

オープンソース・インパクト

～ オープンソースはソフトウェア世界をどう変えるのか ～

Debian Project 武藤 健志 <kmuto@debian.org>

2001年7月30日

1 オープンソースとは何か

「オープンソース (Open Source)」という言葉は、Netscape のブラウザソフトウェア (Mozilla) のソースコード公開に起源を発する。1998 年の Netscape のこの公開声明の動きを支援し、またこのような流れをより一般に戦略として広めることを目的として、Todd Anderson、Chris Peterson、John “maddog” Hall、Larry Augustin、Sam Ockman、それに Eric Raymond といったメンバーによる会議が同年 2 月 3 日に開催され、キャンペーンの中心として発案された。これが「オープンソース」の誕生日とされる。「フリーソフトウェア」のような危険を感じさせる匂いのないこの言葉は、Tim O’reilly や Eric Raymond、Bruce Perens といった著名人を通して、開発者、ベンダー、メディア、そしてユーザーに実に驚くべき速度で浸透していった。いわゆる「オープンソースムーブメント」のはじまりである。

1.1 オープンソースの定義

では、この「オープンソース」とは一体何であろうか。Bruce Perens らはこれを、次のような「オープンソースの定義」としてまとめている。

1. 自由な再配布を許可する。
2. ソースコードを入手できる。
3. 改変したものの配布を許可する。
4. オリジナルソースコード自体の変更に制限をかけたい場合には、その代わりにパッチの存在を認める。

5. 特定の個人あるいは団体に対して制限をかけない。
6. 使用用途に対して制限をかけない。
7. ライセンスはプログラムと一緒に配布され、再配布時に条項を変更しない。
8. ほかの製品とバンドルした特有のライセンスではない。
9. ほかのソフトウェアのライセンスに干渉しない。
10. GNU GPL、BSD ライセンス、X ライセンス、Artistic ライセンス、MPL などがこの定義に沿ったライセンスとして該当する¹。

簡潔にまとめれば「ソースコードおよびバイナリの利用、変更、再配布を誰に対しても許す」ことがオープンソースであると言える。

多くの商業ソフトウェアでは、複製を禁止したりソースコードを非公開にしているほか、バイナリの変更やリバースエンジニアリングも制限している。このようなソフトウェアはオープンソースソフトウェアに対し、「プロプライエタリ (独占的) ソフトウェア」または「クローズドソースソフトウェア」と呼ばれる²。

なお、条項 10. からわかるように、オープンソースには「オープンソースソフトウェアライセンス」なる 1 つの固有のライセンスがあるわけではない。オープンソースソフトウェアにあるのは「ガイドライン」であり、このガイドラインに沿っているものであればそれを「オープンソース」と呼ぼうというのがこの運動のポイントである。

GNU プロジェクトの「フリーソフトウェア」の考えに親しんできた方なら、この定義にとりたてて目新しさを感じないだろう。実際のところ、GNU プロジェクトが訴える「フリーソフトウェア」と「オープンソースソフトウェア」に大きな違いはなく、GNU プロジェクトのイデオロギーを、ビジネスの分野にも親しみやすいよう薄めたもの³とも言えよう。「オープンソース」には政治的配慮が含まれているのだ。

¹このほか、GNU LGPL、Apache ライセンス、NPL がある。IBM Public ライセンスと Mac OS X ライセンスにはその内容に問題があり、擬似オープンソースとして扱う。

²すべての商業ソフトウェアがプロプライエタリソフトウェアであるとは限らないことに注意されたい。また、商業ソフトウェアでなくても、いわゆる「シェアウェア」や Windows に見られる「フリーソフト」はプロプライエタリソフトウェアである。

³GNU プロジェクトの思想はプロプライエタリソフトウェアと対極にある「Copyleft」である。そしてそのための実装の 1 つが GNU GPL である。また同時に、Copyleft は資本主義社会の著作権に基づくものであり、「共産主義」や「著作権の破壊」はまったくの的外れな批判であることも申し添えておく。

1.2 オープンソースの哲学

さらに歴史をさかのぼれば、オープンソースの考え方、哲学は以前より存在する。その典型的な1つが「科学 (science)」である。科学の世界では「発見」と「正当化」のプロセスがその基盤となり、正当化においてはすべての実験経緯が明らかにされ他人がそのプロセスを再現できなければ認められることはない。インターネットのような基盤技術もこのようなプロセスを経ていく中で、さまざまなレビューを受け、改良され、今日の発展につながってきた。人類が生き延び進化していく上で、先人の発見とその情報の共有が大きな支えとなってきたと言っても過言ではあるまい。

オープンソースの今日の発展の理由を考える上で、インターネットの存在も見逃せないだろう。ほぼすべてのオープンソースソフトウェアの開発プロジェクトは、インターネットを介し、人種や国境を越えた世界中の人々の手の協業 (コラボレーション) によって進められている。プロプライエタリなソフトウェアではその開発者は企業という顔の下に隠れてしまうが、オープンソースソフトウェアでは開発者の姿が見えやすく、一般のユーザーが直接コンタクトを取ることも当たり前のように行われている。オープンソースを作るのは結局は人であるし、それを成長させるのもやはり人なのである⁴。そこには「コンピュータネットワークは人間のふれ合いを喪失させる」といった指摘はまったく当たらない。

オープンソースは古くて新しい考え方であり、人類に新たな進化と自由、そして協調の喜びをもたらし得る概念である。

2 オープンソースとビジネス

では、オープンソースは現在のビジネスシーンをどのように変えるのかについて考察する。

2.1 オープンソースを使うメリット

まず、オープンソースソフトウェアを「使う」側の視点で考えることにしよう。先に挙げた定義から明らかのように、オープンソースソフトウェアは使用、改変、配布の自由が保証されている。これには次のようなメリットがある。

- 目的や利用者に対する制限がないため、ビジネス用途にも安心して利用できる⁵。

⁴使い、参加し、貢献してくれる人を集めることができなければ、そのプロジェクトは消失の時を迎えざるを得ない。

⁵Microsoft のシェアードソースライセンスは一見安心して利用できるように思われる

- 表皮だけでなく、その内部まで自分の用途に合うよう修正、カスタマイズできる。もしソフトウェアベンダーがサポートを停止しても、自力でメンテナンスし続けることができる。また、内部まで明らかになっていることで、バックドアのようなセキュリティ、プライバシー侵害の疑いを排除できる。
- 1つのソフトウェアをライセンス数を気にすることなく、いくつものマシンに複製(インストール)できる。ライセンスビジネスモデルに基づくプロプライエタリソフトウェアと反対に、オープンソースソフトウェアではライセンスに対する対価は求められないため、既存のライセンスコストを大きく削減できる。

さらに、オープンソースソフトウェアにはプロプライエタリソフトウェアをしのぐほど質の高いソフトウェアが多い⁶。オープンソースのOSカーネル「Linux」はその代表例であり、IBM や Hewlett-Packard、Intel や Compaq といった著名なソフトウェア/ハードウェアベンダーから絶大な支持を受けている。GNU C コンパイラコレクションはその優秀な機能により、一般のソフトウェア開発から組み込みシステムまで広く使われている。Apache は Web サーバーのシェアにおいて Microsoft の IIS をはるかに凌駕している。Perl や Ruby はネットワークプログラミング言語としての地位を獲得した。BIND は DNS のデファクトスタンダードである。Samba は Windows NT からの乗り換えを促し、Postfix や Sendmail は安定したメールサーバーとして信頼を得ている。オープンソースに縁がないと思っている方でも、実は利用しているいろいろなサービスの裏でオープンソースソフトウェアが存在していることだろう⁷。

2.2 オープンソースで提供するメリット

続いて、「提供する」側の視点で考えてみよう。特に企業として提供する際に重要なのは、「公開することで何を得るか」という明確な戦略である。この戦略が明確であれば、そのライセンスの種類は自ずと決定するし、ソフトウェアと企業の双方に成功をもたらす得る。

が、オープンソースソフトウェアへのウィルスとなり得る危険を孕んでいる(たとえば、オープンソースソフトウェアにコードを流用すると将来 Microsoft から訴訟を起こされる可能性がある)。

⁶オープンソースソフトウェアの開発者のほとんどは優秀な職業プログラマーであり、仕事の手助けあるいは趣味として開発を行っている。このためにモチベーションと技術が共に高く保たれている。

⁷挙げていったソフトウェアからわかるように、高評価を受けるオープンソースソフトウェアのほとんどはサーバーあるいはインフラストラクチャであり、ユーザーフロントエンドのものは少ない。ユーザーフロントエンドの質の向上は今後のキーワードとなるだろう。

企業がソフトウェアをオープンソースとして公開するメリットとしてはいくつか考えられる。公開することで、多くのユーザーの目に触れ、未知のバグの洗い出しやコードがブラッシュアップされることが期待される。停止してしまったソフトウェアの開発プロジェクトの続行をインターネット上の人々に求めることもできる。オープンソースに理解があるという評判が広まることで、企業に対してユーザーの支持を集めることもできる。しかし、これらは企業が戦略として考えるときに第一義とすべきではない。資本主義社会においては、そのソフトウェアを公開することでどのように企業がそこから営利を得るかが重要な点である。これについては成功と失敗の明暗を分けた企業の例を後述する。

改変物のフィードバック (開発者自身だけでなく社会、公共への還元を含む) を義務づけたい場合には、GNU GPL や GNU LGPL が向いている。GNU GPL は「ウィルス性」⁸があるので、より広く別のライセンスのものとも協調させて広げたい場合には GNU LGPL が向いているだろう⁹。GNU のライセンスはプロプライエタリによる「フリーライダー (free rider)」を排除できるように設計されている。

フィードバックはなくてよいのでとにかく広めたい場合には、Modified BSD¹⁰または X ライセンスを採用すると、プロプライエタリにもアプローチできる。BSD ライセンスや X ライセンスは非常に制約がゆるい¹¹ため、Microsoft をはじめとする多くのソフトウェアベンダーには受けがよい。しかし、プロプライエタリな派生物の乱立を招く恐れは否定できない¹²。

このように、企業がソフトウェアを公開する際には、GNU GPL または GNU LGPL のようなライセンスのほうが適していると考えられる。MPL や NPL も類似のライセンスだが、Netscape や Mozilla に強く依存しているためにほかのソフトウェアに流用するのは少々難があるだろう¹³。

2.3 産業への影響

本節の最後として、オープンソースの広がりによって既存のコンピュータ産業構造にどのような影響を与えるのか (あるいは現在与えているのか) を考察してみよう。

⁸GNU GPL 下のソフトウェアのソースコードを流用したり、そのソフトウェアにリンクすると、ソフトウェア全体が GNU GPL の適用範囲となる。

⁹GNU LGPL ではリンクの場合に対象をライセンス適用範囲としない。

¹⁰旧 BSD ライセンスの「University of California, Berkley で開発されたということ」を広告に明記する」という条項を 1999 年の改訂で削除したもの。

¹¹「コピーライトを残す」というだけである。

¹²X Window System の派生物の乱立はその 1 つであろう。

¹³既存のソフトウェアを改造してプロプライエタリなソフトウェアを作りたいという場合には、そのオリジナルのソフトウェアは BSD や X ライセンスが適用されているものを選ぶのが適切だろう。

ソフトウェア開発ベンダー、特に UNIX 向けのプロプライエタリな製品を作っているところにとってオープンソースは脅威となろう。しかし、車輪の再発明だけで過ごしてきた質の低いベンダーは淘汰されるべきであり、市場社会では当然の帰結と考える。競合するオープンソースソフトウェアに対して技術力、競争力、あるいはプロプライエタリならではの企業秘密の取り込みなどで上回れば、VMWare のように成功をおさめることもできよう。あるいは、オープンソースソフトウェアを改良することでビジネスにつなげる展開もあるだろう。

システムインテグレーションやコンサルティングビジネスのようなソリューションビジネスは、オープンソースの利点を最も生かしやすい分野である。オープンソースソフトウェアであればソフトウェアの初期導入コストは無料同然で、目的に合わせたカスタマイズも自由にできる。日本ではまだそれほど大規模なものは現れていないが、IBM や Compaq、Hewlett-Packard をはじめとする大手ハードウェアベンダーが Linux を推進したことと合わせ、今後この分野でのオープンソースの活用がますます拡大していくものと思われる。

Linux ディストリビューション販売ビジネスやサポートサービスは、その多くが厳しい状況にあり、これからさらに淘汰が進むと思われる。Linux ディストリビューションは自社内で開発するにはその開発コストで破綻しかねないし、オープンソースプロジェクトとして行う場合にはネットワーク上での無料配布をベースとするために直接の利益を生み出せない。さらに、Red Hat などのブランドイメージが定着した市場で残されたパイは少ない。サポートサービスも、「サポートの対価」という概念が定着していない日本ではあまり評価されず、刻々と進化する状況に対応だけの技術力のあるスタッフを用意するのも難しい。LPI のような教育ビジネスは今後広がっていくと思われるが、状況の進化についていけなければすぐに価値のないものになってしまう。

ハードウェアベンダーは、オープンソースの Linux を無視できない存在として認めはじめている。オープンソースソフトウェアで弱い点のあった印刷システムも、プリンタベンダーが参加して協議がはじまるなど、脱 Microsoft 依存は着実に進行している。XFree86 バージョン 4 では、ベンダーがドライバを書きやすくまた企業秘密をプロプライエタリのままにできるように設計されている。

3 オープンソースの光と影

本節では、いくつかのプロジェクトを取り挙げ、なぜ彼らはオープンソースで成功をおさめたのか、あるいは失敗に終わったのかを考察する。

3.1 Cygnus Solutions

オープンソースを売りにしつつ最も商業的に成功している企業の1つが、Cygnus Solutions だろう。Cygnus Solutions は得意とする組み込み分野で Red Hat の利益を牽引するほか、GNU C コンパイラを中心に数多くの質の高い GNU ソフトウェアの原作者あるいはコントリビューターとして自他共に広く認められている¹⁴。

なぜ Cygnus Solutions は商業的に成功しているのだろうか。同社の特徴的な商品である GNU C コンパイラを見てみよう。そこには、自社の技術力への絶対的な自信と、オープンソース、特に GNU GPL 下にあるオープンソースの特徴を最大限に生かしていることが見出される。

まず、Cygnus Solutions では創業者の Micael Tiemann をはじめ、極めて優れたコンパイラの専門家を数多く揃えている。他社が同じ土俵に乗るためには同じくらいの技術力で勝負しなければならない。

次に、GNU GPL の永続性により、Cygnus Solutions はその技術力によって改良した GNU C コンパイラのソースコードを他社に対しても公開することになるが、他社が同様に独自の拡張を行っても Cygnus Solutions はその恩恵を受けられる¹⁵。この結果、他社がパイオニアである Cygnus Solutions に対して優位に立つのは極めて難しい。まさにオープンソースだからこそ、Cygnus Solutions は負けることがないのである。

組み込み市場は今後とも廃れることはないであろうし、その頭脳である特殊チップに対応したコンパイラもやはり廃れることはないだろう。今後とも (Red Hat はともかく) Cygnus Solutions 自体のビジネスモデルは手堅いように思われる。

Cygnus Solutions から得られる教訓は次のとおりである。

- オープンソースソフトウェアを使ってのビジネスは可能である。
- オープンソースの世界で優劣を決めるのは最終的には技術力である。
- オープンソースの世界ではパイオニアの優越性が極めて大きい。

Apache Software Foundation、Sendmail、Aladdin Enterprises といった企業もこの方法で成功をおさめている。

¹⁴1999年11月に Red Hat の傘下となったが、Cygnus Solutions の CEO である Michael Tiemann が現在、Red Hat の CTO の座についていることからわかるように、これは一方的な吸収ではなくお互いを補う合併と言えよう。

¹⁵他社が GNU GPL を無視してソースコードを提供しないことも考えられるが、遅かれ早かれこれが露呈したときにはオープンソース支持者から激しい非難を浴び、社会的な制裁を受けることになるだろう。

3.2 Netscape

かつてそのブラウザで一世を風靡した Netscape は今や AOL に取り込まれて見る影もない。オープンソースの間接的な始祖と言える存在であり、次期ブラウザ (Mozilla) のソースコードのオープンソース化によって Microsoft に対して一発逆転を図ったのにも関わらず、なぜ彼らはこのように失速してしまったのだろうか。

いくつかの理由は考えられよう¹⁶。その中でも著者は次の点において失敗があったと考える。

- 公開されたソースコードはコンパイルすらまともに通らないほどの代物 (ガラクタ) であった。このため、ユーザーの多くは早々に見捨ててしまった。
- すでにブラウザはいくつも存在しており、あえて Mozilla を改良しなければならないモチベーションはユーザーになかった。
- Netscape が公開によってどのように利益を得るのかの構造が不明瞭だった。もともと Netscape のブラウザは実質無料で配布されており、利益は Web サーバービジネスに頼っていた。ブラウザをオープンソースにしても、Web サーバービジネスが広がることにはつながらない。

ここから次の教訓が得られる。

- オープンソースにしたことで成功するとは限らない。
- 最初に動くものをリリースせよ。
- 競合するソフトウェアがすでに動作している場合には市場を獲得するのが難しい。
- ソフトウェアをオープンソース化することで自身にどのような利益が予想されるかをよく検討する。

さて、このように惨々な状況であった Mozilla ソースコードだが、現在ではインターネット上の多くのハッカーによって改良が進み、優れたブラウザへと変貌した。Mozilla は Netscape を救うことはできなかったが、新たな生命を吹き込まれ、確実に成長を遂げている。ここにも教訓がある。

- オープンソースは死ぬことがない。

¹⁶Netscape の失敗の分析については、「ハロウィーン文書 I」に詳しい。

3.3 Debian Project

個人ベースのオープンソースのプロジェクトでも、結局注目を浴びないままにフェードアウトしていったものは数多くある。特に次の点には注意が必要である。

- プロジェクトが立ち上がったからといって常に人が集まるとは限らない。
- ユーザー全員が開発意欲を持っているわけではない。

このような中で、完全なオープンソース OS を提供するために活動している Debian Project は比較的的成功をおさめているプロジェクトだろう。オープンソースプロジェクトとしては有数の規模の 700 人以上の開発者を世界中に抱え、さまざまな分野においてその技術力を認められ、泣き所と言われ続けていたサポートも Hewlett-Packard が自社のハードウェアにおけるサポート体制を表明するなど、Debian Project への評価は非常に高く揺るぎないものとなっている。

前述の問題に関しては、初期の段階で Ian Murdock や Bruce Perens らが優れた基本ソフトウェアおよびプロジェクトのデザインを作り上げたことで、たくさんの優秀な開発者を参加させることに成功している。なお、前述の「オープンソースの定義」は、「Debian フリーソフトウェアガイドライン」が基になっている。プロジェクトの開発者が遵守すべきことを社会規約としてまとめ、これに各開発者が同意することを要求しているのも特徴的だろう。

Debian Project のようなネットワーク上のプロジェクトの観察からわかることはまず、プロジェクトの開発者はそのプロジェクトの「雇用者」ではないことだ。オープンソースの開発者は金銭でなく個々のモチベーション¹⁷によって動くので、高圧的、官僚的な管理手法はなじまない。プロジェクトの運営の先頭に必要とされるのは、「マネージャー」ではなく、プロジェクトを自ら引っばっていく「リーダー」である。

できる限りのオープン化も特徴的である。メーリングリストのアーカイブ、開発版や試験版のありのままの姿、バグ報告とその処理状況、そういったものすべてがインターネット上で公開され、誰でも閲覧したりそこに参加したりできるようになっている。

オープンソースプロジェクトで各開発者のモチベーションを均一に保つのは難しいことであるが、Debian Project では最終的なゴールとして「完全なオープンソース OS の提供」を掲げ、そのための段階として Debian

¹⁷自分が楽をしたい、広めたい、人から尊敬されたい、知識向上、研究、不平不満などさまざまである。

GNU/Linux や Debian GNU/Hurd のリリースという中間目標を用意することで解決を図っている。

Debian Project はよく言われるほど一枚岩というほどではない。たび重なるリリースの遅れに対する不満や、新規メンバー登録の方針での対立など、ぶつかり合いはいくつも発生している。しかしそれでも比較的良好に各開発者の意思疎通ができているのは、人と人が協調して作りあげるといふオープンソースの哲学が各自の意識に染み込んでいるからだと言えるだろう。

Debian Project の考察から得られる教訓を次にまとめておく。

- オープンソースプロジェクトでは、マネージャーではなくリーダーが望まれる。
- オープンソースプロジェクトではできる限り「オープン」にする必要がある。
- プロジェクト運営においては多段階の明確な目標を設定する。
- オープンソースは人と人との協調作業なくしては存続し得ない。

4 オープンソースと未来

プロプライエタリの擁護者はそのビジネスモデルがこれまでのソフトウェア産業の成功を支えてきたと主張するだろう。しかし、オープンソースソフトウェアはほぼ 0 の状態からわずか数年で多くのユーザーの支持を受け、池に投げられた石が生み出す波紋のように、静かに—しかし徐々に大きく広がり、ソフトウェア産業と社会にインパクトを与えている。

オープンソースムーブメントは単なるブームではなく、もはや止まることのない「新たな産業革命」なのである。

我々の目の前にある選択肢は、このムーブメントに参加してオープンソースの自由を享受しさらにムーブメントを発展させていくか、あるいはプロプライエタリの大樹の陰で飼いやられて生きるか、である。賽は投げられた。

参考文献

- [1] Karl Fogel 著, でびあんぐる監訳, 竹内里佳訳, “CVS -バージョン管理システム-”, オーム社, 2000, ISBN4274063720
- [2] Debian Project, “*Debian Social Contract*”, Web, 1997, http://www.debian.org/social_contract.html
- [3] The Open Source Initiative, “*history of the OSI*”, Web, <http://www.opensource.org/docs/history.html>
- [4] Jan Sandred, “*Managing Open Source Projects: A Wiley Tech Brief*”, John Wiley & Sons, 2001, ISBN0471403962
- [5] Microsoft Corporation, “*Microsoft Shared Source*”, Web, 2001, <http://www.microsoft.com/BUSINESS/licensing/sharedsource.asp>
- [6] The Open Source Initiative, “*The Open Source Definition*”, Web, <http://www.opensource.org/docs/definition.html>
- [7] Brian Behlendorf 著, ZDNet 訳, “「オープンかクローズか」ではなく、中庸のアプローチを”, Web (ZDNet), 2001, http://www.zdnet.co.jp/news/0107/26/e_behlendorf.html
- [8] Chris DiBona & Sam Ockman & Mark Stone 編著, 倉骨彰訳, “オープンソースソフトウェア”, オライリー・ジャパン, 1999, ISBN4900900958
- [9] 武藤 健志, “オープンソースプロジェクトな日々 -概論-”, 講演資料, 2000
- [10] 武藤 健志, “オープンソース・ムーブメント -静かなる革命-”, 講演資料, 2001
- [11] Eric S. Raymond 著, 山形浩生訳, “伽藍とバザール”, Web, 1999, <http://cruel.org/freeware/cathedral.html>
- [12] Free Software Foundation 著, 八田真行訳, “さまざまなライセンスとそれらについての解説”, Web, 2001, <http://www.fsf.org/philosophy/free-sw.ja.html>
- [13] Eric S. Raymond 著, 山形浩生訳, “ノウアスフィアの開墾”, Web, 1998, <http://cruel.org/freeware/noosphere.html>
- [14] Eric S. Raymond 著, 山形浩生訳, “ハロウィーン文書”, Web, 1999, <http://www.pobox.org.sg/home/hiyori13/freeware/halloween.html>
- [15] Free Software Foundation 著, 八田真行訳, “フリーソフトウェアって何?”, Web, 2000, <http://www.fsf.org/philosophy/free-sw.ja.html>
- [16] Eric S. Raymond 著, 山形浩生 & 田宮まや訳, “魔法のおなべ”, Web, 1999, <http://cruel.org/freeware/magicpot.html>