

第 29 章 オープン・ソース・ソフトウェア

「オープン・ソース・ソフトウェア」とは何か

インターネット上での辞書の 1 つである「e-words」によると、「オープン・ソース・ソフトウェア」は次のように説明されている。

「ソースコードを、インターネットなどを通じて無償で公開し、誰でもそのソフトウェアの改良、再配布が行えるようにすること。また、そのようなソフトウェア。」

普通ソースプログラムはそれを開発した企業の最も重要な機密事項の 1 つであり、それを公開することなどとうてい考えられない。例えばマイクロソフトの word や excel などのソースプログラムが公開されることは、現状では想像もできない。

ソースプログラムの機密を巡っての有名な事件に、1982 年に起きた「IBM 産業スパイ事件」がある。当時 IBM のメインフレーム（大型コンピュータ）と互換性のあるコンピュータを作っていた日立と三菱電機の社員がアメリカで、IBM の OS のソースプログラムについての機密を盗もうとしたとして FBI に逮捕された事件である。FBI がおとり捜査を行ったためやり方が汚いとして「IBM 産業スパイ事件」などと呼ばれているが、この呼び方は IBM には気の毒かもしれない。

同じ頃にやはり IBM 互換機を作っていた富士通はこの事件に巻き込まれることはなかったものの、富士通がユーザに提供していた OS の中に IBM が作ったソースプログラムが一部含まれていたとして、やはり IBM が持つ OS の著作権との関わりで莫大な金額を IBM に支払ったとされている。

この例のようにソースプログラムは一般に非常に機密性の高い、重要なものとして取り扱われている。しかし世の中には、この常識を破るソフトウェアがずっと以前から存在していた。このようなプログラムは、当初は「フリー・ソフトウェア」などと呼ばれていた。この章では、これらのフリー・ソフトウェアについて議論する。

英語の「フリー (Free)」には、「自由の」と「無料の」の 2 つの意味がある。ここではまず、「無料の」ソフトの代表格だった Unix の話から始めたい。

Unix

1960 年代に、ジェネラル・エレクトリック社 (GE)、マサチューセッツ工科大学 (MIT) と AT&T (American Telephone and Telegram) のベル研究所 (ベル研) が合同で始めたプロジェクトがあった。GE-645 という大型コンピュータ上でタイム・シェアリング・システムを稼働させるためのオペレーティング・システム (OS) を開発するプロジェクトで、Multics (Multiplexed Information and Computing System : マルティクス) と名付けられていた。

最終的にこのプロジェクトは目的とする成果物を完成するが、一時期パフォーマンスの悪さからその完成が危ぶまれた時があった。このためその時ベル研からそのプロジェクトに出向いていたケネス (ケン)・トンプソン (Kenneth Thompson) は、呼び戻されてベル研に帰ってきた。しかし戻っては見たものの、ベル研で彼が行うべき当面の仕事がなかった。そのため彼は、誰にも使われずに放置されていた DEC (Digital Equipment Corporation) 社製のミニ・コンピュータ PDP7 を使って、もっとシンプルで、Multics のような OS を実現しようと考えた。

当初彼は PDP7 のアセンブリ言語を使って開発を行ったが、その後やはり当時ベル研でデニス・リッチー¹ (Dennis MacAlistair Ritchie) が生み出した C 言語を使っての開発に切り替えた。Multics が多くのユーザを対象にしていたのとは異なり、彼の OS は 1 人だけのユーザを対象にしていたので、彼はその OS に Unics (Uniplexed Information and Computing System: ユニックス) という名前をつけた。Unics はその後、Unix に改称された。

1970 年代初めには、Unix は PDP7 の後継機であり、「名機」の誉れの高いやはり DEC 社製の PDP11 上で稼働するようになった。しかしその頃米国政府は独占禁止法を厳しく適用しており、IBM は通信のビジネスに参入することを禁止され、同時にベル研の親会社である AT&T はコンピュータのビジネスへの参入を禁止されていた。そのため AT&T は Unix の入手を希望する大学や企業、政府機関などにこのソースプログラムを、コピー代程度を請求するだけで、タダ同然の代金で配布した。

配布を受けた大学などの多くのところはこのプログラムの内容を解読し、それを通して OS の理論とその実践方法を習得し、併せてそれぞれが必要と考える機能を競って付加した。そのため一時期多くの Unix が、互換性も機能の統一もないまま同じ名前を持って多数存在するという一種の混乱状態を生じた。余談だが、この Unix の普及と共に C 言語の普及も進んだ。またこの後で述べるように、Unix のタダ同然の配布が「フリー・ソフトウェア」の考え方を形成することやその実現に、大きな力になった。

この時期 AT&T は、前述の通りほとんどタダのような費用で Unix を配布したけれど、著作権まで放棄したのではないことに留意したい。この Unix の混乱はその後徐々に収束し、最終的に AT&T が開発と配布に責任を持つもの (SVR4 (System V Release 4: エスヴィアールフォー) と呼ばれた) と、カリフォルニア大学バークレイ校が主導するバークレイ版 (BSD: Berkley Software Distribution) の 2 つに集約された。この 2 つは、いずれも有償の Unix だった。

DEC はその後、32 ビットの仮想記憶の機能を持つ VAX11 を発表した。AT&T 版の Unix はこのコンピュータに対応できず、一方の BSD は VAX11 上でその機能を生かして稼働できた。またインターネットの普及と共に BSD はインターネットのプロトコルである TCP/IP も取り扱えるようにしたため、一時期こちらの方が人気が高かった。

その後米国政府の独占禁止法の取り扱い方が変わり、AT&T もコンピュータのビジネスに参入できるようになった。そこで AT&T は Unix でビジネスを展開するよう方向転換し、カリフォルニア大学バークレイ校に著作権上の問題を投げかけて、その開発にクレームをつけた。この結果「Unix は AT&T のもの」という、ある意味ですっきりした形を取り戻した。BSD はその後、フリー (無料) の Unix の方へ方向転換をしてゆく。

AT&T はその後、この Unix の著作権を子会社の USL (Unix System Laboratory) を経由してノベルに売却し、ノベルはそれをさらに X/Open に譲った。今では、Open Group という団体がその著作権を所有している。今は IBM やヒューレット・パカード、日立など多くのコンピュータ・メーカーが自社のコンピュータで Unix を使えるようにするために、多くの商用の Unix が存在している。これらは全て、この Unix の著作権を持っているところから、Unix を開発し、配布する権利を取得したものである。一方で、後で述べる Linux などの、多くの無償の Unix が存在している。

¹ その後でデニス・リッチーも Unix の開発に参画するようになる。デニス・リッチーが生み出した C 言語は、ケン・トンプソンが作った B 言語をベースにしている [Wik291]。

フリー・ソフトウェア

1983 年に、当時 MIT の人工知能研究所でプログラム開発の仕事をしていたリチャード・ストールマン (Richard M. Stallman) は「ソフトウェアはフリーであるべきだ」との信念を持ち、それを実現することに力を尽くすことになる。彼が MIT で開発したソフトウェアを有償で配布しようとした企業があったことが、このきっかけになったといわれている。

具体的には、1984 年に彼は新しいフリー・ソフトウェアの開発のために GNU (GNU is Not UNIX : 「グニュー」と読む) プロジェクトを立ち上げ、さらにどこでもそのソフトウェアを使えるようにすることを目的に「フリーソフトウェア財団 (FSF : Free Software Foundation)」を設立した。

さらに 1989 年には、GPL (GNU General Public License) の最初の版を公表した。GPL は「コピーレフト (Copy Left)」の考え方を具現化したもので、「コピーレフト」とは著作権の英語の単語 (Copyright) をもじって、彼のフリー・ソフトウェアについての考え方を示したものである。

GNU はこれまで、GCC (GNU コンパイラ・コレクション) と呼ばれる多くのコンパイラ群 (ここには C、C++、Java、Ada、FORTRAN、Pascal などのコンパイラが包含されている)、Emacs というエディタ、ビルド作業を自動化する Make、デバッグ・ツールなど、多くのフリーなソフトウェアを開発してきた。また後で述べる Linux も、それ単独では OS としての機能を果たすことができないため、GNU プロジェクトが開発した多くのプログラムと共に使用されており、GNU/Linux と呼ばれることもある。

GPL は今では第 3 版が発行されており、フリー・ソフトウェアの考え方が時間の経過と共に変わってきたことを示している。しかし GPL そのものはまだ完全な「自由」を実現したとはいえない、そのためそれぞれの人が理想とする本当の「フリーなソフトウェア」を実現するために、GPL の他にも多くのライセンスが存在しているのが現状である [Wik292]。

「本当のフリー」の中には、当初のフリー・ソフトウェアを改変したものや、それらを全体の中に一部として組み込んだものを、「無償で配布する自由」だけでなく、「有償で配布する自由」も含まれることになる。

Linux

Linux の起源は 1991 年に、当時フィンランドのヘルシンキ大学に在学していたリーナス・トーバルズ (Linus Torvalds) が独力で OS のカーネルの開発を行ったことに遡ることができる。当時は Intel の 80386 というプロセッサを組み込んだ PC が普及を始め、強力なコンピュータが個人でも入手できる値段で出回り始めていた。彼はこのコンピュータの上で Unix か、それと互換の OS を走らせてみたいと考えた。

しかし商用の Unix は高価だった。BSD 版の Unix は AT&T との係争のため、あまり大きな動きが見られなかった。オランダのアンドリュー・タネンバウム (Andrew Stuart Tanenbaum) 教授が作成した Minix という Unix 風の OS があったが、これも教育を目的とするために機能が大幅に簡略化されていた。このようなことから彼は、Minix をベースにして自分で OS のカーネルを開発することにした。この当初の開発に、4 ヶ月かかったという。

この OS を彼は Minix について議論するメーリング・リスト上に公開し、GPL のライセンスの下で利用や改変ができるようにした。彼と同じようなことを考え、同じような悩みを抱えてい

た多くの人たちが、彼のこの OS をダウンロードし、場合によればそれに手を加えて使用した。

この頃、Minix の生みの親であるタネンバウム教授とリーナス・トーバルスの間で、メーリング・リスト上で、Linux についての論戦が展開された。タネンバウム教授は、「1990 年代にマイクロ・カーネルの方式を採用しない OS は意味がない」と Linux に反対を唱え、トーバルスは「Linux は動いており、それだけで大きな意味がある」と反論した。

図表 29-1 Linux についての経緯

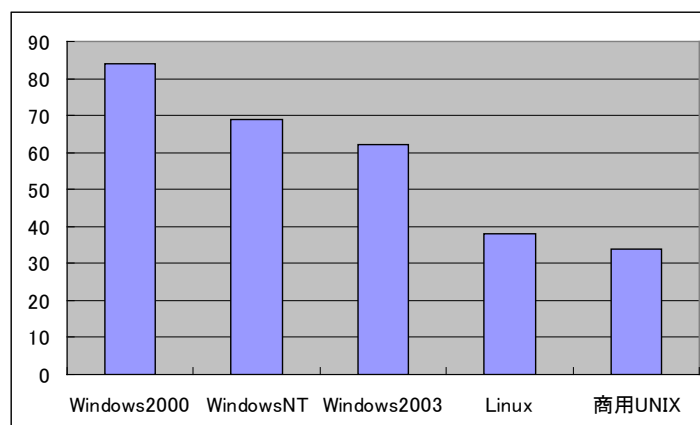
年	Linux バージョン No.	ソースコード 行数	ユーザ数
1991 年	0.01	約 1 万行	1 人
1992 年	0.96	約 4 万行	1000 人
1997 年	2.10	約 80 万行	350 万人
2007 年		約 500 万行	

その後 Linux についてのメーリング・リストが作られ、このメーリング・リストを使って、Linux が改訂の都度公開され、多くの優秀なエンジニアがボランティアでこの開発に参画するという、我々が既によく知っている「オープン・ソース・ソフトウェア方式」の開発が行われることになる[[Wik293](#)]。

Wikipedia 日本語版上に、おもしろい数字が公開されている。それを図表 29-1 に示す。この表上の 1991 年のユーザ数 1 人とは、リーナス・トーバルス自身を指す。

Linux が順調にユーザ数を増やす状況を見て、マイクロソフトはその中にその主力製品である Windows が Linux に負けるのではないかと危ぶみ、状況を分析し、対応を検討した。当然この文書はマイクロソフト内の機密書類だったはずだが、なぜかそれが漏洩し、公表されてしまった。1998 年 11 月初めのことで、公表された日が 10 月 31 日に近いことから、これは「ハロウィーン文書」と名付けられた[[RAY99](#)]。

このような開発方法をとったソフトウェアは、これまでになかった。だから当初はこのような開発の方法ではうまくゆかないと考える人が圧倒的に多く、短時間に品質上の問題を含めて次々と問題を解決し、機能も順調に拡張している Linux に、大きな驚きの目が向けられた。

図表 29-2 サーバ機での OS のシェア ([[JUAS06](#)]より)

エリック・スティーブン・レイモンド (Eric Steven Raymond) はプログラマとしていくつかの GNU プロジェクトに参画し、多くのフリー・ソフトウェアの開発に貢献してきた。彼は、彼が参画したフリー・ソフトウェアの開発方法は「教会の大聖堂」のように秩序正しい、整然としたものであるのに対して、Linux の開発方法は「広場で開かれているバザー」のように統制のない、混沌としたものであると考えた。そして彼がそれまで経験したことがない Linux のような開発方法で、なぜ品質の高いソフトウェアが開発されているのかに興味を持った。

そこで彼は機会を見つけて、彼自身が主導権を取って電子メールに関わるあるソフトウェアの開発をオープン・ソース・ソフトウェアの方式で行った。結果は、見事な成功だった。彼はこの経緯などを「伽藍とバザール」という論文にまとめて、インターネット上で公開した [RAY97]。

Linux の成功は既に常識だが、それではどの程度の成功を収めているのだろうか。MIT のネグロポンテ教授が提唱している開発途上国の子供向けの 100 ドル・コンピュータでは、OS は Linux の使用が提唱されている。しかし現実には、Linux はまだクライアント機での使用は少ない。Linux の使用が高いのは、サーバの領域である。

日本の企業でのコンピュータの使い方を継続的に調査している日本情報システム・ユーザー協会 (JUAS) の「企業 IT 動向調査 2006 報告書」には、図表 29-2 に示す結果が記載されている [JUAS06]。このグラフから分かるように、Linux は既に商用 Unix を追い抜いて、日本でもサーバ OS として広く利用されている。ただしここでの「シェア」は、一部でもその OS を導入してサーバで使用している企業の、調査した企業全体に占める割合である。

「オープン・ソース・ソフトウェア」

1998 年に、マイクロソフトのインターネット・エクスプローラ (Internet Explorer) との競争でその主力製品の 1 つであるネットスケープ・ナビゲータ (Netscape Navigator) のシェアが低下してしまったネットスケープ社は、エリック・レイモンドなどを招いてその対応を検討した。その結果、製品の機能と品質を向上させ、シェアの向上を図るために、技術者が誰でも開発に参画できる Linux のような開発方式を取り入れることにした。併せてこのようなソフトウェアを、「オープン・ソース・ソフトウェア」と名付けることにした。

このインターネットのブラウザの開発も見事に成功し、今ではモジラ・ファイアフォックス (Mozilla FireFox) という名前になって、私を含めて広く世界中の人に使われている。

世界中でオープン・ソフトの開発に参画している技術者は、1990 年代後半では数千人だった [RAY97]。しかし今では 300 万人といわれ、その半数はヨーロッパ、4分の 1 がアメリカからで、圧倒的に人口が多い中国とインドの寄与がこの世界では少ない [UME07]。オープン・ソース・ソフトウェアの開発に参画するソフトウェア技術者の数は直近の約 10 年で、1,000 倍見当に増加したことになる。しかもその参加者は、30 代前半ぐらいまでの若い層が多いという [UME07]。

なぜ優秀なソフトウェア技術者は、ボランティアでオープン・ソース・ソフトウェアの開発に参画するのだろうか。この問題は、結局は「人なぜ働くのか」という問題に行き着く。これまでは、経済学や企業理論の立場から「人は雇用関係や金銭的な契約に基づく『強制力』によって動くもの」と考えられてきた。しかしオープン・ソース・ソフトウェア方式の開発が繰り返され、多くの成功事例を目にするようになって、この古典的な考え方では説明しきれ

ない世界が出現した[UME07]。

梅田氏はこれについて、以下のように述べている。「豊かな先進国の、若い知的階層に所属する優秀なソフトウェア技術者たちは、国籍・性別・年齢などに一切関係なく専門的な知的関心が同じであるという点だけで出会い、一緒に仕事を始める。チームが共演する劇場的空間ともいべき世界で、イノベーションの連鎖が起きる。この間全ての情報が公開・共有され、プロジェクトへの個の貢献は構成物の成果によって評価される。成果が上がれば、自分の達成を深く理解する他者から注目され、賞賛される。そんなオープンなプロセス全体から得られる満足感こそが、無償で『働く』原動力になっている。」 [UME07]

この「人は雇用関係や金銭的な契約に基づく『強制力』によって動くもの」という「原理」が、ソフトウェア技術者の世界では成立しなくなった。このことが、ソフトウェア技術者の管理を難しくしているという側面がある²。

オープン・ソース・ソフトウェアは単に OS やメーラ、ブラウザなどだけでなく、今では ERP (Enterprise Resource Planning) などのアプリケーション・プログラムの領域でも目にすることができるようになった。主なオープン・ソースのソフトウェアには、以下のようなものがある。

- FreeBSD など、BSD 系の Unix (OS)
- Linux (OS)
- Mozilla FireFox (ブラウザ)
- Mozilla Thunderbird (メーラ)
- OpenOffice.org (マイクロソフト・オフィスのようなソフトウェア群)
- GCC (コンパイラ群)
- GNU Emacs (テキスト・エディタ)
- X Window (ウィンドウ・システム)
- Samba (ファイル・サーバ)
- Wiki 関連の多くの製品 (Wikipedia など使われているプログラミング)
- Eclipse (Java の開発環境)
- NetBeans (Java の開発環境)
- PostgreSQL (リレーショナル・データベース管理システム)
- MySQL (リレーショナル・データベース管理システム)
- Apache Tomcat (Web コンテナ)
-

エリック・レイモンドは、次のような性格を持つソフトウェアが、オープン・ソース・ソフトウェアとして適していると述べている[RAY97]。

- 信頼性、安定性、スケーラビリティがとても重要な場合。
- デザインや実装の正しさが、独立ピアレビュー以外の方法ではきちんと検証できない場合。
- そのソフトが利用者のビジネス展開を決定的に左右するような場合。
- そのソフトが、共通のコンピュータ・通信インフラを確立するか、可能にする場合。
- その核になるメソッド (あるいは機能的にそれと同等のもの) が、よく知られた工学的知識の一部である時。

² ソフトウェア技術者の特質は、第 42 章で述べる。

その上で彼は、それぞれの領域での将来のソフトウェアの開発方式について、「魔法のおなべ」という論文の中で次のように述べている[RAY97]。

「OS やインターネット関係のインフラのソフトウェアは、ほとんど全てオープン・ソース・ソフトウェアになるだろう。アプリケーションは、非公開のままでもどまりやすい。データベースや開発ツールなどのミドルウェアは、両者が混ざってくるだろう。」

いずれにしろオープン・ソース・ソフトウェアは、既にソフトウェアの世界ではなくてはならない存在になっている。今後は、さらに重要なものになってゆくだろう。

Ruby

日本発のオープン・ソース・ソフトウェアもある。まつもとゆきひろ（松本行弘）氏がリーダーになって開発している Ruby というソフトウェアである。

彼は筑波大学でソフトウェアを専攻した後、ソフトウェア会社で仕事をする傍ら、20 代後半から、好きでたまらなかつたプログラムの言語の設計を始める。そして 1993 年に「Ruby」というプログラム言語を作って、1995 年から公開する[UME07]。

Ruby はオブジェクト指向のスクリプト言語という範疇に入るプログラム言語だが、同じ範疇には Perl という言語がある。Perl の発音は真珠 (Pearl) と同じであり、真珠が 6 月の誕生石なので、7 月生まれの彼はその月の誕生石であるルビー (Ruby) をその言語の名前にした。

Ruby は、今では世界中で数十万人が使う言語となり、何千もの開発プロジェクトがこの言語を使って動いている[UME07]。まさに、東京大学の坂村健教授が主導したトロン・プロジェクトの CTIRON に次ぐ、日本発の、世界に通用するソフトウェアの 1 つとなった。

いまや 20 代から 30 代前半ぐらいまでの若いソフトウェア技術者に目標とする技術者名を挙げて貰うと、これまでの「チューリング賞」の受賞者などに代わって、まつもとゆきひろ氏の名前が挙がるという。これからの彼の一層に活躍を、期待したい。

キーワード

IBM 産業スパイ事件、フリー・ソフトウェア、Unix、GNU、フリーソフトウェア財団、GPL、コピーレフト、Linux、オープン・ソース・ソフトウェア、ハロウィーン文書、Ruby

略語

AT&T : American Telephone and Telegram
 Multics : Multiplexed Information and Computing System
 DEC : Digital Equipment Corporation
 Unix (Unics) : Uniplexed Information and Computing System
 BSD : Berkley Software Distribution
 USL : Unix System Laboratory
 GNU : GNU is Not UNIX
 FSF : Free Software Foundation
 GPL : GNU General Public License

人名

ケネス (ケン)・トンプソン (Kenneth Thompson)、デニス・リッチー (Dennis MacAlistair Ritchie)、リチャード・ストールマン (Richard M. Stallman)、リーナス・トーバルズ (Linus Torvalds)、アンドリュー・タネンバウム (Andrew Stuart Tanenbaum)、エリック・スティーブン・レイモンド (Eric Steven Raymond)、まつもと ゆきひろ (松本行弘)

参考文献とリンク先

[JUAS06] 日本情報システム・ユーザー協会、「企業 IT 動向調査 2006 報告書」、日本情報システム・ユーザー協会、2006 年。

[RAY97] エリック・スティーブン・レイモンド著、山形浩生訳、「伽藍とバザール オープン・ソース・ソフト Linux マニフェスト」、光芒社、平成 11 年。

この本には「伽藍とバザール」、「ノウアスフィアの開墾」と「魔法のおなべ」の 3 つの論文が掲載されている。その「伽藍とバザール」の部分は、次の URL からダウンロードできる (確認日: 2017 年(平成 29 年)1 月 24 日)。

<http://cruel.org/freeware/cathedral.html>

この本の原書は、以下のものである。

Eric S. Raymond, “The Cathedral & the Bazaar Musings on Linux and Open Source by an Accidental Revolutionary,” O’Reilly, 2001.

またこの「伽藍とバザール」の原文も、次の URL からダウンロードできる (確認日: 2017 年(平成 29 年)1 月 24 日)。

<http://www.catb.org/%7Eesr/writings/cathedral-bazaar/>

(原書も含めてこの本は、先にそれぞれの論文がインターネット上で発表され、それまとめて後で本の形にして出版したという経緯がある。)

[RAY99] エリック・スティーブン・レイモンド著、山形浩生訳、「ハロウィーン文書」

この資料は、以下の URL からダウンロードできる (確認日: 2017 年(平成 29 年)1 月 24 日)。

<http://cruel.org/freeware/halloween.html>

この資料の原文も、以下の URL からダウンロードできる (確認日: 2017 年(平成 29 年)1 月 24 日)。

<http://www.catb.org/~esr/halloween/>

[UME07] 梅田望夫著、「ウェブ時代をゆくーいかに働き、いかに学ぶか」、ちくま新書 687、筑摩書房、2007 年。

[Wik291] 「C 言語」、<http://ja.wikipedia.org/wiki/C%E8%A8%80%E8%AA%9E>

(確認日: 2014 年 (平成 26 年) 6 月 12 日)。

[Wik292] 「GNU General Public License」、<http://ja.wikipedia.org/wiki/GPL>

(確認日: 2014 年 (平成 26 年) 6 月 12 日)。

[Wik293] 「Linux」、<http://ja.wikipedia.org/wiki/Linux>

(確認日: 2014 年 (平成 26 年) 6 月 12 日)。

(2007 年 (平成 19 年) 11 月 30 日 初版作成)

第 29 章 オープン・ソース・ソフトウェア

(2014 年 (平成 26 年) 6 月 12 日 一部修正)

