

フリーソフトウェアで目指すもの

g新部 裕

<gniibe@m17n.org>

独立行政法人 産業技術総合研究所

Thu September 27, 2001

要旨

本論文では、フリーソフトウェアの運動の原点を確認し、その視点から今後の技術的なターゲットを明らかにする。

1 はじめに — 新しい時代を迎えたフリーソフトウェア

21世紀に入り、Linux 2.4 のリリース、GNOME 1.4 のリリース、および GCC 3.0 のリリースと、フリーソフトウェアの開発には大きな進展が見られる。その適用分野は拡大し、大企業のサポートとユーザの増大につれて、開発プロジェクトと開発者が急激に増えてきた。ある種のタブーの感もあり、アンダーグラウンドの活動という側面が強かった前世紀を考えると、フリーソフトウェアは、今、まさに新しい時代を迎えたと言えよう。

しかしながら、後進国の日本では、フリーソフトウェアはその利用に関する動きはようやく活発になってきたが、全体的な技術的競争力はとても低く、開発力では大きく水を明けられている。

この遅れの元凶は、フリーソフトウェアを単に(安価に)「利用する」ものとするところである。フリーソフトウェアとそれが形成するコミュニティは、動的に変化し成長するものであり、そのメリットは、その場の中に「参加する」あるいは「貢献する」ことによって享受できるものなのだ。フリーソフトウェアの最大の利点は、その利用を制限しないことにあり、もともと目的とする直接の研究開発の成果以外に、他のソフトウェアや活動に影響を与え、ソフトウェアを中心として、さまざまな人々を巻き込んで発展・拡大していくことにある。

この利点を理解し、それを最も生かすことが大事だ。

ソフトウェア開発にかかるコストを考え、それに要する費用を削減することを目的としてフリーソフトウェアを導入することは、必ずしも成功しないばかりか、多くの場合失敗する。なぜならば、そういった考え方では、フリーソフトウェアの「互いに発展していく」という利点を生かすことができないばかりか、それを目的外的なもの、あるいは無用のものと位置付けてしまい、阻害してしまうからだ。

「フリーソフトウェア」の「フリー」とは「自由」のことである。これを「只(口ハ)」と考えるのは安易な誤解である。その自由のもたらす結果として、ソフトウェアが安価に流通することは事実であるが、それは「自由」のソフトウェアの目的ではない。

なんらかの利益をおこぼれとして拾うために、外から傍観者として関わり、単に技術的な差異を問題とする視点では、フリーソフトウェアの運動は理解できず、その方向性を見失う。

そこで(見失わないために)、以下では、フリーソフトウェアとは何かを再確認し、これまでの運動を概観し、日本の現状を述べる。そして、今後の展望をこの運動の視点から述べる。

この文章を読んで、(読まなくてもいいから :-), この運動に参加しよう。さあ、今から新しい世界を創り出す一員だ。

2 フリーソフトウェアって何?

フリーソフトウェアは、自由にコピー、研究、修正および再配布できるソフトウェアである [1]。そしてこの自由が、共同作業とその発展をもたらす力の源である。

繰り返すが、それは「値段が只」ということではない。この誤解をちやかす言葉として、フリーソフトウェアの事業を起こした John Gilmore の創ったスローガンがあった [2]:

“We make free software affordable.”

「あなたも買えるフリーソフトウェア」といったところか。(80年代後半は確かにフリーソフトウェアは高価であり、それをリーズナブルなコストでサポートする企業が望まれていた。)あるいは、この誤解を一つの単語をはさむことで明解に避けたスローガンもあった:

”Free the software!”

「ソフトウェアを自由に」と。また、GNOME を成功させたメキシコ人プログラマ Miguel de Icaza は、こう言っている:

英語は壊れている、無料のビールと言論の自由を区別しない。

「只」ではないと念を押したところで、次に移る。

フリーソフトウェアは、他者と共有し協力する自由を、その利用者に与えるものである。この自由により利用者は、時にまた開発者となる。ソフトウェアは利用されて始めて発展するものであることから、この利用と開発の両方を密接に繋げることができる「自由」の仕組みがフリーソフトウェアの重要な構造であると言える。これによって、利用者は直接あるいは間接に開発に関わることが可能となり、ソフトウェアを多くの場に適用させ、その知見を集積して発展することとなる。

そして、フリーソフトウェアの自由とその結果の共同体を創り出す、法的なツールとして、GNU GPL (GNU 一般公共使用許諾) がある。知的財産権に関するライセンスが、その作者の利益を(強く)保護することを目的としているのに対し、GPL はその自由を保護することを目的としている。

フリーソフトウェアの原点は、排他独占的なライセンスあるいはそれをを用いる分断のソフトウェアビジネスに対する反動であったと言える。GNU Project の名前が、否定形であるのは偶然ではない:

“GNU’s Not Unix!”

Unix ではないものをと。(当時)それは不可能と思えるほど壮大な改革のはじまりであった [3]。

Linux の原点が、「使えない」OS (研究) に対する反動であることも興味深い事実である [4]。研究は研究として留まり、それ自身は広く(自由に)利用可能とならないことに対して、自由に開発できる、自由に利用できる OS への渴望が Linux の原点であった。そしてそれは、一学生の有名教授への抵抗であり、物知り顔の既存の権威をものとしないうえ溢れる挑戦であった。

3 フリーソフトウェアのこれまでの流れ

これまでの大きな流れの年表を表 1 に示す。

表 1: フリーソフトウェアの年表

1984	GNU Project が始まる
1985	Free Software Foundation, Inc.
1991	Linux の開発が始まる
1996	GNOME の開発が始まる

改革の運動であるフリーソフトウェアは、ソフトウェアの「自由」を標榜し、その継続的活動により、開発者と利用者のコミュニティを形成するに至った。経済モデルの欠如か

ら、その存続は当初から疑問視されてきたが、インターネットというコミュニケーションの基盤と、既存の価値観に因われないハッカーの熱意に支えられ、特に 90 年代後半、サーバの運用をはじめとする現場での実際の利用に供することにより、発展し大きくなってきた。そして、通常のコンピュータが提供するオフィス環境まで整備しようと進んでいる。

以下、開発環境、ネットワーク環境、デスクトップ環境およびハードウェアの広がりという流れについて述べる。

3.1 開発環境

80 年代中ごろ、フリーソフトウェアは、まず開発環境から始まった。GNU Project のソフトウェアである GNU Emacs (エディタ)、GDB (デバグ)、GCC (コンパイラコレクション)、もっとも古いフリーソフトウェアであり、長く使われ続け、そして発展し続けている。これらのソフトウェアは (自由のソフトウェア以外の開発にも) 多くのユーザに使われてきている。このことは、開発環境の自由が、いかにソフトウェア開発に重要であるかを示すものである。

そして、GNU の開発環境は、今では、もっとも広範囲に適用される何者にも変えがたい技術的集大成となっている。GCC は 8-bit のワンチップ組み込みプロセッサから、最新の VLIW アーキテクチャの 64-bit プロセッサまでをサポートし、いろいろな OS の上で動くことがサポートされている。また、C、C++、Java、Fortran と多くの言語をサポートしている。

これはその自由により適用範囲を広げ、その知見と経験を継続的に集積してきたことによる。

3.2 ネットワーク環境

90 年代に入り、フリーソフトウェアは、OS 全体を構成するまでに広がり、ネットワーク環境で大きく成長した。

Linux というカーネルの実装を得て、初めて OS を構成するすべてのソフトウェアを、自由のソフトウェアとすることが可能となった。そして、これら個々のソフトウェアをまとめる「ディストリビューション」が編纂されるにあたり、利用は大きく広がった。

同時期に世の中では、ネットワークの利用が広がった。このネットワークの広がりとともに、Apache (その名 “a patchy” がフリーソフトウェアの運動を象徴する) は、多くのサイトで利用されるようになってきた。その自由が技術導入を用意ならしめ、また運用の柔軟性とサービスの向上につながった。

3.3 デスクトップ環境

そして、今、フリーソフトウェアの運動は、すべてのソフトウェアを自由のものとする方向に展開している。90 年代後半に始まったデスクトップ環境 GNOME の開発は、一般のユーザに利用されるデスクトップ環境としてフリーソフトウェアを利用することを目標としている。

80 年代において、腕利きのハッカーの道具 (時に道具のための道具となることもあった) であったフリーソフトウェアは、90 年代にサーバ管理者のよき見方となり、ついには、一般のユーザの直接の使いやすさをもたらすまでに至った。

そして、そのすさまじいまでに速い開発は、既存の UNIX ベンダーをして、これまでのプロプライエタリソフトウェアを捨て、フリーソフトウェアを採用させるまでに広がった。

3.4 組み込み環境 — より多くのハードウェアに

また、フリーソフトウェアの運動は、すべてのハードウェアに自由のソフトウェアを稼働させようとする方向に展開している。Linux 2.4 に統合された組み込み分野へのプロセッ

サのサポート、メインフレームのサポート、USB のサポートおよび PC カードのサポートは、フリーソフトウェアが稼働するハードウェアをより広範囲のものとした。

90年代前半は、適用できるハードウェアが少ないことがフリーソフトウェアの欠点であったが、今では、そこにハードウェアがあれば、それはハッカーの挑戦の対象となり、適用分野はどんどん広がっている。特に、組み込み分野での展開では、開発環境を含めた機器のすべての環境が自由のソフトウェアで構成されるという、フリーソフトウェアの「完全なる」形が可能となった [5][6]。

4 日本におけるフリーソフトウェア

さて、日本において、個人、組織、国のプロジェクト等、さまざまな立場でフリーソフトウェアの運動を続けてきた経験を述べ、日本のソフトウェア開発の社会的問題点を上げ、どのようなことが望まれるかを述べよう。

これまで、g新部が行なってきたフリーソフトウェアの活動を以下に列挙する。

- 個人でのフリーソフトウェア
 - mLh, mlh (1990 ~ 1996)
 - Xp-BETA (1996)
 - EGG V4 (1997)
- 日本でのフリーソフトウェア (仕事)
 - 開放型ソフトウェア評価事業 (1993 ~ 1995)
 - ICOT Free Software (1994 ~ 1998)
 - RING プロジェクト (1996 ~ 1998)
 - 未踏ソフトウェア創造事業 (2001 ~)
- GNU Project (芸事)
 - MULE (GNU Emacs) の開発に参加 (1991 ~)
 - GNU C Library の開発に参加 (1996 ~)
 - GNU Guile POSIX Thread Support (1997 ~)
 - GCC/Binutils SuperH port (1999 ~)
 - GNU Patron (1997, 1999)
 - FSF ブース運営 (1999, 2001)
- Linux (芸事)
 - PLIP Driver
 - SIGIO support
 - SuperH support

個人でのフリーソフトウェアは、自分で必要なものを創り、それを公開し共有することで使いやすいものとし自分もその利益を享受するというものという性質がある。入力システムの mLh, mlh および EGG V4, アプリケーションゲートウェイの Xp-BETA とともにそれがどうしても自分で必要だったから創ったものである。特に広がりがあるシステムでもなく、比較的小規模の開発で、開発の初めにおける自由の効果が大きかった。

仕事として取り組んだものには、当時の通産省、今の経産省の関連である。ここでは、以下のような(少々無理な)理屈づけがなされていた。

- ミドルウェアのような基盤となるソフトウェアは公開されて然るべき
- 副産物であり、市場の価値がないが研究の価値が高いから公開する
- ソフトウェアを評価しフィードバックを得るために公開する
- 人材を発掘し、産業の発展に繋げる

このようなプロジェクトがあったおかげで、何らかの形でフリーソフトウェアに関連することができたのも事実である。しかし、それは、「自由」を標榜したものではなかった。そのため、あくまでも既存の排他独占のソフトウェア研究開発をモデルとし、それを修正するだけの「公開」程度であり、残念ながら発展したとは言えない。今後は「自由」を標榜するものに取り組むべきだと考えている。

芸事としては、GNU Project と Linux の開発を行なっている。GNU Project に関しては、開発を進めていく中で、開発に従事する人やそれを支える人、ソフトウェアの自由を保護を貫く人など、多くの人と関わり、FSF の運営支援まで行なうはめになった。:-) Linux は PLIP Driver の書き直しからはじまり、ネットワークレイヤの拡張を行ない、最近では「そこに CPU あれば Linux」との信念に基づき、日立 SuperH 対応を行なっている。日立 SuperH 対応は大きく発展し、サポートビジネスがはじまり、製品が作られるまでになった。GNU および Linux の開発は大規模な国際協調体制であり、ここに日本から参加して、成果を出してきた。この知見を日本に広め、良いソフトウェアを作る基盤を作ろうと考えている。

日本での問題は、(外から見た場合の) 閉鎖性と不透明性であろう。

第一に、日本ではソフトウェア開発が日陰ものの仕事とする向きもあり、特にフリーソフトウェアは公然とそれを行なっているとさえ言えないような状況が未だにある。このため、匿名希望で開発に参加したり、表にでない人もいる。カミングアウトまでには処遇や社会的認知(位置付け)などの諸条件の整備が必要であろうか。

第二に、逆に、自己主張が強過ぎて困る(協調体制としてはやっていけない)ということがある。小さな単位での収支を見ってしまう潔癖な性質では、型破りの挑戦が難しいのであろうか。個々の小さな差異を評価するという「オリジナリティ」の間違った適用が、協力して大きなものを作っていくということを妨げている面がある。このため、開発のコミュニティで望まれる、頻度の高い逐次的な改善や、「誰が書いたのであろうと良いコードを良いものとする」ということの実施が難しいようだ。大きなバッチ、あるいは(相互運用性が欠如した)ソフトウェア一式を、「*私* がやりました。」と持って行って壘盤を買うことがままある。自由のソフトウェアに対して、「このコードは私のもの」という概念を無理に適用するのではなく、互いの仕事を尊重できる環境作りが望まれる。技術的な点以外の情報交換の場や、技術的に切磋琢磨し互いに触発されるような場を作っていくことが課題である。

5 衝突とリーダーシップの必要性

ソフトウェアを自由のものとするれば、すなわちそれで「皆、しあわせ」かということではない。フリーソフトウェアの回りにはさまざまな衝突がある。それを解決あるいは回避するためにリーダーシップが必要となる¹。

衝突の例を以下にあげる。

- (独占的) ソフトウェアビジネスと相容れない「自由」
- ビジネスの展開とソフトウェアの発展
- 個人の欲求と集団の協調
- 企業の利潤の追求と社会性
- 安定性と新規性
- 部長は、ホントは Hacker が大嫌い B-)

「自由」のソフトウェア開発は、これまでの排他独占のやりかたとは異なるやりかたである。それを異なるものだと位置付けずに、ななああでも進めても、本質的に相容れない面があり難しい。ビジネスのために付け加えられる宣伝のための New Feature は、ソフトウェアそのものには無益であり、保守性を損なう。自由であろうとなかろうと個の欲求と集団の中での協調は衝突する。個が一人の人間のこともあり、企業のこともある(集団はコミュニティや社会)。

¹この衝突の存在を隠してあたかもユートピアのような幻想を与えるフリーソフトウェア開発の説明があるが、それは現場とは異なる、頭の中の理想の世界であろう。そういうのがあれば僕もやってみたいものだ。

もっとも顕著な衝突の例は、安定性 (安住) と新規性 (改革) であろう。ソフトウェアに有用性の面から安定を求める声と、その発展と新規性の面から改革を求める声と。

これらの衝突については、いつだって完全な答はないものであり、状況依存で決めていくものである。完全な答がないために、人は何かを必要とする。そのために、リーダーシップによる一貫した姿勢 (たとえ恣意性が高くとも) と、意見集約をはかり調整する仕組みが求められる。安定性と新規性の両立はあきらめて、別立てにして目的別に分ける方法がとられ、Linux では (サブバージョンの) 奇数シリーズが開発版、偶数シリーズが安定版とする運用を行なっている。

フリーソフトウェアの開発では、リーダーシップに求められるのは、(時には非情な) 選択を行ない、“NO” と言うこと、(適切に) 無視することである。複数の開発者が同時に好き勝手に進めることが可能であるため、その品質を保つためには、選択が重要となる。リーダーシップによる眼識が唯一その力である。

6 今後の技術的課題

フリーソフトウェアで目指すものは、ソフトウェアの富を増やすことである。フリーソフトウェアによって多くの開発者を参加させ、競争を促し、さまざまな知識を集積することだ。

これまで、多くのソフトウェアが自由に利用できるようになってきたが、特定の分野では、まだまだ独占排他のものが優位を保っていて、社会として必要とされる富が増えず、社会の要求にソフトウェアの生産が答えられないところがある。この点において、年々急速な勢いで進歩するハードウェアに関連した以下の分野が今後の重要なターゲットとなる。

- 周辺機器サポート (USB 接続機器, PC カード, IEEE 1394)
- 家電製品
- 携帯電話
- ハードウェア開発環境 (CAD/CAM)

この他に、ソフトウェアの適用分野の広がりによって追い付いていない部分として、以下の分野も重要なターゲットとなる。

- 自由に利用できるフォント, 文字を扱う仕組み
- (電子) 印刷, プリンタ

次に、富を増やすための方法論としてのフリーソフトウェアの活動の推進を考える。この場合の課題は、まず第一に、これまでと全く異なるやりかたの、いわば「大規模無計画開発」を技術的にサポートするツールである。これによって作業の効率を高め、高いアウトプットを出す仕組みを実装することが緊近の課題である。現在、ソースコード管理システムとしては、CVS がほとんどのフリーソフトウェアプロジェクトで用いられているが、もともとの設計としてネットワーク利用が前提とされていたものではなく、再設計の必要が出てきている。Subversion や BitKeeper などの研究開発の進展が期待される。

この「大規模無計画開発」サポートの指針を考えるにあたっては、これまでの仮定を捨てるのが最初にすることだ。従来の考え方は、計算機資源やプログラマの人的リソースを貴重なものと認識し、(少数精鋭を前提とし) 計画立てて、できるだけ重複を避け、節約して臨むというものであった。フリーソフトウェアの回りはもはや大きく変わっている。10万人を超える開発者がほっといても適当に好き勝手に進めていく²。リソースが豊富なので、新しいものが生まれる可能性がある限り、重複があろうとも、再発明が繰り返されても、節約などする必要はない。ソフトウェア全体でみた富が増える方向であれば、プロジェクトフォークもあってしかるべきである。従来のソースコードコントロールが進化する、分化することをモデルとし、中心のコントロールを前提としていたが、もっと非構造的なものや、合併する、階層的に昇進するといったモデルも必要となろう。加えて Coding monkey を集めるサル山、優秀なボス猿を選んでいく仕組みが実装されれば、さらに良い。

²SourceForge.Net の登録開発者の数。2001年7月現在

第二に、ソフトウェアの品質をわかりやすい形で計測し、改善の必要な部分を特定し、開発の方向性の決定に役立てる情報を提供するツールが課題である。すでに GNU のソフトウェアでは、機能テストや、これまでに報告された不具合やテストケースを集積し、リグレッションテストを行なう環境とそれを利用した開発体制ができてきている部分がある。また、これと合わせて、ユーザからの不具合の報告を集積するバグトラッキングシステムも利用され始めている。こういったシステムを更に磨き、これまでの活動の知見を集積する形で定着する方向が大切である。Linux の開発では、何度もこの試みが繰り返されたが、いまだに定式化されたシステムがない。GNU C Library の開発においては、IBM の多大な貢献によりテストスイートの整備が大きく進歩した。企業の得意とする分野でもあり、企業の貢献が期待される。

7 まとめ

これまで述べたことをまとめると、ソフトウェアの富を増やすことがフリーソフトウェアで目指すことであり、その直接のターゲットとしては、社会の要求に追いつかないソフトウェアであるとした。急速に進歩するハードウェアに関連した分野、および適用分野が広がっていくところがフリーソフトウェアが発展するところだ。

この展望を広げると、フリーソフトウェアとその活動は、アカデミズムを越えるもの、企業社会を越えるものとなることが可能であり、ひとつの独立の極としてそれぞれを補完しうると言える。

アカデミズムは、細分化により発展したが、知識の整理分類とその利用はどんどん離れてしまい、知識の再編が求められている。フリーソフトウェアによる知識の集積は、「専門分野」という垣根を越え（構造を破壊し）、自由に通り抜ける暴走あるいはレイヤバイオレーションにより行なわれてきており、(いわば素人が勝手に)進めてきているものであるが、知識の有用性という観点からは、これまでのものを明らかに凌駕するものが出てきている [7]。繰り返しになるが、Linux は大学教授に否定されたところから始まったのだ。

企業社会は、その効率追求により発展したが、その特化したやりかたは、環境問題に代表される地球全体の問題や、安全性の問題を引き起こし、自家中毒に陥り、非効率で持続不可能となってしまうおそれがある。フリーソフトウェアのやりかたは、効率を追求しないため、全体の問題としてのアプローチができ、それぞれが勝手に進めることによって、草の根の運動として広がることができる。(企業社会の中で)与えられた仕事とうまくできず、現実逃避として作られたフリーソフトウェアは、実のところとても多い。その力は、その構造の中では発揮されないが、異なる形では大きく役に立つのだ。

8 おわりに

最後に、フリーソフトウェアの重要な点は、その自由にあると繰り返す。

論文、公的資金、および認定試験など、これまでの古い体制の構造に取り込まれてはいけない。論文と言えば、LC2001 は今回からプログラム委員や査読などの仕組みが改善されマトモになってきた。これは果して、良いことだろうか。望まれるのは、当てにいくバッティングではなく、暴走ではなかったか。

公的資金と言えば、未踏ソフトウェア [8]。フリーソフトウェアは世間に認められなくとも、信念でやっていくものではなかったか。提案を公募で集めての選択は、テストでは良い点を取る人やブルペンだけでは調子の良い投手の採用とならないか。

そして認定試験。テストが得意ならハッカーには向かないのじゃないか。その特性は、与えられた問題に答えることではなかろう。そつなく問題をこなすのであれば、逆に適正がないのではないか。ハッカーは多分、決して受からない。

さて、模倣と自己否定と破壊こそが創造の原動力である。フリーソフトウェアは、自由にソースコードを利用し、自由に改変することが可能であることから模倣を積極的に奨励する。数限りなく当てられるパッチ、繰り返されるバージョンアップは自己否定そのものであり、多くの場合破壊的ですからある。

この論文自体も、何らかの形で有用であることを目指し、日本的な最後の方に結論を持っていくというスタイルを試みた。「ふむ。そうか。」などと頷かず、「こんなものではない」と自分自身の考えで自由に飛翔しよう。

参考文献, URL

- [1] Free Software Foundation, “フリーソフトウェアって何?”, 2000年.
- [2] John Guilmore Home page, <http://www.toad.com/~gnu/>
- [3] Richard M. Stallman, “The GNU Manifesto”, 1985, 1983.
- [4] comp.os.minix, “LINUX is obsolete”, 1992.
- [5] g新部 裕, “GNU/Linux on SuperH Project”,
Linux Conference 2000 (LC2000) 論文集,,
日本リヌックス協会, 2000年11月.
- [6] “技術者のための UNIX 系 OS 入門”, インターフェース増刊
TECHI Vol. 5, CQ 出版, 2000年6月.
- [7] g新部 裕, “フリーソフトウェア, そして知識の再編”,
ソフトウェアシンポジウム 2001 (SS2001) 論文集,
ソフトウェア技術者協会, 2001年6月.
- [8] 未踏ソフトウェア創造事業:
<http://www.ipa.go.jp/NBP/13nendo/13mito/koubo13.htm>
- [9] GNU/Linux on SuperH Project: <http://www.m17n.org/linux-sh/>