

星屑



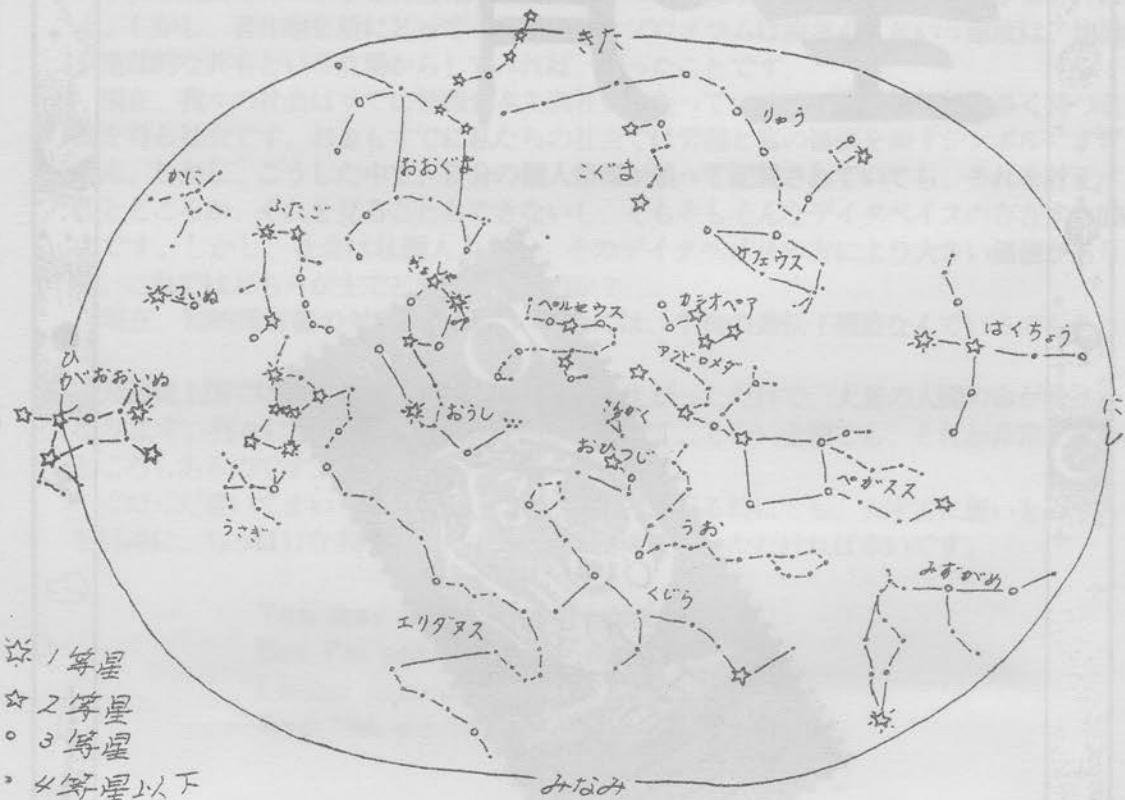
1990年11月号
HOSHIKUZU Vol.189

熊本県民天文台

KEIKOの星空散歩

12月上旬 *** 午後9時00分頃

12月下旬 *** 午後8時00分頃



12月の星座

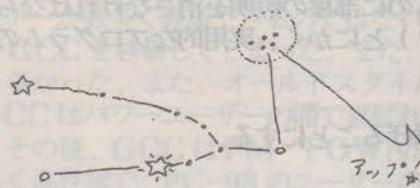
28日 01時50分(1日の夜中まで午前1時50分)スハル食
ても満月なので、良く見えませんか…

12月 ふたご座流星群がこの頃 極大
流星群が沢山。おかげで月が細いので見易い。

22日 冬至

かにまちゅか美味しい。

おうし座



写真でえに

撮,た人... 渡辺 D. さん

撮,た時 11月3日 2時19分~2時44分

105mm F2.8

撮,たと... 古住高原,

おうし座の肩の所にホーッと白い雲の様な所があります。

目の良い方なら明るい星が6個、もっと目の良い方ならいくつも(！)

見えるでしょう。これがあの有名なスバルです。

歌や車になつただけあって望遠鏡で見るとはみ出しますが

きれいな青、色をしています。今なら火星が近くにありますよ。

KUNI*

ASTRO COMPUTING (天文電脳術) Vol.2

§ 2.1 前回のプログラムの欠点

前回のプログラムの評価をしてみよう。この記事を読んでいる物好きの内、前回のプログラムを実行させた変人がいるとすれば、その人には分かることだが、あのプログラムは、とても遅く、そして、画面がとても暗い。画面を見るのに部屋の照明を消さなければならないほどである。（これはこれでリアルともいえるが？）とにかく、実用的なプログラムのためには、高速化と、画面表示の改良が必要だろう。

とりあえず、高速化の方から始めることにする。

前回のプログラムを遅くしていた要因を幾つか挙げることにする。

(1) 浮動小数点演算の遅さ

(2) 画面の外の領域まで計算していた

(3) 暗い星が明るい星を隠さないように、後に打つ点が前に打った点よりも明るくないと点を打たないようにしていた

(4) 星表がテキストデータだったのでバイナリーに直すのに時間がかかった

(1) から(4)のうち、根本的なのは(1)と(2)である。(4)の要因は前回のプログラムの性質からして仕方が無いことであるし、(3)も(4)の要因に付随して起こったことである。ただ、こうしたプログラムは同種の計算を大量にするものなので、一つ一つに少しの高速化を図っただけで、全体としては塵も積ればということもあるので、実用版の場合は、考慮しよう。さて、(2)だが、これは

(2-1) 行う座標変換の逆変換を行って、表示画面領域から、表示され得る赤経・赤緯の範囲を求める

(2-2) 星表を幾つかの領域に分割し、その領域毎に表示判定を行い、まるまる表示される領域は判定なしで表示し、部分的に表示される領域は個々の星について表示判定を行い、表示領域に入った星についてのみ表示する。全く表示されない領域は飛ばす。

(2-1) と(2-2)では、(2-2)の方が技術的にも簡単であるので、(2-2)を採用することにする。(2-1) 方式の良いアルゴリズムが見つかったら、そっちに変更するかも知れないが、とりあえず、(2-2) 方式でいくことにする。(2-2) 方式の、領域分割の方式としては、利用上の利点から考えて、星座毎に分割するというのが良いと思うのだが、星座の境界が正確にはどうなっているのか分からないので、困っている。もし、それが数値的にちゃんと分かったら、星表を星座毎に分類するフィルターを作るのだが。

さて(1)だが、演算の高速化のために、浮動小数点演算ではなくて、固定小数点演算を行うことにする。そもそも、パーソナル・コンピューターのたかだか 1000×1000 dots の粗い画面に表示するための計算に、倍精度浮動小数点演算はオーバースペックである。しかし、固定小数点演算と簡単にいうが、加減乗除は良いとして、座標変換に必要な三角関数はどうすれば良いのかということだが、仕方がないから作るのである。というわけで、次回は固定小数点演算の計算ルーチンを作ることにする。

§ 2.2 GNU

“ASTRO COMPUTING”ではプログラム記述に **GCC (GNU-C)** を使用しているが、これは、DOS ユーザー、中でも、大勢を占めるであろう PC-98 ユーザーには馴染みがないだろう。つまり、これを読んでいる人の大半は知らないということになると考えて良いだろう。

GCC は Richard Stallman が主宰する **Free Software Foundation (FSF)** が進めている GNU プロジェクトの基幹となる C コンパイラーである。GCC はとても性能が良い。機械語中級者が書く位に最適化されたコードを吐き出す。市販の C コンパイラーよりも最適化性能は良い。NeXT は GCC を標準 C としているし、UNIX の世界では GCC を使うことを奨励している場合も多い。また、現在のところ数少ない ANSI 完全対応のコンパイラーである。（実際には、ANSI C のスーパーセットになっている。）世に

ANSI 準拠をうたう C は多いが、実際に完全対応している物は数少ない。GCC は Version 1.37 からは、多バイト長文字（日本語処理等）まで完全サポートしている。

X68000 のユーザーにはパワーユーザーが多い。そして、X68000 はその初期、ソフトウェアの数が少なかった。そんな中で、「GCCさえ移植すれば、GNU のソフトウェアが数多く移植できるのではないか」と考える人たちがあらわれ、それで、幾らかの人々が GCC を移植してしまった。また、X68000 の純正の C は性能が大変悪く、また、バグも多かった。また、オールドスタイルな ANSI 準拠度の低い C だった。そんなわけで、GCC はパワーユーザーの間で喜ばれた。

その後、GCC は FM-TOWNS にも移植された。GNU のソフトウェアはすべからく高性能で、PC-98 のユーザーの中にも移植を行っている人があり、比較的小さなものは実際に動いている。しかし、640 KBytes しか管理できない MS-DOS のもとでは約 500KBytes もある GCC を動かすことは不可能であろう。しかし、現在、80386 マシンが増えてきた。80386 は、事実上 680x0 のようなリニアなメモリー空間を持っている。そして、MS-DOS のもとでこうした広大でリニアなメモリー空間を利用するための DOS-Extender というものが幾つか作成されている。事実、FM-TOWNS はこうした DOS-Extender のひとつの Phar Lap 社の 386+DOS-Extender を採用している。これは、PC-98 用も発売されていて、これを手に入れれば、PC-98 でも GCC を動かすことができる。（多分）また、386+DOS-Extender 互換の無料で配布されている DOS-Extender もあるので、こうした物を手に入れれば PC-98 のもとでも GCC を動かすことができる。但し、あくまで CPU が 80386 以上の機種の話であるが。

GNU プロジェクト (Richard Stallman) は GCC のような高性能なソフトウェアを作り続けているが、真に注目すべきことは GNU が単に高性能なソフトウェアを作る物ではなくて、ある種の思想運動であるということである。そこで、Richard Stallman の「GNU 声明書」を読んでもらおうと思う。

<<<< GNU 声明書 >>>>

The GNU Manifesto

by Richard Stallman
from DDJ March, 1985

GNU (Gnu's Not Unix) は私が今書いている Unix コンパチブル・ソフトウェアシステムです。私はこれを使いたいという人に無償で提供しようと思っています。私のほかにも何人のプログラマが手伝ってくれています。時間、お金、プログラム、機器の寄付が大いに必要です。

すでに我々は Vax, 68000 用の C と Pascal のコンパイラ、Emacs-like なテキスト・エディタ（コマンド記述用の Lisp 付）、yacc コンパチブルなパーサー・ジェネレータ、リンク、その他の約 35 個のユーティリティを準備しています。shell はほとんど完成に近づいています。カーネルとデバッガが完成すれば、（1985 年中に何とかしたいと思っているのですが）GNU をプログラム開発環境に適したシステムとして配布することができます。その後、テキスト・フォーマッタ、Empire game、スプレッドシート等の何百ものツールをオンライン・ドキュメントと共に提供するつもりです。最終的には、Unix に標準的についてくる用なツールのすべて（それ以上）を提供したいと思っています。

GNU は Unix のプログラムを走らせることができますが、Unix そのものではありません。私たちは他の OS の経験上、便利だった点はどんどんとりいけていきます。特に、

より長いファイル名やファイルのバージョン・ナンバー、クラッシュに耐えうるようなファイルシステム、ファイルネーム・コンプリഷン（ファイルネームを途中まで指定しただけであとはシステムが完全なファイル名を用意してくれるような機能）などのほかに、多分、端末独立なディスプレイのサポート、最終的には複数のLispプログラムや通常のUnixプログラムがスクリーンを共有できるようなLispベースのウィンドウ・システムについても計画中です。

システム・プログラミング用の言語としてはCとLispの両方が用意されます。また、通信のためには、UUCP, MIT Chaos Net, Internet の各プロトコルをサポートしようとしています。

GNUはまず、68000/16000クラスの仮想メモリをもったマシンを目指としています。何故ならば、この程度のマシンならば比較的簡単にGNUを走らせることができるからです。もっと能力の小さなマシンの上で走らせるための努力はそのマシンの上で使いたいという人の課題ということになるでしょう。

1.1 なぜGNUを書く必要があるか

もし、好きなプログラムがあるとしたら、それを好きなほかの人とも分けあわねばなりません。ソフトウェア販売会社は、ユーザにソフトウェアを他人と共有しないことを契約させることで、ユーザを分割して、支配しようとしています。私は良識として、非公開とかソフトウェアのライセンス契約とかいったものにはサインできません。何年もの間、私はこのような傾向に抵抗するためにAIラボの内部で活動してきました。結局、私の努力は無駄でした。自分の意志に反してこのようなことをする教育機関に留まることは不可能です。

私は自分の信念をまげることなく計算機を使い続けるために、自分が無償でないようなソフトウェアなしで十分やっていけるような無償ソフトウェアの団体を組織することを決めたのでした。私はGNUを配ることをMITが法律を盾に阻止するのを避けるためにAIラボを辞職しました。

1.2 なぜGNUはUNIXコンパチブルなのか

UNIXは私の理想のシステムという訳ではありませんが、しかし、そんなにひどいものという訳でもありません。UNIXの本質的な特徴はよいものです。私はそのよい点を駄目にせずUNIXに欠けているものを埋めたいと思っています。それ以上にUNIXとの互換性は、移植をしようと思っている他の多くの人々にとって便利であるはずです。

1.3 GNUはどうやって提供されるのか

GNUはパブリック・ドメインには存在しません。誰でもGNUを修正したりそれを再配布したりするのは自由ですが、その人がそれ以上の再配布を禁止するようなことは許されません。言葉を換えれば、所有的な修正は許されないということです。私はGNUのすべてのバージョンが無償であることをはっきり言っておきたいと思います。

1.4 なぜ他の多くのプログラマが支援してくれるのか

私はGNUにひきつけられ、手伝いたいという多くのプログラマを目のあたりにしました。多くのプログラマはシステム・ソフトの商業化に幻滅しています。それは彼等に金儲けをさせはするのですが、同時に他のプログラマを仲間と考えるよりは競争者と考えるように仕向けています。プログラマの間の連帯の基礎となる行動はプログラムの共有です。今行われているような市場協定ははっきりいって、プログラマに他のプログラマを友人として扱うということを忘れさせます。ソフトウェアの購入者は友情のもとで選択を行い、かつ、法に服従しなければなりません。多くの人が友情のほうが大切であると考えるのは自然なことです。しかし、法律の枠組みの中で物事を考える人はしばしば、すぐにそれをきめることができます。そのような人はシニカルにプログラミングは単なる金儲けの方法に過ぎないと考えて

しまうのです。

プログラムを所有するのではなく、GNUで仕事をし、GNUを使うことで私たちは誰に対しても心広くあることができ、同時に法を守ることもできるのです。加えて、GNUは皆を共有に参加するように奮い立たせ、一緒になることの一つのサンプルの役をも果たすでしょう。それは私たちにもし、無償でないソフトウェアを使い続けるならばほとんどありえないようなある種の調和の感情を抱かせることができるでしょう私の話したプログラマのうち、約半数がそれはお金にかえられない貴重な幸福であるといっていました。

1.5 どうしたら参加できるか

私は計算機メーカーにマシンとお金の寄付を求めていました。同時に、個人にもプログラムと労働の寄付を求めていました。

ある計算機メーカーはすでに、マシンの提供を申し出してくれています。しかし、私たちはもっと必要なのです。もし、マシンを寄附してもらえるならば、GNUを早いうちからそのマシン上で走らせることができるという結果になります。そのマシンは、普通の人の住む部屋におけるもので、特殊な空調や電源が必要なものであってはなりません。

多くのプログラマが熱心にGNUにパートタイムで労働を提供してくれています。ほとんどのプロジェクトではこのような部分的で分散した仕事というのはまとめ上げるのが非常に難しいものです。独立して書かれた部分というものは一緒にしてもうまく動かないものです。しかし、UNIXを書き換えるというこの特殊な仕事にとっては、この問題はありません。完全なUNIXシステムは何百というユーティリティ・プログラムをふくんでおり、そのひとつひとつは別々に文書化されています。ほとんどのインタフェース仕様はUNIXコンパチブルに固定されています。それぞれの参加者があるひとつのUNIXユーティリティをコンパチブルに書き換えることができ、それがUNIX上でもとのものと置き換えてちゃんと動けば、それらを集めたものも正しく動くはずです。たとえ誰かがとんでもない問題を発生させても、これらの部品を組み合わせるのは実行可能な仕事です。（カーネルの部分は密接なコミュニケーションを必要とし、小人数の緊密なグループで仕事をしなければならないでしょうが。）

もし、お金を寄附してもらえるならば、私はフルタイムかパートタイムの人を何人か雇うことができるでしょう。給料はプログラマの標準からすれば余り高くはできないかもしれません、お金を稼ぐことと同様に共同体の精神を築くことが大切であると考えるような人を私は探しています。これはGNUに参加してくれる人が別の方法で生活の糧を得る代わりにその全力をGNUの仕事に注ぎ込めるようにするためのひとつ的方法であると思います。

1.6 なぜすべてのユーザにとって利益となるのか

ひとたびGNUが書き上がってしまえば、誰もがちょうど空気のような感覚で良質のシステム・ソフトウェアを無償で手にいれることができるようになります。これは単に人々がUNIXのライセンス料を節約できるということではありません。システム・プログラミングの非常に無駄な繰り返しをせずにすむのです。この努力は代わりに“state of art”を高めるのに費やすことができます。

完全なシステム・ソースが誰でもみられるようになるでしょう。そうすれば、システムに変更を加えようとするユーザは自由に自分で書き換えるなり、誰かプログラマか会社にそれを依頼することができます。ユーザはもはやソースコードを所有するプログラマや会社のお情けにすがる必要もなく、変更を行うことに關しては独立した存在でいられるのです。

学校では生徒たちにシステム・コードを学習し、改良するように奨めることで、最高の教育環境を準備することができます。ハーヴァードの計算機研究室ではソースコードが公開されないようなプログラムはシステムにのせないという方針をもっており、実際にそのようなプログラムのインストールを拒否することでその方針を擁護しています。私はこれに強く感

銘を受けました。

最後にいえることは、誰がシステム・ソフトウェアを所有し、どれがそれに関わる権利があり、またないのかということをいちいち考慮するオーバーヘッドがあげられます。人々にプログラムを使う代価を払わせるという協定（コピー・ライセンスを含む）は人がいくら（つまり、どのプログラムに）支払うべきかを決定するのに必要な厄介な機構を通して、常に社会に大変なコストを負わせています。更に、このようなことを人々に守らせようとするには警察国家でもなければできません。例えとして、空気を作る出すのに膨大なコストがかかる宇宙ステーションのことを考えてみて下さい。空気1リットル毎の呼吸の度に料金がかかるのはいいとして、メーター付の酸素マスクを昼も夜も着けていることはたとえ、全員が料金を支払うようになっていたとしても耐え難いものです。そして、テレビカメラがそれを外したりしないように到る所で見張っているなど、どうかしているとしかいいようがありません。頭割りの料金で、エア・プラントを作り、マスクを外したほうがまします。プログラムを全部とか一部をコピーするということはプログラマにとって呼吸をするのと同じように自然なことなのです。それは無償であるべきなのです。

1.7 GNU の目標に対する反対とそれに対する反証

“ただのものなんか誰も使おうとしないだろう。なぜなら、それはどんなサポートも当てにできないということだからだ。サポートを用意する代わりにプログラムに対する代金をとつたらどうか？”

もし人々がサービスなしで無料のGNUよりも、サービスの付いた有料のGNUのほうがよいというなら、無償でGNUを手に入れた人にサービスを提供する為の会社を作つて儲けることができるでしょう。

本当のプログラミングの仕事の意味でのサポートと単なる手間仕事を区別するべきだと思います。前者はソフトウェア販売会社には期待のできないものです。その問題が多くの人々にとっても問題となっているのでもない限り、販売会社は何もしてくれないでしょう。もしあなたの仕事上、サポートに頼らざるを得ないのであれば、必要となるのはソースとツールだけです。そうすれば、後はあなたの問題を解決してくれる人を雇えばいいのであり、誰のお情けも必要ではありません。UNIXの場合は、大方のビジネスにとってソースの価格の高さがこのようなことを考慮の外におく結果になっています。GNUの場合には何の問題もありません。最近に有能な人間なしでやっていくとしても、配布協定ではなんら問題となりません。GNUは世界の難問をすべて解決しようとしている訳ではないのです。

一方、計算機のことを何も知らないユーザはいろいろな雑事を必要とするでしょう。すなわち、彼等は自分でも簡単にできるようなことを代わりにしてくれる誰かを必要としています。決してどうやったらできるのか、その方法を知りたがっているというわけではありません。そのようなサービスは単純作業と修復サービスのみを行うような会社が請け負うでしょう。もし、ユーザがお金を払ってでもサービス付の製品を手に入れたがっているというのが本当ならば、彼等は製品は無償で手に入れても、そのようなサービスを買いたいと思うでしょう。サービス会社は仕事の質と価格で競争するでしょう。ユーザは特定の会社と結び付けられるということはありません。一方、サービスを必要としないユーザはサービスに対して代価を支払う必要はありません。

“広告なしでは多くの人々に知らせるることはできないだろう。その分でもプログラムの料金をとるべきだ。ただで手に入るプログラムの広告をしてもしょうがないのだから。”

GNUのようなものについて、多くの計算機ユーザに情報を伝えられるような無料か若しくは非常に安い広報媒体にはいろいろな物があります。しかし、広告を使えばもっと多くのマイクロコンピュータのユーザーにまで知らせられるのも事実かも知れません。もし本当にそうだとして、代金をとってGNUのコピーや配送のサービスを広告してくれる企業はその広告に支払った以上のものに見合うだけの成功をおさめなければなりません。こうし

て、広告から利益を受けるユーザーだけがそれに支払うのです。ところで多くの人が友達なんかからGNUを手に入れるとしたら、そのような企業は続いていかないでしょう。つまり、GNUを広めるのに広告がどうしても必要という訳ではないのです。自由市場の擁護者は何故こんなことを市場で自由に決めさせないのでしょうか？

“私の会社はライバルに競り勝つために私的なOSを必要としているのだ。”

GNUは競争の領域からオペレーティング・システム・ソフトウェアを除外しようとしているのです。あなたはこの分野で勝利を得ることはできないでしょうが、同時にあなたの競争者が勝利を得るということもないのです。あなたとあなたのライバルたちは他の分野で競争し、そこで互いに利益を得ることになるでしょう。もし、あなたの商売がOSを売るごとだったら、GNUに好意を抱く訳にもいかないでしょうが。そうでなければ、GNUはOSを売るというような高くつく商売に巻き込まれずにするようにしてあげられるでしょう。GNU開発が多く企業やユーザからの善意に支えられているのをこの目で見たいものです。

“プログラマは自分の創造性に対して報われないのでないか？”

何か報われることがあるとすれば、それは社会的な貢献です。創造性は社会的な貢献ですが、それは社会がその成果を自由に使う限りにおいてなのです。もしプログラマが革新的なプログラムを作りだしたことに対して報われるべきだとしたら、同様にそのプログラムの使用を制限することは罰せられるべきことではないでしょうか。

“プログラマは自分の創造的な仕事に対して、その代價を求めてはいけないのか？”

仕事に対する支払を求めたり、自分の収入ができるだけ多くしようとすることは反社会的な手段を用いない限り悪いことではありません。しかし、今日ソフトウェア開発の現場で習慣となっている手段は破壊的なものに基づいています。プログラムの使用を制限することによってプログラムのユーザーからお金を引き出すことは反社会的な行為です。なぜなら、制限を行うことによってそのソフトウェアの流通量、利用方法を減らしてしまうことになるからです。これは、人間がプログラムから得ることのできる富の量を減少させます。故意に制限しようとするならば、その有害な結果は故意の破壊行為といわざるをえません。まつとうな市民がそのような破壊的な手段に頼らないのは、もし誰もがそんなことをすれば、互いの破壊行為の中で我々全部が貧しくなっていくからです。これはカント的な倫理、黄金律です。皆が情報の私有を行った結果を私は嫌悪するからこそ、このようなことが悪いことであることをよく考えてくれるように私はお願いしているのです。殊に、自分の創造性が報われたいという欲望がその創造性の一部又は全部の一般性の名において世界を奪うことを正当化するものではありません。

“人間には自分の作りだしたものがどのように使われるか管理する権利があるのではないか？”

自分のアイデアの利用についての支配権は実際には他人の生命についての支配権からなりたっています。そして、それはしばしば他人の生存を危うくする方向に用いられるがちなのです。法律家のような知的な所有権についてよく学んだ人は知的な所有物については本質的な権利はないといっています。政府が認めているような仮想的な知的所有権の類は、特殊な目的のための特殊な立法行為によって作りだされたものなのです。例えば、特許制度は発明家達に彼らの発明の細部を公開させるために制定されたものです。その目的は発明家個人を保護することではなく、社会を保護することだったのです。その当時、特許に対する17年の保護期間は技術の程度の発展の割合に較べれば、短いものでした。特許は単に企業家達の間だけでの問題でしたから、ライセンス契約のコストや手間が製造の準備に較べて小さなものであるような人達にとって、特許制度はそれほど害を及ぼさなかったのです。彼等は特許の

ついた製品を使用する個人をそれほど妨害しませんでした。

著作権の概念は、ノン・フィクション作品で著者がしばしば他の著者から長々とした引用を引き写していた古代には存在しませんでした。この慣行は多くの著者の作品が部分的にせよ生きながらえるには有用な、ほとんど唯一の方法でした。著作権制度は明らかに著述業を助けるために作りだされました。その制度が作られた分野（書物、これは印刷するだけで安価に複製できる）では、それはほとんど害をもたらすことはありませんでしたし、本を読む個人を妨害するなどということはありませんでした。

知的な所有物に対する権利というものはすべて、単に社会によって承認された契約にすぎません。なぜなら、いいにつけ悪いにつけ、その契約を承認することによって社会全体が利益を得ると考えられたからでした。しかし、どんなに特殊な状況にあっても私達は問い合わせなければなりません。

「このような免許を承認することは本当によいことなのか？」

「私達はその人になにをすることを認可しているといえるのだろうか？」

今日のプログラムの場合は百年前の本の場合とは全く異なっています。プログラムをコピーする最も簡単な方法はお隣からお隣へまわすだけだという事実。プログラムにはソース・コードとオブジェクト・コードがあってそれぞれ違うものだという事実。プログラムは読んだり、楽しんだりするものではなく使うものだという事実。これらの事実があわざって、著作権を押し通そうとする人間が、物質的にも精神的にも社会に害を及ぼすような新しい状況を作りだしているのです。つまり、そのような状況の中で、人は法律が自分にとってよいものかどうかよく考えていかなければなりません。

“金銭的な刺激なしでは誰もプログラミングなどしないのではないか？”

実際、多くの人々が金銭的な刺激全くなじでプログラムをするでしょう。プログラミングはある人々にとって（そのような人はプログラミングに最も向いている人であることが多い）拒否し難い誘惑なのです。たとえ、音楽で生活の糧を得る望みがないとしても、プロのミュージシャンがいなくなってしまうことがないのと同じです。しかし、実際にはこの質問は（よく質問されるのですが）状況にちゃんと適応していません。プログラマに対する支払は少なくはなるでしょうが、なくなることはないでしょう。そこで正しい質問は次のようにあります。

“金銭的な刺激が減ってもプログラムを書く奴はいるか？”

私の経験からいって、そういう人はいるでしょう。10年以上の間、世界中の最も優秀なプログラマがよそに行けばもらえる筈の給料よりはるかに少ない給料でAIラボで働いてきました。彼等は金銭的ではない多くの報奨を得ました。例えば、名譽、評価といったような。また、創造することは彼等にとって喜びであり、それ自身が報奨であったのです。そして、彼等の多くはもっと多くの給料で同じぐらいに興味深い仕事の話がきたときにはここを去っていました。この事実が示しているのは、人間は富以外の理由でプログラムを書くのだということ、そして、もっとお金を儲けるチャンスがあればそれを受け入れるのだということです。支払のよくない組織は支払のよいところとの競争では負けてしまいがちですが、支払のよい組織がなくなってしまえば、そんなにひどいことをしているわけではないのです。

“プログラマは何とかして生活の糧を得なければならない。”

プログラマがプログラムの使用権を売らなくても生きていける方法はいくらでもあります。例えば、

企業が新しい計算機を導入する時、OSを新しいハードに移植するために支払うだろう。

教育、維持、保守などのサービスを売っている企業がプログラマを雇うだろう。

新しいアイディアをもった人がそれを求めるユーザの寄付によって無償でそのプログラムを配布することができる。私は、何人かの人がこの方法ですでにうまくやっていると聞いている。

関連する要求を持ったユーザがユーザーズ・グループを結成し、料金を支払う。グループはソフト会社と契約し、グループのメンバーが使いたいようなプログラムを書いてもらう。

すべてのプログラム開発に対して、ソフトウェア税のような資金を運用することもできます。

計算機を購入する人はすべてその値段のXパーセントをソフトウェア税として支払うようにする。政府はソフトウェア開発を進める NSF のような政府機関にこれを与える。

但し、もし計算機の購入者が自分自身のソフトウェア開発のために寄付を行う場合には、税金に対する貸し金を得ることができる。寄付を行うプロジェクトは自分で選択することができる。そのプロジェクトの成果を自分で使いたい時などがあるだろう。

税の総合率は税金の支払者の投票によって決めたものに、前年支払った税金に額を勘案して決められる。

結果として

計算機を使用するコミュニティがソフトウェア開発を支援することになる。

このコミュニティはどのレベルのサポートが必要であるか自分で決定できる。

自分の出した分がどんなプロジェクトに費やされるかに興味をもつユーザは自分自身で決定できる。

長い目で見れば、プログラムの無償化は飢餓のない世界——誰もが生活の糧を得るために外にはきつい労働をする必要のない世界——に向けての一歩になります。人々は一週間に10時間、立法や家族協議、ロボットの修理、小惑星の試験などの必要な仕事をしたあと、自由に楽しみの得られる活動（例えば、プログラミングのような）に専心することができます。もう、プログラミングでお金を稼ぐ必要もなくなっているでしょう。

私達はすでに、社会全体がその活動的な生産性のために必要とされる仕事の量を大幅に減らしてきました。しかし、生産的な活動に伴って非常に多くの非生産的な活動が必要とされるために、このうちのほんのわずかが、労働者のレジャーのために振り替えられているだけです。その中心的な理由は、官僚主義と競争に対する闘争心です。無償ソフトウェアはソフトウェア生産の分野でこれらの失費を大幅に減少させるでしょう。私達は、生産性における技術的な進歩を労働の削減に振り向けるためにそれを是非ともやっていかなければなりません。

Copyright 1985 Richard Stallman.

Permission is granted to make and distribute copies of this article as long as the copyright and this notice appear, and the copies are distributed at no charge.

編集後記

今月号も発行が大幅に遅れて本当にすみませんでした。深くお詫びします。1990年12月号は、塩屋君が頑張ってくれるので、すぐに出るはずです。

しかしながら、今月号の『星屑』を読まれた方は、発行が大幅に遅れたことよりも、その内容の方に、より大きな不満を持たれたかも知れません。何故『星屑』に『GNU声明書』なんかを載せたのかと。しかしながら、私はこれを読んで欲しかったのです。

今、世の中では知的所有権の問題がいわれています。確かに、人が一所懸命に作ったプログラムを無断でコピーするのは確かにそれを作った人にすればたまらないことかもしれません。しかし、著作権を盾にとって「金渡さんとプログラムは渡さん」という態度は、知識の全地球的な共有という立場からしてみれば、困ったことです。

現在、我々の社会はすでに情報資本主義社会になっています。これは情報を多く持つ者が富を得る社会です。お金もすでに私たちの社会では労働と富の価値を表すシンボルにすぎません。しかし、こうした中で、自分の個人情報が誤って記載されていても、それを訂正することどころか、それを見ることもできないし、そもそもそんなデータベースの存在すら闇の中です。しかし、社会は私個人よりも、そのデータベースの方により大きい価値があります。これではどちらが主でどちらが従なのか？

現在、知的所有権の対象になっている物には、生物の遺伝子構造なんていうのもあります。

発展途上国では、医薬品の値段がほんの少し上がっただけで、大量の人間の命が救えなくなります。我々にとって10円や100円がどおってことない金額でも、それが非常に貴重なところもあるのです。

ぐだぐだ書いてまいりましたが、できれば、星を見る時にでも、大宇宙に想いをめぐらすと同時に、ちっぽけな我々人間の未来のことも考えていただければ幸いです。

You may say I'm a dreamer
But I'm not the only one
I hope someday you'll join us
And the world will be as one

熊本県民天文台機関誌「星屑」 1990年11月号 通巻189号
発行所 熊本県民天文台

〒861-42 熊本県下益城郡城南町藤山
Tel. (0964)28-6060

熊本県民天文台事務局
〒860 熊本市古京町3番2号 熊本市博物館内
Tel. (096)324-3500
振替口座 熊本 8-24463

編集担当 守岡 知彦