りちゃーど すとーるまん

リチャード ストールマン

(フリーソフトウェア財団)

フリーソフトウェア GNU のグル

コンピュータのソフトウェアは、料理のレシピのように、自由に使いあえるようにすべきだ。そうすれば、コンピュータのユーザが自由に協力しあえる世界をつくれる。こう考えたリチャード ストールマンは、フリーなオペレーティング・システムを開発するために GNU プロジェクトを開始した。プロジェクトは、大きな成功を収め、リナックスカーネルと組み合わせた、GNU/リナックスシステムは、マイクロソフト社のウィンドウズに対抗できるオペレーティング・システムとなった。

マンハッタンで育つ

リチャード ストールマンは1953年4月16日生まれである。ニューヨーク市の五つの区のひとつであるマンハッタン区で子供時代を過ごした。区は英語で言うと borough だが、同じ発音の単語に burrow があり、これは動物が地面に掘る穴のことで、彼は「自分は子供の頃は穴に住んでいた。」という冗談を言っている。ストールマンの父は印刷会社を経営し、母はパブリックスクールの教師だった。

小さい頃から、数学と科学に魅せられていた。すべての部品を見ることできる機械に興味があった。子供のころには夏にボーイスカウトのキャンプに行ったが、そこに来ていたカウンセラーがコンピュータのマニュアルを持っていた。IBM社の7094のマニュアルだったことを覚えている。最初にコンピュータのマニュアルを読んだのは1962年というから、九歳の時ということになる。コンピュータを使う事は出来なかったが、紙の上にプログラムを書くことは出来た。その一つは、整数の集まりを入力し、それぞれの数の平方の和を計算するプログラムだった。

プログラムを書くという考えはストールマンを魅了した。

高校最終学年の時に、初めてコンピュータを使うことが出来た。IBM社の360だった。マンハッタンにあったIBM社の研究所に行ってコンピュータを使った。それまでに、プログラミング言語や他のプログラミングテクニックについての本を読んでいたので、自分でプログラムを書くことが出来た。書いたのはPL／1という言語の機能を拡張するプリプロセッサだった。このプログラムは完成しなかったが、優れたアイデアを得ることが出来たし、とても面白かった。とにかく楽しかった。

ストールマンはハーバード大学で物理学を学んだ。同時にプログラミングも学んだが、ハーバード大学のコンピュータ・コースはあまり強くない。そこで、1971年からアルバイトでMITの人工知能研究所(AIラボ)でプログラミングの仕事を始めた[1]。物理学を学んで理論物理学者になろうと思っていたが、プログラムを書いていると、前進しているのがわかるし、結果も出てくる。その喜びの方が物理学を学ぶ喜びよりも大きくなった。それでコンピュータの仕事の方へ進むことにして、AIラボの仕事を続けることにした[2]。

AIラボのソフトウェア共有コミュニティ

この頃のAIラボでは、当時の有力なコンピュータのひとつであるディジタル・エクイップメント社のPDP-10を使っていた。このコンピュータでは、ラボのスタッフがアセンブリ言語で書いたITS(Incompatible Timesharing System)というタイムシェアリング・オペレーティング・システムを使っていた。何人もの人間が同時にコンピュータを使うことが出来るようにするソフトウェアである。

コンピュータが開発されたときから、コンピュータに自分のやりたいことを実行させるためのソフトウェアを書く人間たちのソフトウェア共有コミュニティがあった。自分が書いたソフトウェアを他の人が使いたいと言えば、自由に使ってもらった。逆に誰かが書いたソフトウェアが面白いと思ったら、そのソフトウェアを使わせてもらったり、中身を見せてもらったり出来た。ほかの人が書いたソフトウェアの一部を使って別のソフトウェアを書くことも出来た。ちょうど、料理のレシピを交換しながら、自分が食べたいものを作るのに似ている。

AIラボにも、こういうソフトウェア共有コミュニティが長年にわたって存在しており、ストールマンもその一員となった。このコミュニティの一員としてのストールマンの仕事は、ITSシステムを改良することだった。「ソフトウェアを書くことを愛し、そのための工夫を楽しむ人々[3]」という

---

[1] Think GNU　（シンク　グヌー）--プロジェクト GNU 日記とソフトウェアの憂鬱--引地　信之・引地　美恵子編著、株式会社ビレッジセンター刊より

[2] Interview by Jeremy Andrews、http://kerneltrap.org/node/4484

[3] Richard Stallman 著、（株）ロングテール/長尾高弘訳、「フリーなソフトウェアと自由

意味での「ハッカー」として仕事をしていたのである。

1980年代の初めに、ALラボ出身者が設立した企業である、シンボリックス社がAIラボのほぼすべてのハッカーを雇い入れた。メンバーを失ったコミュニティは崩壊してしまった。1982年に、AIラボが新しいPDP-10を購入したとき、コンピュータの管理者は、ITSでなくディジタル・エクイップメント社のフリーでないタイムシェアリング・オペレーティング・システムをそのマシンに導入することに決めた。また、それからまもなく、ディジタル・エクイップメント社は、PDP-10シリーズの製造を中止した。新しいアーキテクチャの三十二ビットマシンであるVAXに切り替えたのである。

VAXなどのコンピュータはそのコンピュータのためのオペレーティング・システムを持っていたが、それらは、フリーソフトウェアではなかった。ディジタル・エクイップメント社などのコンピュータメーカは、ハードウェアの能力で競争するだけでは不十分で、自社が開発したオペレーティング・システムなどのソフトウェアの機能で競争するようになっていた。そのため、ソフトウェアの内容は公開せず、コンピュータを指定し、そのコンピュータでソフトウェアを使用する許諾をユーザに与えるというやり方でソフトウェアを供給していた。ソフトウェアのソースコードは、コンピュータ企業にとっての門外不出の最高機密事項として扱われるようになってしまった。コンピュータを使う仕事をするには、メーカのこのやり方を受け入れなければならない。そうすると、お互いが自由に共有しあってきた、ソフトウェアの作り方はできなくなり、コンピュータユーザは分断されてしまう。

フリーなオペレーティングシステムを開発しなければならない

この時点で、ストールマンは、厳しい選択を迫られた。ひとつは、コンピュータの世界から離れる、というもので、これは面白くない選択だった。しかも、コンピュータユーザが分断されてしまうということは、変わらない。コミュニティを復活させるために有効なやり方はないか、自問自答した。まず、必要なものはオペレーティング・システムである。フリーなオペレーティング・システムがあれば、新しい共同コミュニティを築くことが出来るだろう。ストールマンはオペレーティング・システムの開発者だったから、この仕事をするのに適切な技能を持っていた。このような状況を考えれば、自分がこの仕事をするのは道徳的な義務だと確信した。そこで、ユニックス互換システムを作るプロジェクトを開始した。プロジェクトの名前は、GNUとした[4]。”GNU” is not Unixという入れ子のフレーズの頭文字をとった。

オペレーティング・システムというと、1970年代では、中核部分のカーネルだけでなくて、コ

な社会」、アスキー刊、28 ページ

[4]南アフリカ産のウシ科の動物ヌーを意味する英語で gnu という単語があり、「ヌー」と発音するが、GNU の場合は「グヌー」と G を発音する。

マンドプロセッサ、アセンブラ、コンパイラ、インタープリタ、デバッガ、テキストエディタ、メーラなどのソフトウェアを含んでいた。GNUオペレーティング・システムも、これらのソフトウェアを持つものにしようと決めた。1984年1月にMITを辞め、GNUソフトウェアの開発を開始した。MITのスタッフとして残っていたら、MITがストールマンの仕事の所有権を主張したり、独自の配布条件を課したりしてフリーソフトウェア開発の障害になるかも知れないので、そういうことがないようにするためであった。AIラボの所長だったウィンストン教授は、ラボの設備をそのまま使えるように計らってくれた。しかし、オフィスは無くなったので、めったに来ない学生にアサインされているAIラボの部屋を使い始めた。その学生は卒業するか、別のところに移っていったが、その後には誰もその部屋を使うようにアサインはされなかった。1991年ごろにストールマンは、MITのResearch Affiliateということになり、大手を振って部屋を使えるようになった。

2003年にAIラボはLCS（Laboratory for Computer Science）と統合されて、CSAIL（Computer Science and Artificial Intelligence Laboratory）となったが、ストールマンは2006年12月現在でもResearch Affiliateとして職員録に登録されている[5]。

GNU プロジェクト

GNUプロジェクトが目指したオペレーティング・システム全体の開発は、非常に大きなプロジェクトである。ストールマンは可能な限り、既存のフリーソフトウェアを利用するか、修正して使うことにした。コンパイラは、Free University Compiler Kit (VUCK)を使おうと思ったが、大学の名前はFreeだが、コンパイラはフリーソフトウェアではないことがわかった。GNUプロジェクトで開発する最初のソフトウェアは、コンパイラにしようとした。ローレンスリバーモア研究所で開発したコンパイラであるパスカルをつかい、C言語をコンパイルできるように、Cフロントエンドを追加した。68000コンピュータへ移植を始めたが、スタックと呼ばれる領域がパスカルの必要とする大きさより、ずっと小さくて使えなかった。この時点で、コンパイラを最初からつくらなければならなくなった。これがGCCと呼ばれるものになった。Cフロントエンドは、何とか修正して活用した。この仕事に実際に取り掛かったのは、GNUプロジェクト開始後、数年後のことであった。

GNUプロジェクトの最初のソフトウェアとして、ストールマンはまずGNU Emacsを書いた。ストールマンは、Emacsについては先駆者だった。1974年にITSのために最初のEmacsの開発を始め、1976年から配布を始めた。Emacsの名称は、Editor Macroからきており、エディターに様々な機能拡張をしたものである。ユニックスで使えるviやedでもエディターの機

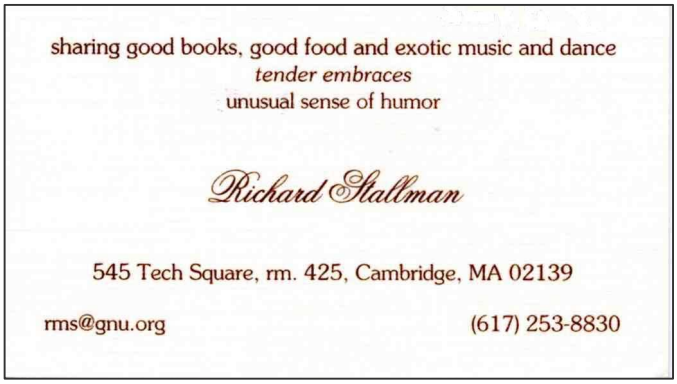
[5] http://www.csail.mit.edu/directory/directory.php

能は使える。しかし、Emacs は拡張性がある。ストールマンは、PDP-10の下で動いていたITS用Emacsと同じ機能を持つ、GNU Emacsを開発することにした。GNU Emacsは、Emacsの新しいバージョンで、ユニックスまたは GNU のもとで動くものである。MIT時代にストールマンが開発したEmacsではなく、ジェームス･ゴスリンがカーネギー･メロン大でユニックス用に開発したゴスリンEmacsをベースに開発した。出来上がると、GNU Emacsを使いたいという人も沢山出てきた。MITコンピュータのFTPサーバから入手できるようにはしたが、インターネットに接続できない人も多く、何か方法を考えなければならなかった。それで、百五十ドルでコピーを販売すると発表した。最初のフリーソフトウェア流通ビジネスを始めたわけである。これが、MITを退職したストールマンが収入を得るのを助けた。

1985年10月にGNUプロジェクトを進めるために、FSF(フリーソフトウェア財団)を設立した。FSFは、コンピュータプログラムの使用、学習、コピー、修正と再配布の権利を増進するための団体である。GNUソフトウェアとフリードキュメントの開発と使用を推進する。ソフトウェアの使用における自由の課題の認識を広げることを支援している。2005年現在、ストールマンが創立以来の理事長で、ボストンの事務所には約十人のスタッフがいる。

プレジャーカード

ストールマンとは、2001年12月の武田賞受賞式で会ったことがある。フリーソフトウェアのグルと言われている人なので、いかめしい人かと思ったら、なかなか魅力的な人物であった。自由に、生きていることを楽しんでいるという面も持っている。最初に会った時に、名刺交換をしたが、彼の名刺は、なかなかユニークであった。下にあるような、ビジネスカードではなく、むしろプレジャーカードというものであった。Sharing が最初にくるのは、共有コミュニティの信奉者らしい。このプレジャーカードは、1989年の秋頃にプライベート用に、特にレディーに渡すために作られたもののようである[6]。この名刺のとおり、美味しい食事をするのは楽しいようで、来日初日にもかかわらず、健啖ぶりを発揮されていた。本拠地のボストンでは、チャイナタウンの飲茶が大好物とのことである。趣味は気取ること、国際フォークダンス、飛行機、



[6] Think GNU （シンク グヌー）--プロジェクト GNU 日記とソフトウェアの憂鬱--
引地 信之・引地 美恵子編著、株式会社ビレッジセンター刊

料理、物理学、リコーダ、言葉遊び、SF 小説ファンであること、それにプログラミングである[7]。飛行機はシングル･エンジンのライセンスを持っている。

手放さないパソコン

ストールマンにとって、インターネットと接続できる環境は、空気のように必須であるようだ。武田賞授賞式のために来日した時も、ホテルの条件として、インターネットの接続が出来るようにして欲しいと言われ、財団が加入していたダイアルアップ回線をホテルの部屋から使ってもらった。食事や授賞式の行事以外の時間は、ほとんどホテルの部屋で、ネットワークに接続して、メールを読んだり、返信を書いたり、ずっと仕事をしていたようだった。

ノートパソコンをいつも肩からかけて持ち歩いており、授賞式のパーティの乾杯の時も、体から離さないほどだった。彼の頭脳は、パソコンを通して外界とつながるもうひとつのチャネルを持っているのかもしれない。記者会見の席上でも、ノートパソコンのキイをたたいており、「質問を受けたくないのではなくて、とてもたくさんの電子メールが来ているのでこの席上でも仕事をしている。遠慮なく質問して欲しい。」と質問をうながしたが、他の受賞者に対する質問よりも少なかったようであった。

受賞メッセージや講演録の確認のための連絡に、うっかりワードの文書を添付したメールを送ったりしたら、必ずフリーソフトウェアの考え方の説明と一緒に、お叱りのメールを頂く。それ以後は、PDF ファイルにして送ることにしている。

コピーレフト

ストールマンは、「コピーレフト」という考え方を作り出した。著作権を使って、多くのフリーソフトウェアのすべての変更、拡張バージョンもフリーソフトウェアにするための方法である。70パーセントのフリーソフトウェアはストールマンが書いたコピーレフトライセンスであるGNU GPL (General Public License) を使っている。

ソフトウェアをフリーにするための、もっとも簡単な方法は、著作権なしの状態で差し出すことである。こうすれば、人々は自由にそのソフトウェアを使えるばかりでなく、ソースコードを見て中身を調べたり、改良したりすることが出来る。その意味では、文字通りフリーである。しかし、このやり方では、この著作権なしのソフトウェアを改良して、改良部分について自分の著作権を主張することが可能である。そうなると、だれもが自由に、中身を理解したり、改良したり出来るようにしたい、という自由が失われる。

そこで、著作権を使って、ユーザの自由を守るために、ストールマンが考え出したのが、著

[7] Richard Stallman's 1983 biography, http://www.stallman.org/#serious

作権を留保する、というコピーレフトという考え方である。だれでも、自由にそのソフトウェアの中身を理解したり、改良したり出来るが、自分が改良したソフトウェアも同じ条件で提供することを条件とする、というものである。その条件に同意できない場合は、そのソフトウェアを改良することを認めない。コピーレフトは、すべてのユーザがソフトウェアにおける自由を持つことを保証してくれる。また、企業や大学は、改良したソフトウェアを放棄するよりも、ＧＰＬにしたがってフリーソフトウェアとしてリリースする方を選ぶだろうから、そこで働いているプログラマが、フリーソフトウェアを改良した場合に、それをコミュニティに提供できるようにすることにも有効である。私有ソフトウェア企業が、著作権(コピーライト)を使って、ユーザの自由を奪うが、ストールマンたちは、著作権を使ってユーザの自由を保証する、わけである。これがコピーレフトである。自由にソフトウェアの中身を理解したり、改良したりするためには、ソースコードが公開されていなければならないので、ソースコードの公開を条件としている。このコピーレフトの考え方に基づいて、法律的な整備をして契約書にしたものがＧＰＬである。

私有ソフトウェア企業の、ソフトウェアの著作権(コピーライト)とコピーレフトの概念の違いを図に表すと、図１と図２のようになる。

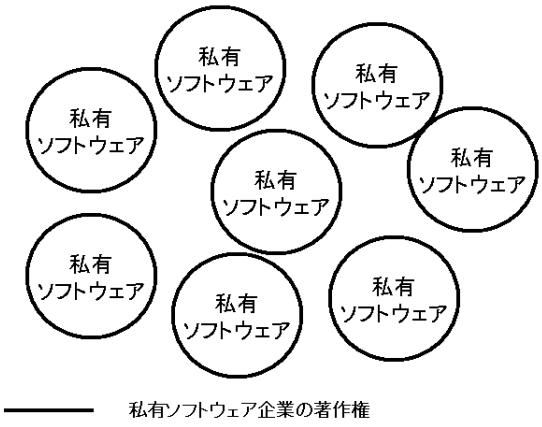


図１　私有ソフトウェア企業の著作権(コピーライト)

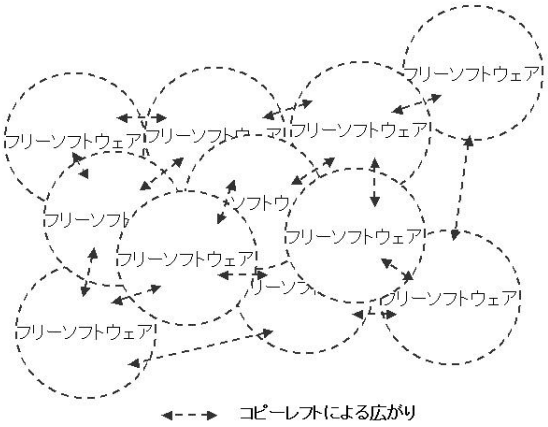


図 2　フリーソフトウェアのコピーレフト

フリーとオープン

ここで、フリーソフトウェアとオープンソース・ソフトウェアについて整理しておきたい。どちらも、そのソフトウェアを改良したり、バグ修正をしたりする条件として、改良後や、バグ修正後のソフトウェアのソースコードをオープンにすることを条件にしている。大きな違いは哲学的な価値観にある。フリーソフトウェアは、倫理的な課題として自由と社会的共有に焦点を当てている社会運動であるのに対して、オープンソース・ソフトウェアは、技術的に優れたソフトウェアを開発する方法論である、と考えるとわかりやすい。つまり、オープンソース・ソフトウェアにとって、ソースコードが公開されていないソフトウェアは、劣位の選択肢であり、組み合わせて使うことは問題とは考えない。フリーソフトウェア運動にとっては、非フリーソフトウェアは、社会問題であり、受け入れられるのは、フリーソフトウェアだけである。

フリーソフトウェア運動の人たちは、オープンソース・ソフトウェアには、反対はしないし、貢献してきたことは認めるが、コミュニティを作ったのは自分たちであり、その主張を聞いて欲しいと思っている[8]。

オープンソース・ソフトウェアとフリーソフトウェアは、組み合わせて使うことが出来るし、現実的な場面で推奨する方向は一致している。オープンソース・ソフトウェアである、リナックスもフリーソフトウェアが生み出したＧＰＬという契約を採用している。今日でもリナックスがひとつのソースコードツリーを維持し、枝分かれしていないのは、ＧＰＬによるところが大きい。

フリーソフトウェアとオープンソース・ソフトウェアのそれぞれが、他のソフトウェアをどのように見ているかを図に表すと図3と図4のようになる。

---

[8] Richard Stallman 著、（株）ロングテール/長尾高弘訳、「フリーなソフトウェアと自由な社会」、アスキー刊、89, 90 ページ

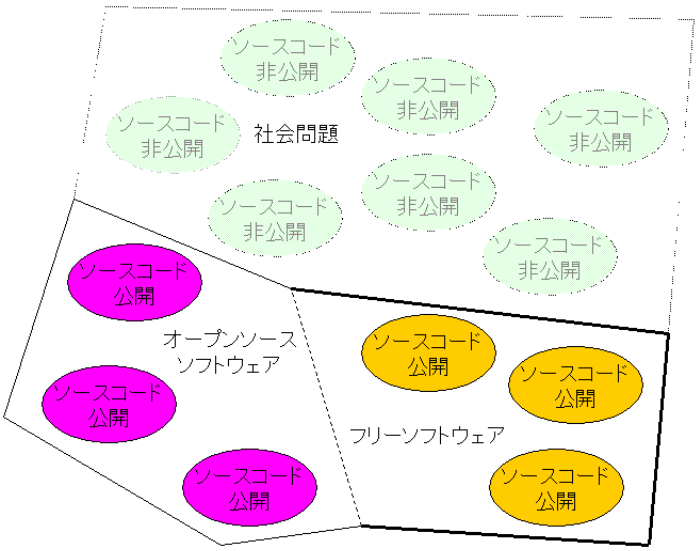


図３　フリーソフトウェアの場合

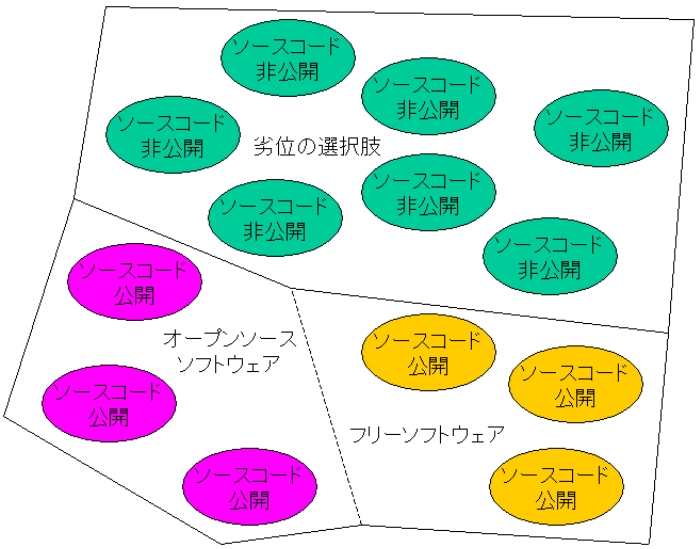


図４　オープンソースソフトウェアの場合

ストールマンは、「自由」に価値を見出さない人は、その「自由」を失ってしまうだろう、と警告している。「自由」を与えるときに、「自由」の価値を教えておかなければ、いずれ与えられた自由はなくなってしまうからである。

GNU／リナックス

リーナス・トーバルズたちの開発したリナックスカーネルは、フリーソフトウェアと組み合わされて使われ、オペレーティング・システムとして必要なソフトウェアの機能を提供することが出来る。このフリーソフトウェアの貢献をはっきり表すために、GNU／リナックスと呼ぶべきだとス

トールマンは主張している。

また、ストールマンはGNU／リナックスはGNU スラッシュ リナックスまたはGNU プラス リナックスと発音すべきだ、と言っている。GNU／Emacsは、GNU が開発したEmacsという意味だから、GNU／リナックスと言うとGNUが開発したリナックスという意味になってしまうからだ。GNU プラス リナックスと言うことでそういう誤解は避けられる。リナックスは、Just for fun で開発されたオペレーティング・システムで、様々な人がリナックスを共有し、自分が使いやすくなるように改良し、改良したものをまた共有している。その結果、実用的で表面的な価値を追求することになってしまっている。この点ではマイクロソフトと同じだ。GNUは自由のために戦うことにした人々のためのシステムである。GNUの考え方とリナックスの考え方の違いをはっきり表現するためには、GNU プラス リナックスと呼ぶべきだ、と言っている[9]。

フリーでないソフトを使うか

旅先などでは、ストールマンでも、フリーなソフトウェアが使える環境にいつもいられるわけではない。このような時にはどうするのだろうか。フリーでないソフトウェアを使うことがあるのだろうか。カーネルトラップというウェブマガジンのジェレミー・アンドリューズ記者のインタビューに次のような答えをしている。「フリーでないソフトが走っているサーバに接続することを拒否はしないし、フリーでないソフトが走っているコンピュータでタイプする事は拒否しない。もし、私があなたの家を訪問していて、あなたがウィンドウズマシンしか持っていないとして、それを使う重要なことがあればそのマシンを使う。ウィンドウズを自分のマシンに入れたいとは思わないし、あなたのマシンにも入れて欲しくないが、そのマシンに触らないことであなたがウィンドウズマシンしか持っていないことが変わるわけではない。」

「もし、フリーでないソフトが走っているマシンに接続しても、あなたの自由が犯されるわけではない。サーバのオペレータの（ソフトウェアに関する）自由は、走らせているソフトの制限で侵されているが。しかし、そのサイトがフリーソフトに切り替えるまでボイコットしようとは思わない。彼があなたにフリーでないソフトを使わせているわけではないから。」

例えばグーグルをオンラインのコンテンツを探し出すのに使うかといえば、グーグルネットワークサーバとコミュニケートすることには反対しない。グーグルのためには、彼らのサーバで走っているソフトについて学び、変更し、配布する自由があったらよいとは思うが、そういう自由を持つ事は義務ではない。グーグルは走らせているソフトウェアを変更して配布しなければならないわけではない。しかし、そうする自由を、つまりストールマンが自分のマシンのソフトについて持っている自由と同じ自由を持つべきだと思っている。

---

[9] Interview: Richard Stallman by Jeremy Andrews, http://kerneltrap.org/node/4484

あなたの仕事が、フリーでないソフトを使うことを要求したらどうするか、と言う質問に対しては、「その仕事を辞める。誰かが給料を払ってくれるからといって反社会的なことをするのと同じことだ。もし、あなたの仕事が街角で人の頭を殴って財布をとるとしたらどうするか。誰かが給料を払ってくれるというのは言い訳にならない。」と答えている[10]。

ソフトウェア特許

ストールマンは、ソフトウェア特許については、非常に憂慮している。個々のプログラムに特許を与えても、それは今までと変わりはなく、現状より事態を悪くするわけではない。しかし、特許は、何らかのアイデアを含んでおり、ソフトウェア特許は、ソフトウェアのアイデアが特許になる。これが、ソフトウェアの開発を危機にさらすと、考えている。

特許と製品の関係は、分野によって大きく異なる。ひとつの化学式、つまりアイデアからひとつの薬品が出来る、という薬品のような分野もあれば、ひとつの製品に多くの特許、つまり多くのアイデアが関係している分野もある。最近では、製薬関係の特許の中には複数の薬をカバーする特許も出ている。ソフトウェアは、多くのアイデアが関係する最たる分野である。それは、ソフトウェアの世界では、概念化された数学的なものを開発しているからである。実体のあるものを開発しているときに心配しなければならない、部品の大きさだとか、発熱だとか、腐食だとか、部品の物理的条件などを心配する必要はない。だから、他の分野に比べたら非常に少ない工数で、膨大な数の部品を持ったソフトウェアを開発することが出来る。逆に言えば、ソフトウェアは膨大な数の部品をもっている。ということは、沢山のアイデアを使って開発することを意味する。それらの多数のアイデアは、世界で始めて考え出されたものもあるだろうが、部品に分解すれば、今まで誰かが考えたアイデアであることも多い。それらのアイデアが、特許になっているかどうか、を全部調べるだけでも、膨大な工数がかかってしまう。一つの製品の開発の費用の大きな割合をしめてしまう。それでも、大きな会社であれば、自分が持っている特許を梃子にして、クロスランセンス契約を結ぶと言うやり方が出来る。自社の持っている特許の使用権を相手に認める代わりに、相手の持っている特許の使用権を認めさせる契約である。そうすれば、自分の開発したソフトウェアで使っている沢山の数のアイデアが、相手の会社が持っている特許を侵害していないかを心配しなくても良くなる。こういう契約をいろいろな会社と結び合えば、大きな会社は、ソフトウェア特許の権利を守りながら、開発をすることができる。しかし、個人でソフトウェアを開発する人や、小さな企業は、自分が所有するソフトウェア特許が少ないから、クロスライセンス契約を結ぶというやり方は出来ない。フリーソフトウェアやオープンソース･ソフトウェアも、クロスライセンス契約を結ぶというやり方は出来ない。

[10] Interview: Richard Stallman by Jeremy Andrews, http://kerneltrap.org/node/4484

また、ソフトウェア特許のアイデアの中には、すでに同じようなアイデアが使われていたり、公開されていたりしていると思われるものもある。このような場合でも、裁判をして特許を無効にするのは、大変な労力と資金を必要とするので、やらないことが多い。音楽の旋律に特許を認めたら、作曲家が多くの旋律のアイデアを組み合わせる交響曲を作曲することがどんなに難しくなるか、を考えてみればわかるように、ソフトウェア開発にとってソフトウェア特許は、大きな脅威であるとストールマンは考えている。

不本意ながらマネージャに

最近、ストールマンは転んで腕を折り、手術を受けた。今では、自分でタイプできるように回復したが、痛みはあるそうだ。自分でタイプ出来ない時期があったせいではないようだが、ストールマンが自分自身でプログラムを書くのは、Emacsについてほんの少しだけになってしまった。マネジメントとアクティビズムが多くなり、プログラミングは他の人にやってもらわなければならなくなったからだ。ストールマンは不本意ながら自分自身をマネージャに昇進させた、と言っている。さらに、「自分がもっとも誇りに思うことは、人々が自由をもってコンピュータを使い、共同して仕事ができるコミュニティを作ったことだ。プログラミングの代わりにマネジメントとアクティビズムをやりたいとは思わないが、コミュニティを発展させていくためには必要なことだ。現時点で私に代わる人がいない。これに私が絶対必要ではなくなるように、これが出来る人を育てようと思っている。現在の自分の役割は、堅固な断固としたリーダであり、説法者だ。」と語っている。

また、「自分は、性格的に悲観主義者だ。多くの人は勝つと期待して戦う。自分は負けると期待して戦う。それでも戦い、時に勝つ。」とも言っている。自分の信念に基づいて行動するフリーソフト運動のグルらしい発言である。

オープンソース・ソフトウェアの開発には、大企業に籍を置く技術者もたくさん参加するようになった。ソースコードが公開されているので、オープンソース・コミュニティーは、ソフトウェア企業にとって、ある種の共同研究所、共同開発部門としての役割を果たすようになってきた。ストールマンが目指した共同コミュニティとまったく同じではないが、ソースコードを公開し、お互いに利用しあうコミュニティは大きな広がりを見せている。この広がりが発展し続けていくためにも、厳しすぎるとも思われるストールマンの哲学的な「自由」という考え方は、貴重であり、大きな波及効果をもたらしたと言えるだろう。

略歴

1953年　ニューヨークのマンハッタンで生まれる

1970年　IBMのニューヨークリサーチセンターでプログラム作成を体験

1971年　ハーバード大学入学、同時にMITでプログラム作成に従事し1984年までMIT勤務

1974年　ハーバード大学物理学科を卒業

1976年　Emacsの配布を開始

1984年　MITを退職、GNUプロジェクト開始

1985年　FSF(フリーソフトウェア財団)設立、理事長となる

1996年　スウェーデンの王立技術研究所から名誉博士号を授与

1999年　ユーリ・ルビンスキー記念賞受賞

2001年　グラスゴー大学から名誉博士号を授与

2001年　武田賞受賞