロでは手におえません。

(2) は L^AT_EX という普通使うマクロパッケージで、「ここから章が始まる」「ここは引用」とか指定していくと、あとは勝手に形態を整えてくれるというものです。

(3) も重要で、MS-DOS パソコンでも UNIX ワークステーションでも、家の 180dpi のドットプリンタでも 3200dpi の写植機でも同等の出力が得られます。家で 180dpi で推敲して、本番はディスクで結果を持ち込んで写植機にかけてもらうとかいうことも可能です。

こんなにいいとばかりの TeX ですが、やっぱりめんどくさそうというイメージが強いんですね。残念なことです。

というわけで、この分科会では、この強力な TeX を自分でつかいこなすべく、解説をこころみていきます。担当は私、「TeX にすっかりのめりこんでしまった」編集長 Aleph-NULL が行います。参加をお待ちしています。

鉄拳分科会

ZERO

主曰く 人は パンのみにて 生きるにあらず
されど また曰く
人民に パンと 自由を！
市民に 娯楽を！

例えどのような状況下にあれ、人は生きることから始めねばならず、生きる為に娯楽を求めることは天地自然の理である。向かいて天に恥じること無し。まして東大生に於いてをや、日頃の勉強に備え健康な肉体と健全な精神を保持する事は神聖なる義務である。その根幹が娯楽に置かれてある限り、これを軽視する傾向は厳に戒められねばなりません。

これから私が語る恐怖の事件、TSG を震撼させたあの忌まわしい事件こそ、その何よりの証であり、学問を追求する東大生と雖も、この人生最大のテーマからは逃れ得ぬ凡俗の徒に過ぎぬことを我々に示すものであります。

全ての事件がそうであるように、それもまた、ありふれた日常の一コマから始まったのです。

そもそもの事件の始まりは、現 3 年生の「基金」により、SONY が世に送り出した傑作ゲーム機・PlayStation が学館 305 に位置する TSG 部室に導入されたことに端を発する。当初は部員の私有物の数日間の滞在こそあれ、これと目玉ソフトの常駐はなかった。

しかし、まず手始めに namco の秀作レーシングゲーム・リッジレーサーが導入され、次いで新入部員（現 2 年生）らが部室に出入りを繰り返すようになったころ、畠山氏（現 2 年生）が彼の好意から鉄拳を部室に持ち込んだ。しかし、このことこそが数人の若き TSG 部員を鉄拳猿へと変貌させた忌まわしい事件の決定的な要因となった。

現在鉄拳分科会を構成する 4 人のうち、M は既にその前から鉄拳を始めていたようなので取りあえず横に置いておくと、K と H は TSG の部室に出入りするようになってか

らあつという間に鉄拳猿へと変貌した。特にHたるや、出なくて良い授業が彼の高校時代に比べて余りにも多すぎることを良いことに、夏学期の間の必修以外の時間はほぼ部室に常駐して鉄拳の修行に勤めた(必修の時間すらも!)

KやMも勿論部室に来れば鉄拳である。現3年生のW氏を中心に鉄拳愛好会(仮称)が結成され、昼夜を問わず鉄拳が部室で行われ、一時期などはPS=鉄拳なる「PSの方程式」が成立するほどPSでは鉄拳のCDしか回らなかった。ZERO DEVIDEも導入されたが結局鉄拳にはかなわず、潰えた。鉄拳こそが正義であり、例えPSで他のゲームをしていようと、鉄拳愛好会員が部室にやってくるや、まるでそれが徴発であり、それが当然であるかのごとく、PSのふたが開けられ、鉄拳に変えられた。そして、ぶよぶよ人間であったGも部室にいれば目に入ってくる鉄拳の画像に洗脳され、現在では立派な鉄拳分科会員である。

鉄拳の修行道場と化したTSG部室。彼らの鉄拳熱は止まるところを知らず、3/29に発売が予定されている「鉄拳2」についても発売当日の導入が予定され、駒場祭打ち上げコンパ時に設立が決定された鉄拳分科会は、事実上その役割を終え、引き続き名称を変えることなく鉄拳2を修練・研究の対象とし、存続することになった。

以上があのおまわしい事件の全貌である。

TSGとは「Theoretical Science Group」の略称であるというのがタテマエだが、実状は「Tennis, Ski & Game」であると語る事情通もいる。しかし、正式には「鉄拳最強グループ」の略称であると語る人物もいる。

その真偽のほどは別として、分科会の人間ははっきりいって異常だ。勿論、その異常の中には強さも含まれるが、彼らの大半は空中コンボの研究家である。一発食らったらもうおしまいなど日常茶飯事で、Hなどはしょっちゅういぢめられている。

こう書くと「鉄拳分科会の人間は異常だ」と思われるかもしれないが、心配には当たらない。何故なら、この分科会の人間は別にキャラクターに異常な愛情を注ぎそれを表現(同人誌・フィギュア等)している訳ではないからである。単純に強さを求める。それが鉄拳分科会の存在意義である。無論、強さを求めるだけではなく、単に遊びたいだけでも構わない。

また、こういったものは上手な人間とやってこそ上達するのである。「家にあるからいいや」と思っているあなた、井の中の蛙で終わってはならない!

例えば、格闘ゲームが苦手なトIIを代表とする2次元格闘ゲーム全盛期には乱入することもおぼつかなかった人間であったHは、鉄拳分科会の前身である鉄拳愛好会のおかげでゲーセンで乱入できるようになった。後にHは語る。「1人でやっているだけでは上達しない!」と。

入り方は至って簡単、部室にあるPSで鉄拳または鉄拳2をプレイすればいい。対戦相手が欲しければ、昼休みや4限や5限の終了後なら誰か必ずいる筈だ。勿論、他の分科会のように決められた曜日や時間などは設定されてはいない。ただひたすら鉄拳をやる、それがこの分科会の活動内容だ。勿論、そのプレイの中で様々な技を研究するのは

格闘ゲームの常であるが。相手が自分の知らない技を出した場合は相手に聞き、それを自分のものにする。それによってみんなで強くなるうというというのがこの分科会の設立趣旨である。

暇な空き時間の多い駒場生活、まずは暇つぶしにでも覗いて見てはいかが。

落ちモノ分科会

竹島

落ちモノ分科会は、ぶよやばずるだま等の落ちモノを遊ぶだけの分科会です。一応、目標は、だれかがぶよ等で遊んでいたら、参加すること、積極的に連鎖を作ることです。つまり遊ぶだけです。305には活動に必要なゲーム機がちゃんと用意されています。また、落ちモノは種類を問いません。(某ばずるだまでも可)ゲーマーな人、こわれている人大歓迎です(もちろん、そうでない人も歓迎します)活動内容は、未定です。(きっとつぶれますね、この分科会)参加者がいたら考えます。やっぱり、どうすれば大連鎖が組めるかを追及することになるんでしょうか。うーむ。やっぱ未定ということで(^^;なんか得体のしれない、存在すらあやしい分科会ですが、とりあえず参加してくださいね:) ということで、おしまい。

TSG のイベント

はじめに

TSG には年間を通して、たくさんの様々なイベントがあります。こうしたイベントのうち主なものについて紹介しましょう。

新入生への説明会

5月の月上旬に TSG に入りたい人、入るかどうかが迷っている人のために説明会があります。分科会の説明や部員の自己紹介が行われる予定です。10月から TSG に入った現2年生もいるくらいですので、説明会に必ず出席する必要はありませんが、なるべく来て下さい。

開催日程は電子メールやハガキでお知らせします。 (部長わたる)

新歓コンパ

説明会の数日後に来るのがどこのサークルでも必ず行われる「新歓コンパ」です。TSG のコンパは一般的に「飲む」為ではなく「食う」為に存在していると言えますので、酒に弱い・酒は嫌いという方でも安心して参加できます。運動会系に見られる「飲みの強制」等は一切存在しません。

一般に、コンパの後は(普通、21時前後に終了します)ゲームセンターへ繰り出す人・喫茶店へと出かける人・根津研(別掲)に消える人・すぐに帰る人など、皆さん様々な行動をとりますので、集団のどれかにくっついていけば心配はない... と思います(爆)

東大はその性格上、1・2年生と3・4年生及びそれ以上(修士・博士)の人間の拠点が駒場と本郷の2カ所に分かれているので(TSGには3年生以上でも駒場にいらっしゃる方もいらっしゃいますが)、コンパや駒祭くらいでしか皆さん一緒に、ということがありません。ですから、新歓コンパはもちろんのこと、その後にある数々のコンパ¹にも、皆さんどうぞふるってご参加下さい。(以上、コンパ委員の(笑)ZERO 記す)

夏合宿

7月の終りごろ、試験が終わった時期に、夏合宿があります。昨年は伊豆大島でした(3泊4日、うち船中1泊)。TSGはコンピュータのサークルですから、合宿といっても特に何もすることはなく、遊ぶしかないわけです(笑)。

¹新歓コンパの他には10月・駒祭総決起コンパ・11月の駒祭打ち上げコンパ・3月の追い出しコンパがあり、さらに1月には恐怖の(^^; ケーキパーティーが...あなおそろしや(^^;

昼間は、自転車で島を一周したり、炎天下のもとテニスをしたり、海で泳いだりしました。夜にはゲームや花火で盛り上がっていました。

3年生以上の人と会える数少ないチャンスですし、1度参加してみるとよいでしょう。
(ライブラリアンげる)

駒場祭

TSG にとっては唯一の活動の発表場所である駒場祭は、TSG 最大のイベントといってもいいでしょう。まず、恒例の展示物として、占いを1年生が中心になって作ります。それだけでは「理論科学グループ」としてはつまらないので、何人かが更にプログラムを作って展示します。去年度の展示物は次の通りです。

- 四柱推命占い
- UNIX 版「平安京エイリアン」
- Windows95 「落ちモノ」
- 微分方程式の数値解法による物理シミュレーション
- フライトシミュレータ
- 多関節モーションエディタとデモ「Marionette」
- 分子軌道の 3D デモ

まず、10月に「駒場祭総決起コンパ」が開かれます。そこで全員が大風呂敷を広げまくり、そのホラを残り1ヶ月で可能な限り現実に近づけるのです。(^^; 終了後は打ち上げコンパも行われます。
(Aleph-NULL)

冬合宿

スキーは全くのドシロートのタカノです。昨年度は2泊3日で志賀高原に行って参りました。冬休みの最後に行われます。

TSG のメンバーはスキーの上級者も多いのですが、ほとんどスキーの経験の無い人でも手とり足とりおしえてもらえるハズです。ボーゲンさえできればあとは何とかできるでしょう。私自身中学校のスキー教室以来のことだったので、はじめから転んでばかりいたのですが、2日目の終わりにはシュテムターンもどきぐらいはできるようになっていました。ただ、初心者の方はコースを間違えないようにしましょう。私は部長と一緒に大回転のコースに行ってしまう立ち往生する羽目になったので...
(高野)

ケーキパーティー（ケーキコンパ）

ケーキコンパはセンター試験の日に駒場キャンパス内で行なわれるコンパです。ただしアルコールが入らないので、むしろパーティーといったほうがよいかもしれません。各自で食べきれだけのケーキを持ち寄り、センター試験を受けに来た受験生を横目にひたすらケーキを食べます。みんなで話しながら楽しいひとときを過ごします。たいていの方はだんだん気持ち悪くなり、コンパの後しばらく体がケーキを受け付けなくなります。人生は何事も経験です。このような経験もためになるでしょう。みなさんもぜひ参加してください。（片岡）

いぬ。BBS とは何ぞや

ったく (Tak@inubbs, SysOp)

はよーん。ったく、でーす。

SysOp やってまーす

1 いぬ。BBSってどういうところ？

というわけで、いわゆる草の根パソコン通信 BBS「いぬ。BBS」とかいうものの運用をやってます。パソコン通信がどういうものか、とかいうのは説明は不要ですよ？

そもそもいぬ。BBS は、コンピュータゲームサークル「あらかわ犬（ウシ）」のソフト開発・サポート BBS として運用すべく、1993 年に開設されました。あらかわ犬というのは、TSG メンバー有志を中心に結成されたサークルで、端的に言えば「ゲームを作ってコミケで売る」とかいうような活動をしています。そういう目的で運用が始まったいぬ。BBS なのですが、運営母体であるあらかわ犬は現在、活動を小休止している状態になってしまっています。

で、現在いぬ。BBS はどうなったかということ、事実上フリートーク中心の BBS として運用されています。って要するに雑談しているわけですね。

いぬ。BBS の会員は、運営母体であるあらかわ犬がもともと TSG メンバー中心だったこともあって、ほとんどを TSG 関係者で占めています。今ちょっと人数を数えてみたら、7 割以上が TSG 関係者でした。うーん。

2 いぬ。BBS でみんななにやってるの？

いぬ。BBS の主な活動は、ずばり雑談です。

世の中には「アニメについての会話専門 BBS」とか「猫についての会話専門 BBS」(なんだそりゃ) などというものも存在していますが、いぬ。BBS は決して「犬についての会話専門 BBS」ではありません(笑)

もっとも、ジャンルを問わない雑談とは言え、なにせメンバーの 8 割は TSG 関係者です。話題も自然と TSG 好みな系統になることが多いようです。

もちろん、年がら年中雑談ばかりしているわけではありません。プログラムの開発・テストなんかをやってる人も結構多いですね。これには、元々いぬ。BBS が「ゲームの開発をしよう」という目論見で開局したという経緯もありますし、TSG に強力なプログラマが多い、というのもあります。かくいう私もいぬ。BBS で、自作フリーソフトウェア “lfd.” の テストをやったりしました。

プログラムだけじゃなくて、例えば絵を描いてる人も結構いたりしますねー。音楽データを作ってる人は今のところ一人しかいないのですが。

ちなみに、去年(1995 年)1 年間の統計データを紹介しますと、ファイルライブラリに登録された「自作プログラム・自作データ」の数は、約 300 本でした。

いぬ。BBS には、TSG の中心メンバーのうちのかなりの数がアクセスしています。そんなわけで、TSG に関する連絡がいぬ。BBS で行われることは多いですね。コンパや合宿の告知なんかは必ず電子掲示板に書き込まれるようです。

掲示版での告知だけでなく、例えば TSG メンバーの間の連絡には、電子メールがよく使われているみたいです。学校に行かなくても自宅から連絡が取れるって、予想以上に便利がいいんですね。

3 いぬ。BBS にアクセスするには？

いぬ。BBS にアクセスするには、当然パソコン通信をするための機材が必要です。パソコンとモデムと電話線ですね。そりゃそーだ。最近はモデムも安くなったし、敷居はとっても低くなりましたよねー。

準備ができれば、おもむるにいぬ。BBS に電話をかけます。回線がつながると ID の入力画面になりますから、そこで new [リターン] と入力するだけで、自分の ID を発行してもらえます。簡単ですね

いぬ。BBSは、「積極的に宣伝はしてないけど一応公開BBS」です。TSGの人じゃないと入れない、とかいうことは全然ありません。ゲストアクセスでも制限はほとんど無いですから、一度ためしにアクセスしてみてください。

とゆーわけで、みなさまのお越しをお待ちしております。

いぬ。BBS

Tel. 03-3468-4008 300/1200/2400/9600/14400/28800(V.34/V.FC) (bps)

(1回線、24時間運用)

mmm Rev. 4.1 オンラインサインアップ可、会費無料

ったく

Tak@いぬ。BBS (SysOp)

taka@is.s.u-tokyo.ac.jp

注) この案内文、統計部分を若干修正してある他は、去年のオリパンフに載せた原稿ほとんどそのままだったりするのだな～

3年からの楽しみ：根津中央研究所

Face

新入生の皆さん、こんにちは。根津研責任者のFaceです。ここまで読んできた皆様には、TSGがどういう集団なのか、だいたいわかって頂けたと思います。簡単にいうと、普段は305でわいわいがやがや楽しくやって、コンパや合宿といった年間行事で楽しんで、その成果を部報とかたちで残すのがTSGで、コンピュータに限らず楽しいことをとことん追求する団体と言えるでしょう。

しかし、ここで東京大学の持つ問題点が浮き彫りになってきます。それは、学部生は3年から、キャンパスが本郷に移ってしまうため、3年生以上は駒場の学生会館に足を運ぶ機会が減ってしまうという点です。実際、東京大学にあるサークルの中には、本郷に進学した人はもうサークルに参加しないという、属に言う「本郷解散」という現象が起きているサークルも少なくありません。しかし、「1年毎に、一定数の新しい人間が加わって、同時に一定数の人間が去っていく」という性格を持つ大学のサークル活動において、それが駒場の2学年のみから構成されるとすると、それまで培ってきたものの半分を失うことになり、1年毎に全く別の集団になると言っても過言ではありません。

そこで、3年生になってもTSG部員(以後TSGerと略す)としてTSGの活動を続けるための場として、「根津中央研究所」(通称根津研)というものがあります。「あります」と書くと、どっかから湧いて出てきたみたい印象を受けますが、実際は、過去のTSG

部員が本郷生用の活動スペースとして設けたものです。こういった場を何もないところから作り上げるのは非常に大変だったと思いますが、私達はそういったことをあまり実感することなく、会費を払うだけでその恩恵をこうむることが出来ます。これもサークルという組織の持つ力と私は考えます。

では、いよいよ根津研について説明します。根津研というのは、根津駅から歩いて数分の所にある、アパートの一室です。ここには、ATマシンや98MATEをはじめとして、PS、SS、SFC、MDにNEO・GEOと、ゲーム機も見事に揃っています。また、ビデオデッキもあるので、見逃したアニメがあっても、1週間くらいなら根津研でリカバー出来ます。また、根津研はアパートの部屋なので、便所は当然として、流し台に冷蔵庫、ガスコンロに炊飯器といった生活に必要なものも揃っています。さらに、部屋は絨毯張りなので、ねっころがってゲームという事も出来て、夏は冷房、冬はこたつとストーブとホットカーペットと空調も完璧で、305よりはるかに快適です。また、駅と根津研の間に銭湯があり、根津研には毛布と布団があるので、根津研で一晩あかすというのも最近では珍しくありません。下手に外泊することを考えると、それだけで十分根津研会費の元を取ることが出来ます。

根津研というのは、根津研会員と言われる TSGer の共同出資によって運営されています。根津研にかかる費用の大部分は家賃（月当たり 5 万 8 千円）ですが、これに電気水道ガス電話が加わって、結同年当たり 85 万円くらいかかります。これを根津研出資賛同者の人数で割ったものがその年の年会費となります。ちなみに、95 年度は年会費は 4 万円で、今年度は年会費は 3 万 5 千円です。

新入生の皆さんには 3 万 5 千円というのかなりの額に思えるかも知れませんが、実際それだけのお金をアルバイトして稼ごうと思ったら、それほど大変ではないことがわかるはずですよ。もしいいバイトが見つからなかったら、直接僕のところに言いに来てくれれば、いいバイトを紹介します。

根津研というのは先ほど言ったとおり、本郷生のためのスペースです。従って、駒場生は根津研にお金を払う必要はありません。駒場生のかたで根津研がどんなところか見てみたいという方がいましたら、根津研に電話をかけてみましょう。そして、いまから行くということを伝えておきましょう。そうしないと、普通は鍵がかかっているの、鍵を持っていない人は、中に誰かいないかぎり中には入れません。鍵は、年会費を払った人間に渡されます。本郷生でない人で、どうしても鍵を持っておきたいという人がいたら、会計までご連絡下さい。考えます。

だいたいパターンとして、コンパのあと（ゲーセンを経由するかもしれないが）、根津研になだれ込むということがよくあります。このときに、しっかりと根津研の場所をおぼえておきましょう。そして、根津研の快適さを身をもって感じて下さい。あの快適さを知ってしまうと、305 が窮屈でたまらなく思ってしまうかも知れません。しかし、これも 2 年間の辛抱。めでたく本郷生になったあかつきには、すばらしい根津研生活がまっています。みなさん、是非駒場は 2 年で卒業しましょう。あれ、何の話だったっけ(^_^;

根津中央研究所：(03) 3821 - 3384

305

Tellur

人民食堂の裏にひっそりと建つ我がサークルの根城、学生会館。玄関をくぐり、事務室の横を歩いてすぐ右手の階段を一気に3階まで登ると、305号室の入り口が正面に見える。扉には「化学部」「生物研究会」「自然科学研究会」「理論科学グループ」という、4つの理系の極致に達した名を持つサークルの看板が貼ってある。しかし、これらのサークルの活動は謎に包まれている。

左手手前にテーブルと冷蔵庫の置いてあるところが「化学部」のスペースである。彼等は土曜になると「実験をする」と言って、いずこへともなく消えていくが、暇になると象牙様の直方体に竹のようなものをはめ、妖しげな文様を刻印したものをかき混ぜているようだ。

左手奥に居を構えているのが「生研」である。彼等は珍しい生き物を求めて山深く分け入るのを生業とするが、時々酒を片手に深く語り合っているのを見ることがある。

右手手前に陣取るのが、305の最も新しいメンバーである「自然研」である。少年、青年漫画の名作を豊富に揃え、普段はFMVでゲームをしたり、様々な話題で熱く語り合ったりしているようだ。ときどき、怪しげな電気工作を遂行しているらしい。

これらのサークルについては他にも、猫の糞が落ちてた、テーブルタップが漏電して火花を出していた、プラナリアが繁殖していた、謎の展覧会が開かれたなどの怪現象が報告されているが、あまりにも長くなりそうなので割愛する。

右手奥にある魔窟こそが我が「TSG」の本拠地である。まず目に付くのが窓の上のパイプに引っかけるようにして置いてある段ボール箱の群である。地震にでもなれば確実に部員の頭上を襲う¹様に並べられているが、何の目的があるのか未だに不明である。

中央にぼっかり空いた空間には幾つものパイプ椅子が置いてあるが、それらのいくつかは座り心地が極めて悪くなるように改造されている。部員はより良き椅子を求めて静かなる椅子取りゲームを常に行っている。

左手前のロッカーは一部の部員にしか知り得ない謎が詰められている。このロッカーには「借り物帳」なるノートが下げてあり、部員は物を借りる度にここに記帳するよう義務付けられている。そして、その横には懐かしのFM TOWNSが鎮座ましましている。今でこそファミコンの真似ばかりしているが、かつては、駒場祭でTSGの危機を救った英雄である。

TOWNSの隣の黒い棚にはTSGの暗黒面が濃縮している。TSGの歴史や活動内容と深い相関があるのか、それは永遠の謎であろう。これらの資料を総て閲覧したとき、「ヲ」による精神汚染を受けるとされる。その右の木の棚には、開発のための資料が並べられていることになっている。テレビの裏の本棚にも同様な傾向が見られる。

¹こないだ頭直撃されました(T_T) (編集長)

棚を下から支えるスチール机には TSG の古き良き思い出がいっぱい詰まっていると云う。今でも高密度のジャンクの中には、ハード使いにしか判らない「おたから」が眠っているのかも知れない。

窓際の白い台は、古のすちーむ暖房の遺跡である。これが活動していた頃は、冬の午後には部室中に倦怠感が漂っていたもので、呪われた暖房は封印されて久しい。しかし、今封印を解いても活動を始めるかどうか怪しい。

木のテーブルの上には 2 台の 98(及び互換機) が並んでいる。左の 9821Xa は Pentium 90MHz を搭載したマシンである。世にも怪しげな OS、Windows95 を入れてあり、プログラムや文書を書いたり、巷ではやっている蜘蛛の巣をテストしたりできるようになっているそうだ。右の 486GR は 486SX 25MHz を載せた、今となっては古臭いマシンであるが、数々の周辺機器が繋がっていて、データのやり取りや謎の 86 音源の探求に向いているらしい。

窓際には CD ラジカセが置いてある。305 に来てから 6 年は経つ古株であるが、一応完動品であり、たまに怪しげな音楽を奏でている。下の CD ラックには変な CD ばかりが置いてあるそうだ。ここ数年の部員の煩惱がつまっていると言われる。

自然研との境界には開かずのロッカーがある。この中は実は本棚になっていて、今では読まれなくなった漫画や、TSG が過去に行った活動の痕跡が入っているらしい。かつて自然研が来る前は、そこにいたサークルが半ば幽霊サークルと化しており、我等が TSG はロッカーをずらすことによってそのサークルの領域を侵食していたらしい。尤もそのサークルの部員が誰も来ない間は、そこでまあじゃんを勝手にやっていたという伝説もある。

さて、開かずのロッカーとテーブルに挟まれた所にはテレビが積まれている。片方にはアンテナが接続しており、何故かテレビ東京が最も映りが良くなるように調整されているという専らの噂である。これらにはさらに「こんしゅーまあ機」と呼ばれるものがそれぞれ繋がっていて、時々、ある者はしゅーていんぐの道を窮め、またある者は互いに色合わせや人形操りの腕を競い合っている。

だがしかし、これらを載せているテーブルが実はとんでもない代物である。木のテーブルの上にガラス板を乗せてあるのだが、中央にブラウン管が覗いている。横から電源プラグが出ており、つなげば古めかしい電子音とともに不思議な色彩を持った架空の都市が画面に現れるのである。まさに Tennis & Ski & Game の内の Game の第一歩を印した貴重なモニュメントである。筐体を含めたゲーム一式は部の至宝であり、今となってはこのゲームが置いてあるゲーセンは数少ない。(渋谷会館という所にはあるそうだ) ゲーム史に詳しい方ならこれだけのヒントで名前を当てられるだろう。

TSG の天井には何枚かのポスターが貼られている。最も美しかった頃の南野陽子や宮沢りえの写真は、今となっては貴重品なのかも知れない。また、天井からぶら下がる滋養強壯剤のモニュメントは、かつての駒場祭の前夜に、窮地に陥ったプログラマ達が、これを飲んで宿敵「ばぐ」を打ち破った故事に由来するものである。洗礼名としてその名

を頂いた騎士もかつて居たという伝説も残っている。

宙に浮いた謎の銀色の円盤は、TSGer 一流のゆーもあの表れである。当時のメーカーにもし剛気が備わっていれば、DVD はこのサイズで既に売られていたのであろう。それだけの可能性を秘めながら、この円盤の一族は今猶「アの鉄人」達の神器としてしか使われない。

部室の奥深くにある神棚には、TSG が初めて製作したコンピュータ「くみこむ様」が奉られている。「クミコさん」なる女性にちなんで命名されたこのコンピュータは、今から比較するに本当に初歩的なことしか出来ないマシンであったが、理論科学を探究し、山登りに精を出したこのサークルに、一大転機を迫るものとなった。この時より TSG は日本のコンピュータ裏面史に名を輝かすようになり、今ではコンピューターとは斬っても切れない縁を結ぶに至っている。

天井中央から窓よりにあるシミは、「生研」に災いをなす「雨漏り」の跡である。築後 30 年以上経つこの学生会館は外装として打ちっぱなしのコンクリートを有するにも関わらず、予算の都合上外装の維持費が計上された試しがないようである。「雨垂れ石を穿つ」、屋根より出づる石中の水脈は我達が 305 の天井にまで達し、長雨が続けば天からの恵みが生研の鍋に澄んだ音色をもたらす。

TSG の生命線は未申の方角から延びている。この「えれくとろん」の源は自然研と共用しており、総ての機械に火を灯した時、305 は滅亡すると言い伝えられている。

最後に、この部室のどこかに「あけてびっくり」なる秘宝が隠されているという。部室を清めたときのみ現れるらしいが、中を覗いた者は数えるほどしかいないそうだ。

*この文章は古老の言い伝えを書写した物で、必ずしも真実を語っているわけではない。真実は自らの手で確認しなければならない。

inter-talk

八重樫 剛史

italk の開発者として、オリパンフ号にひとつ原稿を書かされることになってしまいました。でもいつものことながら、大変原稿の提出が遅れてしまったようです。

ここに書いてあること以上のことを知りたかったら、ぜひ理論科学グループに入部しましょう。入部しない人も、ぜひ italk はやってみてください。やりかたはこの原稿の中に URL が書いてある italk ホームページなどを参照して下さい。いつでもお待ちしております。

1 inter-talk の紹介

inter-talk(以下 italk と略します)というのは、Internet TELNET プロトコルを利用したチャット環境のことであり、その実体は UNIX 上で daemon として動作する italk サーバプログラムです。

italk の利用者は、サーバの動作しているマシンに telnet(1) コマンドや専用クライアント(italk.el など)を用いてアクセスし、他の利用者とリアルタイムに会話を楽しむことができます。インターネットの情報空間をダイブしまくっているTM われわれにとっては、非常に身近なシステムなのです。

とりあえず、僕が普段工学部 8 号館で italk をしている時の画面をお見せしましょう。これをみれば、italk がどんなものか、チャットするとは具体的にどういうことかがだいたいわかると思います。

この画面は 3 月 17 日の深夜に xv で撮影されたもので、画面中に現れているのは、現在 italk で活躍中の TSG のぞぎたちです。このとき自宅から電話をかけている人と、本郷あるいは駒場にいる人がいました。また、動作しているクライアントプログラムは、はらださん¹ が作った Mule 上の italk.el という Emacs Lisp プログラムです。

⁰八重樫: 機械情報工学科, t50311@mech.t.u-tokyo.ac.jp

¹基礎科学科第 2, harada@graco.c.u-tokyo.ac.jp

2 inter-talk の現状

現在 italk にはふたつのバージョンがあります。ひとつは italk1(サーバプログラム italk-server) であり、もうひとつは italk2(サーバプログラム italkd) です。

italk-server は僕が UNIX プログラミングを始めたころの産物で、性能的にも問題があるプログラムなのですが、まだこちらが用いられています。現時点 (96 年 3 月) ではくりすさん² によって、不具合を修正したり、伝言機能の追加など、機能的にいくつかの改良が加えられたもの (italk+, version 0.12) が動いています。

italkd は昨年から作る作るといいつつなかなか作られなかったのですが、今年の春休みに入ってからようやく完成に近付きつつあります。italk2 は italk1 の機能と大幅に拡張したものであり、またサーバプログラム italkd も、italk-server に較べて性能が改善されています。きっと新入生のみなさんがやってくるころには、完成して正式運用が行なわれていることでしょう。

italk2 は italk1 に較べて、大きくって次の点で機能が拡張されています。

²情報科学科, ことし就職されました。

- 多チャンネル化

italk2 ではサーバ内に「部屋」と称するものが多数存在し、異なる内容の会話を、ひとつのサーバで分けることができます。会話のログは各部屋ごとに別々に保存されます。これによってクライアントプログラムの作成が数段難しくなったものと思われる。

- 認証

サーバに認証ができるようになりました。これはユーザ名とパスワードにより、ログインしてくるユーザを個別に認識するというものです。これにあわせて伝言などの機能もつきました。

その他、ユーザコマンドなどで大幅に機能が拡充しています。結果として italk2 は italk1 とはまったく別物といえるものに仕上がりました。中には逆に使いにくくなったという人もいますが、これは機能を増強する上ではある程度やむをえないことです。

3 inter-talk に参加するには

東大の学生はみな ECC の計算機のアカウントがもらえますから、東大生ならだれでも italk サーバにログインしたり、自分用の italk サーバを立ちあげたりすることができますということになります。詳しい手順は駒場にいる先輩たちに聞くか、あるいは italk ホームページ (URL は次に書いてあります) を見てください。

チャットの持つ中毒性というのは知っている人は知っていると思いますが、すでに italk なしでは生きていけなくなった人が多数存在します (こういう人たちはさると呼ばれます)。この世界に足を踏み入れる時には十分気をつけてください。家から利用する人は特に電話代に気をつけましょう。ECC に対するテレホーダイは非常に有効ですが、つなぎっぱなしは他のユーザを困らせることになりますのでほどほどにしましょう。

さて、italk に関するもっと詳しい説明、最新情報については、こちら (italk ホームページ) をごらんください。

<http://www.ecc.u-tokyo.ac.jp/~t50311/italk/>

ここには italk の使用法、italk2 のチュートリアルや、あやしげな用語解説などがあります。

次号では、この italk ホームページにアクセスできない人 (いるのだろうか)、したことのない人のために、その内容を一挙公開しましょう (部報のページ稼ぎともいわれるが...)。

4 inter-talk の位置付けと今後の展開

さて、オリパンプ的な内容は実をいうとこれでおしまいです。ここからは italk に関する僕の考え方や、計画を書こうと思います。その多くは僕のたわごとですので、おかしいと思う人は笑いとはしてもらってかまいません。

italk は昨年の夏あたりから開発を始めてきましたが、italk2 サーバ italkd の完成により、そのソースプログラムを公開・配布する予定でいます。しかしなにも考えずに公開しても、みんなに使ってもらえないかもしれません。すでに一般に普及しているチャット環境があるからです。

Internet 上のチャット環境として非常に有名なものに、IRC(Internet Relay Chat) があります。当初 UNIX の talk プログラムの代替としてフィンランドで開発がスタートしたもので、すでにその RFC もあります。その機能は、複数のサーバ間でメッセージをバケツリレーのように配送することにより、全世界をリアルタイムに(?)³ つなぐというものです。

IRC と italk がリアルタイムなチャット環境を提供するという点ではどちらも同じです。IRC で用が足りてる人は italk は必要ないかもしれません。このような状況ではだれも italk を使ってくれないかもしれない、と僕を含めた多くの人は考えています。

italk を普及させるには italk にはあって IRC にはない点をなにか探す必要があるのですが、とりたてて italk が (ここでは italk2 とします) IRC より優れている点を指摘するとすれば、次があげられると思います。

- 会話のログがサーバに残ります。
以前行なわれた会話のログをサーバに要求して、読み出すことができます。また、会話の途中に入ってきて、以前どのようなことが話されていたのかわかります。
- 特別なクライアントプログラムを必要としません。
インターフェースが単純ですので、Internet につながっているマシンであれば、italk サーバには telnet(1) コマンドでそのままアクセスできます。また、別に UNIX を使っている必要もありません。
- 認証機能を提供します。
italk2 では認証機能があり、ログインしてきたユーザを個別に識別でき、これを用いたメッセージの伝言機能があります。ログインしていない相手にメッセージが確実に伝えられます。
- ローカルなチャット環境を提供します。
IRC とちがって全世界に接続できるわけではありませんが、italk サーバプログラム

³発言の返事が数十時間後に帰ってくることもあるそうです。

は非常に小さくて軽いものです。IRC にくらべてサーバの導入も非常に簡単です。

このように、italk にあって IRC にはない特徴は結構多いものです⁴。これなら、italk のような小さなシステムでもその需要をほりおこすことは可能だと考えられます。IRC と住み分けることは十分に可能ではないでしょうか。

今後の italk の開発および配布の展開として、次のような計画を立てています。

- サーバプログラム italkd の公開

italkd が完成ししだい、ソースを fj.sources あたりで公開することにします。これが第一歩です。

- www-talk の完成

italkd はそのログがサーバで保存されます。これと CGI 機能を利用して、WWW ブラウザからアクセス可能な www-talk というシステムを作ります。すでにこれを実現した WWW ページが存在し、これは

<http://www.ecc.u-tokyo.ac.jp/~t50311/italk/wwwtalk.html>

にて参照可能です。

- BBS 的な性質をもたせる

ログ保存や認証システムなどを活用して、サーバごとの BBS 的な性質を強めていくつもりです。

- サーバ間中継機能の追加

(別に勝つつもりはないのですが) IRC に勝とうと思ったら、これをやらなくてはならないでしょう。ログを保存という要求がこの機能の実現を難しいものにしています(逆に、ログをとらなければ実現は比較的容易です)。

ここまできたら italk の RFC も作らなくてはならないでしょう。しかしいったいいつにやることやら...

そういうわけで、これからの道のりはやはり長くけわしいようです。これからも、みなさんの変わらぬ支援をよろしくお願いします(なんだそりゃ)。しかしこれからは本業の勉強のほうもやらなくちゃいけないので、これらが本当に達成されるかどうかは、だれにもわかりません⁵。

⁴ただし、これらの点のいくつかは IRC でも bot と呼ばれる IRC クライアントプログラム(italk である“人工無能”のようなもの)を用いれば、原理的には実現できます(italk1 でも、いくつかの機能をそうやって実現していました)。しかしかなり面倒で不便なものとなるでしょう。

⁵ごく一部で、僕の卒論が italk サーバの開発になるなどという話がまことしやかに語られていますが、これは完全なデマですので、信用してはいけません。

一般記事

PCM への誘い

あじ

1 はじめに

最近、音関連で PCM という言葉をよく耳にしますね。

DAT や CD、MD の録音形式から電話回線、Windows の WAVE ファイルに MIDI 規格音源。一昔前まで、やれアナログだの FM 音源だの言ってたのものは、ことごとく PCM 方式になってしまいました。

PCM とは一体何なのでしょう？

PCM とは「Pulse Code Modulation」の略で、音はアナログの連続量なのですが、これを瞬時値 (pulse) の離散的な連続に変換し、それぞれの瞬時値をデジタル (code) で表す変調¹ (modulation) だということです。²

簡単に言えば、音の波形をやたらと沢山の数字の羅列で表すわけです。

何でこんなもんが流行っているかといえば、数字の羅列にするとコンピュータで扱えるので、色々と便利なことがあるからです。³

ま、これが TSG の部報に載ってる記事だと言うことから明白ではありますが (笑)

ここまで聞いて、「そんな『変』調かけたら雑音が載るんじゃないか、それに再生はどーすんだオイ」と思った人はいるでしょうか？

いたら、その人はかなり鋭い人です。鋭いけど、かなりの天邪鬼ですね (笑)

CD を聞いたりとか、「通話料が安くて音がいい」デジタルホンを使ったりとかしたことはないんでしょうか(^^;

別にいいですけど (笑)

実際問題として、色々な理論や技術に支えられて音を PCM として扱うことができるようになったわけなのですが、画像と違って音は目で見て分かるわけではないし、アニメ絵のような扱いやすいデータを作ることもできないので、PCM を扱うというのはかなり敷居の高い作業になってしまっています。

¹ 便利な形に信号を変換すること

² パルス変調には他にも PAM, PNM, PPM, PWM などがあります。

³ 音質が劣化しない？ そりゃ、単純に保存と再生だけ考えりゃそんな気がするかもしれませんが、TSG 部員がそんな悲しいことを言っちゃダメですよ！)

しかし、信号処理独特のものの見方に慣れれば、PCM のとても面白い世界が見えてくるはずですよ。

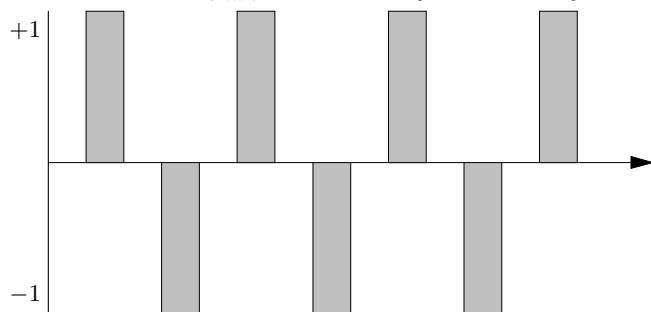
本記事は「ほんのさわり」でしかありませんが、これをきっかけにあなたのプログラミングライフに「愉快的 PCM (笑)」が加わることになれば幸いです:)

2 パルス変調ということ

さて、PCM では一定時間毎に波形の瞬時値をひろってくる (サンプリング) わけですが、この変復調について考えてみましょう。

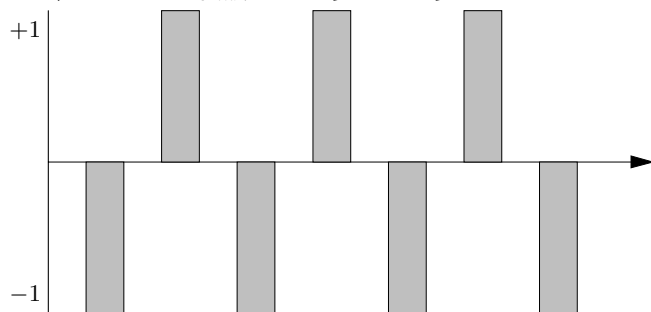
例として、瞬時値が 0 になった時点から 1 秒間に 8 千回の頻度でサンプリングする場合を考えてみます。

ここに 2kHz の正弦波を入れたらどうなるでしょう？



ですね。

では、6kHz の正弦波ではどうでしょう？



おや？

2kHz と同じ形になってしまいました。

よ～く考えれば、これは単に

$$\sin\left(\frac{1}{4} \cdot 2\pi t\right) = -\sin\left(\frac{3}{4} \cdot 2\pi t\right) \quad t = 0, 1, 2, 3 \dots$$

なだけなので、当たり前ですね(^_^);

同様に 1kHz, 7kHz, 9kHz... とか 3kHz, 5kHz, 11kHz... などとも区別が付きません。
 というわけで、詳しい説明は省きますが

標本化定理 「 $0 \sim f$ の周波数成分⁴のみを含む信号は、サンプリング周波数 $2f$ の標本
 点で一意に決まる」

という大定理があるわけです。

これを踏まえて、PCM 関連のデバイス、ソフトウェアは全て

「サンプリング周波数が $2f$ の時、 f までの周波数成分しか含まれていない」

ように作られています。

変調器には必ず高周波成分をカットするローパスフィルタが入っているし、復調器は
 $0 \sim f$ の周波数成分のみを持つ信号を復元するように色々と工夫がなされているわけ
 です。

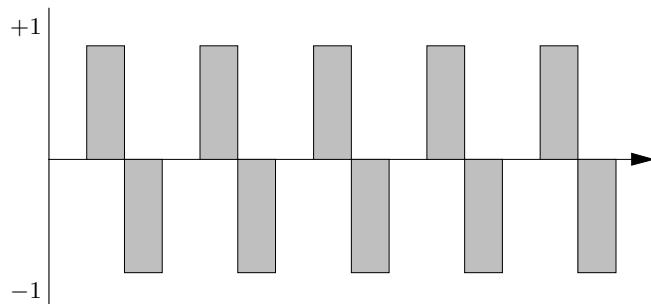
ちなみに、 f より大きい周波数成分を周波数 $2f$ でサンプリングすると高周波成分は
 $0 \sim f$ に綺麗に折り畳まれてしまうので「折り返し雑音」と呼ばれます。

さて、変復調のイメージもつかめた所でちょっと練習:)

8kHz の矩形波 $\{-1, +1\}$ を、まともなサンブラで 24kHz にてサンプリングしま
 した。

サンプリングされたデータはどのようになるでしょう？

答. 8kHz の矩形波は 8kHz, 16kHz, 24kHz ... の周波数成分を含みます。⁵ 12kHz より
 大きい高調波はサンブラに除去されるので、結局サンプリングされるのは 8kHz の
 正弦波だけになります。



⁴信号を、周波数の異なる正弦波の和に分解した時の係数のことです。新入生にはきつかな(^_^);

⁵矩形波をフーリエ変換してみましょう。フーリエ変換とは信号を正弦波の周波数領域に写す変換なのですが、詳しくは数学の本でも見てください(オ

3 デジタルとして

波形の瞬時値を取ってくるのはいいのですが、まだこの段階では「時間的に離散なアナログ量」です。これをコンピュータが扱えるようにするには A-D 変換⁶ を施す必要があります。

A-D 変換にも色々な技術があるのですが、ここではそれは置いてうまいこと離散量になったとして(^^;、そこから先の取り扱いについて考えてみます。

アナログ入力が $-1 \sim +1$ の時、これを $\delta = 1/128$ のきざみで離散量に変換したとすると、うまく一番近い値が出力されていたとしても $\delta/2 = 1/256$ の誤差がでてしまいます。これを量子化誤差と言います。

一般的にデジタルの音が良いと言われるのは、デジタル回路を多少いい加減に作っても、誤差を理論的に押さえ込む事ができるからです。アナログだと精度が回路の出来や環境に左右されますからね:)

ま、デジタルでも電磁波とか 線⁷ とかで誤動作する可能性はあるんですが(笑)

この原稿を読んでいる人には浮動小数点の扱いなどでお馴染みだと思うのですが、コンピュータで離散量を扱う以上、量子化誤差や \sin , \log などの無限項必要な計算の有限項での打ちきりによる誤差、似たような信号の差での S/N 比⁸ の悪化などは避けられません。

詳しい所は数値解析の教科書に譲りますが、PCM を扱うには誤差についての知識も必要なわけです。

4 音と音声

ちょっと一服して、音について考えてみましょう。

人間は一体、音をどういう風に知覚しているのでしょうか？

生理学的なことはよく知らないのですが、数十 Hz ~ 十数 kHz の「周波数成分」を感じ取っているのだそうです。

CD のサンプリング周波数は 44.1kHz、DAT や MD は 48kHz なのですが、これは可聴域の周波数成分を取り込むのに十分な速度であるわけです。

⁶何の事はない、アナログ-デジタル変換です。

⁷他のものはモールドを工夫したりとかで何とかありますが、IC 自身の原子崩壊でできる 線は素子を大きくして誤動作の確率を下げる以外どうしようもないですね。(^^;

⁸信号対雑音比

ところで「音の信号処理」と「音声処理」と言う言葉を聞いた事ありませんか？
これらの違いは一体何なのでしょう。

人間が声を出すというのは、「声道(口)の形を変化させ、それにあわせて音源(声帯)を振動させて共鳴を起こし、空気中に指向性を持った音波を発射する」行為で、その波形は 100 ~ 300 Hz の特定波形の繰り返しになっています。

これを人間が感じる周波数成分から見れば、ある基本波とその高調波の組み合わせパターンで音声識別されていることが分かります。

「音声処理」というのは、発音されている音声に対しては短いけれども波形を解析するには十分長い部分を切り出して、様々な波形を解析することをベースに行う処理なのです。

一方、「音の信号処理」では音の遷移なども芸術性のうちなので、連続的な処理を行います。とはいえ、部分的に切り出して波形の解析・操作を行う処理は便利なので、うまく繋ぎ合わせて戻せるようにする研究も色々と行われています。

それから、「音の信号処理」では周波数成分によって位相の遅れ方の異なるような処理は当然御法度になります。⁹

5 何かやってみる

ここまでうだうだと PCM について話をしてきましたが、コンピュータサークルたる理論科学グループとしては、この辺でコンピュータによる処理について話をしなければいけない(?) でしょう。

というわけで、コンピュータで PCM データをいじっているとちょくちょく出てくる処理として、サンプリング周波数のダウンコンバートについて考えてみます。

まず、真正面から考えます。

周波数領域で作業するのだからフーリエ変換を行った後、低域成分だけで逆変換してやればいいはずですが。

しかし！

数十 MB もあるような PCM データ全体に対してフーリエ変換を行うことは、必要な精度、計算量とも悪夢のようなものであることは容易に想像がつきます。

細切れに処理して繋ぎ合わせる？

まあ、かなり難易度は高いでしょうができないことはないでしょう。

もっと簡単な処理でできないでしょうか？

⁹周波数成分によって遅れが変わらない性質を直線位相特性と言います。位相の遅れで見れば、周波数に比例しているからですね。

幸い、PCM データを一定個数置きに間引くという処理に関しては、折り返し雑音以外は入らないことが知られています。これを利用して何とかしましょう。

まず、元データのサンプリング周期単位でアナログ信号モデルを設定して繋ぎ合わせたものを考えます。

ここでは、次のサンプル時刻まで値をホールドするモデルを考えましょう。

すると、考えたアナログ波形は直観的に

$$(\text{元データの成分}) + (\text{サンプリング周波数}/2 \text{ を基本波とする成分})^{10}$$

であることが分かります。

これより、アナログ波形のモデルから 変換先サンプリング周波数 x_n (n は正整数) の周波数でサンプリングしたデータを作り高周波成分を除去 間引き、という方針が成り立ちます。

ただし、このモデルに対してサンプリングの前処理としての高域除去はできないので、 n を十分大きくして折り返し成分を十分小さくします。¹¹

次に高周波成分の除去ですが、これはデジタルのローパスフィルタで行います。¹²

しかし間引きを行うためには、変換先サンプリング周波数/2 以下の成分は殆ど完全に通り、それより大きい成分は殆ど完全に遮断しなくてはなりません。それでいて遮断周波数が $1/n$ のフィルタというのは恐ろしく実現が難しいので、ここでは 2 段階に分けてみます。

n を 4 の倍数¹³ とし、初段では $4/n$ まで高域除去&間引きを行います。こうすれば、初段のフィルタは $1/n$ 以下の低域を完全に通り、 $7/n$ 以上の高域は完全に遮断する様なフィルタでさえあれば $1/n$ 以下の低域は折り返し雑音に汚染されないわけです (笑)

で、次段では $1/4$ の急峻なフィルタを気合を入れて作ります。

初段のフィルタでは入力信号は一定時間ホールドされているので、係数を予めまとめておけば劇的に計算量が減ります。また、計算誤差も小さくなります:)

次段も、間引き処理とセットになっているから出力に必要な項だけ計算すればいいので、それ程計算量はいいりません。¹⁴

これだけ工夫すれば、Pentium クラスの CPU や DSP なら実時間の処理も可能になりますね:)

¹⁰ AM 変調がかかっているとみなせば側波帯を持つことが予想されます。DPCM の包絡線を考えると、値をホールドするモデルでは結構側波帯が広いですね (^_^;。基本波の下側波帯はノイズになって残ってしまうので、モデルをもう少し工夫して側波帯を小さくした方がいいかもしれません。

¹¹ 高調波は周波数が高いほど小さくなります。

¹² フィルタは入力データに適当な係数を掛けて足し合わせて出力するというだけのものですが、信号処理の重要な基礎技術です。デジタルフィルタの入門書は本屋に沢山置いてあるので一度読んでみましょう。ちなみに、音の信号処理では直線位相特性が必要なので FIR フィルタを使います。

¹³ 別に 3 でも 5 でもいいんですが (^_^;

¹⁴ こういうフィルタをデシメーションフィルタと言います。方式の A-D コンバータでは、64 倍の周波数のデルタ変調信号の積分に $1/4$ のデシメーションフィルタを 3 段がける、というのが一般的だったりします。

6 おわりに

現在のパソコンは、高度な信号処理を実用的な時間で行うにはあまりにも非力です。教科書通りの処理ではまさしく日が暮れることもあるでしょう(^^;

しかし 5 章で紹介したように、信号処理の基礎程度のローテクでも工夫次第では予想以上に良い結果が得られる事があったりします:)

PCM 関連ソフトウェアの実装では音質と計算時間のトレードオフがたまたまなく愉快です。是非一度 PCM 関連のツールを作ってみることをおすすめします。

大容量デバイス、高速な CPU、Windows、インターネット、CD で供給されるゲーム(^^; など、パソコンで PCM を扱う環境は急速に充実してきていますしね

7 あとがき

新入生向けの入門記事ということで、出来るだけとつきやすいようにと専門用語はなるべく平易な日本語に置き換えるように努力しましたが、どんなものでしょうか？

5 章ではいきなり怪しい内容になったような気がします、気にしないことにしましょう(^^;

質問等がありましたら、メールをくれるなり、いぬ。に書き込むなり、根津研に押しかけるなり (笑) してください。僕に分かることなら何でも OK です。:->

ごみ.JPG

TEA

1 はじめに

どもTEAです。6月13日に98用のビデオ取り込みボードを買いました。カノープスのIMAGEPAQ-98なのですが、それでいろいろ取り込んでいたら、jpegローダーが作りたくなりました。ずっと前から、いつかは作りたいなあと考えていたのですが、仕組みがよくわからないし、PC-286だと16色しかでないので作る気が起きませんでした。

jpegローダーを作ろうと決めたのは、東京BBSでしゅいらさんやARGONさんが小さいjpegローダーを作っていて、実はjpegは簡単なのかなと思ったからですが、その他にも、去年の駒祭のころにICHI君がMATE用のjpegローダーを作っていて、256色でも結構見られるんだなあとしたこととか、ぼくがMATEを買って256色モードが使えるようになったこともあります。あと、IMAGEPAQについてきたjpegローダーがかなり高速で、もえ~だったというのがあります。

というわけで、これから何回かに分けて、jpegローダーの作り方を書こうと思います。jpegの表示には、ハフマン符号の復号、IDCT、RGBへの変換、減色が必要ですが、まず1回目は、一番時間がかかるとされる、IDCT(逆離散コサイン変換)について書こうと思います。

2 IDCT

IDCTの処理では、周波数で表している波を普通の2次元の画像に戻します。jpegでは、この処理を8×8ドットのブロックで行います。IDCTは、次のような式で表されます。

$$S'_{yu} = \sum_{v=0}^7 (F_{vu}/8) \cdot C_v \cdot \cos \frac{(2y+1)v\pi}{16}$$

$$S_{yx} = \sum_{u=0}^7 S'_{yu} \cdot C_u \cdot \cos \frac{(2x+1)u\pi}{16}$$

$$C_v = \begin{cases} 1 & (v=0) \\ \sqrt{2} & (v \neq 0) \end{cases}$$

S'_{yu} と S_{yx} の式はほとんど同じなので、後者だけ考えます。三角関数を、 $c v = \sqrt{2} \cos(v\pi/16)$ とすると、 S_{yx} は次のように表せます。

$$\begin{pmatrix} S_{y0} \\ S_{y1} \\ S_{y2} \\ S_{y3} \\ S_{y4} \\ S_{y5} \\ S_{y6} \\ S_{y7} \end{pmatrix} = \begin{pmatrix} c4 & c1 & c2 & c3 & c4 & c5 & c6 & c7 \\ c4 & c3 & c6 & -c7 & -c4 & -c1 & -c2 & -c5 \\ c4 & c5 & -c6 & -c1 & -c4 & c7 & c2 & c3 \\ c4 & c7 & -c2 & -c5 & c4 & c3 & -c6 & -c1 \\ c4 & -c7 & -c2 & c5 & c4 & -c3 & -c6 & c1 \\ c4 & -c5 & -c6 & c1 & -c4 & -c7 & c2 & -c3 \\ c4 & -c3 & c6 & c7 & -c4 & c1 & -c2 & c5 \\ c4 & -c1 & c2 & -c3 & c4 & -c5 & c6 & -c7 \end{pmatrix} \begin{pmatrix} S'_{0y} \\ S'_{1y} \\ S'_{2y} \\ S'_{3y} \\ S'_{4y} \\ S'_{5y} \\ S'_{6y} \\ S'_{7y} \end{pmatrix}$$

よって、この式のまま計算すると 64 回の掛け算が必要となります。そこで、この式を変形して計算を簡単にしたいわけです。

まず、 $c4 = 1$ ですから、ここでは掛け算はいりません。また、この行列に現れる係数は、 $c1, c2, c3, c5, c6, c7$ の 6 種類しかないので、掛け算用のテーブルを作ることできます。次に、上の行列を偶数列と奇数列に分けると、次のようになります。

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} c4 & c2 & c4 & c6 \\ c4 & c6 & -c4 & -c2 \\ c4 & -c6 & -c4 & c2 \\ c4 & -c2 & c4 & -c6 \end{pmatrix} \begin{pmatrix} S'_{0y} \\ S'_{2y} \\ S'_{4y} \\ S'_{6y} \end{pmatrix} \quad \begin{matrix} S_{y0} = a_0 + b_0 \\ S_{y1} = a_1 + b_1 \\ S_{y2} = a_2 + b_2 \\ S_{y3} = a_3 + b_3 \end{matrix}$$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} c1 & c3 & c5 & c7 \\ c3 & -c7 & -c1 & -c5 \\ c5 & -c1 & c7 & c3 \\ c7 & -c5 & c3 & -c1 \end{pmatrix} \begin{pmatrix} S'_{1y} \\ S'_{3y} \\ S'_{5y} \\ S'_{7y} \end{pmatrix} \quad \begin{matrix} S_{y4} = a_3 - b_3 \\ S_{y5} = a_2 - b_2 \\ S_{y6} = a_1 - b_1 \\ S_{y7} = a_0 - b_0 \end{matrix}$$

つまり、もとの行列の上半分と下半分が全く同じか符号が違うだけなので、一度行った計算が他のところに使えるようになり、掛け算を減らすことができるわけです。偶数列

に対しては、同じように 4 で割った余りで列を分ければさらに掛け算が減ります。奇数列も同じようなことができますが、掛ける数の種類が増えてしまうのでやめておきます。

以上のようにすると、 S_{y0}, \dots, S_{y7} の計算は 20 回の掛け算でできることになります。え？ 掛け算は 11 回ですむはずだって？ うーむ。まあいいでしょう (笑)。

3 いらぬ計算を無くそう

さて、ここからが本題だったりします。次回以降に説明しますが、jpeg では、周波数のデータは量子化 (スケーリングと言ったほうがわかりやすいかな) されているので、 F_{vu} のなかには 0 になっているものが多いわけです。そして、 F_{vu} は、次の順番でビットストリームに入っています。

$$\begin{pmatrix} 0 & 1 & 5 & 6 & 14 & 15 & 27 & 28 \\ 2 & 4 & 7 & 13 & 16 & 26 & 29 & 42 \\ 3 & 8 & 12 & 17 & 25 & 30 & 41 & 43 \\ 9 & 11 & 18 & 24 & 31 & 40 & 44 & 53 \\ 10 & 19 & 23 & 32 & 39 & 45 & 52 & 54 \\ 20 & 22 & 33 & 38 & 46 & 51 & 55 & 60 \\ 21 & 34 & 37 & 47 & 50 & 56 & 59 & 61 \\ 35 & 36 & 48 & 49 & 57 & 58 & 62 & 63 \end{pmatrix}$$

つかれた (笑)。つまり、 $F_{00}, F_{01}, F_{10}, F_{20}, F_{11}, F_{20}, \dots$ という順番に入っています。そして、終了の符号のあとはすべて 0 になっています。たとえば 35 (F_{70}) で終わりだとすると、行列の右下半分はすべて 0 になっているわけです。よって、データの数によって計算するルーチンを分ければいいのです。

例えば、 F_{00} だけが 0 でないとすると、 $S'_{00} = S'_{01} = S'_{02} = S'_{03} = S'_{04} = S'_{05} = S'_{06} = S'_{07} = F_{00}$ となり、ほかの S'_{yu} は 0 になります。jpeg で圧縮率が良い絵の場合、この周波数のデータが 3 つしかないというところがわりとあります。ですからデータによってルーチンを分ける方法はかなり有効のようです。

さて、ぼくのプログラムがどうなっているかです。まず F_{vu} から S'_{yu} を計算する部分ですが、64 通りのルーチンを作るとなんかプログラムがでかくなりそうでいやなので (なんか昔と考え方が違うな)、行に関して上から何番目まで 0 でないデータが入っているかによって 8 通りのルーチンを作り、それを 8 回呼ぶようにしました。例えば、0 から 5 までデータが入っている場合は、1 列目が 3 個、2 列目が 2 個、3 列目が 1 個なのでそれぞれに対応するルーチンを呼ぶようになっています。

S'_{yu} から S_{yx} を計算する部分は、上とは違って各列ごとに 0 でないデータの数が違うなんてことはないのですが、ある行より下は全て 0 になっていることがあります。上の 0 から 5 までの例では、 F_{vu} が 3 列目までしかないので、 S'_{yu} は 3 行目までしかないということになります。よってこちらは 64 通りではなく 8 通りに分けるわけです。なお、こうすることで S'_{yu} に現れる 0 をメモリに書かなくてすむことにも注意してください。ついでに、 F_{00} だけが 0 でないばあいは S_{yx} は 64 個全てが $F_{00}/8$ と等しくなるので、この場合だけはサブルーチンを 8 + 1 回呼ぶようにはせずに専用のルーチンにしています。

4 実際のプログラム

F_{vu} に 0 以外のデータが 6 つ (0 から 5) のときのプログラムの一部を載せます (核爆)。

```

even
de05:
        push    offset idctsub22
        push    offset idctsub10
        push    offset idctsub11
        jmp     idctsub12

even
idctsub22:
        assume ds:_tbl,es:_tbl,ss:_dat
        mov     di,blk
        mov     si,offset dcttmp
        mov     di,dctob[di]
        mov     k,8
id221:  mov     bx,[si+2*16]
        add     bx,bx
        mov     cx,[si+0*16]
        mov     ax,dctmul2[bx] ; c2
        mov     bp,dctmul6[bx] ; c6
        mov     bx,[si+1*16]
        add     bx,bx
        add     cx,ax
    
```

```
mov     dx,dctmul1[bx]
add     cx,dx
mov     [di+0*2],cx
sub     cx,dx
sub     cx,dx
mov     [di+7*2],cx
add     cx,dx
sub     cx,ax
sub     cx,ax
mov     dx,dctmul7[bx]
add     cx,dx
mov     [di+3*2],cx
sub     cx,dx
sub     cx,dx
mov     [di+4*2],cx
add     cx,dx
add     cx,ax
add     cx,bp
mov     dx,dctmul3[bx]
add     cx,dx
mov     [di+1*2],cx
sub     cx,dx
sub     cx,dx
mov     [di+6*2],cx
add     cx,dx
sub     cx,bp
sub     cx,bp
mov     dx,dctmul5[bx]
add     cx,dx
mov     [di+2*2],cx
sub     cx,dx
sub     cx,dx
mov     [di+5*2],cx
add     si,2
add     di,16
dec     k
jnz     short id221
```

ret

5 次回の予定

今回は行列とかソースばかりで、文字が少なかったですね。次回はハフマン符号の復号のところを書こうと思います。

6 連絡先

感想などは、sada@is.s.u-tokyo.ac.jp までお願いします。また、ぼくの作ったプログラムは <http://naomi.is.s.u-tokyo.ac.jp/~sada/> にありますので使ってみてください。

初心者のための 8086 講座

高野

1 パソコンの脳味噌

パソコンの脳味噌に当たる部分を CPU (Central Processing Unit) と言います。CPU は単に数値の計算を行うほか、データの読み書きや判断、入出力等の処理を行います。現在では様々な CPU が開発されていて、有名な物としては Intel 社の Pentium やモトローラ社の Power PC などがあります。ここでは今日パソコンにもっとも多く使われている Intel 社製の Pentium の元となった 8086 について解説します。

私たちはキーボードやマウスなどによってパソコンを操作しますが、これはキーボードやマウスを使うためのプログラムがあらかじめ用意してあるためです。そのプログラムはマシン語と言われるもので書かれています。コンピュータはそのマシン語でしか働くことができません。マシン語は普通の人間にはとてもわかりづらいので、様々なプログラムを作るときには、C 言語やベーシックと呼ばれる、人間にわかりやすいプログラミングのための言語を使ってプログラミングをします。そしてそのままではコンピュータはそのプログラムを解読することはできないので、コンパイラやインタプリタと呼ばれるマシン語変換プログラムによってコンピュータを働かせることができます。

2 クロック

CPU の処理速度を測る上で重要なファクターとなってくるのが、このクロックです。コンピュータは CPU 以外の部品とタイミングを取るために、クロックという同期信号によってシステム全体の動作のタイミングを取っています。一般にこのクロック周波数が高ければ高いほどそのコンピュータは高い処理能力を発揮するわけです。また、CPU の命令はその命令ごとに実行に必要なクロック数が決まっており、クロック数が少ない命令ほどその命令の負担は軽いこととなります。

3 2進法

我々は物を数えるとき、10を一まとまりとして扱うことが多いのですが、コンピュータの世界では0と1しか存在しません。数字をすべて2進法としてあつかっているわけです。0と1の二つの状態しか持たない情報量のことをビット(Bit)と言います。コンピュータは情報を伝えるときにはこのビットを単位としています。つまり、信号線においては電圧の高低という二つの状態で区別されて情報が処理されているわけです。信号線が n 本ある時 2^n 乗通りの区別ができます。その時の情報量を n ビットの情報量があると言います。8ビットの情報を1つの基本単位として、1バイト(Byte)と呼びます。

4 16進法

このように普通マシン語は2進法で書かれているのですが、一般に我々がマシン語を見るとこれを16進法として見る人が多いです。16進法というのは一つの桁で0から15までの16種類の数値を表します。一桁で10から15の値も表さなくてはならないので、AからFまでの英文字を数字として使います。たとえばAは10、Bは11と言う具合です。16進法であることを明確にするために後ろに「H」(Hexadecimal notation)をつけたり「0x」を前につけたりします。

5 2進法と16進法

ではなぜいちいち16進法にするのでしょうか？2進法において4ビットは0から15までの数を区別することができます。だから2進法では1バイトを8けた必要なのに対し、16進法は2桁ですむわけです。たとえば2進法において10010111は16進法では97Hと表され、非常にすっきりと見やすくなります。

10進法と2進法と16進法の対応を表にしてみます。

10 進法	16 進法	2 進法
0	0H	0000
1	1H	0001
2	2H	0010
3	3H	0011
4	4H	0100
5	5H	0101
6	6H	0110
7	7H	0111
8	8H	1000
9	9H	1001
10	AH	1010
11	BH	1011
12	CH	1100
13	DH	1101
14	EH	1110
15	FH	1111

16 進法と 2 進法の相互の変換は、2 進法で 4 桁ごとに区切って考えて、上の表に対応させるだけなので非常に簡単です。

6 キロとメガ

普通 1 キログラムというと 1000 グラムのことを指します。しかしコンピュータにおいて 1 キロビットと言うと普通 1024 ビットのことを指します。これはコンピュータの場合、 2^{10} で 1024、16 進法で 400H と非常に区切りのいい数字になるからです。これと同様に 1 メガバイトとは 1000 キロバイトのことではなく、1024 キロバイトのことを指します。

7 メモリ

CPU はデータの受け渡しを「データバス」と呼ばれる信号線をとおして行われます。電線がゲジゲジの IC から何本も出ていることを見たことがあるでしょう。何本かの電気信号が通る道の束と思っていただければいいです。

データの受け渡しが行うには、どこにデータが存在するかが、わからなくてはなりません。例えば、NHK の集金係の人がお金をもらいに行くのにも、その人のうちの住所（アドレス）を知らなくてはお金を集めてくることができません。データのアドレスを、今度は「アドレスバス」と呼ばれる信号線をとおして、データの場所を指定してやることで、CPU は初めてデータを読み書きできるわけです。データの場所は数字で指定されます。（もちろん 2 進法です。）

データは常にメモリに記憶されています。メモリには大きく分けて ROM と RAM というものがあります。前者は Read Only Memory の略称で、その名の通り読み込み専用のメモリです。後者は Random Access Memory の略称で、読み込みもできれば書き込みもできるメモリです。ここで注意して欲しいのは、メモリはフロッピーディスクやハードディスクとは全く別物であるということです。ROM は読み込むことしかできない代わりに、パソコンの電源を切っても記憶は消えません。一方 RAM は読み書きは両方ともできるのですが、電源を切ると消えてしまいます。一般に、ROM はコンピュータに電源をいれたとき一番はじめに実行されるプログラムや、BIOS (Basic Input Output System) と呼ばれる基本的な入出力を行うプログラムが納められています。RAM は CPU のユーザープログラムや、処理するための各種データエリアなどに使われます。

8 ちょっと寄り道

RAM には大きく分けて SRAM と DRAM に分類されます。SRAM は非常に読み書きのスピードが速いのですが、構造上の問題から集積度があげにくく、一般に高価です。DRAM は構造が非常に単純なので集積度はあげやすいのですが、そのかわり読み書きのスピードが遅いのです。最近では DRAM のこの欠点を解消するために、普通の DRAM の代わりに EDORAM などが使われ始めています。

9 I / O

フロッピーディスクやハードディスクは電源を切っても内容は保存されますが、RAM や ROM とは違って直接 CPU に繋がっているのではなく、I/O ポートを通してデータのやりとりを行っています。では I/O ポートとはいったい何なのでしょう？

I/O とは日本最古のパソコン雑誌のことではありません。Input/Output System の略称、すなわち入出力装置のことです。CPU は大部分の時間をメモリ間の処理に費やしますが、それだけでは何の結果も知ることができません。モニターやプリンタなどに出力することで、はじめて結果を知ることができます。このような信号のやりとりの接点に当たる部分が I/O ポートと言います。I/O ポートには普通モニタやプリンタなどのほかに、フロッピーディスクやハードディスクなどが接続されています。ゲームなどの音楽に使われる FM 音源ボードもそうです。8086 の場合、これらの周辺機器にアクセスするには、I/O アドレスと呼ばれる I/O 専用のアドレスでアクセスします。8086 は 16 ビットの I/O アドレスを持っています。メモリのアドレスとは全く別物なので混同しないようにして下さい。(モトローラ社の 68k の CPU はメモリのアドレスと I/O のアドレスが混在しています。)

10 CPU の中身はどうなっているのか

以上で簡単にパソコンの中身の構造を簡単に説明しましたが、CPU の中身はどうなっているのでしょうか？

CPU はバスを介してメモリにつながっていますが、このメモリとは別に CPU の内部にもメモリが存在し、それをレジスタ (Register) とよびます。レジスタには 8 ビット、16 ビット、32 ビットのデータを格納できるようになっています。レジスタは RAM などのメモリと違ってアドレスで区別しません。レジスタには個別に名前が付けられており、8086 系ではそのレジスタごとの役割が決められています。(80386 以上の CPU ではだいぶ自由になりましたが...) CPU は普通、データをいったんレジスタに読み込んでからメモリや I / O に転送したり、演算を行ったりします。8086 系には一般に 14 種類のレジスタが存在します。それぞれの役割を簡単に示します。

《汎用レジスタ群》

レジスタ	名称	機能
AX	アキュムレータ	主に演算。
BX	ベースレジスタ	特定のメモリを指し示すポインタ。
CX	カウントレジスタ	その名の通り転送や繰り返しの回数のカウンタ。
DX	データレジスタ	データの一時記憶や AX と一緒に 32 ビット演算に用いられます。

《インデックスレジスタ群、特殊レジスタ群》

SI	ソースインデックス
DI	ディスティネーションインデックス
BP	ベースポインタ
SP	スタックポインタ
IP	インストラクションポインタ

上のレジスタは汎用レジスタと似たような機能を持っているのですが、直接レジスタの値をそのまま利用するのではなく、メモリのレジスタを指定するためのポインタとして用いられます。SP と IP はユーザーが直接プログラム中でアドレスを指定することができません。これらはプログラムの実行などを制御するためのレジスタとして、間接的に利用するからです。

《セグメントレジスタ群》

CS	コードセグメント	プログラムの格納されているセグメントアドレスを示します。
DS	データセグメント	データの格納されているセグメントアドレスを示します。
ES	エクストラセグメント	DS 以外のデータを格納しているセグメントアドレスを示します。
SS	スタックセグメント	スタック専用のセグメントアドレスを示します

これらは上のインデックスレジスタや、特殊レジスタと組み合わせることによってメモリのアドレスを示すのに利用します。このセグメントの使用法については、8086 系独自のメモリ管理方式を理解する必要がありますが、それは後で説明します。

《フラグレジスタ》

このレジスタは、プログラムによって変化する CPU の状態を表すためのレジスタとして利用されます。たとえば演算結果がゼロになったり、演算で

桁あふれが生じたときなどこのレジスタを調べることでわかります。

11 セグメント

8086 は 1 メガバイトのメモリを使うことができます。これは最近ではおせじにも大容量とはいえませんが、当時主流だった 8 ビット CPU 8080 や Z80 などは、64 キロバイトしか一度に扱うことができませんでした。このことを考えてみると、非常に高性能な CPU だったことがわかるでしょう。8086 は、8080 や Z80 などのソフトウェア資産を移植しやすいようにと、メモリの管理をこのセグメントという概念を用いて管理しています。このセグメントという概念は 8086 のマシン語のプログラミングをやる上で非常に大切な概念です。また、8086 のプログラミングを難しくしている一因でもあります。

8086 は 1 メガバイトのメモリを操作するために 20 ビットのバス幅を持っています。しかし 8086 のレジスタは 16 ビットの幅しかありません。どのようにして 1 メガバイトのメモリにアクセスしているのでしょうか？

8086 は 2 つのレジスタを足し算することで、20 ビットのアドレスを実現しています。2 つのレジスタとはセグメントレジスタとインデックスレジスタ、またはセグメントレジスタと特殊レジスタのことです。単に 2 つのレジスタを足し算しただけでは 16 ビットのままですから、まずセグメントレジスタを 4 ビット左にずらしそれをインデックスレジスタまたは特殊レジスタとを足し算します。16 進法で言えば、一桁左にずらして足し算すればよいのです。

(例)

SS が F04AH で SP が 159BH の時、CPU (8086) が実際にアクセスするアドレス(つまりアドレスバス上にあらわれるアドレス)は

```

F 0 4 A H
 1 5 9 B H
F 1 A 3 B H

```

となります。

セグメントレジスタに納められたアドレスをセグメントアドレス、特殊レジスタまたはインデックスレジスタに納められたアドレスをオフセットアドレスと言います。8086 がメモリに対してアクセスするアドレスは物理アドレスと言います。

8086 は 1 メガバイトのメモリを扱うことはできると言っても、実際は 64 キロバイト単位でしかメモリを扱うことができません。つまり連続して 1 メガバイトのメモリ空間をアクセスできないわけです。

12 ちょっといい話 (謎)

20 ビットで 1 メガバイトをアクセスできるのですが、それでは 16 ビットでは何キロバイトまでアクセスできるのでしょうか？ 8 ビット CPU の 8080 や Z80 はアドレス幅は 16 ビットでした。16 ビットでは 2 の 16 乗、すなわち 64 キロバイトまでしかアクセスすることができません。勤の良い人はもうお気づきかと思いますが、実はこれが 8086 がセグメントを導入した理由なのです。

64 キロバイトごとにメモリを使い分けることで、プログラムを作りやすくしようと考えたわけです。しかしこのセグメントの概念は「64KB の壁」と言われるようになり、後のプログラマたちを苦しめることになりました。

8086 はよく言えば 8 ビットのソフトウェア資産を考えた CPU でしたが、わるく言えば 8 ビットの呪縛から逃れられないままの 16 ビット CPU だったのです。(この呪縛は 80286 にも引き継がれていますが、80386 以降はかなり強力なメモリ管理機能が付くようになりました。)

セグメントアドレスは DOS 上でプログラミングする場合は、DOS が自動的に設定してくれるので、セグメントアドレスはプログラマ自身が設定することはまずありません。しかしプログラムで 64 キロバイト以上の連続したアドレス空間が必要な場合、プログラマ自身が設定してやらなくてはならないこともあります。

13 スタック

ところで SS (セガサターン もといスタックセグメント) は「スタック専用のセグメント」と説明しましたが、スタックとはいったい何のでしょうか？

マシン語のプログラムで、処理するデータや処理に必要なデータを格納するためのメモリ領域を、ワークエリアと呼びます。プログラムを制作する過程で、レジスタの内容を一時的にデータを RAM に退避させなければならない場合が多く出てきます。たとえば、『AX レジスタを演算に使いたいのだが、今現在のレジスタの内容を保存させておいて演算後にもう一度使いたい。』と言う場合などです。そこでこの「スタック」という概念が生まれました。

14 スタックのメリット

スタックは、簡単にいえばデータを積み上げて、使うときになったら上から順番に(つまり一番最後に積み上げられたデータから)取り出すことです。本を積み上げのことを例に取れば、積み上げた本を取るとき一番上に載っている本、最後に載せた本しかとれないわけです。このスタックの概念を用いることで、プログラマはデータを格納する際、メモリのアドレスを全く考慮せずにデータを出し入れできるようになるわけです。しかも高速にメモリに転送できるというメリットもあります。

SS と SP (スタックポインタ) には、このスタックしてあるデータの一番上のアドレスが入っています。だから普通、SS と SP は常に一緒に使われます。プログラマはスタックの出し入れに関する命令を用いる際、スタックポインタがいったいどこを指しているのかという心配は不要です。というのは、スタックの出し入れを行う命令では、スタックポインタの操作が自動的に行われるからです。

15 まとめ

以上で大体の 8086 の基礎を説明しましたが、次回は具体的にマシン語を使ってプログラムを説明していきます。

情報棟を活用しよう!

東大 ECC ことはじめ

Aleph-NULL

1 はじめに・情報棟事始め

新入生のみなさんは、理系は1学期に、文系は2学期に情報処理の講義を受けることとなります。初回を除いて(クラスによっては初回も)、正門の左手にある電車から見えるひとときわ新しい建物、「情報教育南棟」で授業を受けます。まずは1階の自習室に入って見ましょう。「X 端末」と呼ばれる端末が山ほど並んでいます。このシステムは、UNIX の一種である SunOS を用いたネットワークシステムになっていて、ここから電子メールのやり取りやネットワーク・ニュースの読み書き、それに WWW (World Wide Web, 通称 Mosaic) へのアクセスもできます。東大のネットワーク UT-net2 を通じていわゆる Internet に通じているので、世界中の情報にアクセスすることもできます。

すこし予備知識に触れておきます。

2 絶対やってはいけないこと

- 他人にパスワードを教えること(どんなに親しくても)
- ログイン時の画面と、実名・パスワードの変更(後述)以外たとえ要求されてもパスワードを入力しないこと(とくに WWW (Mosaic) には気をつけてください)
- ログインしたまま長時間席を離れること
- 館内での飲食・喫煙・ゲームなど

3 UNIX と MS-DOS の大きな違いは？

MS-DOS が UNIX からいろいろな機能を取り込んだことはよく知られています。階層ディレクトリとか標準入出力の概念とかです。しかし、根底に流れる思想は全く異なるといういいでしょう。簡単にその辺りを説明します。

UNIX では、複数のユーザが同時に 1 つのコンピュータを使うことができます。また、1 人のユーザが同時に複数のプログラムを動かすこともできます。それぞれの動作しているプログラムのことを「プロセス process」といいます。そして、各プロセスは「完全に独立に」動作することが可能です。MS-Windows のように、「1 つのプログラムが作業をしている間はほかのプログラムは動作できない」ということもありません。また、ユーザのファイルもきちんと公開・非公開の設定が可能です。

また、システムレベルでネットワークがサポートされており、ネットワークを通じて別のコンピュータにログインして作業をしたり情報を取得することも比較的容易です。

4 ログインしよう

手続きの際に、学生証の下に変な紙切れが付いていたと思います。そこには、g64**** (理 1 の場合) とかいうユーザ ID (g + 学生証番号) と、アルファベット・数字の羅列が印刷してあると思います。

情報棟のコンピュータは 3 万人近い人数が利用できますから、まず使い始めに自分が誰であるか、本当にその本人であるかをコンピュータに示さなければなりません。

適当に開いている端末を探して、もし電源が入っていなければ (キーボードの上の方の Power ランプがついていなければ) 端末前面左の電源スイッチを押してください。白地に黒文字でいろいろメッセージが表示された後、グレーバックにユーザ ID とパスワードの入力を求める画面が表示されるはずですが、電源が入っていて、そのような状態に初めからなっているかもしれません。

カーソルはユーザ ID のところにあるはずですが、ここにあなたの ID を入力してください。Return キーを押すと、パスワードの欄に移るはずですが、パスワードは画面に表示されませんので気をつけて入力してください。

正しく入力できたらなんか「GAME 禁止」とか画面に出た後、指示どおりクリックすると、しばらくして 3 つのウィンドーと郵便受けが表示されるはずですが、

g6?????@xss??> とか出ている 2 つを「シェルウィンドー」とか「コンソール」とか

呼びます。下の方に反転した *scratch* とか出ている方をとりあえず「Mule」(みゅーる)と呼んでおきましょう。

この画面は、UNIX での標準的なウィンドーシステムである X-window の画面です。自分でカスタマイズすると見栄えをいろいろと変えることが可能ですが、それはおいておきましょう。

まず、何か処理をする際には、マウスカーソルを処理をするウィンドーに入れることから始めます。枠線の色が変化して、カーソルが黒くなるはずです。

先に終了の方法を説明しておきましょう。何もウィンドーが無いところ(「ルートウィンドー」といいます)にカーソルを持って行って、マウスの右ボタンを押しっぱなしにすると、メニューが出てくるはず。一番下の「Exit X-Window」かなにかを反転させてマウスを離すと終了します。でも結果の保存は忘れずに。Mule を使った時は Mule をきちんと終了させてからの方が賢明です。

まず、シェルウィンドーの広い方にカーソルをいれて、「finger」と入力して Return を押して見てください。これを今後「finger コマンドを実行する」と表現します。すると、よほどすいていなければ何人もの人が表示されます。実は皆さんの手前にある X 端末は実は画面の処理しかしていません。実際の処理は別の部屋にある「端末サーバ」にネットワークで接続して行っています。駒場では 6~8 台の X 端末が同時に 1 つのサーバを使っていて、いま表示されたのはその同じ端末サーバを使っている人の一覧です。

5 準備

次に進む前に一回準備をしておきましょう。

実名の登録とパスワードの変更を行います。

シェルウィンドーから chfn コマンドを実行してください。あなたの名前の入力が必要されるはず。メールやネットニュースなどでも使われるので、ローマ字で実名を入力してください。次にパスワードを要求されますから、入力してください。

```
“... changed at ecc-ad00”
```

とか表示されればきちんと実名が登録されました。

```
“password file busy - try again later”
```

とか出て来た場合は後で再トライしてください。「パスワードが合わない」とかいわれた場合はパスワードの入力をミスったのでしょう。やり直して見てください。

次に passwd コマンドを実行してください。ここで、「忘れにくく」「他人に勘付かれない」「英数字 6~8 字」のパスワードを決めてください。間違えても自分の名前とか「tsg」

とか「inu」とか「tokyo」とか誕生日とかは駄目ですよ。脈絡のない英記号 2 つと数字 1 つの組み合わせとかでもいいでしょう。決めたら指示に従い新しいパスワードを 2 回、これまでのパスワードを 1 回入力してください。あとは前の名前の場合と同じです。

パスワードの変更は受け付けられてから 1 日程度かかる事があるので、変えたばかりの新しいパスワードがログインの時に受け付けられない時は古い方を試してみてください。

パスワードは最低でも 2ヶ月に一度は変更することが望ましいと考えられます。管理には十分気をつけてください。普段持ち歩く手帳には書かない方がいいでしょう。

6 Mule と電子メール

「Mule」は、文章やプログラムなどを書くためのプログラムです。「情報処理入門補遺」にも若干使い方が書いてありますが、カーソルキーと Backspace キーを使うとなんとか英文の編集くらいはできるはずなので、まずはいじって使い勝手を試してみてください。大文字やキーボードのキーの上の方に書いてある文字は Shift キーを押しながらキーを押すことで入力できます。とりあえず一通り記号を入力して見ましょう。なにかの拍子に画面の反転行の下にカーソルが行ってしまったら、Ctrl を押しながら g を押せば戻るはずです。(このような操作を C-g と略記します。)

一通り慣れたら電子メールの練習をして見ましょう。

シェルウィンドーから「~g540001/bin/mailtest」を実行して見てください。長いので入力を間違えないように。このコマンドは練習用にあなた自身にメールを送るようになっています。

実行して少しすると、「ピッ」とかいう音とともに郵便受けが黒く反転するはずですが、これがメールが届いた合図になります。

メールが届いたら、Mule のウィンドーから「ESC x mh-rmail Return」と入力してください。最初の ESC x で画面の下には M-x とかでてるはずですが、その後ろに mh-rmail と打ち込んで Return を押すのです。うまく行かなかったら C-g で戻ってもう一回試して見ましょう。

この操作をすることで、システムの「郵便受け」のメールが「自分の手元」に取り込むことができます。

画面が切り替わって上半分にはメールの一覧が、下半分には本文が見えているはずですが、読む時のキー操作を書いておきます。

Space 本文のページをめくる
メールを選ぶ
. 選んだメールを表示
ESC R 古いメールも含めて一覧表示する
q mh-rmail を終了する
i mh-rmail の最中に届いたメールを取り込む

次に、試しに自分にメールを出して見ましょう。mh-rmail を起動した状態で、m を押します。画面の下の方に、To: とか表示されているはずですが、ここに送り先の ID を書きます。ここでは自分の ID を入力して Return を押してください。次に cc: ができますが、メールの写しを送っておく相手がいないければ Return で結構です。最後の Subject: にはタイトルを入力します。そうすると、画面が切り替わって、

```
To: g*****  
cc:  
Subject: *****  
-----
```

とかなっているはずですが。(多少違ってもいいです)線の下になにかを打ち込んで、To: がちゃんと自分になっていることを確かめたら、C-c C-c (Ctrl を押しながら c を 2 回押す)で送信します。mh-rmail に戻りますから、郵便受けが反転したら q で終了してもう一度 mh-rmail を起動するか、i でメールを取り込むといま送信したメッセージが取り込めるはずですが。できましたか？

日本語入力の方法は「情報処理入門 補遺」などに記載されています。日本語が入力できるようになったら、友人宛にメールを出して見ましょう。本文の画面になってから、気が変わって送信をやめるには C-c C-q です。また、線の上の To: を編集すると宛先を後から変更することも可能です。

7 Netscape

いま世の中で大流行している WWW にアクセスして見ましょう。

シェルウィンドーから “netscape &” と入力して Netscape を起動して見ましょう。

しばらくして(ライセンスの同意を要求されるかもしれない)駒場情報教育棟のページが表示されるはずですが、文字化けが激しいかも知れません。

最初に設定が必要です。¹

メニューのから、次の順序で設定を変えていってください。

Options – Network Preferences – Cache

Memory Cache Size を 2000 kbytes に

Disk Cache Size を 0 kbytes に

Clear Disk Cache にカーソルをあわせてマウスの左ボタンを押す

OK

Options – Document Encoding – Japanese (auto-select)

Options – Save Options

そうしたら一回 [Reload] とかかれたボタンを押してください。これで正常に立ち上がらなかったら、Menu の File – Quit で一旦終了して、もういちど立ちあげてください。

各ページには何ヶ所かの下線の引かれた文字 (アンカー) があり、画面の一番上にはなんか <http://www.komaba.ecc.u-tokyo.ac.jp/> とかかれた文字欄があるはずです。この文字欄に URL と呼ばれるページの指定を打ち込んだり、下線の引かれた文字列の上で左ボタンを押したりしていくとどんどんページが変わって行きます。ネットサーフを愉しんでください!

なお、TSG のホームページは

<http://www.komaba.ecc.u-tokyo.ac.jp/~g541119/TSG/TSGhome.html>

です。こちらもぜひご覧ください。

8 コマンドについて (MS-DOS ユーザへの注意)

「UNIX 入門」も書くつもりですが、とりあえずちょっとだけ...

コマンドは DOS とはかなり違いますが、いじっていくうちになれるとは思いますが。参考書をなにか求めるのがいいでしょうが、先に DOS ユーザがよく陥る注意を述べておきます。

- ワイルドカードの * は、「全ファイル」を意味します。unix のシステムは拡張子を強く意識はしないので、このような仕様になっています。*. * では、「何等かの拡張子のあるファイル (. のあるファイル名) になります。DOS と違って、*1 で「最後が 1 で終わるファイル名」などという指定もできます。

¹ 新学期までに直るといいんだけど...

- mv コマンドで拡張子だけの変更などはできません。また、mv は指定先のファイルが存在しても上書きしますので要注意です。とくに移動先にワイルドカードを指定することはできないと思った方がいいでしょう。悲惨な目に遭います。同様のことは cp にも言えます。
- rm * とかやっても確認はして来ません。いきなりカレントディレクトリを全削除します。UNIX には undelete は無いので一度消えたらそれまでです。rm には注意しましょう。
- 駒場での個人のディスク容量は一人 15000KB (約 15MB) に制限されています。これを超えてしばらくいるとログインができなくなります。ときどきホームディレクトリから du コマンドを起動して、最後の数字が 15000 を超えていないことを確認してください。

P.S. 実は Netscape のところで Disk Cache を 0 にしたのはこれの対策でもあります。その分、駒場ではシステムの方で Web ページをキャッシュしているので速度はそんなに変わりません。

よく使うコマンドだけ DOS と対照してあげておきます。

dir	ls -l	type	cat
dir/w	ls	cd	cd
copy	cp	md	mkdir
del	rm	rd	rmdir
ren, move	mv		

コマンドラインの文化は圧倒的に UNIX の方が上です。これについては「UNIX 入門」で触れます。

9 今後

次の新入生歓迎号では、Netnews の扱い方とエチケットについて、それと「UNIX 入門」と題してコマンドを通じて UNIX と情報棟を 256 倍とはいわずとも 16 倍くらいは使う方法に入ります。² ぜひ TSG に入って次号を読んでください(^^);³

² といって勝手に連載を増やして自滅の道を歩んでるかも...

³ 実は先の TSG のホームページで部報は読めます。この辺の号の記事はなるべく早く掲出するので、そこで読んでくれてもいいんですが...まァ TSG ほど制約の少ないサークルもめずらしいから、(週 1 回も来ない奴も多い) 入って損することは多分無いです。

編集後記

春休みなので部員の多くが実家に帰ってしまい、サークルオリの準備は大変でした。東京に残っている人も朝が遅くてなかなか学校に来ないし。せっかく午前9時に電話で起こしてあげても、それからまた寝ちゃう人とかいるし。
(--;)

部室を見に来たい人はいつでもいいですよ。(^^)/ けれど弁当を買って昼休みにやってくるのが一番いいかな。 (わたる)

つかれた...

まあ今回は原稿の集まりは悪くなかったのですが、量の多いこと多いこと...

それはさておき
閑話休題。

TSG は非常に束縛の少ないサークルです。コンピュータ好き、ゲーム好きの連中の寄り合いのようなものです。

皆さんの参加をお待ちしております。 (Aleph-NULL)

理論科学グループ 部報 198号 [オリパンフ号]

1996年4月1日 初版 第1刷発行

1996年4月11日 初版 第2刷発行

発行者 金子 済

編集者 大岩 寛

発行所 理論科学グループ

〒153 東京都目黒区駒場 3-8-1

東京大学教養学部内学生会館 305

Telephone: 03-5454-4343

(C) Theoretical Science Group, University of Tokyo, 1996.

All rights are reserved.

Printed in Japan.

理論科学グループ部報 第 198 号
— '96 オリバンフ号 —
1996 年 4 月 1 日

THEORETICAL SCIENCE GROUP.