

SOFTICの研究会報告書

オープンソース・ソフトウェアの
現状と今後の課題について

この報告書は、ソフトウェア情報センター（SOFTIC）における委員会での検討内容をまとめたものであり、IPA（情報処理振興事業協会）による「電子商取引関連基盤技術開発実証事業」の成果である。

Copyrights (C) 情報処理振興事業協会（IPA）2003

目次

はじめに	1
本報告の構成について	3
<u>オープンソース・ソフトウェアをめぐる現状について</u>	
1部 オープンソース・ソフトウェアの定義	5
2部 オープンソース・ソフトウェアの動向	7
1 オープンソース・ソフトウェアの歴史	7
1.1 フリー・ソフトウェア運動	7
1.1 Linuxとオープンソース	7
2 代表的なオープンソース・ソフトウェア	8
2.1 オペレーティング・システム (OS)	8
2.2 インターネット・サーバ	9
2.3 データベース	10
2.4 デスクトップ統合環境	10
2.5 デスクトップ・ソフトウェア	10
2.6 スクリプト言語	11
2.7 GNU	11
3部 日本のオープンソース・ソフトウェアの動向	12
1 一般的なオープンソースの流通	12
1.1 オープンソースというソフトウェアの利用構造	12
1.2 利用構造の変化	12
1.3 利用構造と情報循環	14
2 日本発のオープンソース・ソフトウェア	14
3 日本でのオープンソースの流通	17
4部 オープンソース・ソフトウェアの活用実態と問題点	18
1 ユーザーによる活用実態と問題点	18
2 ビジネスにおける活用実態と問題点	26
2.1 オープンソース・ビジネスの形態	26
2.2 ビジネスにおける失敗例	29
2.3 個別企業の活用例	30
3 利用形態・ビジネス形態別の活用実態と問題点の整理	43
4 今後の展望	44
<u>オープンソース・ソフトウェアに関する今後の課題</u>	
5部 オープンソースをめぐる視点	45
1 利用者側からのオープンソース・ソフトウェアに対する誤解	45
2 開発スタイルとしてのオープンソース	45
3 オープンソース・ソフトウェアに関する法的枠組み	46

6部	オープンソース・ソフトウェアの導入検討ガイドライン	47
1	一般利用者（一般ユーザー）向け導入検討ガイドライン	47
1.1	コスト	47
1.2	サポート，瑕疵への対応	47
1.3	バージョンアップへの対応	48
1.4	アプリケーション・ソフトウェアの可用性	48
2	ITサービス企業（ビジネス・ユーザ）向け導入検討ガイドライン	49
2.1	コスト	49
2.2	システムの安全性	49
2.3	ソフトウェアの公開	49
2.4	ソフトウェア自体の瑕疵，法的責任への対応	50
2.5	サービス・レベル	50
3	ソフトウェア開発者（開発者）向け導入検討ガイドライン	50
3.1	収益性	50
3.2	新規参入容易性	50
3.3	公開によるメリットとデメリット	51
3.3	組み込み機器における優位性	51
7部	オープンソース・ソフトウェアのライセンス契約	52
1	各種オープンソース・ソフトウェアのライセンス概観	52
1.1	GPL (GNU general public license)	52
1.2	LGPL (GNU lesser general public license)	53
1.3	X ライセンス	55
1.4	BSD ライセンス	55
1.5	MPL (Mozilla public license)	55
1.6	その他のオープンソース・ソフトウェア・ライセンス	55
2	オープンソース・ソフトウェア・ライセンスの比較・類型化	55
3	それぞれの類型に対する注意点について	58
3.1	GPL 類型	58
3.2	MPL 類型	59
3.3	BSD ライセンス類型	60
3.4	使用目的に応じたライセンスの比較	60
4	今後の検討の方向	60
8部	GPL に関する法的問題の整理	61
1	GPL初心者に向けたQ&A	61
2	GPL 法律的検討（論点別）	63
2.1	FSF の思想とGPL 文言解釈の関係	63
2.2	契約と諸法の関係	66
2.3	法律上問題となり得る技術の解説	82
2.4	個別的論点	87
	おわりに	98
	委員会メンバー一覧	99
	付録	
	GNU General Public License	103
	参考サイト	111

はじめに

オープンソース・ソフトウェアは、ソース・コードが公開され誰でも自由に改変することができるソフトウェアである。その代表格ともいえるべきLinuxは市場で有効であると認められ、この数年急激に普及してきている。

しかし、残念ながら、我が国ではオープンソース・ソフトウェアが市場に占める割合も少なく、またソフトウェア開発の現場で積極的に取り組もうとする動きも弱い。LinuxがOS市場において占める割合も数%にすぎず、多くのアプリケーション・ソフトウェアの市場でもオープンソース・ソフトウェアの活用は見られない。

その主たる原因はオープンソース・ソフトウェアの利用にまだに多くの不安が残されていることである。第1に、①システム導入後のサポート、②品質上の問題、などの利用者側の一般的な不安である。第2に、①オープンソース・ソフトウェアは本当にビジネスになるのか、②オープンソースのソフトウェアを事業化することによって自社製のソフトウェア商品までが市場を失うことにならないか、といった開発者側の不安である。第3に、いわゆるGPL問題である。オープンソース・ソフトウェアのライセンスに関しては一定した考え方があるわけではなく、また、それぞれの適用範囲や、各国法との適用関係においても明確でないとの指摘がある。このため、法的に不明確な点が多くオープンソース・ソフトウェアを活用したくとも怖くて手が出せないという不満は、開発者、利用者双方に見られることになる。

仮に、ソフトウェアをめぐる技術的知見や優秀な技術者の力を集結する手段としてオープンソースという形態が優れており、また、そうしたオープンソース・ソフトウェアへの取り組みが我が国では単なる不安や誤解で遅れているのだとすれば、ソフトウェア産業や政府、国民生活全般にとって大きな損失になるといえよう。

そこで、本報告は、オープンソース・ソフトウェアをめぐる不安をいくらかでも払拭できれば有益であろうという目標の下に作成された。まず、問題点の一般的解説をし、GPLの法的問題を取り上げる。すなわち「オープンソース・ソフトウェアをめぐる現状」と「オープンソース・ソフトウェアに関する今後の課題」の大きく2つに分けつつ、①ソフトウェアの利用者の視点、②ソフトウェアの開発者の視点、③法的問題の視点（特にGPL問題）、の3つの視点を整理した。

特に、オープンソース・ソフトウェアはLinux以外にもさまざまな分野で既に活用されつつあること、オープンソース・ソフトウェアを活用したビジネスにもさまざまな形態があること、を網羅的に示すとともに、オープンソース・ソフトウェア本来のメリット/デメリットの整理、を試みた。

また本報告では、法律専門家等の知見を動員し、一般に「分かりにくい」「不明確」との批判が多いオープンソース・ソフトウェアのライセンスにつき、Linuxなどに用いられるGPLに焦点を当ててその問題を整理し、初心者向けのガイドラインの作成、専門家向けの法的分析とガイドを試みた。

具体的には、法曹界、学界、企業、業界関連団体などから専門家を招請し、経済産業省の商務情報政策局を事務局スタッフとしつつ、2003年（平成15年）1月～2月に、ソフトウェア情報センター内に研究会を組織して検討を行い、まとめられたのが今回の報告である。

本報告が、オープンソース・ソフトウェアに関する広範な現状と課題を網羅的に取りまとめたものとして、多くの関係者に読まれ、それに対する誤解や不安を払拭し、我が国におけるオープンソース・ソフトウェアに対する取り組みの活性化に貢献することができれば大変ありがたいことである。

本報告は、当然ながらGPLなどの有権的解釈ではあり得ない。GPLを調査の対象として一般的な知見により、その妥当な解釈を目指したものである。もとより限られた時間内の作業であり、内容の完璧を期し得ない。執筆も分担しており各執筆者間の内容的調整も十分にはできていない。また、各執筆者の論考は、必ずしもその所属する団体等の意見ではない。このような前提をお含みの上、本報告を活用していただければ一同大いに幸いである。

本報告の構成について

オープンソース・ソフトウェアを理解する上で重要なことは、

- オープンソース・ソフトウェアには、Linux以外にもいろいろな種類が既に市場に普及している
- オープンソース・ソフトウェアには、Linuxが使っているGPL以外にもさまざまな種類の契約に基づく「オープンソース」がある

ということである。本報告の前半、「オープンソース・ソフトウェアをめぐる現状について」では、上記2点を隠れテーマに、実態を整理する。

まず最初に「1部 オープンソース・ソフトウェアの定義」(pp. 5-6)を示す。Linuxだけがオープンソースではない。改変後のソフトウェアにも厳密に公開を要求しているわけではない。これに続いて、Linuxも含めて、世の中で流通しているオープンソース・ソフトウェアにはどのようなものがあるのかを、その歴史とともに「2部 オープンソース・ソフトウェアの動向」(pp. 7-11)として整理する。OS以外にも、インターネット対応するためのソフトウェア、例えばWWWサーバ・ソフトや電子メール・ソフトなどを中心に、既にさまざまなソフトウェアが市場に普及している状況を示す。ここでは、オープンソース・ソフトウェアだから品質の良しあしが決まるのではなく、オープンソースというソフトウェアの開発形態が必要な技術者のサポートを集めることができた分野では、結果としてその成果物としてのソフトウェアも市場で普及していることが観察される。

こうした普及状況は、日本でも欧米でも、その傾向に大きな変化はない。しかし「オープンソースという流儀を用いたソフトウェアの開発がどの程度行われているか」「日本のイニシアチブに基づいたオープンソース・ソフトウェアがどの程度あるのか」という構造面に焦点を当てると日本は大きく遅れている。そのため「3部 日本のオープンソー

ス・ソフトウェアの動向」(pp. 12-17)において、我が国でのオープンソース・ソフトウェアの開発コミュニティの状況を整理しつつ、日本発のオープンソース・ソフトウェアがいかに少ないか、国際的なオープンソース・コミュニティとのつながりが日本でいかに低いかを検証する。

次に、実際にオープンソース・ソフトウェアがどのような形で活用されているのかを「4部 オープンソース・ソフトウェアの活用実態と問題点」(pp. 18-44)において整理する。オープンソース・ソフトウェアの活用には、①IT業界がオープンソース・ソフトウェアを活用することでどのようなビジネス・モデルをつくるか、②情報システムのユーザーがエンド・ユーザーとしてオープンソース・ソフトウェアをどう活用するか、という2つの視点がある。本報告ではそれぞれについて、研究会で集めた豊富な事例を統一フォーマットで整理し、網羅的に紹介する。

- オープンソース・ソフトウェアと商用ソフトウェアの境界が判然としないことが、オープンソース・ソフトウェアを活用したビジネス・モデルの設計を難しくしていること
- ユーザーにとっても、その境界の問題がさまざまな法的リスクをもたらしていること

などから、ディストリビューター型に始まる既存のオープンソース・ソフトウェアのビジネスが苦勞している状況をレビューする。その上で、今後のビジネスの形態としては、オープンソース・ソフトウェアを自由に使いこなせるエンド・ユーザーやビジネス・ユーザーを増やすための人材育成の分野とオープンソースを組み込み商品としてパッケージ化したシステム、特に機器への組み込み分野などでの活用の進展が期待されることを概観する。

本報告の後半は「オープンソース・ソフトウェアに関する今後の課題」である。

まず最初に、今後の課題を整理するに当たっての視点を

「5部 オープンソースをめぐる視点」(pp.45-46)で包括的に整理する。ここでのポイントは、①オープンソースを活用する側には、まだまだ不要な誤解が多い、②オープンソースを利用することばかりでなく、オープンソースというスタイルを活用したソフトウェアの開発を促進することこそ、今後の検討課題である、③上記の①にとっても②にとっても、オープンソース・ソフトウェアのライセンス契約の分かりにくさ、不透明さが大きな障害となっている。このため、ライセンス契約問題に対する明確な指針が必要である、の3点である。

これを受け「6部 オープンソース・ソフトウェアの導入検討ガイドライン」(pp.47-51)では、①オープンソース・ソフトウェアをエンド・ユーザーとして利用する者、②オープンソース・ソフトウェアをITビジネスに利用する者、③オープンソース・ソフトウェア自体の開発を進める者、以上3つの立場から、導入に当たって検討すべき諸点を分析する。

いずれの場合も、結論としてはオープンソースが絶対的に優位か不利かという結論は得られない。しかし、それぞれが検討する際に見落としてはいけない視点について、ここも網羅的に整理する。

続いて、GPL問題を中心とするオープンソース・ソフトウェアのライセンス契約に関する問題に触れる。内容的には、オープンソース・ソフトウェアのライセンス契約全体を見渡した「7部 オープンソース・ソフトウェアのライセンス契約」(pp.52-60)と、そのうちGPLに的を絞った「8部 GPLに関する法的問題の整理」(pp.61-97)とに分かれる。

7部では、GPL以外のさまざまなライセンス形態の紹介、それぞれのライセンス形態の相違点を明確化することによって、オープンソース・ソフトウェアのライセンス契約に関する今後の在り方を浮き彫りにする。

これに対し8部では、実用的に最も困難に直面することの多いGPLに的を絞り、初心者に向けたGPLガイドラインを整理するとともに、専門家による法的な分析を詳細に行い、専門家によってもGPL問題が一筋縄には解決されない状況を厳密に解析する。

これにより、1部～4部はオープンソース・ソフトウェア

をめぐる現状の整理、5部～6部はオープンソース・ソフトウェアの導入検討ガイドライン、そして7部～8部はオープンソース・ソフトウェアを活用するための法的なガイドラインの提示を本報告では行う。

1 部

オープンソース・ソフトウェアの定義

オープンソース・ソフトウェアとは、簡単に言えばソース・コードが公開され、誰でも自由に改変することができるソフトウェアである。最近では、Linuxがとくに有名であるが、そのほかにも多種多様なソフトウェアが市場で既に流通している。以下に、どういう要件を満たせばオープンソース・ソフトウェアと呼ばれるのか、オープンソース・ソフトウェアを推進するOSI (Open Source Initiative) による、オープンソース・ソフトウェアの公式な定義を紹介する。

なお、オープンソース・ソフトウェアの由来、オープンソース・ソフトウェアにどのようなものがあるかについては、2部で触れる。

①自由な再頒布

ライセンス契約は、幾つかのプログラムの集合の一部として販売・無償譲渡することを制限してはならない。またロイヤルティー、あるいはその他の料金を要求してはならない。

②ソース・コード

プログラムはソース・コードを含まねばならず、オブジェクト・コードに加えてソース・コードによる頒布を許さねばならない。ソース・コードが頒布されない場合は、手ごろな料金でソース・コードを入手できる手段（無料でインターネットによりダウンロードできるのが望ましい）を用意しなければならない。ソース・コードを故意に分かりにくくしてはならない。プリプロセサなどの中間出力も許されない。

③派生物

ライセンス契約は、改変、派生物の作成とこれらの同じ条件での頒布を許さなければならない。

④開発者のソース・コードの同一性保持

パッチ・ファイルの配布を許すのであれば、改変したソース・コードの頒布を禁止してもよい。ライセンス契約は改変されたソース・コードから作られたソフトウェアの頒布を明示的に許可しなければならない。派生物が元のソフトウェアと異なる名称または版数を持つように要求することができる。

⑤個人・団体に対する差別禁止

ライセンス契約はいかなる個人や団体に対しても差別してはならない。

⑥使用分野に対する差別禁止

ライセンス契約は特定の分野におけるプログラムの使用を制限してはならない。例えば、ビジネス用の使用、バイオ技術研究における使用を制限してはならない。

⑦ライセンスの継承

プログラムに付与された権利は、そのプログラムが再頒布されたすべての者に対して、さらにライセンス契約を結ぶことなく与えられなければならない。

⑧ライセンスは製品依存であってはならない

プログラムに付与された権利は、そのプログラムが属する特定の頒布方式に依存するものであってはならない。プログラムがその頒布方式から取り出されて使用されたり頒布されたりした場合でも、再頒布を受けた者は、元の頒布

方式により与えられたのと同じ権利を与えられなければならない。

⑨ライセンスは他ソフトを制限してはならない

対象ソフトとともに頒布される他ソフトに制限を加えてはならない。例えば、同じ媒体で頒布されるソフトウェアはすべてオープンソースでなければならないといった制限を付けてはならない。

⑩ライセンスは技術に対して中立でなければならない

いかなるライセンス条項も特定の技術やインタフェース形態に依存してはならない。

オープンソース・ソフトウェアという呼称が生れるまでの経緯については2部で述べるが、オープンソース・ソフトウェアはそれ以前の「フリー・ソフトウェア」^{注1)}に代わる呼称として提唱され、Linuxなどの普及に伴って、世の中に受け入れられた。従って、オープンソース・ソフトウェアとフリー・ソフトウェアはほぼ同じ意味と考えてよい。

注1) 無償で流通する「フリーウェア」と異なり、この場合の「フリー」は制約からの「自由」を意味する。

2部

オープンソース・ソフトウェアの
動向

1 オープンソース・ソフトウェアの歴史

オープンソース・ソフトウェアであるか否かのメルクマールは、自由に利用でき、自由に改変でき、自由に再頒布できるかどうかにある。その理由を知るためには、オープンソースをめぐる議論の背景と経緯を理解する必要がある。

1.1 フリー・ソフトウェア運動

「オープンソース・ソフトウェア」という用語が登場したのは1990年代後半であるが、ソース・コードの公開や自由な流通は「フリー・ソフトウェア」として以前からごく一般的に行われてきた。フリー・ソフトウェアの考え方は1983年にRichard M. Stallman氏が発表した「フリー・ソフトウェア宣言」にさかのぼる。その目標は、あらゆるソフトウェアを誰もが使える「自由なソフトウェア」として開発し、「自由でないソフトウェア」を使わなくても済むような世界をつくり上げることであった。この運動は「GNUプロジェクト」と呼ばれた。この目標の実現のために、自由なソフトウェアのライセンス「GPL (GNU general public license)」を作成し、「FSF (Free Software Foundation)」を設立した。

かつて、特に研究者の間ではソフトウェアは原則的に公開され、互いに自由に利用できるのが一般的であった。しかし、徐々に商用ベンダーがソフトウェアのソース・コードを非公開にすることが増え、簡単にはほかのソフトウェアを利用できない状況が発生し始めた。実際にStallman氏もソフトウェア研究を進める上でソース・コードの非公開により障害を来すこととなった。そこで同氏はソフトウェアの発展には「ソフトウェアの自由」が不可欠であると思いついた。具体的には、ソフトウェアの利用・再頒布の自

由、ソース・コード入手・改変・再頒布の自由などである。このフリー・ソフトウェア運動は、文化的・社会的な運動でもあり、そのことが求心力となりメール・サーバ「Sendmail」やDNSサーバ「BIND」など、現在でも利用されている数多くの著名なフリー・ソフトウェアを生み出すことにつながった。

特に、インターネットに関連した分野では、当時、商用ソフトウェアが手薄であり、草創期を支えた開発者が共同で手作りしていったこと、インターネットというさまざまな環境への対応を必要とする分野では今でいうオープンソースという開発スタイルが特に有効に機能したことなどから、多くの優れたフリー・ソフトウェアが生み出されるようになった。

1.2 Linuxとオープンソース

1990年代に入りインターネットの普及に伴って、フリー・ソフトウェアの協調的で組織的な開発が急速に広まった。特に注目を集めたのがフィンランドの学生だったLinus Torvalds氏が開発した「Linux」である。従来、OSのような大規模ソフトウェアはごく少数の天才的プログラマーが慎重に組織的に開発しなければならないと信じられてきた。しかし、Linuxの開発は同氏が中心になったとはいえ、細部まできっちり管理・統括しているわけではなかった。しかも、開発途中で頻繁にβ版のリリースを繰り返し、多くの開発者がパッチやバグ報告を同氏に届けるという新しいスタイルで開発が進んでいった。そしてこの開発スタイルがどうやら非常にうまく機能しているという事実に皆が気づき始めた。Eric S. Raymond氏は著書「伽藍とパズール」において、旧来の組織的な開発スタイルを「伽藍モデル」、Linuxのような誰もが自由市場に集まり（ある意味）好き

勝手に、しかし協調しながら進める開発スタイルを「バザールモデル」と名付けた。

1990年代後半にはLinuxのビジネス的成功と米国のITバブルに後押しされ、Linuxに代表されるバザールモデルに注目が集まることとなる。しかし、FSFの掲げる「すべてのソフトウェアは人類の共有財産であるべきだ」というフリー・ソフトウェアの理念は、本質的にビジネスとは相容れないと考える人も増えてきた。そこで、Raymond氏やBruce Perens氏が集まり「OSI (Open Source Initiative)」を設立し、フリー・ソフトウェアの思想を受け継ぎながら、バザール型のソフトウェア開発モデルに焦点を当てた「オープンソース・ソフトウェア」という用語を定義した。オープンソースはフリー・ソフトウェアとほとんど似通っているが、オープンソースを使う人々は技術的利点を強調する傾向にあり、フリー・ソフトウェアを使う人々は他者支配からの自由や倫理的側面を強調する傾向がある。ただし、フリー・ソフトウェアの中心的理念である「コピーレフト」はオープンソースよりも制約が強い。それはGPLが定義する「フリー・ソフトウェアの派生物もまたフリー・ソフトウェアである」という「伝搬性」である。伝搬性がなければ、オープンソース・ソフトウェアを利用したクローズドな商用ソフトウェアの販売などさまざまなビジネスがやりやすくなる。ただし、Linux自体はGPLであり、フリー・ソフトウェアであるからこそ、これだけ広まり成長することができたという事実を見逃してはならない。

2 代表的なオープンソース・ソフトウェア

オープンソース・ソフトウェアといえば、Linuxが有名であるが、実際にはさまざまな分野で既にオープンソース・ソフトウェアが広がっている。オープンソース・ソフトウェア開発の中心的ホスティング・サイト「SourceForge」には6万を超えるプロジェクトが登録されており、登録ユーザーは60万人を超える。日本語サイトでも約500以上のソフトウェア開発のプロジェクトが登録され、登録ユーザーは約5000人（2002年6月現在）に上る。

Linux以上に注目すべきは、WWWサーバやメール・サ

ーバなど、システムをインターネットに接続するために必要なソフトウェアの分野では、オープンソース・ソフトウェアが既に市場の主流を占めているという事実である。これは、そもそもインターネットの草創期に、それを支えた開発者たちが必要に迫られてこの開発スタイルを採用したことが背景にある。インターネットを共用すべき人たちが持てる知見を持ち寄るという意味でも、インターネットが接続すべきさまざまな環境に適合していくという意味でも、オープンソースという開発スタイルは必要な知見の収集に極めて有効であったと考えられる。その結果、優れたソフトウェアができ、それがまた市場で普及するという好循環をたどったものと推察される。

オープンソースではOSやサーバ・ソフトウェアに注目が集まっているが、デスクトップ環境でも徐々に機能性や完成度が向上し、実用性が高まっている。

以下に、代表的なオープンソース・ソフトウェアを紹介する。

2.1 オペレーティング・システム (OS)

①Linux

1991年、フィンランドの学生だったLinus Torvalds氏がたった1人でLinuxの開発を始めたことは有名である。その後、多数の開発者が加わり、Red Hat社やSuSE社などのディストリビューターが数多く登場するに至って、ビジネスでも利用されるようになった。さらに、最近ではIBM社やHP社、日本では富士通、NECなどの大手ベンダーがこぞってLinuxサポートを打ち出した。UNIXやWindowsの安価な代替品としてだけでなく、ミッション・クリティカルな用途にすら利用され始める。特にWWWサーバOSとしてのシェアは、Windowsが49.6%、Linuxが29.6%、BSDsが6.1%（Netcraft社調べ、2002年6月時点）であり、既に巨大市場を有している。一方でデスクトップOSとしてはわずか1.7%のシェアしかないが、2002年にデスクトップ環境が急速に整ってきた。Linuxなどのコンシューマー向け製品も出始め、徐々に普及の兆しが見える。

②FreeBSD

FreeBSDは、オープンソースUNIX OSとして最初のリリースが1993年に発表された。UC（カリフォルニア大学）のBerkeley校で開発された、長い歴史を持つUNIXのBSD（Berkeley software distribution）版を、Intel社製マイクロプロセッサを搭載したパソコンで動作するように改良したものである。ほかのBSDとしてはNetBSDとOpenBSDがあるが、FreeBSDが最も普及している。BSD系のPC UNIXは日本でもNECのPC-9801シリーズに移植され、1990年代前半においてはIBM AT互換機でしか動かないLinuxよりも、BSD系の方が日本では普及していた。BSDライセンスは商用利用に対する制限が最も少ないオープンソース・ライセンスの1つであり、FreeBSDを改良してクローズドな商用ソフトウェアを販売することもできる。FreeBSDはApple社の最新OSであるMac OS Xのベースとなったことは有名である。コア開発者であるJordan K. Hubbard氏は現在同社に勤務し、FreeBSDコミュニティと同社の橋渡し役を務めている。

③ Darwin

Darwinは、Apple社のMac OS Xのオープンソース版である。Mac OS XはUNIX OSの1つであるMachとFreeBSDの流れを組む。1999年に同社はコンピュータ・メーカーとしては初めて自社OSのソース・コードを公開した。ただし、公開されているのはMac OSのコアの部分であり、ユーザー・インタフェースなどの周辺部分は除かれている。しかし、最近、Mac OS X上で動作するX Windows Systemが公開され、もう1つのFree PC UNIXの期待が高まっている。

2.2 インターネット・サーバ

① WWWサーバ「Apache」

Apacheは現在世界で最もシェアの高いWWWサーバである。Netcraft社によると2002年9月時点で、世界のWWWサーバの66.04%がApacheである。オリジナルはNCSA（National Center for Supercomputing Applications）が開発したフリー・ソフトウェアのWWWサーバである。1990年代前半にはNCSA版が広く利用され、多く

の改良パッチがユーザーから寄せられていた。しかし、1995年に主要な開発者がNetscape社に移籍後、プロフェッショナル・ユーザーが不満を持ち、Apacheとして開発を継続することになった。1999年にApache Software Foundationが設立され、オープンソース・ソフトウェアの代表的成功例として、現在に至っている。

② DNSサーバ「BIND」

DNSサーバとは、IPアドレスとドメイン名/ホスト名の対応関係を与えるインターネットにおける最重要の基幹サーバである。オープンソースのDNSサーバであるBINDは、現在ほぼすべてのUNIXに含まれている。1984年に最初のリリースがあり、1993年よりISC（Internet Software Consortium）が管理している。ISCにはSun Microsystems社、HP社、Compaq社、IBM社、SGI社などの主要コンピュータ・ベンダーが参加している。2000年の調べでは、BINDのDNSサーバのシェアは95%に及ぶ。

③ メール・サーバ「Sendmail」

電子メールを中継するメール・サーバとして、長期間高いシェアを誇るのがオープンソースのメール・サーバのSendmailである。2001年の調べでは、世界シェアが42%である。Sendmailは1981年にUC Berkeley校のEric Allman氏が開発した。同氏は現在も主要開発者であり、1997年にSendmail社を設立した。同社はSendmailをオープンソースとして提供するとともに、管理ツールなどを含む商用版を販売している。

④ アプリケーション・サーバ「Zope」

Zopeは、もともとプロプライタリ・ソフトウェアとして開発された。1995年にRob Page氏とPaul Everitt氏が設立した企業は、WebアプリケーションのプラットフォームであるPrincipiaを販売しようとした。しかし、Webアプリケーション・サーバ市場は既にIBM社のWebsphereやBEA社のWeblogicに席巻され、新参企業に入るすき間がなかった。そこで1998年にライセンス販売モデルからオープンソース・ビジネス・モデルに切り替えることにした。

PrincipiaをZopeの名でオープンソースとして提供し、ユーザー・コミュニティの形成に力を注いだ。その結果、特にコンテンツ管理やポータルサイト管理のソフトウェアとして、市販ソフトウェアと競合するまでに成長した。現在、社名もZope社に変更し、Zope関連サポートを提供している。欧州では数社がZope関連サービスを提供するEurozope協会を設立し、事業を展開している。

⑤ファイル共有サーバ「Samba」

Sambaは、UNIXプラットフォームにおけるWindows向けファイル・サーバとプリント・サーバとして、1993年にAndrew Tridgell氏によって開発された。その後SambaはWindowsの多様なプロトコルに対応し、ほとんどのLinuxディストリビューターで採用されている。現在開発はSamba Teamへ移行している。Apacheとともにインストール数の最も多いサーバ・ソフトの1つである。

2.3 データベース

①データベース「MySQL」

MySQLは、スウェーデンのTcX dataconsult AB社（後のMySQLAB社）が開発したリレーショナル・データベースである。1994年に最初のリリースがあり、2000年にGPLとしてオープンソース化されている。GPLだけでなく商用利用できるライセンスも提供している。Yahoo!社やMotorola社などの巨大企業もMySQLを採用している。

②データベース「PostgreSQL」

PostgreSQLは、オブジェクト指向リレーショナル・データベースである。その起源はUC Berkeley校にあり、1996年にオープンソースのSQLデータベースを開発しようとしたことに始まる。当初は学内利用をメインに考えており、ドキュメントの不足やサポートの悪さのためあまり広まらなかった。しかし、2001年にGreat Bridge社にコア開発メンバー6人のうち3人が入社し、品質やサポートも充実してきた。現在は、PostgreSQL社など多くの企業がサポートしている。日本ではSRAがPostgreSQLに注力している。

2.4 デスクトップ統合環境

①デスクトップ統合環境「GNOME」

GNOMEは、KDEと並ぶオープンソースのメジャー・デスクトップ統合環境である。1999年にGNOME 1.0がリリースされた。Eazel社がGNOMEの開発を支援したが、2001年に断念する。しかし、チーフ開発者のMiguel de Icaza氏が率いるXimian社がビジネス・ユーザーをターゲットに開発を続けた。現在はSun Microsystems社、IBM社、HP社なども積極的に支援している。

②デスクトップ統合環境「KDE」

KDE（K desktop environment）プロジェクトは、1996年に始まった。その目標は、UNIXにおける使いやすいGUIの統合環境を実現することである。KOfficeなどのオフィス・ツールも含み、Windowsとデスクトップ市場で競合する可能性を秘めている。GNOMEがすべてGPLで提供されるのに対し、KDEはライブラリをLGPLで提供することにより、KDE上の商用アプリケーションを許容し促進している。本拠地ドイツをはじめ、欧州ではKDEの普及率が高い。ドイツ政府もKDEを積極的に支援している。

2.5 デスクトップ・ソフトウェア

①ブラウザ「Mozilla」

Mozillaは、WWWブラウザのNetscapeのオープンソース版である。NetscapeのシェアがInternet Exploreに押されて低下する中、1997年、Netscape社からブラウザのソース・コードが公開された。しかし、同社の強いコントロールがコミュニティに敬遠され、コア・コンポーネントのレイアウト・エンジンを書き直したりするために、新版のリリースは何度も延期された。紆余曲折を経て2002年にMozilla 1.0をリリースしたが、シェアを奪い返すまでには至っていない。また、MozillaをベースにしたNetscapeもリリースされている。

②オフィス・スイート「StarOffice/OpenOffice.org」

StarOfficeは、Sun Microsystems社が開発したオフィ

ス・スイートであり、StarOfficeのオープンソース版がOpenOffice.orgである。Microsoft Officeとの高い互換性で欧州でシェアを伸ばし始めている。デスクトップLinuxのキラー・アプリケーションとして期待され、同社は世界各国で教育機関に対し、2億ライセンス（57億米ドル相当）を寄付した。日本では商標の関係からStarSuiteと呼ばれている。

③グラフィックス・エディタ「GIMP」

GIMPは、イメージ・ファイルを編集するグラフィックス・ソフトウェアである。高価であるAdobe Systems社のPhotoshopの代替品と位置付けられている。

2.6 スクリプト言語

UNIX環境においては、スクリプト言語は日常的なルーチンワークから、本格的なシステム開発まで多種多様な目的で利用される。最近では、特にWebアプリケーションの実装言語として注目されている。

①Perl

現在、最もメジャーなスクリプト言語はPerlである。Perlはテキスト・ファイルを検索、操作、整形印刷する目的で、1987年にLarry Wall氏により開発された。その後、ネットワークやシステム管理ツールとして進化し、現在は100人以上の開発者がいる。WebアプリケーションのCGIプログラミング言語として地位を確立した。CPANと呼ばれる数百の広範な分野のモジュールが整備されている。

②Python

Pythonは、Guido van Rossum氏により1991年にリリースされた、オブジェクト指向のスクリプト言語である。Java言語と完全に統合されたJPythonもある。2001年にはPSF (Python Software Foundation) が設立された。開発者の同氏は、Zope社でPython Labsのリーダーを務めている。

③Ruby

Rubyは、まつもとゆきひろ氏が開発した日本発のオブジ

ェクト指向スクリプト言語である。海外にもユーザーが多数おり、数あるスクリプト言語の中で一定の地位を築いた。Web日記システム「tDiary」などRuby上で開発されたオープンソース・ソフトウェアも数多く存在する。

④Tcl/Tk

Tcl/Tkは、UC Berkeley校のJ. Ousterhout氏が開発したスクリプト言語である。TclはPerlやPythonと同様のスクリプト言語であるが、TkというGUI (graphical user interface) の開発向け言語とセットになっていることが特徴である。

2.7 GNU

GNUとは「GNU is not UNIX」の再帰的略語であり、1983年にMIT (マサチューセッツ工科大学) のRichard M. Stallman氏が始めたフリー・ソフトウェア運動の一環として開発された、一群のソフトウェアである。GNUC/C++ Compiler, GNU Emacs, GNOME, GnuPGなど、カーネルを除くほとんどのUNIXソフトウェアがそろっており、Linux上の基本的なソフトウェアはほとんどがGNUのソフトウェアである。

(2部 比屋根委員)

参考文献

- 1) Free/Libre and Open Source Software : Survey and Study (FLOSS Survey), <http://www.infonomics.nl/FLOSS/report/>, 2002.
- 2) 青山幹雄, 「オープンソースソフトウェアの現状」, 『情報処理』, Vol.43, No.12, 2002年.
- 3) Linux協会監修, 『Linux白書2003』, インプレス, 2002年.
- 4) SourceForge, <http://sourceforge.net/>, <http://sourceforge.jp/> (日本語).

3部

日本のオープンソース・ソフトウェアの動向

「2部 オープンソース・ソフトウェアの動向」(pp. 7-11) で見たように、オープンソース・ソフトウェアにはさまざまな形態のものがある。しかし、一般ユーザーに身近なデスクトップ・コンピュータ領域では、まだあまり普及していない。さらに、ライセンス数を誰かが一括して管理しているわけでもないで、普及の実態が一般ユーザーには分かりにくい。

しかし、WWWサーバの約半数がオープンソース・ソフトウェアで稼働していることが知られている (Netcraft 社の実数調査「Web Server Survey」) ほか、実際には企業の内部で利用されていたり、プロプライエタリ・ソフトウェアの一部でライブラリやプロトコル・スタックなど部分的に利用されていたりするケースもある。さらにネットワーク指向の家電製品などの組み込み機器でも利用されるケースがあり、我が国でも利用者から意識されない形で広く浸透し始めている。

こうした利用面での傾向を見ると、世界と日本での動向に大きい差はないと思われるが、ソフトウェアの流通や利用の構造的な側面が見える。というのも、ソフトウェアの開発に焦点を当ててオープンソースの利用を見ると、欧米での一般的なケースと比較して、日本が大きく遅れているからである。この3部では、こうした欧米と日本との構造的な違いを紹介する。

1 一般的なオープンソースの流通

1.1 オープンソースというソフトウェアの利用構造

オープンソースの本質は新しいソフトウェア技術ではなく、ネットワークと一体化したソフトウェアの開発・品質

保証・流通・マーケティング・利用構造である^{注1)}。

ソフトウェア技術者がオープンソースとして開発・頒布する動機は、意識的/無意識的なものを含めてさまざまなものがある。相互扶助によるコストダウン、ネットワーク上の仲間意識、自己宣伝、政治実現といったものが代表的である^{注2)}。いずれにしてもネットワーク上のコミュニケーションが刺激となってソフトウェアは開発・頒布されオープンソースとなる。従って生産・利用の構図でどのような集団がどのような経路で情報伝達しているのかがソフトウェアの生産・普及・利用構造を決定しているといえよう。

1.2 利用構造の変化

初期のオープンソースの世界では、開発者と利用者はほとんど同一だった^{注3)}。すなわち、利用者は自己の目的にかなうソフトウェアを作成し、相互扶助を目的に小規模のグループ内部で相互に公開し合っていたのである。自分でソフトウェアをコンパイルするなど環境に合わせる作業が可能な者だけが使用できるため、開発者に近いソフトウェア技術を持つ人間だけが利用可能だった。現在ではこのような相互扶助的集団を「コミュニティ」と呼んでいる。明確なコミュニティの発祥は1960年代のMIT AIラボまでさかのぼる。ここでは、複数の端末を接続した大型計算機自体が今日のインターネットのような流通媒体としての役割を果たしていた^{注4)}。1970年代、1980年代にはLANやBBSなどを流通媒体として複数のコミュニティが発展・成長し、最終的には現在の単一のインターネットに内包されていった。

インターネットやGNUツール、Linuxといったものがより一般的になってからは、利用者の構造が拡大し、オープンソースにかかわる集団は複数に分かれていると考えるこ

とができる。各グループの分離が明確になってきたのは、
だいたい1998年以降である。

図1はソフトウェア流通を軸として、オープンソースに
かかわる集団を区分したものである。グループ間の距離が
コミュニケーションの距離を表している。

●一般ユーザー

一般ユーザーは、真にソフトウェアを利用するだけの普
通の人々である。このグループは特に組織化されている
わけではない。

●ユーザー・コミュニティ

ユーザー・コミュニティは、一定の貢献意志を持って
オープンソースに接し、小規模の修正は行うものの、基
本的に開発行為をしない人々の集団である。このグルー
プはネットワーク上での存在感が大きく、集団化傾向が
強い^{注3)}。

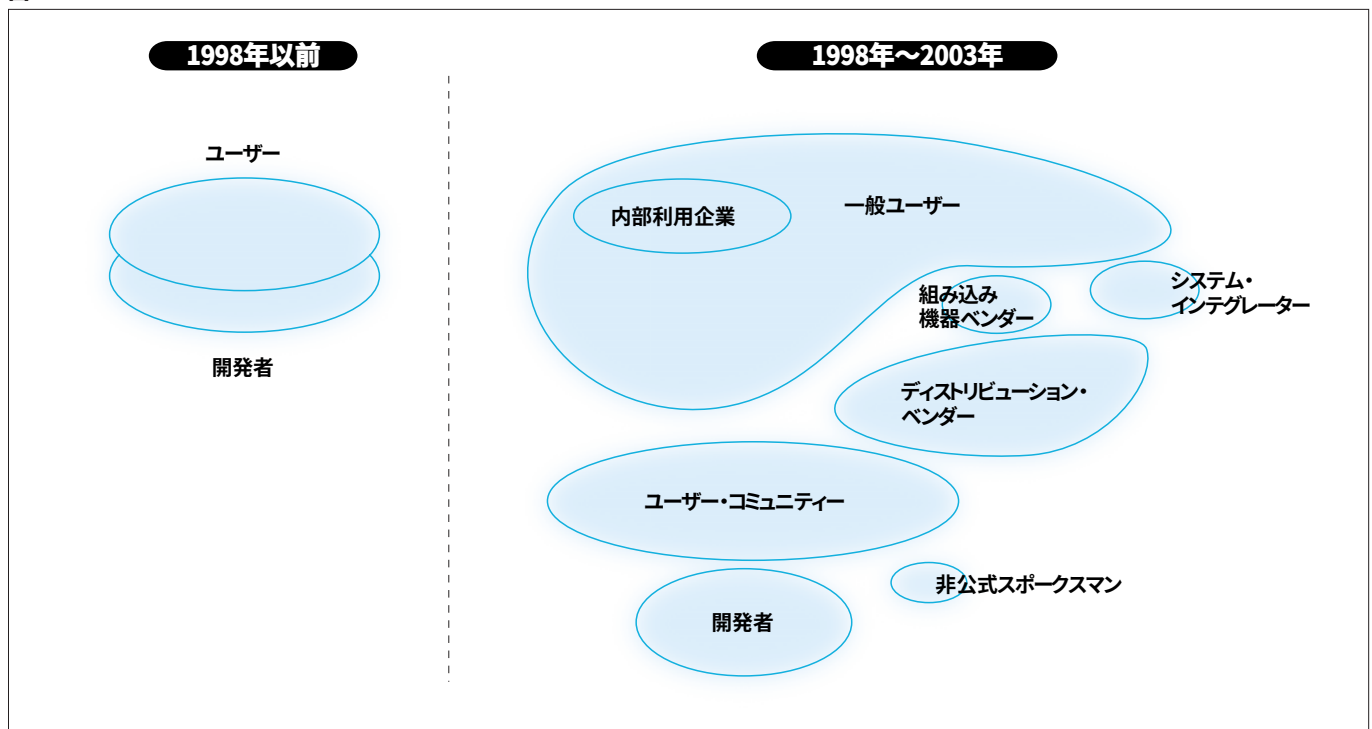
●開発者

開発者は、ユーザー・コミュニティほど集団化傾向も
なく寡黙であるが、ソフトウェア技術を持っており、実
際にソフトウェアを生産している人々である。

●ディストリビューション・ベンダー

ディストリビューション・ベンダーは、ユーザーを対象

図1



注1) 実際にオープンソースに使われるソフトウェア技術は、古典的なものと先進的なものが混在する。基本的には古典的な技術が用いられるが、ネットワーク分野ではアーキテクチャ間の接続性がソフトウェアの存在意義となるため、オープンソースに先進的な技術が注ぎ込まれる。代表的なものはプロトコル・スタック「IPv4/IPv6」やW3C規格に従う「Apache/Mozilla」などである。

注2) 一例としては、Linux Journal誌に掲載された40人のLinuxの初期開発者に対するインタビューを挙げることができる。この「なぜLinuxの開発をしようと思ったのか」という問いに対する回答によると、最も多い理由は「自分で必要な機能を作成した。特に理由がないから配布した」というものだった。

注3) 「オープンソース」という用語そのものが登場したのは1998年で、それ以前はこの用語は存在しなかった。オープンソースという言葉の代わりに使われていたのは「フ

リー・ソフトウェア」という言葉だが、これは広い意味で使うとソース・コードを公開しないソフトウェアも含んでしまい、逆に狭い意味で使うと今度は社会運動としてのニュアンスを含んでしまうという不便な状況だった。多くの人々が真に意図するところは社会資産たる公開ソフトウェアのことだったので、これを指すためにオープンソースという言葉がBruce Perens氏とEric S. Raymond氏らのグループによって生み出された。周知の通りこの言葉は、今日では「フリー・ソフトウェア」という言葉よりもずっと広く使われている。

注4) 井田昌之、「Gnuの誕生まで」、Bit誌草稿、http://www.sipeb.aoyama.ac.jp/~ida/books/gnu_rms.html

注5) プロプライエタリ・ソフトウェアの場合でも、このような性質のグループを「影響力の強いユーザー」ととらえ、マーケティング上のパートナーと見なす場合がある。

としてオープンソースのソフトウェアを編集・開発し、通常は商用のサービスを添付して販売する人々である。一般に法人組織であり、領布に関する責任を負っている。

●非公式スポークスマン

非公式スポークスマンは、オープンソースに関連しない集団に対して「コミュニティからのメッセージ」を発信し、ユーザー・コミュニティや開発者に対して、思想的なリーダーシップをとる政治的な人々である。ユーザー・コミュニティ同様に存在感が大きい。ごく少数の人が独立して行動している。

1.3 利用構造と情報循環

ソフトウェアは下から上のグループへと流通する。当初、開発者グループ内で開発・利用されたソフトウェアは、ユーザー・コミュニティによって吟味される。一般ユーザーは、ユーザー・コミュニティやディストリビューション・ベンダーの情報を基にテスト済みのソフトウェアを入手し、利用する。非公式スポークスマンは対外的なメッセージを発信する。ユーザー・コミュニティは、開発者に対して動作結果や賛辞といったものを伝達する。

このような情報循環が、オープンソースの流通構造となっている。ユーザー・コミュニティと開発者が重なることはもちろんあり、両者をまとめて「コミュニティ」と呼ぶこともあるが、通常、開発者とユーザー・コミュニティは文化的に異質な集団となっている。また開発者は実際に著作権を持ち、オープンソース利用でトラブルがあった場合の最終的な法的義務が彼らに対するものであることから、分けて考えなくてはならない。

また企業利用については、領布に関する責任を直接持つディストリビューション・ベンダーを除くと、多くは一般ユーザーの立場に近い。

2 日本発のオープンソース・ソフトウェア

既にオープンソースを使用している多くのユーザーやソフトウェア技術者は、漠然と日本語のドキュメントの少なさやリーダーシップの欠如を感じているものと思われる。

ここではこのような実感を定量化するために1つの調査を行い、リストを作成した。

表1、表2は、①日本人（または日本の法人）がリーダーシップをとっていること、②一定規模であること、③オープンソース・ソフトウェアであること、④よく使われていること、という基準に従って調査されたソフトウェアのリストである。ソフトウェアの単位としては独立したリソースを一定期間行っている（あるいは行っていた）ものを一単位とした。従って、厳密にはソフトウェアではなく開発単位（プロジェクト）が一単位となっている。

①リーダーシップをとっていること

これは、開発の方針の決定権を半分以上持っている（推測される）ことを意味する。例えば、FreeBSDはオープンソースとしては多くの日本人が貢献していると考えられる。それでも、開発の責任者のうち日本人の割合は12%なので、日本人によるオープンソースと見なすことはできない^{注6)}。

②一定規模であること

これは複数のファイルから構成され、ソース・コード・レベルで5000行以上のソフトウェアを指す。これで簡単なスクリプトのようなソフトウェアや小さいフィルタ・プログラムを除外した。

③オープンソース・ソフトウェアであること

これは、ソース・コードを入手することができ、改変が可能なソフトウェアを指すものとした。UNIX環境用のオンライン配布ソフトウェアは通常ソース・コードが領布され、典型的なライセンスのうちいずれかが添付されるため、オープンソースか否かは明解である。Windowsのフリー・ソフトウェアでは再領布条件が明示されていないことも多々あるが、ソース・コードを公開している場合はオープンソースに含めた。従って本リストの全体は厳密な意味でのオープンソースよりも広範である。特にOSI (Open Source Initiative) によってオープンソースと見なされる、あるいはFSF (Free Software Foundation) によってフリ

表1 UNIX環境用ソフトウェア

○はOSI (Open Source Initiative)によってオープンソースと見なされる,あるいはFSF (Free Software Foundation)によってフリー・ソフトウェアと見なされると考えられるソフトウェア。

国際化/日本語化	○	mule (emacs/xemacs)	マルチスクリプト ^{*1} 化されたLISPベースのテキスト・エディタ
	○	Citrus	XPG4ライブラリのNetBSD上での実装
	○	Qt国際化	ウィジェットの国際化
	○	kon2	マルチスクリプト・コンソール
	○	lv	マルチスクリプト・テキスト・ビューワ
	○	kterm	マルチスクリプト・ターミナル・エミュレータ
	○	ASCL pTeX (日本語化)	組版ソフトウェア
	○	dvipsk (日本語化)	TeX/PostScript変換フィルタ
	○	xdvik (日本語化)	TeXプレビュー
日本語入力	○	Canna	仮名漢字変換
	○	Wnn	仮名漢字変換
	○	kinput2	日本語入力フロントエンド
ネットワーク	○	KAME	BSDでのIPv6実装
	○	Usagi	LinuxでのIPv6実装
	○	delegate	Proxy
	○	PPxP	PPPクライアント
	○	w3m	キャラクタ・ベースのWWWブラウザ
電子メール関連	○	mew	メーラ (MUA)
	○	sylpheed	GUIメーラ
	○	Wanderlust	メーラ (MUA)
	○	fml	メーリング・リスト
言語	○	ruby	オブジェクト指向スクリプト言語
OS関連	○	grub	ブート・ローダ
その他	△ ^{*2}	午後のこ〜だ	高速MP3エンコーダ
		LHa for UNIX	UNIXアーカイバ
	○	timidity	ソフトウェアMIDIプレーヤ
	○	murasaki	ホットプラグ・エージェント
	○	namazu	全文検索エンジン
	○	VFLib2/3	フォント・レンダリング・ライブラリ

^{*1} 「マルチスクリプト」とは、日本語スクリプト (平仮名・片仮名・漢字) だけでなく、 冬の言語のスクリプトに対応しているというソフトウェアの特徴を意味する。「多言語」と呼ばれることもある。

^{*2} 午後のこ〜だは著作権者からはGPLに従って頒布されているが、第三者 (Fraunhofer IIS社) のソフトウェア特許の技術を含んでいる可能性が指摘されているため、日本でも限定的な頒布形態が採られている。

表2 Windows用ソフトウェア

○はOSI (Open Source Initiative)によってオープンソースと見なされる,あるいはFSF (Free Software Foundation)によってフリー・ソフトウェアと見なされると考えられるソフトウェア。

ネットワーク	○	FFFTP	FTPクライアント
		BlackJumboDog	イントラネット用簡易WWW/メール/Proxyサーバ
電子メール (MUA)	○	nPOP	電子メール一覧取得ユーティリティ
		MailDivide ef	携帯用電子メール分割ユーティリティ
	○	BookSync	ブックマーク変換ツール
	○	WWWC	WWW更新チェッカー
言語		ひまわり	スクリプト言語
その他	○	IP Messenger	IMの一種
	○	Galactica proxy	リンク3D化ツール
	○	Namazu for Win 32	検索エンジン
		caldix	DLL自動ダウンロード・ユーティリティ
		窓立て2	ウインドウ立体化ツール
		nLaunch	キーボード・ランチャー

注 6) 「The FreeBSD Developers Core Team」 (http://www.freebsd.org/doc/en_US.ISO8859-1/articles/contributors/staff-committers.html, http://www.freebsd.org/doc/en_US.ISO8859-1/articles/contributors/staff-core.html) によると339人がソース・コミット権を持つ。名前から判断すると、このうち41人が日本人と推測される。参考ま

でLinuxでは、240人のメンテナーがいるが日本人はわずか3人 (1.2%) である。これは、Linuxカーネルのソースに含まれる中心的な開発者リスト (maintainer list) のデータによる (2002年12月11日現在)。

ー・ソフトウェアと見なされると考えられるソフトウェアには、表1に○を付して区別した。

またソフトウェアとしては、フォントや文書・画像といったデータのソフトウェアは含めず、プログラムだけに限定した。

④よく使われていること

これは、頒布物のダウンロード数や実際に使用されているかどうかを判断することは困難であることから、以下の基準を用いた。ここではMacintoshとPDA（PalmOS/Zaurusなど）用ソフトウェアについては省略した。

●UNIX環境のソフトウェア

日本で流通している多くのLinuxディストリビューションやフリーUNIXに収録されている場合、雑誌などに複数回収録される場合によく使われているものと判断した。

●Windows環境のソフトウェア

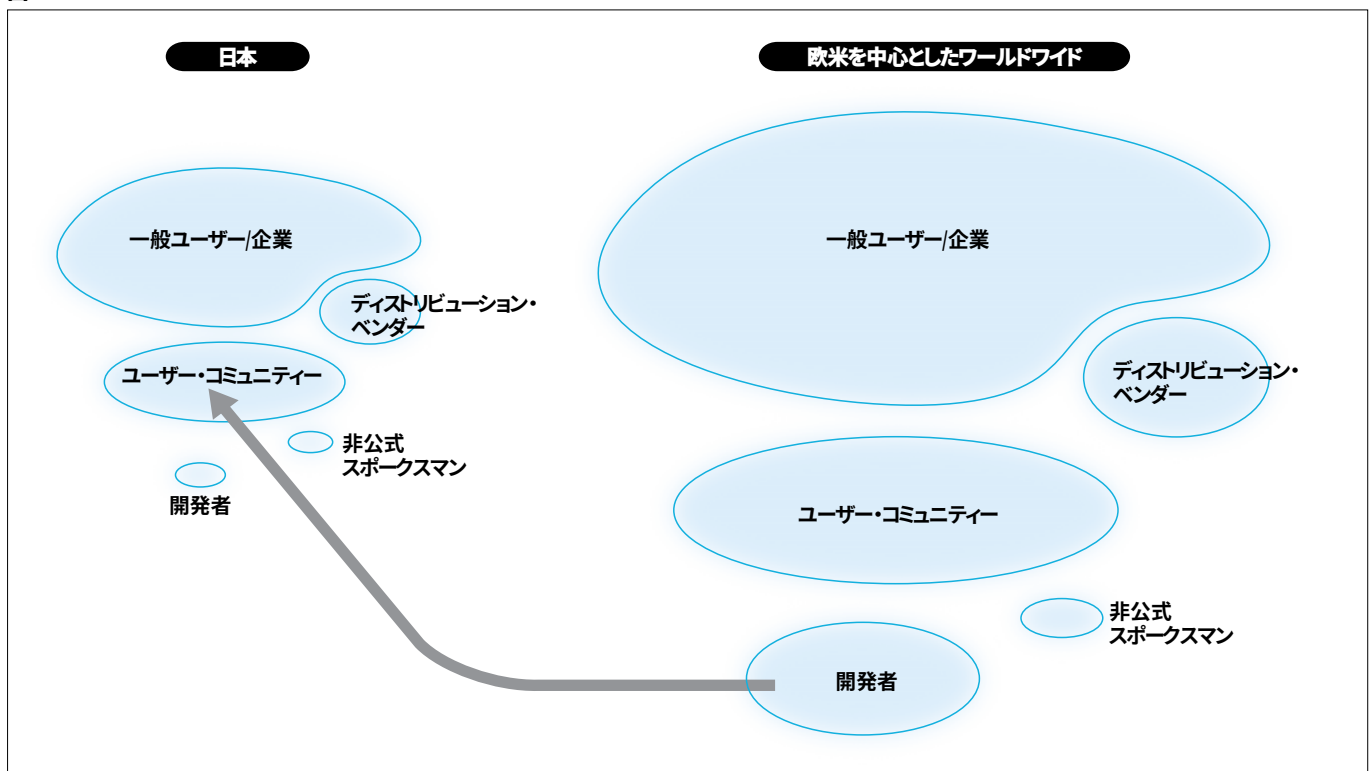
ダウンロード・サイトやインプレスの「窓の杜」に掲載

されているオンライン・ソフトをよく使われている基準とした^{注7)}。オンライン・ソフトはオンラインで頒配布可能なソフトウェアを指す用語で、シェアウェアとフリーウェアが含まれる。647本のオンライン・ソフトが調査対象となった。

他国についての同様の調査・統計はないものの、目安としてLinuxディストリビューション製品が平均的に1000前後のオープンソース・ソフトウェアを含むことを考えると、一先進国としては非常に少ないと考えざるを得ない。

オープンソースの生産の主力を担うのは、典型的には大学生から大学院生までの理学・工学系の学生であるため、コンピュータ・サイエンスの教育総量が少ないことが第1原因なのではないかと思われる。このほかにもインターネット黎明期の回線コストの高さ、西欧中心（英語）的な文化、コミュニケーションにおける時差、人口の差といった環境要因も考えられる。

図2



3 日本でのオープンソースの流通

既に「2 日本発のオープンソース・ソフトウェア」(pp. 14-16) で見たように、日本のオープンソース・ソフトウェアの生産は著しく少なく、多くは日本語処理ソフトウェアを代表として「利用者＝生産者」という古典的なモデルに従って生産・流通している。

そのほかのほとんどのオープンソース・ソフトウェアは、ワールドワイドで成立している生産・普及・利用のモデルを輸入型として流通している。

すなわち海外で生産されたオープンソース・ソフトウェアは、国内ユーザー・コミュニティ、少数の国内開発者のいずれかの経路を経て国内一般ユーザーに流通し、国内からのフィードバック、小規模の修正といったものが海外開発者に返されている。また日本では開発者層が薄いため、非公式スポークスマンのメッセージの正統性が薄く、ミスリーディングが起りやすいという特徴もある。このように、日本のソフトウェア開発コミュニティは今のところ大変脆弱であり、日本を中心としたオープンソース・ソフトウェアの開発コミュニティはなかなか立ち上げにくい状況にある。

「1 一般的なオープンソースの流通」(pp. 12-14) 同様の流通スキームを描くと図2のようになる。

図中の2つの島の距離は、言語や文化の壁であると考えていただきたい。

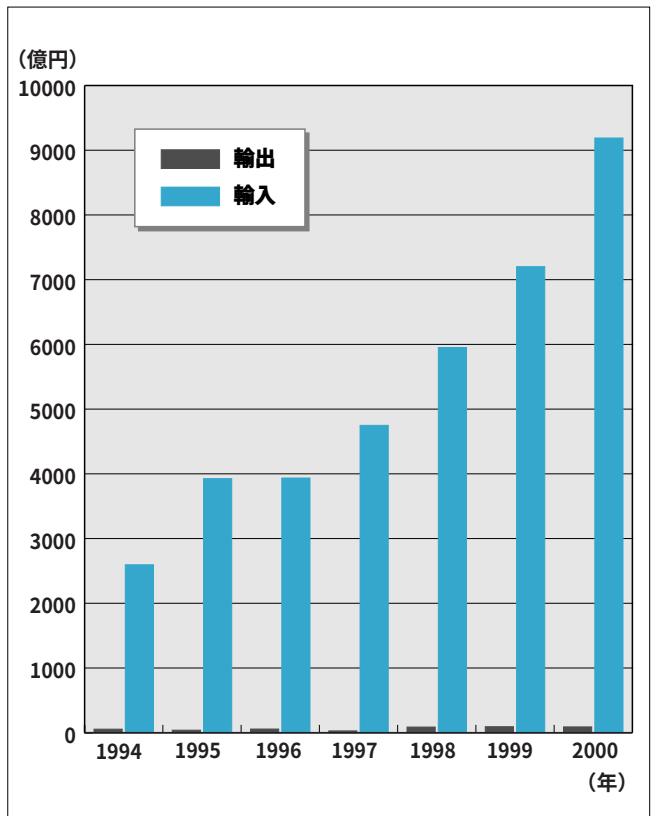
ちなみに従来のプロプライエタリ・ソフトウェアの世界でも、日本はソフトウェア輸入国である。一例として2000年までJEITAによって作成されたソフトウェア輸出入統計を図3に掲載する。

さらに多くの根拠を挙げることは困難だが、輸出入ギャップとオープンソースの内外生産量比は似たような傾向にある。従って日本がことさらオープンソース対応に失敗しているというわけではなく、従来の海外依存構造がそのままオープンソースに持ち込まれていると考えるべきだろう。

また、このスキームに従って日本国内でオープンソース利用のトラブルが発生するケースを想定するならば、まず最初に、国内ユーザー・コミュニティにおいて「ライセ

図3 ソフトウェアの輸出入統計

出典は「JEITA ソフトウェア輸出入統計調査1999年実績」(<http://it.jeita.or.jp/statics/software/1999/>)、「同2000年実績」(<http://it.jeita.or.jp/statics/software/2000/>)。



ンスの違反問題」が喧伝され、次に海外の開発者に通達され、最後に海外の著作者または国内の代理人による訴訟、という形が典型的なステップとなるものと予想される^{注8)}。

オープンソースに関連するニュースは多く、得てして情報に振り回されがちであるが、よりスムーズで確実なオープンソース活用を行うためには、このような日本の地域性を踏まえた上で判断することが必要である。

(3部 大熊委員)

注7) ほかにベクター (<http://www.vector.co.jp/>) の「定番」を対象にすることも可能であろう。

注8) 国内企業については訴訟に至った例はないが、FSFから警告状が出された例がある。

4部

オープンソース・ソフトウェア
の活用実態と問題点

ここでは、研究会での調査を通じて得られた我が国におけるオープンソース・ソフトウェアの代表的な活用事例を紹介する。

1 ユーザーによる活用実態と問題点

ここでは、オープンソース・ソフトウェアがユーザーにどのような形で活用されているか、事例を紹介する。代表的な14の事例を取り上げたが、幾つか共通の傾向を指摘することができる。

- 品質面で大きな問題は発生していない。細かく見ると、

期待以上という評価もあれば、トラブルが生じる場合もある。

- バージョンアップの際に、周辺のソフトウェアとの整合性を含め予想外に苦勞したとの指摘もある。

全体的には、既にある程度リテラシーの高い企業を中心に導入が進んでいることもあり、大きなトラブルは発生していない。中には事例14のように、活用できるオープンソースは全面的に取り入れて、システムのコストダウンにつなげているようなケースもある。

(4部 1 JEITA, 片瀬委員, 古賀委員, 斉藤委員)

事例1 ソフトウェア会社がLinuxの開発環境にgccを活用

1. オープンソース・ソフトウェアの利用形態

①利用しているオープンソース・ソフトウェアの種類

- 名称：gcc (GNU Compiler Collection)
- 機能：各種プログラミング言語用コンパイラ
- ライセンス条件：GPL
- 選択理由：Linux上の標準コンパイラのため。

②具体的な利用形態（事例）

- 使用場所：社内（事業所）
- 使用目的：Linux版製品の開発のため。
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：FTP版（Linuxディストリビューションに標準バンドル、無償）、FTP版（Linuxをダウンロードして入手、無償）

- ◆サポート契約の有無およびサポート内容：なし

2. メリット

- 別段、認識が変わった点はない。
- 無償のため、開発環境作成のコストを節約できる。

3. デメリット

- 別段、認識が変わった点はない。
- トラブルも特になし。

事例2 一般的なユーザー企業がLinuxをシステムのOSとして活用**1. オープンソース・ソフトウェアの利用形態****①利用しているオープンソース・ソフトウェアの種類**

- 名称：Linux
- 機能：OS
- ライセンス条件：GPL
- 選択理由：入手コストの低さ。

②具体的な利用形態（事例）

- 使用場所：社内（事業場）
- 使用目的：社内用小規模サーバ
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし
- オープンソース・ソフトウェアの入手・利用方法：FSFのWWWサイトより入手。

◆有償/無償の別：無償

◆サポート契約の有無、およびサポート内容：なし

2. メリット

- 利用しにくい、安定しないなどと評判を聞いていたが、そのようなことはなかった。
- OSの入手にコストが掛からない。

3. デメリット

- ディストリビューター経由ではないので、利用のノウハウに関して相談する相手がいない（コミュニティーの掲示板はあるが目的の情報を探すのが大変である）。
- 今のところトラブルはない。

事例3 一般的なユーザー企業がLinuxをシステムのOSとして活用**1. オープンソース・ソフトウェアの利用形態****①利用しているオープンソース・ソフトウェアの種類**

- 名称：Linux（主にRed Hat社製）
- 機能：OS
- ライセンス条件：GPL
- 選択理由：コストダウン、安定性。

②具体的な利用形態（事例）

- 使用場所：イントラネット・サーバ、インターネット関連サーバ多数
- 使用目的：Apache, Squid, Sendmailなどのプラットフォームとして。
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし

●オープンソース・ソフトウェアの入手・利用方法：

◆有償/無償の別：Red Hat社の商用パッケージ（7.1, 7.2 Professional, 有償）

◆サポート契約の有無、およびサポート内容：サポート契約は締結せず、初期サポートのみ。

2. メリット

- 安定している。
- パッケージで導入しているため、初期導入コスト削減効果は大きい。

3. デメリット

- バージョンアップ時に、導入方法が問題になることが多い。

事例4 一般ユーザー企業がWWWサーバ・ソフトとしてApacheを活用

1. オープンソース・ソフトウェアの利用形態

①利用しているオープンソース・ソフトウェアの種類

- 名称：Apache
- 機能：WWWサーバ
- ライセンス条件：Apache Software License
- 選択理由：コストダウン、安定性、商用パッケージのフロントエンドとして推奨されているから。

②具体的な利用形態（事例）

- 使用場所：間接業務システム（人事、購買、会計など）の一部、顧客管理システムの一部、イントラネット・サーバなど
- 使用目的：WWWサーバとして、アプリケーション・サーバのフロントエンドとして。

- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：ApacheのWWWサイトからFTPで入手、構築を担当したベンダーから入手。

2. メリット

- 過去に利用していた商用製品より安定している。
- セキュリティ情報やパッチ対応が考えていたより早い。

3. デメリット

――

事例5 一般ユーザー企業がシステムの一部としてTomcatを活用

1. オープンソース・ソフトウェアの利用形態

①利用しているオープンソース・ソフトウェアの種類

- 名称：Tomcat
- 機能：Javaサーブレット・エンジン
- ライセンス条件：Apache Software License
- 選択理由：コストダウン、安定性、商用パッケージのフロントエンドとして推奨されているため。

②具体的な利用形態（事例）

- 使用場所：間接業務システム（人事、購買、会計など）の一部、イントラネットの一部
- 使用目的：Javaサーブレットの稼働環境として。
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：ApacheのWWWサイトからFTPで入手、構築を担当したベンダーから入手。

2. メリット

――

3. デメリット

- Java仮想マシン（JVM）のダウンが頻発、現在は自動リスタートの処理を組み込むことで対応（未解決）。
- JVMのダウンが頻発、アプリケーションのバグ修正で対応。

事例6 一般ユーザー企業がシステムの一部にPHPを活用**1. オープンソース・ソフトウェアの利用形態****①利用しているオープンソース・ソフトウェアの種類**

- 名称：PHP4
- 機能：スクリプト・エンジン
- ライセンス条件：The PHP License
- 選択理由：開発の容易性、LDAPサーバとの親和性。

②具体的な利用形態（事例）

- 使用場所：イントラネット
- 使用目的：Javaサーブレットの稼働環境として。
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：php.netからFTPで入手、Linuxディストリビューションの一部として入手。

2. メリット

――

3. デメリット

- バージョンアップにより言語仕様が若干が変わった結果、アプリケーション・プログラムを修正する必要があった。

事例7 一般ユーザー企業がメール・サーバ用にSendmailを活用**1. オープンソース・ソフトウェアの利用形態****①利用しているオープンソース・ソフトウェアの種類**

- 名称：Sendmail
- 機能：メール・サーバ
- ライセンス条件：Sendmail License
- 選択理由：メール・サーバのデファクト・スタンダード。

②具体的な利用形態（事例）

- 使用場所：インターネット関連サーバ
- 使用目的：電子メールの稼働環境の一部として。
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：sendmail.orgからFTPで入手、Linuxディストリビューションの一部として入手、商用UNIXサーバの一部として入手。

2. メリット

――

3. デメリット

- OS付属のSendmailを利用した場合、セキュリティ・ホールへの対応が遅い場合があった。現在はソース・コードをコンパイルして利用しているケースもある。

事例8 一般ユーザー企業がネーム・サーバ用にBINDを活用

1. オープンソース・ソフトウェアの利用形態

①利用しているオープンソース・ソフトウェアの種類

- 名称：BIND
- 機能：ネーム・サーバ
- ライセンス条件：不明
- 選択理由：メール・サーバのデファクト・スタンダード。

②具体的な利用形態（事例）

- 使用場所：インターネット関連サーバ
- 使用目的：インターネット関連サーバの稼働環境として。
- 使用に当たっての改変の有無，（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：isc.orgからFTPで入手，Linuxディストリビューションの一部として入手，商用UNIXサーバの一部として入手。

2. メリット

――

3. デメリット

- OS付属のBINDを利用した場合，セキュリティ・ホールへの対応が遅い場合があった。現在はソース・コードをコンパイルしたものを利用しているケースもある。

事例9 一般ユーザー企業がメール・マガジン用にfmlを活用

1. オープンソース・ソフトウェアの利用形態

①利用しているオープンソース・ソフトウェアの種類

- 名称：fml
- 機能：メーリング・リスト・サーバ
- ライセンス条件：GPL
- 選択理由：無料で入手可能なメーリング・リスト・サーバのうちで最も構築・運用が容易。

②具体的な利用形態（事例）

- 使用場所：WWWサーバの一部（暫定運用）
- 使用目的：メール・マガジンの発行など。
- 使用に当たっての改変の有無，（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：fml.orgからFTPで入手。

2. メリット

- 導入時に調査した時点では商用のソフトウェアは少なく，高機能で高価（数十万円）なものしかなかった。

3. デメリット

- バージョンアップ当初は，WWWの管理インタフェースが当社のプラットフォームではうまく動作しなかったため，そのインタフェースの利用を断念し，元からあった管理インタフェースを利用した。

事例10 一般ユーザー企業がメール・システム用にqpopperを活用**1. オープンソース・ソフトウェアの利用形態****①利用しているオープンソース・ソフトウェアの種類**

- 名称：qpopper
- 機能：メール・サーバ
- ライセンス条件：Qpopper License
- 選択理由：POPサーバのデファクト・スタンダード。

②具体的な利用形態（事例）

- 使用場所：インターネット関連サーバ
- 使用目的：一部のユーザーの電子メール環境
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし
- オープンソース・ソフトウェアの入手・利用方法：eudora.comからFTPで入手、構築を担当したベンダーから入手。

2. メリット

—

3. デメリット

- 導入当初はダウンロードするデータ・サイズとクライアント環境によってはサーバ・プロセスが暴走することもあったが、バージョンアップで解消された。
- メール・ボックス使用量がQuota上限に達すると、電子メールがダウンロードできなくなった。設定を修正し対応。

事例11 一般ユーザー企業がデータベースとしてPostgreSQLを活用**1. オープンソース・ソフトウェアの利用形態****①利用しているオープンソース・ソフトウェアの種類**

- 名称：PostgreSQL
- 機能：データベース
- ライセンス条件：—
- 選択理由：無償で入手可能なDBMS（データベース管理システム）のうちで国内での導入、実績が最も高かった。

②具体的な利用形態（事例）

- 使用場所：イントラネットの一部
- 使用目的：簡単なWebアプリケーション用のDBMS

- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし
- オープンソース・ソフトウェアの入手・利用方法：postgresql.orgからFTPで入手、Linuxディストリビューションの一部として入手。

2. メリット

—

3. デメリット

- バージョンアップに伴い、バイト数の数え方が変わったため、アプリケーションの修正が必要になった。

事例 12 一般ユーザー企業が検索エンジンとして Namazu を活用

1. オープンソース・ソフトウェアの利用形態

①利用しているオープンソース・ソフトウェアの種類

- 名称：Namazu
- 機能：全文検索エンジン
- ライセンス条件：GPL
- 選択理由：無償で入手可能な全文検索エンジンのうちで最も高性能だった。

②具体的な利用形態（事例）

- 使用場所：イントラネット、WWWサーバの一部
- 使用目的：HTMLやOffice文書などの検索のため。
- 使用に当たっての改変の有無、（および改変者は）改変した事例：なし

- オープンソース・ソフトウェアの入手・利用方法：namazu.orgからFTPで入手。

2. メリット

- イン트라ネットでの利用開始以降、ほかでの利用実績が高くなったため、公開用のWWWサーバでも利用されることとなった。

3. デメリット

- インデックス更新処理が完了しないことがある。根本的な解決はできなかったため、再発時には処理を打ち切る仕組みを組み込むことで対応。

事例 13 一般ユーザー企業が掲示板システムに Phorum を活用

1. オープンソース・ソフトウェアの利用形態

①利用しているオープンソース・ソフトウェアの種類

- 名称：Phorum
- 機能：掲示板システム
- ライセンス条件：The Phorum License
- 選択理由：無償で入手可能なデータベースと連携可能なほぼ唯一の掲示板ソフトだった。

②具体的な利用形態（事例）

- 使用場所：一部の部門サーバ（試行中）
- 使用目的：認証機能付き掲示板
- 使用に当たっての改変の有無、（および改変者は）改変した事例：あり。認証機能、ACLの組み込み、LDAPサーバとの連携機能組み込み、添付ファイル機能の組み込み。

- ◆ 改変者：SIベンダー

- オープンソース・ソフトウェアの入手・利用方法：phorum.orgからFTPで入手。

2. メリット

――

3. デメリット

- ベースとなるPHP、PostgreSQLのバージョンアップに追従するための工数が予想以上にかかった。

事例14 一般ユーザー企業がLinuxをはじめとしたオープンソース・ソフトウェアを大幅に活用し、システムを構築した例

1. オープンソース・ソフトウェアの利用形態

3年ほど前から、情報システム部門が社内インフラにおけるオープンソース・ソフトウェアの利用を推進し、現在、オープンソース・ソフトウェアを利用した16の社内システムが稼働。

①利用しているオープンソース・ソフトウェアの種類

- 名称：Linux, Apache, PHP, PostgreSQL
- 機能：OS (Linux), WWWサーバ (Apache), 開発言語 (PHP), データベース (PostgreSQL)
- ライセンス条件：GPL (Linux), BSD系の独自ライセンス (Apache), BSD系のライセンス (PHP), BSD系 (PostgreSQL)
- 選択理由：いずれもLinuxベースのオープンソースでスタンダードなものであり、特にPHPについてはエンジニアの初期教育が容易であるため。

②具体的な利用形態 (事例)

- 使用場所：全社
- 使用目的：(情報系) 社員情報, 文書共有, 会議室予約など, 8システム / (基幹系) 販売管理, 固定資産管理, 会計など, 8システム
- 使用に当たっての改変の有無, (および改変者は) 改変した事例：なし
 - ◆理由：使用しているオープンソースがミドルウェアのものであり、その上にアプリケーション・システムを開発して使っているだけなので、改変は特に必要ない。オープンソースはソフトウェアの機能追加やバグ修正が頻繁に行われているため、仮に改変による独自の修正を加えた場合、オープンソースに機能追加やバグ修正が行われる都度、独自に行った修正との整合性をチェックする必要が出てくるので、システムの維持が大変になる。
- オープンソース・ソフトウェアの入手・利用方法：雑誌付録などのCD-ROMを利用したり、インターネットから無償で入手したりしている。

- ◆サポート契約の有無, およびサポート内容：なし。ただし、通常の会社ではサポートが提供されている場合、サポート契約を締結してサポートを受ける必要があるだろう。当社においてサポートを特に必要としていない理由は、Linuxに詳しいエンジニアが社内によく、インターネットを通じてサポートに関する情報を得て、自前で解決することができるからである。

2. メリット

- データベースのライセンス料, 保守料が不要である。例えば、有償データベースを利用した場合は、当社(1200人規模)の場合で、使用料が16CPU (サーバ) で3000万円ほど掛かり、年間保守料がさらに650万円ほど掛かる。

3. デメリット

- ソフトウェアのバグなどに関する情報で、該当するものが容易に見つからず苦労したことがある。セキュリティ関係のパッチが多く、システムの維持にコストが掛かる。

4. その他

- 社内システムに利用する場合に必要な技術者の能力, およびバージョンアップのための能力については、オープンソースの上にアプリケーションを開発するに当たり、通常のシステム開発技術以外にオープンソース・ソフトウェアをよく知っている必要は特にない。商用ソフトウェアを使うときに必要となる知識と同等の知識があればよい。バージョンアップは、個別については問題発生ごとに行い、システム全体については2年ぐらいの期間でOSを含めて行うことになると考えられるが、この場合の動作確認作業はかなり負荷が大きくなる。同時に、Linuxに詳しいエンジニアがいなければまず不可能と思われる。なぜなら、単にLinux上で開発を行うことができるだけでなくカーネルのバージョンアップや再構築が必要になったりするからである。

2 ビジネスにおける活用実態と問題点

「1 ユーザーによる活用実態と問題点」(pp. 18-25)では、主にオープンソース・ソフトウェアをユーザーとして利用している場合の現状を見てきた。これに対し、一般に難しいといわれるオープンソース・ソフトウェアを活用したビジネスの状況について、ここでは概観する。

2.1 オープンソース・ビジネスの形態

Linuxの成功は、オープンソース・ソフトウェア関連ビジネスを飛躍的に拡大させた。よくある誤解だが、オープンソース・ソフトウェアは無償であるとは限らない。有償で販売することもGPLによって禁止されているわけではない。しかし、オープンソース・ソフトウェアは、そのソース・コードをさまざまな形で入手することができることから、そのソース・コード自体の頒布だけでは利益率の高いビジネスにはつながらない。このため、信頼性の高いパッケージ群にまとめた上でマニュアルを付けて販売する、オープンソース・ソフトウェアを活用したシステム全体を提案する、といったような何らかの複合的なビジネスにせざるを得ない。

こうしたオープンソース・ソフトウェア関連ビジネスには表1に示すように幾つかのタイプがある。オープンソース関連企業の多くは、同時に複数のタイプに属している。

1990年代後半はディストリビューター型が注目され、ITバブルとともに株価も急騰したが、現在では利益の捻出に苦勞している。

一方、2000年前後には、IBM社やHP社などがLinuxト

ータル・ソリューションを打ち出し、一大ビジネス市場を切り開いた(フラッグシップ型)。オープンソースが注目されLinuxへのニーズが高まるにつれ、現在ではシステム・インテグレーター型やアプリケーション型の企業も着実に業績を伸ばしつつある。同時期にビジネスを拡大した自社のソフトウェアをオープンソース化し、それと密接に関連するその他のサービスで利益を上げるサービス・プロバイダー型は、全般的に苦勞している。

今後については、フラッグシップ型のビジネスが引き続きどこまで成長するか注目されているほか、組み込みシステムに搭載する組み込み型や、需要の高まるオープンソース技術者を養成する教育型のビジネスなどに注目が集まりつつある。

以下では、各ビジネス類型を簡単に概説する。

①ディストリビューター型

Red Hat社やドイツのSuSE社などのLinuxディストリビューターは、Linuxおよび関連ソフトウェアを導入しやすくするパッケージ販売とそのサポート・サービスを事業としている。Red Hat社は年間売上高が7900万米ドル(2002年2月締め)の巨大企業に成長した。欧州では、SuSE社やフランスのMandrake Software社が有名であり、アジアではTurbolinux社が強い。2002年5月にはSuSE社、Turbolinux社、Caldera社、Conectiva社の有力4社が、ビジネス向け統一ディストリビューションのUnitedLinuxでRed Hatに対抗すると発表した。また、日本独自のディストリビューションとして、Oracleデータベースに密着したMIRACLE LINUXや、デスクトップ・ユーザー向けの

表1 オープンソース・ビジネスの形態

ビジネスのタイプ	業務内容	企業など
①ディストリビューター型	パッケージ販売、サポート・サービス	Red Hat社、SuSE社、Turbolinux社、Vineなど
②フラッグシップ型	トータル・ソリューション提供、コア技術開発	IBM社、HP社、富士通、NECなど
③システム・インテグレーター型	受託開発システムにオープンソースを利用	SRA、テンアートニなど多数
④アプリケーション型	キラー・アプリケーションを中心としたサポート、受託開発	Sendmail社、Zope社、SRAなど
⑤プラットフォーム型	特定用途向けLinux PCの販売	Netproject社、デジタルチャイナなど
⑥サービス・プロバイダー型	一般消費者へソフトウェアを提供し、別のサービスで回収	Netscape社、RealNetworks社、Apple社など
⑦組み込み型	組み込み機器にオープンソースを利用	Monta Vista社、Lineo社、シャープ、ソニーなど
⑧教育型	資格認定試験とトレーニング・コース	Leading Edge社、Linux Academyなど

Vine Linuxがある。

②フラッグシップ型

Linuxを世界的に認知させる上で大きな役割を果たしたのがIBM社である。同社は2000年にLinuxに10億米ドルを投資すると宣言し、実際メインフレームまでも含む数々のLinux関連製品を世に送り出した。また、Apacheなどの代表的なオープンソース・ソフトウェアの主要開発者を取り込み、オープンソースでの開発にも大きく寄与している。最近ではHP社もまたLinuxへの取り組みを本格化させ、日本では富士通、NECなどの主要ベンダーもLinuxを主要製品の1つとして提供し始めた。

③システム・インテグレーター型

Linuxに対するニーズが高まるにつれ、企業や政府からの受託システム開発にオープンソース・ソフトウェアを利用するSI（システム・インテグレーション）ベンダーが急増している。「Linux白書2003」によれば、1000万円未満のSI案件におけるLinux採用率は半数を超え、1億円以上の案件でも1/4程度まで利用率が高まっている。SIベンダーの8割以上が今後Linuxへの取り組み件数を増加させたという意向を持っている。

④アプリケーション型

特定のオープンソース・ソフトウェアに注力し、技術力で差別化を図るベンダー企業である。メール・サーバのSendmail社やWWWコンテンツ管理のZope社などは、これらのキラー・アプリケーションの主要開発者が設立した企業で、当該ソフトウェアの保守・サポートのほかに受託システム開発を事業の柱にしている。日本ではPostgreSQLデータベースのSRAがある。

⑤プラットフォーム型

特定用途向けのカスタムLinux PCを販売するハードウェア・ベンダーである。イギリスのNetproject社はセキュリティ機能を強化した安価なカスタム・パソコンを提供し、イギリス警察などに納入している。デジタルチャイナは政

府・企業向けに中国独自プロセッサとLinuxで安価なデスクトップ・パソコンを実現し販売している。Sun Microsystems社もLinuxを搭載した企業向けシン・クライアントを発表した。

⑥サービス・プロバイダー型

自社のソフトウェアをオープンソースとして公開し、サービス事業でコストを回収する企業である。オープンソース・コミュニティを利用し、特にクライアント・パソコンでのインストール数を増やすことを目的にすることが多い。その先駆けはWWWブラウザのソース・コードを公開したNetscape社であり、最近ではストリーミング・ソフトのReal Networks社やOS自体を公開したApple社がこのタイプに含まれる。

⑦組み込み型

安価で多様なチップをサポートする必要のある組み込みソフトウェアの分野では、Linuxの採用が急速に進みつつある。組み込みLinuxディストリビューターとしてMonta Vista社やLineo社がある。シャープはPDAやホーム・サーバにLinuxを採用した。ソニーのハード・ディスク・レコーダ「CoCoon」にもLinuxが搭載されている。今後、情報家電市場とともに成長が期待される分野である。

⑧教育型

Linuxのビジネス化が進むにつれ、Linux技術者養成のトレーニング・コースや技術者認定試験も増えている。Red Hat社は1998年ごろから「Red Hat Certified Engineer (RHCE)」プログラムを実施し、Turbolinux社も「TurboCE」を始めた。1999年には主要Linuxベンダーがスポンサーとなり「Linux Professional Institute (LPI)」を設立し、認定試験を実施している。こうした認定試験を目標にLinux技術者を育成する専門学校や教育コースが増えつつある。

図1

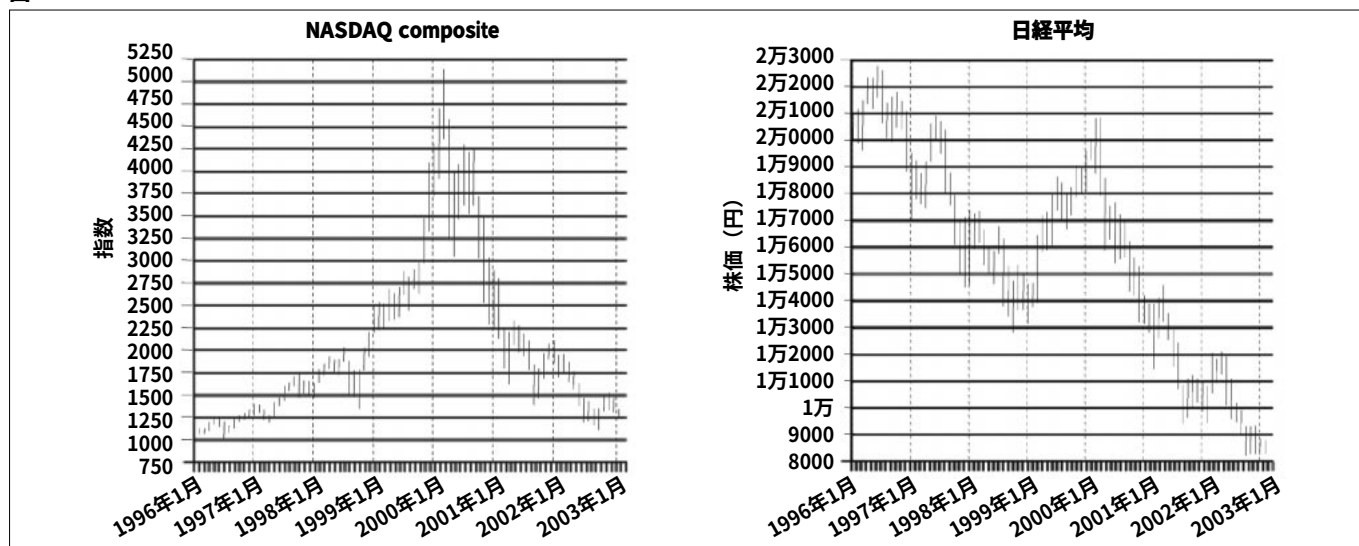


表2 上場企業

(a) Red Hat社

(単位:千米ドル)

	2002年2月期	2001年2月期	2000年2月期
売上高	78937	103429	43428
損益	-140216	-86716	-39925

*Nasdaq,Fundamentals「Annual Income Statement」より

(b) VA Software社(旧VA Linux社)

(単位:千米ドル)

	2002年7月期	2001年7月期	2000年7月期	1999年7月期
売上高	20385	134890	120296	17710
損益	-91038	-525268	-94658	-14512

*Nasdaq,Fundamentals「Annual Income Statement」より

- 1999年12月9日 上場
- 2000年2月3日 Andover.net社を買収
- 2001年6月 サーバ事業から撤退

(c) Caldera社

(単位:千米ドル)

	2002年10月期	2001年10月期	2000年10月期	1999年10月期	1998年10月期
売上高	64241	40441	4274	3050	1057
損益	-24877	-131357	-26923	-9367	7963

*Nasdaq,Fundamentals「Annual Income Statement」,Caldera社「Annual Report」より

(d) Mandrake社

(単位:千米ドル)

	2001年後期	2001年前期	2000年後期	2000年前期	1999年後期
売上高	2331	1851	1705	1279	1717
損益	-3668	-6049	-7546	-4839	-1951

*Mandrake社「Financial Press release」より

- 2003年1月15日 破産申請(現在6カ月の破産保護期間中)

(e) Corel社

(単位:千米ドル)

	2001年11月期	2000年11月期	1999年11月期	1998年11月期
売上高	134320	157487	243051	246827
損益	-11635	-55348	-16716	-30448

*Nasdaq,Fundamentals「Annual Income Statement」より

- 2000年8月17日 Corel社のMichael Cowpland氏がCEOを辞任
- 2001年1月 Linux事業からの撤退をアナウンス

2.2 ビジネスにおける失敗例

「2.1 オープンソース・ビジネスの形態」(pp. 26-27) で見た通り、市場には、当初注目を集めたディストリビューター型から、フラッグシップ型、サービス・プロバイダー型などさまざまなビジネス類型がある。

そのうち少なからぬビジネスは、ネットバブルと呼ばれた市場の波に乗って一時期高い事業資産価値を認められたが、最近では、急速に厳しい状況に置かれつつある。本節では、ここまでオープンソースに取り組んできた企業が現在のどのような状況に置かれているか、概観する。

まず、2003年初頭現在のオープンソースのビジネス界を総括するならば、ネットバブルの清算が完了しつつある時期といえる。あるものは合併・リストラによる合理化を進め、あるものは資金が尽き倒産・整理された。上場・非上

場企業を問わず、ネットバブルから2年～3年を経過し事業の収益性の評価が進んでいる。参考までに図1にネットバブル期前後の株価推移を記す。

表2、表3、表4では公開されている財務資料、ニュース・リリースに従って、オープンソースをうたう企業の事業性について俯瞰する。

ここで取り上げた、経営成績が公開されている企業の全体的な傾向をまとめるならば、以下のようになる。

- 売上額と同規模の赤字額がある。
- 3期以上にわたって（通常創業以来常に）赤字を続けている。
- オープンソースを新技術と表明し発展を正面から引き受けている。

表3 事業を停止した非上場企業

社名	事業の種類	経緯
デジタルファクトリ・Kondara	ディストリビューション・ベンダー	●1999年12月1日 Kondara MNU/Linux 1.0発売 ●2000年8月 米国での販売開始 ●2002年6月28日 デジタルファクトリLinux事業を終了 ●2002年7月1日 Kondaraプロジェクト解散
Eazel 社	デスクトップ・パーツの開発会社	●2000年2月 設立 ●2001年5月 解散
SourceXchange社	オープン・プロジェクト開発仲介業	●1999年 O'Reilly社、HP社の資金支援によって設立 ●2001年4月 事業停止
Cosource.com	オープン・プロジェクト開発仲介業	●1999年 開始 ●停止日時は不明だが現在は少なくとも停止
Great Bridge社	PostgreSQLの開発者らによるRDBサポート会社	●2001年9月 事業停止
Stormix Technologies社	ディストリビューション・ベンダー	●2000年3月9日 日本で英語版の発売を開始 ●2001年2月 営業停止
Loki社	Linuxゲーム移植ベンダー	●1999年5月19日 数タイトルの販売開始 ●2000年3月10日 3Dオーディオ・ライブラリを公開 ●2002年1月31日 営業停止

表4 その他の非上場企業

(a) ターボリナックスジャパン

(単位:億円)

	平成13年12月期	平成12年12月期
売上高	7.54	4.79
純利益	-5.07	-9.6

(b) Progeny社(Debianを推すソフトウェア会社)

●2001年10月15日 distro開発中止のアナウンス

* SRAのプレス・リリース「米国Turbolinux, Inc.の営業の一部譲受けおよびターボリナックスジャパン株式会社の株式の取得(子会社化)に関するお知らせ」
(<http://www.sra.co.jp/public/sra/ir/others/20020820.pdf>) より抜粋

これらの事業はNPO的な風土を持つ集団が株式会社の形態を取って、オープンソースのソフトウェアを生産する上でリスクを取り、オープンソースを新技術であるとして投資家から資金を得て、資金が尽きるまで運営を続ける、といったスキームになっており、企業経営としては全く健全な状態ではない。

ここで取り上げた企業は選択したものではない。オープンソースと正面から取り組む事業はリスクが非常に高く、現在まで企業による「オープンソース事業」の成功例は知られていないのが実状である^{注1)}。

このような事業としての失敗の原因は、ソフトウェア・コード・人材・ノウハウといった企業リソースの流失のしやすさにある。オープンソース自体は先に述べたように新しいソフトウェア技術ではなく、単なる開発・流通・利用の形態であるため、技術の差異化要因となりにくい。

オープンソースのソフトウェアを生産し社会に貢献することはたたえられるべき行動ではあるが、ユーザー以外にも多くのステークホルダーを持つ企業人にとってオープンソース振興は貢献の1つと割り切るべきだろう。オープンソースに取り組む企業は、他企業にタダ乗り（freeride）を許せる範囲で無理なく活用することが賢明である。

本節の筆者は一業界人でもあり、業界事情を説明することには心苦しさは伴うのだが、日本のソフトウェア産業がネットワーク時代に対応して健全に発展していくためにも過去の失敗は広く踏まえられべきであると考え、ここではあえて失敗例として紹介させていただいた次第である。

(4部 2.2 大熊委員)

2.3 個別企業の活用例

ここでは、オープンソース・ソフトウェアを活用したビジネス展開の具体的な事例を個々具体的に紹介する。分類は「2.1 オープンソース・ビジネスの形態」（pp.26-27）で紹介したビジネスの分類に従っている。その全体的な評価は「3 利用形態・ビジネス形態別の活用実態と問題点の整理」（pp. 43-44）でまとめる。

(4部 2.3 JEITA, 大熊委員, 古賀委員,

斉藤委員, 中島委員)

注1) 確かに局所的な成績の好調をうたう発表資料は散見されるが、中期的以上にわたる収益性について具体的な財務データを出した企業はないと思われる。

事例15 <ビジネスにおける活用実態>ディストリビューター型（その1）**1. オープンソース・ソフトウェアの利用形態**

ディストリビューター型（「ARMA 2.1 aka Omoikane GNU/Linux」の例）

①利用しているオープンソース・ソフトウェアの種類

- 名称：Linux/gccほか
- 機能：OS
- ライセンス条件：領布物なので多様
- 選択理由：一般的なものを編集。

②具体的な利用形態（事例）

- 使用場所：オモイカネで開発，ソフトバンクコマースなど一般流通で流通
- 使用目的：ユーザーのオープンソース導入の便宜を図るため。
- 使用に当たっての改変の有無，（改変者は）改変した事例：改変する場合がある。

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：
 - ◆有償/無償の別：――
 - ◆サポート契約の有無，およびサポート内容：直接ダウンロードによって入手し，弊社が実体的な責任を持って配布する。

<提供関係>

- ライセンス契約の提示方法：――
- 提供先のユーザーとのサポートの有無など：――

- 追加保証はあるか？：――

- ソース・コード公開の問題：――

◆公開の有無（オープンソース・ソフトウェア別），公開方法：――

◆公開部分との切り分け基準は：――

- その他これまでの状況で特筆すべき点：

本製品は世界的にメジャーなDebian GNU/Linuxという一流派のディストリビューションを日本語環境に適した形でチューンアップしたもので，保守性，日本語セキュリティ情報などを特徴とした製品である。

ライセンス契約は基本的にシュリンク・ラップ形態になる。複数のオープンソース・ソフトウェアと商用ソフトウェアを含むため，商用ソフトウェアのライセンスのみを記し，オープンソース・ソフトウェアについては「個々のソフトウェアのライセンスに従うこと」と簡略化した契約書を添付する。

提供ユーザーに対してはサポート権が添付され，サポートの追加も可能となっている。ソース・コードは現在は製品に直接添付され，FTPサイトなどでの公開は行っていない。

2. メリット

- 特になし。

3. デメリット

- 特になし。

事例16 <ビジネスにおける活用実態>ディストリビューター型（その2）

1. オープンソース・ソフトウェアの利用形態

当社子会社で、TurboLinuxの名称でLinux OSのディストリビューション業務を行っている。

①利用しているオープンソース・ソフトウェアの種類

- 600種類ほどのオープンソースのソフトウェア群から構成されたOSであり、主要なライセンスはGPL, LGPL, BSD, Perl, MIT, MPLである。
- 選択理由：
 - ◆ライセンスが明確で、TurboLinuxに取り込んで頒布しても問題がないもの。
 - ◆頒布形態がLinux/UNIXの規格に沿ったもの。
 - ◆市場シェアが高い（ユーザーが多い）。
 - ◆コミュニティーや開発が活発である（情報が多い、メンテナンスがきちんとして行われている）。
 - ◆将来性のあるプロジェクトである（先のないソフトウェアをユーザーに提供すべきではない）。

②具体的な利用形態（事例）

- 使用に当たっての改変の有無、（および改変者は）改変した事例：Linuxソフトウェア（群）から、OSとして使い勝手が良いように、選択・組み合わせを行うことがディストリビューターの主たる役目であり、必要に応じて機能拡張やセキュリティ・フィクスなども行っている。そのほか、画面回りなど独自のソフトウェアも付加している。

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：各オープンソース・ソフトウェアは、それぞれのオリジナルのWWWサイト、FTPサイトからダウンロードして入手。
- ◆サポート契約の有無、およびサポート内容：ライセンス元とのサポート契約は特になし。コミュニティーでの情報交換、リリースされたソフトウェアの入手を行っている。

<提供関係>

- ライセンス契約の提示方法：個々のソフトウェアのラ

イセンス契約はCD媒体に記録。ディストリビューターのライセンス条件はパッケージ内に提示（CD媒体のパッケージを開封するに当たってディストリビューターのライセンス契約に同意すること、また個々のオープンソース・ソフトウェアの使用も、それぞれのライセンスに従って使用するべきである旨を記載）。

- 提供先のユーザーとのサポートの有無など：
 - ◆製品を購入したユーザーに対して技術サポートを提供している。
 - ◆OEMに対してサポート契約に基づきサポートを提供している。
 - ◆セキュリティ・アップデートなど、更新したソフトウェアについては、製品を購入した顧客に限定せず、一般に公開している。
 - 追加保証はあるか？：それぞれのオープンソースのライセンス条件に従うものとし、独自のプログラムもFSFへ登録してGPLで提供しているため、これも含めて追加保証はしていない。
 - ソース・コード公開の問題：
 - ◆特別な商用ソフトウェア（製品中に含まれる日本語変換ソフト、フォント、バックアップ・ソフトなど）を除き、基本的に改変、複製、頒布自由なソフトウェアを収集し、調整しているため、非公開にしなければならないソフトウェアはない。
 - ◆オープンソースのソフトウェアの改変は、すべてそれぞれのライセンスに従っており、GPLであれば、改変したものをソース・コードを含めて第三者へ提供できる状態にしている。すなわち製品の場合は、製品ボックスにソース・コードもあらかじめ収録し、FTPサイトでもソース・コードを公開している。
- ### 2. メリット
- 顧客側から見れば、同じく自由度が利くUNIXのOSに比べて価格が非常に安く、利用価値の高いソフトウェアが手に入る。
 - パッケージ製品のすべての内容ではないが、オープン

ソース・ソフトウェアのものはFTPサイトに公開してあるため、必要であればそこから無償でダウンロードして利用できる。

3. デメリット

- もともと大半が無償のものを利用したビジネスであること、およびソース・コード公開を義務付けているGPLに基づくライセンスであるということから、ディストリビューションとして販売しているパッケージをそのまま無償でビジネス的に利用できると誤解され、トラブルが生じることがある。

4. その他

- 調達そのものは無償であるが、最適なOS環境を独自

に作成し、市場提供し、それをさらに維持していくためには多大なコストが掛かる。当然、パッケージ製品に当たってはユーザーが使いやすいように有償の商用ソフトウェアも併用している。相当のコストを掛けて改変し、パッケージ化しても、GPLのライセンスが適用されるソフトウェアについてはライセンス自体を有償にすることはできない。再頒布に必要な料金しか取ることができない。従って、ディストリビューターが頒布するどのブランドのLinuxであればどこまで顧客にとって使い勝手が良いとか、サポートはどこまで期待できるかといったことがビジネスの基本であり、ブランドに対する顧客の信頼が重要となる。

事例17 <ビジネスにおける活用実態>ソリューション・ビジネス型（その1）

1. オープンソース・ソフトウェアの利用形態

ソリューション・ビジネス型（受託システム開発においてオープンソース・ソフトウェアを利用）

①利用しているオープンソース・ソフトウェアの種類

- 名称：Linuxおよび以下のオープンソース・ソフトウェア

apache, bash, bind, binutils, cron, dhcp, dhcpcd, dump, e2fsprogs, egcs, ftp, fileutils, findutils, gcc, gdb, glibc, grub, imap, inetd, ipchains, iptables, kernel, ld.so, libc, libelf, lilo, lpr, make, modutils, mount, mt-st, net-tools, nfs, pam, passwd, perl, pop, portmap, pump, pwdb, rmt, rpm, rsh, samba, sendmail, strace, sysvinit, tar, tcl, tcp_wrappers, tcpdump, telnet, tk, util-linux, wu-ftp, xinetd, yp-tools など

- 機能：OS, WWWサーバなど
- ライセンス条件：GPL, LGPL, Apache, BSDなど
- 選択理由：顧客の要望。

②具体的な利用形態（事例）

- 使用場所：顧客の事業所
- 使用目的：Linux上で稼働する顧客データ管理、ソリューションを提供する。
- 使用に当たっての改変の有無、（改変者は）改変した事例：なし

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：顧客が入手する。顧客からの依頼があれば、インストールのサポートも行う。
- ◆有償/無償の別：有償/無償の両方あると思われる。
- ◆サポート契約の有無、およびサポート内容：サポート契約のある場合、ない場合の両方があると思われる。

<提供関係>

Linuxやオープンソース・ソフトウェアに関しては、顧客が自社内で使用する限り、顧客において、以下のような問題を考慮する必要性はさほどないと思われる。一方、Linux上で稼働するソフトウェア（ソリューション）に関しては、顧客への提供の際に公開の必要性の有無をライセンス条件に従って判断することになるが、公開は必要ないと判断できる事案が多いようである。

- ライセンス契約の提示方法：――
- 提供先のユーザーとのサポートの有無など：――
- 追加保証はあるか？：――
- ソース・コード公開の問題：――
 - ◆公開の有無（オープンソース・ソフトウェア別）、公開方法：――
 - ◆公開部分との切り分け基準は：――
- その他これまでの状況で特筆すべき点：――

2. メリット

- ◆利用前と認識が変わった点：Linuxやオープンソース・ソフトウェアを含めてシステム全体についてのサポート体制が整っている場合には、顧客に対して同一のインタフェースでサポートを提供できる。
- ◆コスト面でのメリット：顧客のLinuxやオープンソース・ソフトウェアの入手の仕方によっては、コストが節約できる場合もあると思われる。

3. デメリット

- ◆利用前と認識が変わった点：一般に、Linuxの進化が早いため、ディストリビューター各社のサポート期間が短いという不安を顧客は持ちやすい。
- ◆これまでのトラブル/クレーム事例：特になし。

事例18 <ビジネスにおける活用実態>ソリューション・ビジネス型(その2)**1. オープンソース・ソフトウェアの利用形態**

カスタム・ソフトウェア開発におけるオープンソース・ソフトウェアの利用。このケースでは、顧客のシステム担当者がオープンソースに慣れていたこと、顧客の要望（コスト、開発期間、性能など）に合わせて最適なものを求めたら、自然にオープンソースを使うことになったということが、当該開発にオープンソースを利用した理由である。

①利用しているオープンソース・ソフトウェアの種類

- 名称：FreeBSD, Apache, PostgreSQL
- 機能：OS (FreeBSD), WWWサーバ (Apache), データベース (PostgreSQL)
- ライセンス条件：FreeBSD (FreeBSD), Apache (Apache), PostgreSQLライセンス (PostgreSQL)
- 選択理由：それぞれのオープンソース・ソフトウェアが、それぞれの機能について定番であるため。

②具体的な利用形態(事例)

- 使用場所：顧客社内およびインターネット上
- 使用目的：出版会社の書籍閲覧・購入システム
- 使用に当たっての改変の有無、(改変者は)改変した事例：当該システムでは改変はしていない。ただし、構築するシステムに必要と思われる場合、例えば機能が足りない場合、インタフェースを合わせる場合、性能が出ない場合などには改変を行う必要が出てくることがある。その他アプリケーション開発以外の場合の例としては、コンピュータへの移植作業、組み込み機器への搭載のための機能削減(メモリ使用量を減らすため)などの場合にも改変が必要となる。

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：
 - ◆ 有償/無償の別：無償。FTPサイトからダウンロードする人が多い。ただし顧客への納品条件によっては、顧客の名前でディストリビューターから購入するケースもある。
 - ◆ サポート契約の有無、およびサポート内容：オープンソース提供者とサポート契約は特にしていない。

<提供関係>

- ライセンス契約の提示方法：顧客の使用目的により、契約内容の確認をするようにしている。
- 提供先のユーザーとのサポートの有無など：顧客に対しては、当社が提供しているサポートに入ってもらっているが、必要でなければサポートなしの場合もあり得る。予算、規模、システム要件などによる。
- 追加保証はあるか?：していない。
- ソース・コード公開の問題：このケースでは、顧客社内の利用だけなので公開の問題は発生しなかった。また、オープンソースがFreeBSDの利用であれば、原則として公開義務を負わないと考えられているが、単にFreeBSDを使用しているから、公開義務がないと考えてしまうのは危険である(下記「4. その他」を参照)。
 - ◆ 公開部分との切り分け基準は：今回は、開発したコードにGPLソフトウェアをリンクしていないので公開部分との切り分けの問題は出てこないが、開発したコードがGPLのソフトウェアへリンクする場合、そのリンクのさせ方でGPLの適用が変わることはないと考えている。そこでGPLソフトウェアを使用した場合、公開部分との切り分け基準は以下のように考える。
 - ① 公開が必要となる基準：GPLソフトウェアのソース・コードを製品の一部に使用している、またはGPLソフトウェアのライブラリを製品にリンクしている。
 - ② 公開は必要ないと考えられる規準：GPLソフトウェアのプログラム(実行形式)を起動する/呼び出すように製品が作られている。
- その他これまでの状況で特筆すべき点：一般に、開発するものの一部としてオープンソースのコードを「流用」する場合などには、特にオープンソースのライセンスを意識することになる。そこで開発においては、顧客への納入物件にオープンソースのコードが混入しないようにしている。例えばオープンソースの開発ツ

ール(コンパイラ: GNU compiler, エディタ: Emacs, Vimなど)を利用する場合には, それらのコードが納品物件に混入しないように注意している。

2. メリット

- データベース(DBMS)は確かに無償だが, どのケースでも, 必ずコストが安い, 開発期間が短くて済むとは限らない。
- むしろ商用ソフトウェアを利用する場合は, システム構築において自由度に欠けるのに対し, オープンソースのソフトウェアの場合は, システム構築にそれぞれ最適なソフトウェアの利用が可能である。

3. デメリット

- 商用ソフトウェアの場合, そのパッケージなどで仕様

が明確にされているが, オープンソースの場合, かつては動作環境などを開発者自身が探さなくてはならないという状況があった。ただし, 現在は情報が豊富で問題はなくなっている。

4. その他

- オープンソース・ソフトウェアが頒布されているソフトウェア単位で全く同じライセンスが設定されているとは限らない。一部モジュールについて別のライセンスが設定されていたりすることがしばしばあり, 例えば, FreeBSDの頒布物の中にもGPLのものがたくさんあれば, BSDでもGPLでもないものもあるので注意が必要である。従って, 顧客に納品する製品の使用目的によって個々にライセンスの確認が必要である。

事例19 <ビジネスにおける活用実態>ソリューション・ビジネス型(その3)**1. オープンソース・ソフトウェアの利用形態**

オープンソース・ベースのアプリケーション・パッケージ(当社製品)を利用したシステム開発。

①利用しているオープンソース・ソフトウェアの種類

- 名称：Linux, Apache, ApacheJServ, OpenSSL, mod_ssl, PostgreSQL
- 機能：OS (Linux), WWWサーバ (Apache), Java Servletエンジン (ApacheJServ), SSLエンジン(暗号化) (OpenSSL, mod_ssl), データベース (PostgreSQL)
- ライセンス条件：GPL (Linux), Apache (Apache, ApacheJServ), BSD系 (PostgreSQL)
- 選択理由：アプリケーションのインフラに掛かるコストを抑えるための利用だが、個々のソフトウェアについては、以下の通り。
 - ◆LinuxはWindowsより信頼性/安定性があり、商用UNIXより安価。
 - ◆Apacheは世界で最も使用されているWWWサーバであり、機能も豊富である。
 - ◆ApacheJServは、Apacheで動くJava Servletエンジンとしては当時これが唯一だったため。
 - ◆SSLエンジンはApacheで動作する最も一般的なものである。
 - ◆PostgreSQLは、商用データベースに比べて機能的にさほど遜色なく、無償であるため。

②具体的な利用形態(事例)

- 使用場所：顧客の社内システム、およびインターネット上
- 使用目的：オンライン・ショップ・システム。オンラインから閲覧注文されたものを受注するとともに仕入先へ在庫確認(→出荷)、出荷センターから出荷するシステム。
- 使用に当たっての改変の有無、(改変者は)改変した事例：なし

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：Linuxはベンダーから購入。その他はそれぞれのWWWサイトからダウンロード。
- ◆サポート契約の有無、およびサポート内容：なし

<提供関係>

- ライセンス契約の提示方法：アプリケーション・パッケージ・ソフトを顧客に購入してもらう際に、これらのオープンソース・ソフトウェアも合わせて一式バンドルで提供したが、パッケージ・ソフトのライセンス契約のみ顧客と締結し、オープンソース・ソフトウェアのライセンス条件は提示せず。
- 提供先のユーザーとのサポートの有無など：あり。当社所定のサポートを提供している。
- 追加保証はあるか?：特にしていない。
- ソース・コード公開の問題：インフラはオープンソースで公開できるものだが、アプリケーション・パッケージ・ソフトは第三者ライセンサーよりソース・コード・ライセンスされているものをカスタマイズして使用しているため、非公開。いずれにしても社内システムとしての利用のため、ソース・コード公開の問題は発生していない。

2. メリット

- インフラが無償で、アプリケーション開発費用しか掛からない。

3. デメリット

- セキュリティ・ホールが発見された場合に、世界中の開発者により即座に対応が行われるのはよいが、頻度が多く、その対応のため(重要度にもよるが)本番稼働中のシステムを止めてパッチを当てなければならないという手間が発生する。パッチが次々と出されることは良いことだが、顧客からは、当然開発側が対応してくれると期待されてしまうことがある。

事例20 <ビジネスにおける活用実態>アプリケーション型

1. オープンソース・ソフトウェアの利用形態

アプリケーション型（パッケージ製品「TERMINATOR パーフェクト2.0」の例）

①利用しているオープンソース・ソフトウェアの種類

- 名称：libc, ncurses ライブラリ
- 機能：特定アプリケーションの起動
- ライセンス条件：LGPL/BSD
- 選択理由：DOSレベルのアプリケーション動作に必須な機能を持たせること。

②具体的な利用形態（事例）

- 使用場所：オモイカネが受託開発し、顧客企業であるアルファ・オメガソフト社がパッケージ製品として流通
- 使用目的：DOSレベル・アプリケーションの移植
- 使用に当たっての改変の有無、（改変者は）改変した事例：なし

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：
 - ◆サポート契約の有無、およびサポート内容：直接ダウンロードによって入手。入手時のサポートはない。

<提供関係>

- ライセンス契約の提示方法：――
- 提供先のユーザーとのサポートの有無など：――
- 追加保証はあるか？：――

●ソース・コード公開の問題：

◆公開の有無（オープンソース・ソフトウェア別）、公開方法：――

◆公開部分との切り分け基準は：――

●その他これまでの状況で特筆すべき点：

本製品（TERMINATOR パーフェクト2.0）は、パソコンの廃棄時に使用するハード・ディスク抹消ソフトウェアで、従来DOSの1FDで起動していた単体機能アプリケーションであったものをLinuxに移植したものである。このような形態でのDOSからの単機能商用パッケージの移植事例としては、国内最初の事例と思われる。

契約関係については、ソフトウェア請負開発のため顧客企業が最終的なユーザーに対する責任を持ち、通常の請負開発に関する契約のみを結ぶ。顧客自身がソフトウェア・ベンチャーであることもあり、特に追加要求がない限りはサポートや追加保証は行わない。

ソース・コードは納品時に顧客企業に対して譲渡されるが、LGPL, BSD ライセンスのみを使用し、顧客企業はソース・コードの公開は行わない。

2. メリット

- 特になし。

3. デメリット

- 特になし。

事例 21 <ビジネスにおける活用実態>オープンソース・ソフトウェアをサポートしたサーバの販売型**1. オープンソース・ソフトウェアの利用形態**

- オープンソース・ソフトウェア（Linuxなど）が動作することを確認したサーバの顧客への販売と、当該ソフトウェアのインストール代行サービスの提供
- オープンソース・ソフトウェアに対するサポート・サービス

①利用しているオープンソース・ソフトウェアの種類

- 名称：Linuxなど
- 機能：OSなど
- ライセンス条件：GPL
- 選択理由：多様なプラットフォームの提供、顧客の指定

②具体的な利用形態（事例） *以降は、Linuxをケースとして記述

- 使用場所：顧客の事業場
- 使用目的：業務用サーバOSまたは組み込みOS
- 使用に当たっての改変の有無、（改変者は）改変した事例：あり。自社ハードウェアへの対応（ただしGPLに登録し、自ら販売せずディストリビューターより供給）。

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：ディストリビューターより入手し、ユーザーへ購入代行サービスを提供。
 - ◆有償/無償の別：有償。基本的に実費
 - ◆サポート契約の有無、およびサポート内容：あり。技術サポート契約（ユーザーの購入の意志に任せている）

<提供関係>

- ライセンス契約の提示方法：開封前に読める形態の紙、および電子マニュアルやWWWサイトに記載。
- 提供先のユーザーとのサポートの有無など：あり。Linux導入前サポートとして、コンサルティング（構成相談、インストール作業支援ほか）、問い合わせ（導入・運用に関して）。Linux導入後サポートとして、問

題解決支援、問い合わせ（Linuxの基本的動作や使用方法に関して）、情報の提供（問題の発生を未然に防ぐ情報の提供）、（ディストリビューターからの）改良版の提供サービス。

- 追加保証はあるか？：拡張サービスの枠で個別対応。
- ソース・コード公開の問題：
 - ◆公開の有無（オープンソース・ソフトウェア別）、公開方法：あり。WWWサイトで公開。
 - ◆公開部分との切り分け基準は：GPLに従う部分をすべて公開。
- その他これまでの状況で特筆すべき点：特になし。

2. メリット

- ユーザーにとっては、他のシステムよりも低価格で相応の動作を見込めるシステム構築が可能。
- Linuxに取り組む企業にとっては、ソフトウェア企業として最先端への取り組みを発信できる。
 - ◆利用前と認識が変わった点があれば特に記入：特になし。
 - ◆コスト面でのメリット：ユーザーにとっては、Linuxを無償または安価に利用できる。ただし安定稼働を目指すのであれば相応のテストを経た上でのシステム構築が必要。

3. デメリット

- 改変可能なため、ユーザーにはLinuxで独自仕様を要求できると思われてしまいがちな点。改変を実施した時点でLinuxではなくなってしまう、ベンダー独自のOSとなってしまうので通常は実施しない。特に行う場合は相応の手続き（GPL登録、ディストリビューターへの提供依頼など）が必要。
 - ◆利用前と認識が変わった点があれば特に記入：特になし。
 - ◆これまでのトラブル・クレーム事例：特になし。

事例22 <ビジネスにおける活用実態>組み込み機器型

1. オープンソース・ソフトウェアの利用形態

組み込み機器型

①利用しているオープンソース・ソフトウェアの種類

- 名称：Linux
- 機能：OS
- ライセンス条件：GPL
- 選択理由：モジュールの豊富さ（コスト、開発スピードの短縮）。

②具体的な利用形態（事例）

- 使用場所：購入者の事業場
- 使用目的：業務用機器のOS
- 使用に当たっての改変の有無、（改変者は）改変した事例：開発技術者による改変あり。機器のハードウェアにマッチングさせるためにカーネルを改変した。

<入手関係>

- オープンソース・ソフトウェアの入手・利用方法：ディストリビューターより入手。
- ◆有償/無償の別：基本的には無償。ディストリビューター開発プログラムに関しては有償もあり
- ◆サポート契約の有無、およびサポート内容：あり。ソフトウェアのアップデート、テクニカルサポート、トレーニング。

<提供関係>

- ライセンス契約の提示方法：利用ソフトウェアに関しては、GPLなどのライセンスをその他のドキュメント（説明書）と同梱。
- 提供先のユーザーとのサポートの有無など：通常の製品に対するメーカーとして機器のサポート（ソフトウェア込み）。
- 追加保証はあるか？：通常の製品に対するメーカーとして機器の保証（ソフトウェア込み）。

●ソース・コード公開の問題：

- ◆公開の有無（オープンソース・ソフトウェア別）、公開方法：あり。ソース・コード提供の旨を書面で提示。
- ◆公開部分との切り分け基準は：自社オリジナル開発と派生物となるかならないか。これに関し、開発ごとに個別に判断する。
- その他これまでの状況で特筆すべき点：なし

2. メリット

- ソース・コードが入手できるため、ソフトウェア本体をシステムに合わせて改変でき、機能の付加やバグの修正が容易である。
- 他の商用ソフトウェアよりも低いハードウェア性能で相応の動作を見込める場合が多い。
- ◆利用前と認識が変わった点があれば特に記入：別段、認識が変わった点はない。
- ◆コスト面でのメリット：相当実績があり安定したソフトウェア（OS）を無償または安価に利用できる、公開された豊富なモジュールやライブラリの利用による開発期間の短縮。

3. デメリット

- ソース・コードの公開と発注顧客の秘密情報、もしくは守秘義務を伴う規格などとの整合性に特に注意し、開発しなければならない。
- ◆利用前と認識が変わった点があれば特に記入：別段、認識が変わった点はない。
- ◆これまでのトラブル・クレーム事例：今のところ、トラブル・クレームはない。

事例23 <ビジネスにおける活用実態>サポート・保守サービス型(その1)**1. オープンソース・ソフトウェアの利用形態**

サポート・保守サービス型(超高速無線LANドライバ開発・保守の例)

①利用しているオープンソース・ソフトウェアの種類

- 名称: Linux カーネル
- 機能: カーネル
- ライセンス条件: GPL
- 選択理由: 改変が容易であるため。

②具体的な利用形態(事例)

- 条件など(記載できる範囲で): 特にオープンソース・ソフトウェアのライセンスとの関連条項
- 保証について: ——
- その他これまでの状況で特筆すべき点: ——
- ソース公開の問題: ——
- ライセンス契約の提示方法: ——
- 使用場所: オモイカネ社内で開発, 通信総合研究所にて開発利用
- 使用目的: 無線LAN研究の実証実験
- 使用に当たっての改変の有無, (改変者は) 改変した事例: 10nsレベルの時間計測等カーネルの機能も拡張。

<入手関係>

- オープンソース・ソフトウェアの入手と利用方法: 直接ダウンロードによって入手。
 - ◆有償/無償の別: ——
 - ◆サポート契約の有無, およびサポート内容: 入手時のサポートはなし。

<提供関係>

- ライセンス契約の提示方法: ——

- 提供先のユーザーとのサポートの有無など: ——
- 追加保証はあるか?: ——
- ソース・コード公開の問題: ——
- その他これまでの状況で特筆すべき点:

これは、ミリ波という新しい周波数を使った無線LANで、P-MP型として上り方向と下り方向の合計312Mビット/秒を実現する、世界最高速レベルの無線LAN用ドライバ・ソフトウェアである。

システム負荷も数%以内に収まっており、Linux応用のネットワーク研究開発として画期的な事例の1つとなっている。

契約関係については、ソフトウェア請負開発なので通常の請負開発に関する契約のみを結ぶ。オープンソースのライセンスについては、直接の顧客である通信総合研究所担当者に説明を行い、同意の上、請負開発を行う。

サポートについてはドライバ・ソフトウェアの最適化を含めた追加の保証を行っている。ソース・コードについては納品時に譲渡されるが、実験用の専用のハードウェアを用い研究所内で使用するものがあり、研究所内の使用にとどまっている。

2. メリット

- ◆利用前と認識が変わった点があれば特に記入: ——
- ◆コスト面でのメリット: ——

3. デメリット

- ◆利用前と認識が変わった点があれば特に記入: ——
- ◆これまでのトラブル・クレーム事例: ——

事例24 <ビジネスにおける活用実態>サポート・保守サービス型（その2）

1. オープンソース・ソフトウェアの利用形態

オープンソース・ソフトウェアを対象とするサポートと保守サービスの提供

①利用しているオープンソース・ソフトウェアの種類

- 名称：Linux, PostgreSQLデータベース・エンジン本体, PostgreSQLアプリケーション・インタフェース (C言語), GNUソフトウェア
- 機能：OS, データベース, アプリケーション・インタフェース言語
- ライセンス条件：主として GPL (Linux), BSD (PostgreSQL), GPL (GNU)
- 選択理由：社内にコア技術者がいるため。

②具体的な利用形態（事例）

- 条件など（記載できる範囲で）：
 - ◆システム開発時のQ & Aから導入後の運用までをサポート。
 - ◆顧客からの障害報告を受け、保守対象としたソフトウェアを起因とする障害と確認したものについて、原因調査、不具合修正、その回避策を提供する。
 - ◆GNUソフトウェア (GCC, GDB, binutilsなど) のカスタマイズ (OS, CPUへの移植作業含む) サービス。
 - ◆バグ修正などの際には、ソフトウェアの改変も生じ得る。
- 保証について：なし
- その他これまでの状況で特筆すべき点：保守では、Linuxのカーネルのバージョンアップやパラメータ値を変えて再構築をするというようなことがよくある。また、ライセンスによっては一切商用利用は行っていない、あるいは個別に許可が必要というものもあり、そういう場合はサポートを行うことができなかったり、

サポート後も面倒な問題が生じたりすることになるので避けざるを得ない。

- ソース・コード公開の問題：顧客社内で動くシステムのサポート・保守の場合にはソース公開の問題は発生しないが、オープンソース・ソフトウェアをベースにしたパッケージ・ソフトを販売する顧客に対して保守を行う場合には、ソース・コード公開の問題が発生する。

2. メリット

- 商用ソフトウェアのサポート・保守が提供できる者は、開発元や特定の代理店などに限られていることが多く、しかも主としてQ & Aとバージョンアップの提供が一般的である。従って、問題発生時にタイミングの良い修正（パッチ）が提供されとは限らないのに対して、オープンソース・ソフトウェアの場合は、ソース・コードが見えるという点で個別ユーザーに対して原因調査を行ったり解決手段を提供したりすることが可能である。
- 商用ソフトウェアに対するサポート・保守との明確な違いは、商用ソフトウェアの場合は、通常、サポート・保守に必要な多くの情報が頒布元にしか存在せず、利用するベンダー側へのノウハウの蓄積が限られるのに対して、オープンソース・ソフトウェアでは、サポート・保守に必要な情報が頒布元に限られることなく数多く豊富に提供され、いわゆる問題点もオープンにされるという点でユーザーに情報が早く届き、問題解決を速やかに図ることが可能である。

3. デメリット

- オープンソースに関する更新情報は豊富であるのに対して、業務としての利用の歴史がまだ短いこともあり、サポート・保守を行うことができる技術がまだ十分でない。

3 利用形態・ビジネス形態別の活用実態と問題点の整理

「1 ユーザーによる活用実態と問題点」(pp.18-25), および「2 ビジネスにおける活用実態と問題点」(pp.26-42)に示した活用実態調査の結果から、利用形態・ビジネス形態別のメリットとデメリット(問題点)を整理した(表5)。

ユーザーから見た問題点として挙げられたものには、技術的な課題がほとんどである。これは、複製、改変、頒布を行わないユーザーにとって、GPLに掲げられたライセンス上の制約が直接影響を及ぼしていない、及ぼす段階にまで至っていないことを表しているものと思われる。

これに対し、ベンダーから見た問題点としては、契約や

ライセンスに関するものが多い。一般的に、複数のオープンソースを組み合わせたものをパッケージとして、販売、あるいはソリューションの提供を行っているが、数種類のライセンスが混雑しているため、使用目的に支障のないライセンスのソフトウェアを集めて、パッケージを構成しなければならない。さらには、そのビジネス形態によっては、幾つか問題点に特徴が表れている。

まず、ディストリビューター型にとって、パッケージを購入した顧客がそのまま無償でビジネスに利用できると誤解している場合が多い。つまり、オープンソース・ソフトウェアを利用したビジネス(販売など)を実施する場合には、別途ディストリビューターであるベンダーとライセンス契約が必要であることを理解してもらにくいという問

表5

利用形態・ビジネス形態	メリット	デメリット(問題点)
ユーザー	●コストを節約できる。特にデータベースでは大きく節約できる ●安定している。過去に使用していた商用製品より安定している ●セキュリティ情報・パッチ対応が早い	●自前でサポートする場合、相談相手がいない ●OS付属のプログラムを利用した場合、セキュリティ・ホールへの対応が遅いことがあった(現在はソース・コードをコンパイルすることにより対応している) ●セキュリティ関係のパッチが多くシステムの維持にコストが掛かる ●Linuxに詳しいエンジニアが必要(Linuxのバージョンアップ、再構築に際して)
ディストリビューター型	●価格が安い ●無償で入手できる ●入手の仕方によってはコストが節約できる	●ディストリビューターが販売している製品まで無償と誤解されることがある ●ディストリビューターのサポート期間が短いことにユーザーが不安を持つ
ソリューション・ビジネス型	●システム構築に最適なソフトウェアを利用することができる	●インフラが無償である ●セキュリティ対応のパッチなど、開発側が当然対応してくれるものと顧客に期待されることがある
アプリケーション型	●特になし	●特になし
オープンソース・ソフトウェアをサポートしたサーバの販売型	●ユーザーはほかのシステムよりも低価格で相応の動作を見込めるシステム構築が可能	●改変が可能なため、ユーザーはLinuxに独自仕様を要求できると思ってしまう
組み込み機器型	●ソース・コードが入手できるため、機能の付加やバグの修正が容易である ●商用ソフトウェアよりも、低性能のハードウェアで相応の動作を見込める場合が多い ●実績があり安定したソフトウェア(OS)を無償または安価に利用できる ●公開された豊富なモジュールやライブラリの利用により開発期間が短縮	●ソース・コード公開義務と発注顧客の秘密情報、または守秘義務を伴う規格などとの整合に注意し、開発しなければならない
サポート・保守サービス型	●商用ソフトウェアのサポート・保守が提供できるのは開発元や特定の代理店などに限られ、しかも主にQ&A対応とバージョンアップなどに限られる。しかしオープンソース・ソフトウェアの場合は、ソース・コードが見えるため原因調査や解決手段の提供が可能である	●サポート・保守ができる人材が少ない

題がある。ベンダーが用意したソフトウェアが無償のものであるという前提から、購入したソフトウェアを自由に使用してもよいと拡大解釈し、ベンダーの権利を失念している傾向が強い。

また、ソリューション・ビジネス型では、ソリューション・ビジネスのためのシステムの開発、導入、そして^{かし}瑕疵の修正までが契約で締結されているが、あくまでもベンダーが開発した部分に関する保証にとどまっている。しかし、顧客はその契約によってカーネルのパッチ修正までを無償で要求することがある。本来、ライセンス上、カーネルに関してはベンダーに修正する義務はないのだが、顧客としてはパッケージ全体が保証範囲と認識していることに原因がある。パッチ修正を依頼するためには、別途、保守・サポートの契約を締結する必要がある。

同様に、サポート・保守サービス型においては、ソフトウェアの変更を伴った瑕疵修正を行う場合に問題が生じることがある。オープンソースでは、自社開発のパッケージではないソフトウェアの保守を請け負うことも多いが、ライセンスによって変更が禁止されているソフトウェアでは代替品を探したり、あるいは新規に開発したりすることもあり、純粋な保守・サポートの枠を超えることがある。このようなときには、新たに開発契約を結ぶ必要があるが、市販ソフトウェアのような無償サポートに慣れている顧客にとっては納得しにくい傾向が強い。

このように、オープンソースという漠然とした言葉によってそのビジネスの成果物があいまいになっているために、当初の契約内容から大きく外れた要求であってもそのことを顧客が理解できず、また、顧客に明確に説明できないこともある。

オープンソース・ソフトウェアを活用したビジネスの活性化を図るためには、関係者間での法的問題に関する理解を深めることが極めて重要である。また、個々の契約書作成の際にも、その契約内容を事細かに書面にしておくよう努めることが必要である。

4 今後の展望

これまでに見た通り、オープンソース・ソフトウェアは、その頒布自体がビジネスになりにくいことに加え、それを組み込んだシステムの開発、それを応用したサービスの提供など、そのほかと複合的に組み合わせたビジネスでも、その成果物の範囲があいまいであり、ビジネスの組み立てが難しい場合がみられる。

オープンソース・ソフトウェアを組み込んだシステムを提供するフラッグシップ型のビジネスは、引き続き堅調に推移すると見込まれる一方、考えられる新たな展開としては、①技術力のあるユーザーが自らの判断により積極的にオープンソースの活用を進める、②パッケージ化されたソフトウェア商品、機器への組み込み分野など、コモディティ化された商品の部品やパーツとしてオープンソース・ソフトウェアを積極的に活用する、といった方向性が考えられる。

こうした傾向を踏まえると今後のビジネスの形態としては、オープンソース・ソフトウェアを自由に使いこなせるエンド・ユーザーやビジネス・ユーザーを増やすための人材育成の分野と、機器などへの組み込み分野における一層の市場成長が期待される。

5部

オープンソースをめぐる視点

これまで、オープンソース・ソフトウェアの活用が徐々に活性化しつつある現状を取りまとめてきた。また、その中でも触れたが、オープンソース・ソフトウェアの今後の動向についても、代表格ともいえるLinuxをはじめ、その成長を予測する声が多い^{注1)}。

しかし、本格的な活用促進に向けて残された課題はまだ多い。5部以降では、大きく3つの視点から今後の課題を整理してみたい。

1 利用者側からのオープンソース・ソフトウェアに対する誤解

第1に、利用者側のオープンソース・ソフトウェアに対する誤解や不安の問題である。

現実にユーザーの声を確認すると、オープンソース・ソフトウェアの利用には不安の声も多い。中には、①システム導入後に必要となるサポートが得られないのではないか、②ウイルスが混在しているのではないか、③品質面で劣っているのではないか、など半ば先入観に基づくと思われるものもある。

また、オープンソース・ソフトウェアといえば、OSソフトとしての「安全性」や「経済性」が頻繁に話題になる。しかし、オープンソース・ソフトウェアのメリットは、何

も「安全性」^{注2)}や「経済性」^{注3)}だけにあるわけではない。むしろ、それは利用者の能力や状況によっても変わるものである。

確かに、先進的な利用者の間でも①サポートできる技術者が少ない、②オープンソース・ソフトウェアを組み込んだシステムに関するサービスのレベルが定義できない、など深刻な問題が発生しているのは事実だ。しかし、オープンソース・ソフトウェアに関して注目されるべきは、先端的研究者の知見をオープンソースという開発スタイルを通じて効率的に集約した成果物が、既に数多く残されているという事実である。市場では、OSソフトに限らずさまざまな種類のオープンソース・ソフトウェアが提供されているし、ライセンス条件についても細かく見ればさまざまな形態がある。単なる不安や誤解といったレベルの理由でオープンソース・ソフトウェアの活用が進んでいないとすれば、大変残念なことである。

このため、6部「1 一般利用者（一般ユーザー）向け導入検討ガイドライン」（pp. 47-49）において、利用者向けにその導入検討ガイドラインを策定する。

2 開発スタイルとしてのオープンソース

第2に、本来、オープンソース・ソフトウェアは、利用

注1) Linux関連市場予測としては、例えば以下のようなものがある。世界OS市場におけるLinuxの売り上げは、2001年に8000万米ドルだったが2006年には2億8000万米ドルに成長する（2002年7月、IDC社調べ）。世界サーバOS市場におけるLinuxのシェアは、2002年に9%だったが、2005年には25%に成長する（2002年、Gartner社調べ）。国内オープン系OS用リレーショナル・データベース・システムのソフトウェア市場におけるLinuxのシェアは、2001年に1.6%だったが、2006年には9.7%に成長する（2002年8月、IDC社調べ）。

注2) 例えば、米国防省が採用しているソフトウェアの安全性評価基準であるTCSEC (trusted computer system evaluation criteria) では、通常のLinuxもWindows MEも共にD評価。セキュリティ機能を強化したLinuxやWindows XPはC評価となるなど、ほぼ互角の評価となっている。

注3) 例えば、WindowsとLinuxをTCO (total cost of ownership: 情報システム全体に掛かるコスト) で比較すると、どちらが有利かは判然としない。Linux優位を伝える調査結果としては、Cybersource社が2002年6月に公表したレポートがあるが、それによると「従業員250人の中堅企業で3年間運用すると、Linuxの方が25%～35%安い」とする。これに対し、IDC社が2002年12月に公表したレポートによれば「1台のネットワーク・サーバで100ユーザーを5年間サポートした場合、Linuxは1万3263米ドル、Windowsは1万1787米ドル」と、Windowsが優位であると伝えている。加えて、パッケージ構成によっても、利用者自身の技術力によっても、ここで挙げる保守運用費用は大きく変化するため、オープンソースの経済性を一般化して比較する意味は乏しい。

者側ではなく開発者側の文脈でこそ語られるべきであるという視点である。

オープンソース・ソフトウェアは、本来、商用的なソフトウェアの開発・市場化に対する1つのアンチテーゼとして始まったフリー・ソフトウェア運動に端を発している。開発したソフトウェアの成果物がライセンスという行為の下に特定企業の管理下に置かれるのは間違っており、その成果はそれを望むすべての人に公開され、自由に活用されるべきであるという考え方がその核にある。この活動は、ソフトウェアを産業として育成するという観点からは難しい論点を提起する。しかし、研究者の知見を効果的に集約し、ソフトウェアの開発者・利用者にそのメリットをリアルタイムでフィードバックするという意味では、優れた開発スタイルというべきであろう。

特にIT分野ではOSソフトのみならず、コンテンツへのアクセス制御技術や認証技術など、複数の企業や主体がその研究成果を共有しながら次に進まないと解決しない課題が確実に増えつつある。残念ながら、オープンソース・ソフトウェアに取り組む開発コミュニティのほとんどは海外にあり、我が国ではまだまだこうした開発スタイルは定着していない。我が国でも、利用側からの文脈ではなく、これから行うソフトウェアや情報システムの開発の中でこそ、オープンソース・ソフトウェアというスタイルをどのように使っていくのか検討していくべきであろう。

このため、6部「2 ITサービス企業（ビジネス・ユーザー）向け導入検討ガイドライン」（pp. 49-50）、および6部「3 ソフトウェア開発者（開発者）向け導入検討ガイドライン」（pp. 50-51）において、ビジネスにおけるオープンソースの導入検討ガイドラインを整理することとする。

3 オープンソース・ソフトウェアに関する法的枠組み

第3に、オープンソース・ソフトウェアにかかわる法的枠組みの透明性の問題である。

有用であるにもかかわらず、利用・開発両面において我が国でいま一つ本格的なオープンソース・ソフトウェアが進まない共通の理由として、「ライセンス形態が多様であり内容が分かりにくい」「法的にリスクが高い」など、その法的枠組みが明確でないことも指摘されている^{注4)}。例えば、Linuxを活用するために派生的に開発されたソフトウェアがオープンソースとして扱われるのか、商用ソフトウェアとして扱われるのかは、専門家でも判別しにくいケースが多い。オープンソース・ソフトウェアが徐々に商用ソフトウェアの領域を浸食していくようなイメージがあることから、こうした性格をviral characterと呼ぶ向きもある。オープンソース・コミュニティ発祥の地である米国では、法体系がコモン・ロー的な性格であることから政府レベルでの検討は見られないが、欧州では2002年11月に、その法的解釈や法的枠組みの在り方をめぐってレポートが公表され^{注5)}、パブリックコメントに付されている。

このため本報告では、オープンソース・ソフトウェアをめぐる法的枠組みについて、現状を俯瞰的に整理し、理解するためのガイドラインを用意したい。

具体的には「7部 オープンソース・ソフトウェアのライセンス契約」（pp. 52-60）において、オープンソース・ソフトウェアをめぐる多様なライセンス契約の現状を整理する。さらに、「8部 GPLに関する法的問題の整理」（pp. 61-97）では、Linuxなどに用いられているGPL（GNU general public license）に絞って、その問題を整理したい。

注4) この問題を早い段階で取り上げた分析としては、三宅ほか、「Linuxをどう使う、燃え上がる『GPL問題』」、日経エレクトロニクス、2001年12月17日号、no.811、pp. 67-76（NIKKEI ELECTRONICS 2003.7.21 特別編集版、pp. 5-14に再掲）がある。

注5) <http://europa.eu.int/ISPO/ida/export/files/en/1115.pdf>を参照。

6部

オープンソース・ソフトウェア
の導入検討ガイドライン

オープンソース・ソフトウェアとしての最も大きな特徴は、そのソフトウェアのソース・コードが公開されているという点にある。ここでは公開されているオープンソース・ソフトウェアを入手して、実際に活用する立場の者から見た導入検討ガイドラインを示す。

1 一般利用者（一般ユーザー）向け導入検討ガイドライン

初めに、利用者、つまり改変・頒布を行わず単に使用する者が、導入時に検討すべき代表的なメリットとデメリットを列挙する。

一般的な傾向を見ると、活用しようとするオープンソース・ソフトウェアのコミュニティに十分な人材が集まっている分野のものであって、自らにも相当のサポート能力がある場合は、オープンソースの採用は優位に働く可能性が高い。他方、自らにサポートする能力が乏しく、その分野についてはコミュニティが必ずしも手厚いとはいえない場合は、さまざまなリスクを考慮する必要がある。

いずれにせよ一般ユーザーの場合は、オープンソース・ソフトウェアを含むシステム全体の構築やサポートを行う企業と別途契約するケースが多いと考えられることから、そうしたサービス企業の実力や契約条件なども十分吟味し、総合的な判断を下すことが必要となろう。

1.1 コスト

一般的に、市販されている商用ソフトウェアとオープンソース・ソフトウェアの入手に掛かる費用（購入価格など）を比較した場合、オープンソース・ソフトウェアの方が無償あるいは実費のみで提供されていることもあり、安価で

あることが多い。さらに企業の場合、商用ソフトウェアを使用するためには、使用する社員の人数分のクライアント・ライセンスを購入する必要があるのに対し、オープンソース・ソフトウェアでは複製が認められていることから、最初の1本の入手に掛かる費用以外には必要ない。このようにオープンソース・ソフトウェアには、個人から企業規模に至るまで、導入（購入）時のコストを削減できるというメリットがある。

しかし、保守・運用まで含めた全体のコストは、そのユーザーの利用目的や利用能力によっても大きく異なる。このため導入時に当たっては、次に掲げる②～④を念頭に置きつつ、初期導入コストにとどまらずサポート費用まで含めた十分な調査が必要である。その全体の費用は、調査の結果によってオープンソース・ソフトウェアが優位とするものもあれば、不利とするものもあり、一概にいずれかが優位とは言い切れない。

1.2 サポート、瑕疵への対応

普段の日常的な作業を行っているときは、商用ソフトウェアであってもオープンソース・ソフトウェアであっても、その差異を感じることはない。しかし、全く未知のソフトウェアを使い始めるとき、つまり、その使用方法を詳しく学ぶときに差異が生じる。商用ソフトウェアについては、著名なソフトウェアでなくても、ベンダーが用意したマニュアルや手引き、一般の書店に並ぶ入門書などの書籍、またはパソコン教室など、学ぶための手段が数多く存在しており、初心者も比較的安心して学ぶことができる。

しかし、オープンソース・ソフトウェアについては、ベンダーやコミュニティなどサポート体制がしっかり整っている大規模ソフトウェアなどを除いて、作者が添付した

マニュアルしか存在しない場合があり、またそれは初心者では理解できないことも多い。すなわち、オープンソース・ソフトウェアには、それを使いこなすために相応の知識が要求されるというデメリットがある。

また、ソフトウェアに^{かし}瑕疵が見つかった場合でも、商用ソフトウェアと異なり、ベンダーによるサポートや修正対応がない。オープンソース・ソフトウェアについては多くの場合、そのライセンス内で「無保証」をうたっていることもあり、作者に修正する意志がなければ自己で行わなければならない。一般的には、利用者自身がソース・コードのバグ取りを行うか、専門家（システム・インテグレーター）に委託することになるが、この場合、瑕疵を解決できない、もしくは、高額の委託費用が必要になるといったデメリットが生じる可能性もある。

なお、これらのデメリットの影響は、オープンソース・ソフトウェアに関して利用者がどの程度の知識を有しているかによってかなり左右される。特に、そのオープンソース・ソフトウェアに関して高度な知識（技術）を有している場合、デメリットがなくなるばかりではなく、現状のオープンソース・ソフトウェアをカスタマイズして、自分の好みの機能を追加したり、企業独自の仕様に変更したりできるというメリットが漸たに出てくる。

1.3 バージョンアップへの対応

ソフトウェアではバージョンアップが行われることがあるが、商用ソフトウェアの場合、これにより旧バージョンのサポートが近いうちに打ち切られてしまうこともある。このため、サポート体制を維持するために、特に重要な機能の追加がなかったとしても、バージョンアップを実行しなければならない。これに対し、オープンソース・ソフトウェアでは、サポートが自己自身で実施したり、または直接委託契約したりするため、追加された機能に魅力がなければバージョンアップする必要がなく、そのことによる不利益はない。従って、オープンソース・ソフトウェアには本当に必要なバージョンアップのときにだけ対応すればよいというメリットがある。

ただし、一部の商用サービス・ベンダー・ソフトウェア

とは異なり、バージョンアップがあった際の告知やサービスが自動的に提供されることはない。また、開発やサポートを行うコミュニティが何らかの事情で解散してしまった場合には、自分自身で周囲の環境変化に応じたバージョンアップを行わなければならない場合も少なくない。このため、自社自身でオープンソース・ソフトウェアの内容を理解しメンテナンスできない場合には、こうした点も考慮に入れたサービス・ベンダーとの契約や社内の人材育成などが必要となる。

1.4 アプリケーション・ソフトウェアの可用性

オフィス業務系ソフトウェアの文書作成プログラムや表計算プログラムのように、連携して利用すると便利なソフトウェア群がある。このようなソフトウェア群について、商用ソフトウェアでは同じベンダーの開発したもの同士でなくては互換性がない場合が多い。このようなとき、A社の文書作成プログラムを使っていた場合、表計算プログラムもA社を選ばざるを得ず、さらには別の用途のものすべてのプログラムにA社製を選ばざるを得なくなってしまう。そして、このことによってA社の拘束力をまた大きくすることになる。

このように特定のベンダーのソフトウェアに独占された状態がベンダー・ロックインといわれるものである。ほかのソフトウェアへの変更が困難になるため、半強制的なバージョンアップなどによりコストの増加を招くことになる。

これに対して、オープンソース・ソフトウェアではデータ・ファイルの構造や連携を図るためのインタフェースの仕様が公開されているため、その仕様にのっとって開発されたソフトウェアがほかにも存在している可能性がある。特に、需要の高いオフィス業務系の分野に属するソフトウェアにおいては、同種のソフトウェアが数多く開発されている。その中から、利用者の好みに合ったソフトウェアを選択することができる。このように、オープンソース・ソフトウェアには、ベンダー・ロックインを回避することができるというメリットがある。

ただし、市場で支配的なアプリケーション・ソフトウェアが、すべてオープンソース・ソフトウェア向けのシステ

ムに対応しているわけではない。ユーザーにとっては、オープンソース・ソフトウェアであるか否かにとらわれず、自らのデータの利用環境をよく検討の上、自分が必要な機能を実現するためのソフトウェアがオープンソース・ソフトウェアにあるかどうか、もしくは連携して動くソフトウェアを開発・入手できるかどうかを十分に検討する必要がある。

2 ITサービス企業（ビジネス・ユーザー）向け導入検討ガイドライン

ITサービス企業が、ソリューションなど外部に対してシステム構築サービスを提供する業務において、オープンソース・ソフトウェアを利用する場合のメリットとデメリットを列挙する。

一般的な傾向を見ると、一般ユーザーの場合と事情は大きくは変わらない。すなわち、活用しようとするオープンソース・ソフトウェアのコミュニティーに十分な人材が集まっている分野のものであって、自らにも相当のサポート能力がある場合は、オープンソースの採用は優位に働く可能性がある。他方、自らにサポートする能力が乏しく、その分野についてはコミュニティーが必ずしも手厚いとはいえない場合は、さまざまなリスクを考慮する必要があるであろう。

2.1 コスト

まず、ソリューション提供用のシステムに必要なコストを比較する。

商用ソフトウェアかオープンソース・ソフトウェアを問わず、既存のシステムをそのまま購入する場合には、個人使用と同様に、導入に掛かるコストはオープンソース・ソフトウェアのメリットになると思われる。

しかし、新規にシステムを開発するには、自社もしくはシステム・インテグレーターなどの委託先にて開発をすることになるが、この場合のオープンソース・ソフトウェアのメリットとデメリットを判断することは非常に困難である。つまり、既存のオープンソースのライブラリを用いる

ことによる開発費削減のメリットと、独自に開発したライブラリを無償で公開することによる権利放棄のデメリットとのバランスで判断されるためである。

一般的に、ソリューションが独創的であればあるほど、システムも独自に開発することになり、開発費用の回収ができないオープンソースではデメリットが大きくなると思われる。

2.2 システムの安全性

外部に対してソリューションを提供する以上、安定して運用する責任を果たすために人員、スタッフをそろえておく必要があり、これは商用ソフトウェアかオープンソース・ソフトウェアを問わず必須である。しかしながら、実際にトラブルが発生した場合の対応が大きく異なる。

商用ソフトウェアの場合、ソフトウェアの瑕疵やセキュリティ・ホールが発見されたとしてもスタッフにはどうすることもできず、ベンダーがパッチを準備するまではシステムを停止しておくことしかできない。

これに対し、オープンソース・ソフトウェアで瑕疵などが見つかった場合、スタッフ自身が修正パッチを検索、もしくは修正作業を実施することになる。従って、オープンソース・ソフトウェアではスタッフに相応の技術が要求されるというデメリットがある反面、システムの停止時間が短くて済むというメリットもある。

ただし、オープンソース・ソフトウェアは世界的規模で修正パッチが作成されているため、その発行頻度が商用ソフトウェアと比較できないほど高い。従って、システムを頻繁に停止しなければならないというデメリットを持つことになる。しかしこれも裏を返せば、それだけ瑕疵、セキュリティ・ホールへの監視の目が厳しくなり、最善の状態が保たれているというメリットになる。

こうした点も、それぞれの実情に応じて個別にその比較優位を検討する必要がある。

2.3 ソフトウェアの公開

個人使用と異なり、企業がソリューションを提供する場合、使用しているソフトウェアが何であるかを公開してい

ることになる。このとき、そのソフトウェアに特定の特許が含まれていたり、ライセンスに不手際があったりした場合、その権利者からライセンス料の請求や使用差し止めを求められる可能性がある。従って、特許を含むライセンス問題については、十分な調査・整理が必要である。

2.4 ソフトウェア自体の瑕疵、法的責任への対応

オープンソース・ソフトウェアのライセンスでは開発者は「無保証」をうたっている。

特に、オープンソース・ソフトウェアは、複数のソース・コードを組み合わせ、さらに何代もの改変が行われてきた結果、多くの人間の手を経て、複数のライセンスが絡み合っている。このため、特許や著作権などの法的な問題が介在しやすいというデメリットがある。

このため、ソフトウェアに残っている未知の瑕疵によって顧客が損害を受けた場合、自社のサービスとしてどこまで顧客に対応を保証するか、契約条件を決める際に十分に検討する必要がある。

2.5 サービス・レベル

自らがユーザーにどのようなサービス・レベルを約束するかという問題に加え、自分自身がどのようなサービスを受けられるかという問題がある。というのも、一般的に、オープンソース・ソフトウェアは「無保証」の下で開発されているため、そのサポートもボランティア的な思想で行われている部分があり、信頼性に乏しいという意見があるからである。

確かに、個人レベルで開発しているオープンソース・ソフトウェアについては、サポートが無かったり、打ち切られたりする。しかし、OSに絡むような基盤ソフトウェアに関しては、コミュニティと呼ばれる集団によってバージョンの管理・瑕疵チェックが行われており、サポート体制の維持が図られ、高い信頼性を得ている。また、クライアント・アプリケーションと呼ばれる爆発的な人気を誇るソフトウェアについては、その需要の多さからシステム・インテグレーターが有償でサポートを行っているものもある。

従って、一般には、オープンソース・ソフトウェアも何

ら遜色のないサポートを受けられる可能性があり、サポートが特段デメリットになることもないと思われるが、導入しようとするオープンソース・ソフトウェアの開発コミュニティの状況とライセンス契約については、あらかじめ十分調査しておく必要がある。

3 ソフトウェア開発者（開発者）向け導入 検討ガイドライン

次に、ソフトウェアを開発する者から見たメリットとデメリットについて述べる。以下から分かるように、ソフトウェアを開発する側から見ると、それぞれの点においてメリットとデメリットが均衡している。

従って、オープンソース・ソフトウェアの採否は、それによってどれだけ技術的に優位な知見や人材を収集することができ、強力なビジネス全体の差異化につなげられるかどうかに懸かっているといえよう。ちなみに、ソフトウェアそのものに対する対価という形で収益を得る必要の乏しい組み込み機器においては、今後、その活用が特に有望になることが推察される。

3.1 收益性

まず、収益性について検討する。

商用ソフトウェアは、その開発に多大なコストが掛かるが、競合製品がない場合には単価を高く設定することができる。これに対して、オープンソース・ソフトウェアでは、既存のオープンソースを利用することで開発費を抑えることは可能であるが、製品の売買という形ではなく開発受託やメンテナンスの請負という形でのみ利益を受けることになるため、収入は少ない。

従って、オープンソース・ソフトウェアは低支出・低収入であり、メリットとデメリットが競合している。

3.2 新規参入容易性

ビジネス的には、商用ソフトウェアという形態では、先進組の占める市場に後発組が浸食していくことは困難であるのに対し、オープンソース・ソフトウェアでは、先行す

る開発ベンダーに対する対抗力をアピールしやすいため、後発組であってもサポートの内容によっては新たに市場に参入できる可能性がある。

また、権利管理技術や認証技術など、広くさまざまなソフトウェアやハードウェアで、今後共有すべき技術がある。こうした新しい分野でも、オープンソースは1つの重要な選択肢となるだろう。

なお、オープンソース・ソフトウェアにおいても、他の追従を許さないほど高機能なソフトウェアを開発し、速攻で市場に浸透させることができたならば「キラー・アプリケーション」と呼ばれてその分野の市場を独占することができる可能性もある。現状ではそのようなキラー・アプリケーションの数は少ない。

従って、オープンソース・ソフトウェアの市場においては、先進組にとってはシェアを失いやすく、ベンダー企業としての存続能力が衰えてしまうというデメリットがある一方、後発組にとっては新規参入や市場創出が容易になるというメリットが生じる。

3.3 公開によるメリットとデメリット

オープンソース・ソフトウェアを開発した場合、そのソース・コードを公開することが必要となる。

メリットとしては、自社の技術力を誇示できるとともに、類似のソフトウェアを販売しているベンダーに対して自社も正当な開発者であることを証明し、不要な訴訟を牽制できる。しかし、逆に技術力を露呈させ、他ベンダーに盗用されてしまうというデメリットにもなる。

また、自らが品質保証管理のできないソフトウェア製品が出回ることにもなる。

3.4 組み込み機器における優位性

ベンダーとして利益を得る方法は幾つかあるが、オープンソース・ソフトウェアについては、そのソース・コード自体の販売から直接利益を得ることはできない。しかし、これをハードウェアに組み込み、組み込み機器として販売する場合には利益を付加することは可能である。

まず、メリットとして、組み込むソフトウェアの開発に

ついては、独自にソフトウェアを開発する場合に比べてオープンソース・ソフトウェアの方が安価で、しかも短期間で行うことができると思われる。また、キラー・アプリケーションを組み込むことで、その組み込み機器の売り上げを伸ばすことも期待できる。

それに対して、デメリットとしては、オープンソースであるため、他のベンダー機器にも搭載されることを制限できず、そのソフトウェアの搭載を独自の特徴として維持することができない。

また、これから組み込みソフトウェアをオープンソース・ソフトウェアに切り替えようと考えているベンダーにおいては、ハードウェア開発・設計もオープンソース・ソフトウェアに対応させなければならないというデメリットを考慮する必要がある。過去の製品との互換性を維持しつつオープンソースの採用を計るのか、オープンソースだけに絞るのか、ベンダーごとにその判断基準が異なってくるものと思われる。

7 部 | オープンソース・ソフトウェアのライセンス契約

2部「2 代表的なオープンソース・ソフトウェア」(pp. 8-11) で見た通り、オープンソースと呼ばれるソフトウェアには、さまざまな種類のものがある。OSソフト、データベース・ソフト、電子メール用ソフトなどとさまざまだが、オープンソース・ソフトウェアのライセンス契約面でも、公開すべきソース・コードの範囲や条件など、細かく見ればオープンソース・ソフトウェアの数だけ異なる契約がある。

「8部 GPLに関する法的問題の整理」(pp. 61-97) で詳細に説明する通り、オープンソース・ソフトウェアにかかわるライセンス契約については、課題が指摘されている。

第1に、開発者の法的なリスクである。例えば、オープンソース・ソフトウェアと連動して動く商用ソフトウェアを開発した場合、どこまでがオープンソースでどこからが自分の商用ソフトウェアなのか、その境界が今のGPLの文言からは判然としない。ソース・コードの開示が要求される範囲も、組めるビジネス・モデルも、その境界によって大きく変わってくる。

第2に、ユーザー側の法的リスクである。例えば、オープンソース・ソフトウェアを組み込んだシステムを構築した場合、一部商用ソフトウェアをオープンソース・ソフトウェアだと偽って開発業者に混入され著作権侵害を起こしても、ユーザー自身には発見が難しい。また、自分の業務上のノウハウに合わせるためにオープンソース・ソフトウェアを加工して社内の関係部署に配布したら、加工後のソース・コードも一般に公開せよと言われたとか、予想もしないような対応を求められることもある。

こうした課題は、特にLinux向けに使われているGPL (GNU general public license) で課題となっている。しかし、オープンソース・ソフトウェアのライセンス形態はGPL

だけではない。GPLでは加工した後のソフトウェアの改変部分についても厳しく公開が求められるが、例えばFreeBSDと呼ばれる契約では、改変後のソース・コードの公開は必要なく、独自に商用で使うこともできる。オープンソースという開発・流通スタイルを積極的に広げていくという観点からは、我が国の実態にあったオープンソース・ライセンスを模索していく必要がある。

この7部では、こうしたあるべきオープンソース・ライセンスの検討に入る前段階の作業として、まず一般に「オープンソース」というくられ方をするライセンス契約に複数の種類があることを明らかにし、それぞれの契約類型の概要と注意点を明らかにする。

1 各種オープンソース・ソフトウェアのライセンス概観

オープンソースというライセンス形態は、必ずしも一通りではない。「1部 オープンソース・ソフトウェアの定義」(pp. 5-6) で見た通り、その定義によって「改変の自由が保証されている」など、一定の枠組みがある。しかし具体的な契約内容は、個々のオープンソース・ソフトウェアごとによっても詳細が異なることが多く、多種多様である。

ここでは、代表的なオープンソース・ソフトウェアとそれが活用するライセンス形態の概要について整理する。

1.1 GPL (GNU general public license)

①概要

オープンソース・ソフトウェアで最も代表的なライセンス契約形態が、GPLである。これは、Richard M. Stallman氏による「フリー・ソフトウェア宣言」を具現化するため

に考案されたライセンス契約であり、コンピュータ・ソフトウェアに対する占有権の根拠となる著作権法を逆手に取り、対象となるソフトウェアが何者かに占有されることを排除し、誰でも「自由に」使用できることを保証した、極めて巧妙な仕組みである。

すなわち、このライセンス契約はライセンシーに対し、改変部分のソース・コードを公開し、同一条件で誰でも使えるようにすることを条件に、対象ソフトウェアの複製、改変、頒布を許諾する。ライセンサーは対象ソフトウェアの著作権を保持し、ライセンシーが契約条件に違反した場合は契約解除となり、ライセンシーが複製、改変、頒布すれば著作権侵害となる。

通常のプロプライエタリ・ソフトウェアを開発する企業がLinuxなどのGPL対象ソフトウェア周辺で開発を行う際には、企業秘密・資産であるソース・コードの公開が場合に依りて要求されるが、その境界が不明確なため、さまざまな問題を引き起こしている（いわゆる「GPL問題」）。

従来からUNIX系のオープンソース・ソフトウェアの多くはGPLに基づいて頒布されていたが、ビジネス環境におけるLinuxの利用が普及し、Linux周辺での企業活動が盛んになるにつれて、GPLのビジネス環境における問題点が一躍脚光を浴びるようになった。その詳細については、次の「8部 GPLに関する法的問題の整理」(pp. 61-97)で詳細に紹介する。

②契約の成立

対象プログラムを改変または頒布することにより、契約承認の意図が表示されたものとされる（第5条）。

③対象範囲

対象プログラムおよびその派生物（「derivative work」すなわち「work based on the Program」）の複製、改変、頒布が対象（第1条、第2条）。

対象プログラムを改変した場合、その部分（section）が対象プログラムから派生するものでなく、独立した別個の著作物（work）であると合理的に判断され、別個の作品として頒布される場合は、本ライセンスの対象とならない。

しかし、その部分がwork based on the Program全体の一部として頒布される場合は、その全体が本ライセンスの対象となる（第2条）。

④ライセンスの継承

対象プログラムのソース・コードを複製して頒布することができるが、本ライセンス契約のコピーも同時に引き渡さなければならない（第1条）。

対象プログラムを改変して頒布することができるが、本ライセンス契約に基づいて、頒布しなければならない（第2条）。

⑤ソース・コードの開示

オブジェクト/実行形式で頒布することも許されるが、その場合はソース・コードを同梱するか、最低3年間、頒布に必要な費用を上回らない料金をソース・コードを頒布することを申し出た書面を同梱しなければならない。オブジェクト/実行形式が（WWWサイトなどの）所定の場所へのアクセスにより頒布される場合は、ソース・コードへの同等なアクセスを提供すればよい（第3条）。

⑥料金

対象プログラムのソース・コードを複製して頒布することができる。頒布費用を徴収できる。また有償で保証を提供してもよい（第1条）。

対象プログラムを改変して頒布することができるが、すべての第三者に対して無料で、本契約に基づいてライセンスしなければならない（第2条）。

1.2 LGPL

(GNU lesser general public license)

①概要

GPL対象のプログラムと他のプログラムとをリンクさせ1つのプログラムにすると、他のプログラムを含む全体がGPLの対象となってしまう。GPLの大きな問題として指摘されてきたこの点を解決するために、コピーレフト性を弱めて商用プログラムとの共存を可能にしたのがGNU LGPL

である。

②契約の成立

対象プログラムを改変または頒布することにより、契約承認の意図が表示されたものとされる（第9条）。この点は、GPLと同様である。

③対象範囲

対象ライブラリおよびその「ライブラリに基づく著作物（work based on the Library）」の複製、改変、頒布が対象（第1条、第2条）。

対象ライブラリを改変した場合、その部分（section）が対象ライブラリから派生（derive）するものでなく、独立した別個の著作物（work）であると合理的に判断され、別個の著作物として頒布される場合は、本ライセンスの対象にならない。しかし、その部分がライブラリに基づく著作物全体の一部として頒布される場合は、その全体が本ライセンスの対象になる（第2条）。

以上の点はGPLと同様だが、以下の点が明記されているところはGPLとは異なる。

対象ライブラリの派生物を含まず、対象ライブラリとともにコンパイルまたはリンクして動作するように設計されたプログラムは「ライブラリを使用する著作物（work that uses the Library）」であり、それ自体は、本ライセンスの対象外である。しかし、そのプログラムを対象ライブラリとリンクしてできる実行形式プログラムは、ライブラリの派生物（derivative work）であり、本ライセンスの対象になる（第5条）。

「ライブラリを使用する著作物」が対象ライブラリの一部であるヘッダ・ファイルを使用する場合、例えばソース・コードは対象ライブラリの派生物にならなくても、オブジェクト・コードは派生物になるかもしれない。このようなオブジェクト・コードが数値パラメータ、データ構造レイアウトとアクセッサ、および小さなマクロや10行以下の小さなインライン関数しか使わない場合、それが法的に派生物であるかどうかにかかわらず、オブジェクト・ファイルは制限なしに使用することができる（第5条）。

④ライセンスの継承

対象ライブラリのソース・コードを複製して頒布することができるが、本ライセンス契約のコピーも同時に頒布しなければならない（第1条）。

対象ライブラリを改変して頒布することができるが、本ライセンス契約に基づいて、頒布しなければならない（第2条）。

これらの点は、GPLと同様である。

⑤ソース・コードの開示

オブジェクト/実行形式で頒布することも許されるが、その場合は、ソース・コードを同梱しなければならない。オブジェクト/実行形式が（WWWサイトなどの）所定の場所へのアクセスにより頒布される場合は、ソース・コードへの同等なアクセスを提供すればよい（第4条）。

この点は、GPLと同じである。

⑥ライブラリを含む著作物の頒布

本条項がGPLと異なるもう1つの大きなポイントである。

「ライブラリを使用する著作物」と対象ライブラリをリンクした著作物については、ユーザー自身による使用のための改変と、デバッグのためのリバース・エンジニアリングを許可する限り、どのようなライセンス条件で頒布してもよい。

対象ライブラリが使われていること、およびライブラリとその使用は本ライセンスで規定されることを表示し、本ライセンスのコピーを添付しなければならない。さらに、ライブラリのソース・コードおよび「ライブラリを使用する著作物」のオブジェクト・コードまたはソース・コードを添付し、ユーザーがライブラリを改変した場合に再リンクすることにより、改変された実行形式プログラムを作成できるようにしなければならない（第6条）。

⑦料金

対象ライブラリのソース・コードを複製して頒布することができる。頒布費用を徴収できる。また有償で保証を提供してもよい（第1条）。

対象ライブラリを改変して頒布することができるが、すべての第三者に対して、無料で、本契約に基づいてライセンスしなければならない（第2条）。

1.3 Xライセンス

Xコンソーシアムライセンス（Xライセンス）は米国政府の資金により開発されたX Window Systemに使用されているライセンスであり、著作権表示と許諾条件表示のみを条件とする、極めて制限の緩いライセンスである。

著作権表示と許諾条件表示さえしておけば、Xライセンスのコードを他のプログラムに組み込み、しかも組み込み後のソース・コードを非公開にできる。しかも、Xライセンス以外のライセンスに基づいて頒布することもできる。この2点は、GPLと異なる大きな特徴である。

次に述べるBSDライセンス、ApacheライセンスはXライセンスの変形である。

1.4 BSDライセンス

UC（カリフォルニア大学）のBerkeley校により開発されたBSD（Berkeley Software Distribution）系UNIXや、その他のソフトウェアに使用されているライセンスであり、再頒布時に、著作権表示と再頒布条件表示と無保証・免責宣言を行うことのみを条件とする、極めて制限の緩いライセンスである。

著作権表示と再頒布条件表示と無保証・免責宣言さえしておけば、BSDライセンスのコードを他のプログラムに組み込み、しかも組み込み後のソース・コードを非公開にできるため、企業からすれば商用化のしやすいライセンスであるが、一方、オープンソース開発者の立場から見れば、オープンソースとして開発したコードがクローズドになってしまう危険がある。

ちなみに、初期のBSDライセンスには派生物の広告に初期開発者名を表示することが条件として盛り込まれており、多数の開発者が携わった場合、広告にすべての開発者名を表示しなければならないという問題を抱えていたが、現在はこの条項は削除されている。FSFではBSDライセンスを使用する場合には、この「修正済BSDライセンス」を使用

するように薦めている。

なお、BSD UNIXのパソコン版であるFreeBSDにも同じライセンス（FreeBSD Copyright）が使われている。

また、WWWサーバ・ソフトであるApacheに用いられているApacheライセンスも、基本的にXライセンス、BSDライセンスと同じである。

1.5 MPL (Mozilla public license)

Netscape社がNetscapeブラウザのオープンソース化を始めた時に作られたNPL（Netscape Public License）が基になっており、自己開発部分のソース・コードを非公開とすることができ、GPLほどコピーレフト性が強くない。また特許への対応、準拠法・裁判管轄の規定など、GPLの不備も解決されている。

MPLをベースとしたライセンス契約は多くの企業で使われており、InterBase Public License、SUN Public Licenseは基本的にMPLと同じものであり、Apple Public License、IBM社のCommon Public License、IBM Public LicenseもかなりMPLに似たものとなっている。

1.6 その他のオープンソース・ソフトウェア・ライセンス

その他のオープンソース・ソフトウェア・ライセンスについては個々に紹介しない。概要についてはライセンス比較表（表1）を参照されたい。

2 オープンソース・ソフトウェア・ライセンスの比較・類型化

オープンソース・ライセンスといわれるものには数多くあり、OSI（Open Source Initiative）のWWWサイトには約40件のOSI公認オープンソース・ライセンスがリストアップされている。今回この中から主要なライセンスである17件を取り上げ、ライセンス条件の比較を行った結果を表1に示す。

オープンソース・ソフトウェア・ライセンスには数多くの種類がある。しかし、表1に示した各種オープンソース・

表1 各種オープンソース・ライセンスの比較

○はイエス, ×はノーを意味する。

ライセンス種別	契約同意を示す動作 <>内は参照条文	頒布する場合ライセンス 継承が必要か <>内は参照条文	改変部分のソース・コード 開示が必要か <>内は参照条文	他ソース・コードと組み合 わせた場合 ^{*1} ,他ソース・ コードにもライセンスが 波及するか <>内は参照条文
GPL	改変,頒布<5>	○<1>	○<3>	○<2>
Q Public License	頒布<1>	○<2>	○<4.b>	○<6>
GNU LGPL	改変,頒布<9>	○<1>	○<4>	×<2,5,6>
MPL (Mozilla Public License)	規定なし	○<3.1>	○<3.2>	×<3.7>
Interbase Public License	規定なし	○<3.1>	○<3.2>	×<3.7>
SUN Public License	規定なし	○<3.1>	○<3.2>	×<4>
Apple Public License	ダウンロード,使用 <前文>	○<2.1>	○<2.2>	×<1.b>
Common Public License	使用,複製,頒布 <前文>	○<3>	○<3.b) iv>	×<1.b>
IBM Public License	使用,複製,頒布 <前文>	○<3>	○<3.b) iv>	×<7,8>
Artistic License (Perl License)	規定なし	○<1>	× ^{*2} <3.c) 3.d) 4.c) 4.d)>	×(規定なし)
BSD License	規定なし	○<1,2>	×(規定なし)	×(規定なし)
FreeBSD Copyright	規定なし	○<1,2>	×(規定なし)	×(規定なし)
MIT License	規定なし	○<全体>	×(規定なし)	×(規定なし)
X11 License	規定なし	○<全体>	×(規定なし)	×(規定なし)
ZPL (Zope Public License) V2.0	規定なし	○<1,2>	×(規定なし)	×(規定なし)
Apache Software License V1.1	規定なし	○<1,2>	×(規定なし)	×(規定なし)
ISC LicenseV1.0	規定なし	○<3,7,10>	×(規定なし)	×(規定なし)

^{*1} MPLなどの場合combine,GPLの場合a part of a whole which is a work based on the Programとすることを意味する。

^{*2} ①改変したモジュールの名前を変え,標準モジュールとの違いを明記するか,②別の頒布方法について著作権者を取り決めた場合は,改変部分のソース・コードを開示しないことも可能。

^{*3} LGPL対象ライブラリとリンクした実行形式プログラムについては,ソース・コードを開示しないことも可能。

^{*4} 顧客の自己使用のための改変,デバッグのためのリパース・エンジニアリングは禁止できない<6>。

^{*5} 当該ライセンス対象外の他のソース・コードと組み合わせ (combine) で拡張製作物 (a larger work) を開発・頒布する場合,他のソース・コードのライセンス条件は頒布者が自由に決められると考えられる<3.7>。

^{*6} ①独自のライセンス契約に基づいて本ライセンス対象プログラムに関連して提供されるソフトウェア・モジュールであって,②本ライセンス対象プログラムの派生物 (derivative work) でないものは本ライセンスの対象ではない<1.b>。

^{*7} 頒布者が開発して本パッケージにリンクしたサブルーチンは本パッケージの一部とは見なされない<7>。

^{*8} 特許条項ありとは,特許権をユーザーにライセンスする条項があることを意味する。

^{*9} GPLと共存可能とは,GPL対象モジュールと組み合わせ (combine) て1つのプログラムとすることができることを意味する。

	改変して頒布する場合のソフトウェア全体について、以下のことが可能か		他のコードと組み合わせて頒布する場合、他のコードについて以下のことが可能か		特許条項はあるか ^{※8} <>内は参照条文	GPLと共存可能か ^{※9}
	ソース・コードを非開示とする	改変を禁止とする	ソース・コードを非開示とする	改変を禁止とする		
	×	×	×	×	×	○
	×	×	×	×	×	×
	×	×	○ ^{※3}	○ ^{※4}	×	○
	×	×	○ ^{※5}	○ ^{※5}	○<2.1 (b) ,2.2 (b) >	×
	×	×	○ ^{※5}	○ ^{※5}	○<2.1 (b) ,2.2 (b) >	×
	×	×	○ ^{※5}	○ ^{※5}	○<2.1 (b) ,2.2 (b) >	×
	×	×	○ ^{※5}	○ ^{※5}	○<2>	×
	×	×	○ ^{※6}	○ ^{※6}	○<2.b) >	×
	×	×	○ ^{※6}	○ ^{※6}	○<2.b) >	×
	○	○	○ ^{※7}	○ ^{※7}	×	○
	○	○	○	○	×	○
	○	○	○	○	×	○
	○	○	○	○	×	○
	○	○	○	○	×	○
	○	○	○	○	×	×
	○	○	○	○	○	×

ソフトウェア・ライセンスの比較から分かる通り、オープンソース・ソフトウェア/フリー・ソフトウェアの本質であるソース・コード公開条件およびコピーレフト性の強さ^{注1)}に着目することにより、おおよそ次の3カテゴリに類型化することができる。

● GPL類型

コピーレフト性の最も強いライセンスであり、改変部分のソース・コード公開が必須である。またGPL対象コードと他のソース・コードを組み合わせることで1つのプログラムとした場合、他のソース・コードについてもライセンス条件が波及し、ソース・コード公開やその他の義務が生じる。

● MPL類型

改変部分のソース・コード公開は必須であるが、コピーレフト性は強くなく、他のソース・コードと組み合わせても、他のソース・コードまでライセンス条件が波及することはなく、ソース・コード公開を要求されることはない。

● BSDライセンス類型

改変部分を含めソース・コード公開は必須でなく、オープンソース・ソフトウェアのプロプライエタリ・ソフトウェアへの組み込みも可能。

以上の3カテゴリとオープンソース・ソフトウェア以外

のソフトウェアとの比較を含めて表に示すと、表2のようになる。

今回調査対象としたオープンソース・ライセンスを上記3カテゴリに従って分類すれば、表3のようになる。

3 それぞれの類型に対する注意点について

3.1 GPL類型

GPLはフリー・ソフトウェア運動を具現化したライセンス契約であり、いったんフリー・ソフトウェアとして誰でも改変・頒布できるようにしたソフトウェアがプロプライエタリ・ソフトウェアになり、一部の人の占有物になることがないように考えられている（コピーレフト）。すなわち、対象ソフトウェアを改変して頒布した場合、対象ソフトウェアの派生物（米国著作権法に基づく derivative work）もGPLの対象となり、改変者はGPLの条項に基づいてソース・コードの開示などを行わなければならない。

個人がフリー・ソフトウェアとして開発・頒布する場合には特に問題はないと考えられるが、企業が既存のGPL対象ソフトウェアの改変・頒布を行う場合には、企業活動の成果を無償で公開し、改変も許可することになるのを認識しなければならない。

ここで問題になるのが、どの範囲がGPLの適用対象となるか、すなわち「派生物（derivative work）」の解釈であるが、GPLでは対象プログラム（の全体または一部）を他

表2

○はイエス、×はノーを意味する。

類型	複製,再頒布可能	改変可能	改変部分のソース・コード公開が必要	他のソース・コードと組み合わせた場合,他のソース・コードのソース公開が必要
GPL	○	○	○	○
MPL	○	○	○	×
BSDライセンス	○	○	×	×
フリーウェア	○	×	---	---
プロプライエタリ・ソフトウェア	×	×	---	---

のコードと組み合わせて1つのプログラム（work）とした場合、プログラム全体がGPLの対象となるとしており、派生物の解釈が拡張されているといえる（いわゆるviral character：ウイルス性）。企業がデバイス・ドライバを開発した場合とか、組み込み機器にLinuxを使用したような場合、どの範囲までがGPL対象となり、ソース・コード公開などの義務が発生するのか重要な問題となるが、明解な判断基準はない。この問題については「8部 GPLに関する法的問題の整理」（pp. 61-97）で詳述する。

3.2 MPL 類型

オープンソース・ライセンスとして最も新しく、準拠法、裁判管轄、特許への対応についても明示されており、法的問題はほとんどないといえる（そもそも契約として成立するかという問題はあるが、この点はGPLも同様であり、「8部 GPLに関する法的問題の整理」（pp. 61-97）における検討を参照されたい）。GPLほどコピーレフト性が強くなく、オープンソースを基にして開発したソフトウェアの実

行形式プログラムを、企業独自のライセンスにより頒布することができる（ただしソース・コードは開示しなければならない）。企業が何らかの理由により新たにオープンソース・ソフトウェアを開発する場合は、本ライセンスをベー

表3

GPL類型	GPL
	Q Public License
MPL類型	GNU LGPL
	MPL (Mozilla Public License)
	Interbase Public License
	SUN Public License
	Apple Public License
	Common Public License
	IBM Public License
	Artistic License (Perl License)
BSDライセンス類型	BSD License
	FreeBSD Copyright
	MIT License
	X11 License
	ZPL (Zope Public License) V2.0
	Apache Software License V1.1
	ISC License V1.0

表4

(a) 新たにソフトウェアを開発する場合

類型	個人での使用		企業による使用	
	メリット	デメリット	メリット	デメリット
GPL	プロプライエタリ・ソフトウェアに組み込まれる恐れがない	---	---	ソース・コードの公開が必要。ライセンスが無償
MPL	プロプライエタリ・ソフトウェアに組み込まれる恐れがない	---	---	ソース・コードの公開が必要。ライセンスが無償
BSDライセンス	---	プロプライエタリ・ソフトウェアに組み込まれる恐れがある	---	プロプライエタリ・ソフトウェアに組み込まれる恐れがある

(b) 既存のソフトウェアを改変して頒布する場合

同一機能のソフトウェアが複数あったり、同一ソフトが複数のライセンスにより頒布されたりする場合には、ソフトウェアやライセンスを選択する必要がある。

類型	個人での使用		企業による使用	
	メリット	デメリット	メリット	デメリット
GPL	自分の改変部分が他者のプロプライエタリ・ソフトウェアに組み込まれる恐れがない	---	---	ソース・コードの開示範囲が広がる恐れがある
MPL	自分の改変部分が他者のプロプライエタリ・ソフトウェアに組み込まれる恐れがない	---	ソース・コードの開示範囲が限定される	---
BSDライセンス	---	自分の改変した部分が他者のプロプライエタリ・ソフトウェアに組み込まれる恐れがある	他者のソフトウェアを自社のプロプライエタリ・ソフトウェアとすることが可能	自社による改変部分が他社のプロプライエタリ・ソフトウェアに組み込まれる恐れがある

スにしたものを採用するのがよいと考えられる。

3.3 BSDライセンス類型

3類型の中では最も制約が弱く、ソース・コード公開の義務もない（実際にはソース・コードを頒布するのが普通である）。利用する者にとっては都合がよいが、ソフトウェア開発者にとっては自分の開発・改変したソース・コードがプロプライエタリ・ソフトウェアに組み込まれてしまうリスクがあることを認識しなければならない。

3.4 使用目的に応じたライセンスの比較

以上3類型について、使用目的（ソフトウェア開発者の立場）に応じた比較を表4に示す。

4 今後の検討の方向

本報告書では、GPLを所与のものとして、その法的問題の解明およびそれを踏まえたGPLの使いこなし方（ガイドライン）を提示することを目的とした。

しかし、これまで見てきた中でも分かる通り、GPLには運用が難しい問題が多く、オープンソース・ソフトウェアという開発スタイルを今後我が国に定着させるに当たって、必ずしも最適なライセンス形態とは言い切れない。

このため今後は、我が国においてGPLを使用する場合の法的問題を解決した日本版GPLというべきものの必要性について検討し、新たにオープンソース・ソフトウェアを開発する人々に提示していくことが必要であると考えられる。

このため「8部 GPLに関する法的問題の整理」(pp. 61-97)では、まずGPLを誤解なく扱えるよう、対象を数多くあるライセンス契約の中からGPLに絞り、初心者向けのガイドラインを提示するとともに、その法的な課題についてさらに彫り込んだ整理を行いたい。

注1) ライセンス対象のソース・コードと他のソース・コードを組み合わせると1つのプログラムとした場合、他のソース・コードにまでライセンス対象範囲が拡張されるかどうか。

8部

GPLに関する法的問題の整理

ここでは、数多くあるオープンソース・ソフトウェアのライセンス契約の中でも、特に多くの議論を呼んでいるGPLに絞って、その理解のポイントと注意点を整理する。

まず、GPLを正確に理解してもらうための初心者向けガイドラインを提示する。このことによって、オープンソースが特に怖いものでも、不可解なものでもないことを明らかにする。

次に、そうはいっても現実にはさまざまな不安を招くGPLに対する専門家の解析を提示する。この「2 GPL法律的

検討（論点別）」(pp. 63-97) で、GPLの何が問題で、どういう考え方で課題を整理していくべきかの道筋が提示される。

1 GPL初心者に向けたQ & A

まずGPLの扱いに慣れていないユーザーに対して、GPLとは何かを理解してもらうために有効であると思われる12個のQ & Aを紹介したい。

【初心者向けGPLガイドライン】

Q1 : GPLとは何ですか？

A1 : オープンソースの対象となるソフトウェアを使う際に、ユーザーが守るべき条件を定めた契約の一種類です。LinuxなどGPLを契約として採用しているソフトウェアについては、本契約に違反する形で利用すると、著作権侵害などで訴えられる可能性があります。

Q2 : どのような行為がGPLで制約されるのでしょうか？

A2 : GPLの対象となるソフトウェアは、複製し頒布したり、加工したりする自由が認められている一方、GPLが定めた条件を守らなければなりません。また、自分が頒布した相手にもGPLライセンスを守らせることが要求されます。代表的な条件は、次の3つです。

- 自分が加工した後のソース・コードも必ず公開すること
- 自分が頒布した相手にも自由な加工を認めること
- 頒布する際には、必ずGPLの写しを添付すること

ただし、私的な、または企業内部での使用や加工はGPLの対象外であり、自由に行えます。この場合、関係会社に頒布するなど事実上企業外部に公開してしまったのと同じ状態にならないよう、注意する必要があります。

Q3 : 契約はどのような形で結べばよいのでしょうか？

A3 : 特別に契約行為を行う必要はありません。また私的に使用したり、企業内部で活用したりしている限りは、そもそも対象外ですので、何ら問題は発生しません。ただし、第三者に渡したり加工したものを公開したりすると、GPLの契約に明示的に合意したと自動的に見なされますので、GPLの条件に合致するように気を付けなければなりません。

Q4 : 契約を守らなかったらどうなるのでしょうか？

A4 : 契約違反が明らかになれば、FSF(Free Software Foundation : フリー・ソフトウェア財団) というGPLに基づくオープンソース・ソフトウェア(GNU, Linux など)の著作権を持つ推進団体から、その他の場合は各ソフトウェアの著作権者から警告状が来ることになります。

Q5 :本当に訴えられたケースはありますか？

A5 :FSFから警告状が来る例は多くあります。我が国では、今のところ判決まで至った訴訟はありませんが、国際的に訴訟になったものとしては、スウェーデンのMySQL AB社 対 米NuSphere社の訴訟があります。

Q6 :使っているソフトウェアはどこから入手できるのでしょうか？

A6 :ソフトウェアは関連するインターネット上の開発サイトからダウンロードすることができます。

手軽に使用する場合はディストリビューションと呼ばれるソフトウェア集がよく使われています。これはコンパイルなどの作業なしにすぐに使えるよう、あらかじめ編集が施された2次的な頒布物です。ディストリビューションは雑誌添付のCD-ROM、非営利の頒布団体のWWWサイトからのダウンロード、商用の頒布業者のパッケージ製品などによって入手することができます。

ソース・コードを直接の頒布元から入手する場合は、検索エンジンやSourceForge (<http://sourceforge.net/>)などで開発サイトを探することができます。例えばLinuxのカーネルは<http://www.kernel.org/>から、GNUツールは<ftp://ftp.gnu.org/>から、というように直接入手することができます。

Q7 :トラブルが発生した場合、どうすればいいのでしょうか？

A7 :原則、誰もサポート・保守をしてくれる人はいません。インターネットなどで情報を入手して、自分あるいは自社で対応する必要があります。

ですから、自分や自社だけでは対応に自信がない場合は、サポートや保守を担当してくれる業者とサポート・保守契約を締結して、トラブルが発生した場合にサポート・保守を依頼できるようにしておく必要があるでしょう。

Q8 :著作権侵害の恐れは？

A8 :人からコピーをもらっても、人にいくらコピーを渡しても、著作権表示、無保証の表示、GPL契約書をそのまま付けて渡す限り著作権侵害にはなりません。ただし、自分がオープンソース・ソフトウェアだと思っていなくても第三者の商用ソフトウェア

が全く混ざっていないとはいえない場合があります。入手先にその点を十分に確認しましょう。

Q9 :特許権侵害の恐れは？

A9 :GPLの対象であるソフトウェアが他人の特許権を侵害することはあり得ます。過去にも他社の特許権に抵触するので、プログラムの書き換えを余儀なくされた例があります。しかし、GPLの対象として流通しているソフトウェアは、既にそうしたリスクを回避してきたソフトウェアです。特に、オープンソース・ソフトウェアの個々の使用者が特許権侵害で訴えられるということはまずないでしょう。

ただし、自らがオープンソース・ソフトウェアを改変するなど商業的に活用する場合には、競争相手が特許戦略を行使してくるリスクも含め、その過程で他人の特許を侵害しないように十分注意する必要があります。

Q10 :加工したソフトウェアは公開が必要ですか？ その結果加工部分に入ったノウハウが流出してしまうのではないですか？

A10 :自分や自社で使うだけなら公開は不要です。ただし、関連会社に使わせる場合も含め、他人へ頒布する場合には、加工部分も含めてソース・コードの公開が必要になりますので、特にGPLの場合は注意が必要です。

Q11 :人に渡すとき無保証を条件にしても問題はありますか？

A11 :自分が他人に渡したものに重大な^{かし}瑕疵があれば、GPLが無保証をうたっている、民法や消費者契約法の規定により責任を負わなければなりません。しかし、実際に、通常の品質のソフトウェアが問題になることは極めて少ないと思われます。

Q12 :有償で販売してもよいのでしょうか？

A12 :複製、頒布に掛かる実費程度のお金を取ることは許されています。また、有償で保証サービスを提供することも許されています。従って、有償で販売してはいけないということはありません。

2 GPL 法律的検討（論点別）

ここでは、さらに専門家向けにGPL問題の全貌を概説する。ここで示す解釈は、研究会での議論をベースに各有識者が個人的見解としてまとめたものであり、オープンソース・ソフトウェアを推進する者や権利者に個別にその妥当性を確認したものではない。

しかしながら、オープンソース・ソフトウェアに詳しい相当の専門家が、現時点で出し得る平均的な見解を示したものと、実務の参考に資するものと思われる。

2.1 FSFの思想とGPL文言解釈の関係

GPL問題の多くは、どこまでがGPLの対象範囲なのか、が判然としない点に端を発する。特に、GPLが規定する「派生物 (derivative work)」の範囲が不明確であり、かつ、多くの国の著作権法が規定する派生物や2次著作物の概念よりもさらに広い概念であることが、専門家による法的な議論ですら混乱してしまう原因となっている。

しかし、こうした「派生物」の考え方は、GPLが生まれた背景にさかのぼって、その活動の狙いを正確に理解することなくして、ただ単に法的に理解することは難しい。このためここではまず、法的な議論の第一歩としてGPLが生まれた背景から「派生物」の解釈を整理したい。

■フリー・ソフトウェア運動「GNUプロジェクト」

欧州連合の加盟国間におけるオープンソース・ソフトウェアの共同利用を推進する欧州委員会IDA (Interchange of Data between Administrations) は「オープンソース・ソフトウェアのプーリング (POOLING OPEN SOURCE SOFTWARE)」と題する報告書を、2002年6月に公表した^{注1)}。

同報告書は「オープンソース・ソフトウェアの核心は、『コードではなくライセンス』である。多くのITに熱心な人々やフリーの開発者たちは失望するかもしれないが、これが現実である。同時にそれは法的な枠組みについての分析がなぜ非常に大切なのか、という理由でもある」と述べている。

しかしコンピュータの黎明期には、ソース・コードの提

供・共有はごく当然のこととされていた。ソース・コードがオープンであることの意義は取り立てて意識されていなかったもので、ライセンスをもって対処すべき性格の事柄であるという発想自体が乏しかった。

これを共有する意義が高度に法的なライセンス形態へと昇華した世界初のケースとして、Richard M. Stallman氏の主宰で1983年^{注2)}に開始されたフリー・ソフトウェア運動「GNUプロジェクト」を掲げることができる。GNUプロジェクトの推進組織として1985年にフリー・ソフトウェア財団のFSFが設立された。

GNUプロジェクトは、オープンソース・ソフトウェア全般について現在に至るまで多大な影響を与えているので、この運動が開始された背景を、最初に俯瞰しておく必要がある。

1970年代に入るとソフトウェアの開発が産業として隆盛し始め、当時Stallman氏が属していたMIT (マサチューセッツ工科大学) の人工知能研究所から、優秀なプログラマーの大部分がソフトウェア産業界に引き抜かれた。米国著作権法にも、1976年改正を経て1980年改正によって第101条に「コンピュータ・プログラム」の定義規定が設けられた結果、ソフトウェア・プログラムに対し著作権法による排他的独占権を付与することが明文化された。

こうした状況の中でStallman氏は前記運動を開始して「GNU宣言」^{注3)}を発表し、合衆国憲法は学芸の発展促進が著作権保護の目的である旨を明記しているから、著作権制度は当然の権利ではなく人工的なものにすぎないと主張した。その背景には、著作権保護の目的に関する英米法流のインセンティブ論が存在している。そして、ソース・コードの共有によって友情を守り、プログラムの入手が容易になるだけでなく、無駄なシステム・プログラミングの重複を避けて、労力を技術水準の進歩に向けることができる

注1) European Commission IDA, 「POOLING OPEN SOURCE SOFTWARE」, <http://europa.eu.int/ISPO/ida/export/files/en/1115.pdf>

注2) GNU プロジェクトの最初の声明は1983年に書かれた Richard M. Stallman, 「Initial Announcement」, <http://www.gnu.org/gnu/initial-announcement.html> であるが、Richard M. Stallman, 「The GNU Project」, <http://www.gnu.org/gnu/thegnuproject.html> では同氏は1984年1月にMITを退職してGNUソフトウェアを書き始めたとしている。

注3) Free Software Foundation, 「The GNU Manifesto」, <http://www.gnu.org/gnu/manifesto.html>

とした。

■フリー・ソフトウェアの定義と著作権法

FSFは、運動対象の概念規定として「フリー・ソフトウェアの定義」^{注4)}を公表して、ユーザーには、ソース・コードを含めてソフトウェアを自由に使用、改変、頒布を行う権利が認められなければならないとしている。

この運動が守ろうとするユーザーの権利内容をより具体的に示すと、①目的を問わずプログラムを実行する自由(第0の自由)、②プログラムの動作を研究し必要に応じて改変を加える自由(第1の自由)、③コピーを再頒布する自由(第2の自由)、④プログラムを改良しコミュニティ全体が恩恵を受けられるよう改良点を公衆に発表する自由(第3の自由)、という4種類の自由が掲げられている。

このうち②および④の前提として、ソース・コードの入手可能性が確保されていなければならないとされている。また、これらの自由が本当のものになるには、ソフトウェアの開発者自身であっても以上の自由を撤回できないものとしなければならないという。

前記の「定義」は、ある意味で著作権法に対するアンチテーゼたる意義を有する。分析のために米国著作権法第106条と対比すると、②の「第1の自由」は2次的著作物を作成する権利(日本法の翻案権に対応)、③の「第2の自由」は頒布権(日本法の譲渡権に対応)、④の「第3の自由」は2次的著作物の頒布権(日本法の譲渡権に対応)に、それぞれ対応しており、①ないし③はすべて権利者の許諾を要する行為である(米国著作権法第107条)。

これに対し、これらの自由の中には複製権に対応した複製の自由は明記されていないが、当然の前提とされているものと思われる(その半面、①が意味する「プログラムを実行する自由」に対応すべき著作権法上の権利は存在しないが、日本法と異なり、プログラムをメモリにロードする行為を複製概念でとらえる米国著作権法上においては明記すべき意味があるし、これを複製権概念でとらえない日本法の場合でも、使用するためにハード・ディスク装置へのインストールが必要なソフトウェアが一般化している現在では、同様に複製権との関連で自由であることを明らかに

する意味がある)。

FSFが提唱するフリー・ソフトウェア概念における「フリー」とは、以上の通り「無料」ではなく「自由(フリーダム)」の意味であって、Stallman氏の著作においても、フリー・ソフトウェアであってもこれを無料で頒布するだけでなく、ソフトウェア開発資金などを得る機会として手数料を徴収して頒布することも可能であるとされている。

なお、この運動が米国を本拠地としているためであろうか、ベルヌ条約・大陸法的な著作権人格権に対する言及がなされておらず、また、著作権に関する世界知的所有権機関条約第8条が認める公衆への伝達権についても「定義」の公表後に策定されたものである関係上、「フリー・ソフトウェアの条件」においては対象とされていない。

■コピーレフトとGPL

次に、フリー・ソフトウェア運動の目的たるフリー・ソフトウェアの普及を実現するために、Stallman氏によって考案された概念が「コピーレフト」^{注5)}である。

すなわち、無方式主義の下で創作により自動的に付与される著作権をあえて放棄して^{注6)}、パブリック・ドメイン・ソフトウェア(PDS)としてソース・コードを公開して無償提供すれば、頒布を受けた者は著作権法の下で無償提供されたソース・コードに改変を加え、原作者の意図に反してStallman氏のいう「proprietary(財産的)」なソフトウェアとして頒布することができ、営利目的のためにクローズドな商用ソフトウェア(commercial software)へと勝手に転用されてしまう恐れがある。これを避けるためにGNUプロジェクトでは、ソフトウェアのパブリック・ドメイン化に代えて、著作権を放棄することなく保有して主張し続けた上で、頒布条件として、そのソース・コードおよびそれから派生したいかなるソフトウェアに対しても、使用、改変、そして再頒布の権利を与える。これを再頒布する人にも、この頒布条件を変更しないことを条件として、改変の有無を問わず、頒布される人にもそれをコピーして改変を加える自由を与えなければならないとする手法が採用された。

この手法は、米国法で著作権を意味する「コピーライト

(copyright)」を逆利用したものであったので、Stallman氏によって「コピーレフト (copyleft)」と命名された。同氏は、コピーレフトでは著作物の私物化を目的とする権利としてではなく、それを自由にしていく権利を保護するために著作権を使用するものであり、この手法によって「ソース・コードと自由とが法的に分離不能なものとなる」と述べている。

なお、こうした性格は「伝搬性」と呼ばれることもあるが、この言葉はStallman氏やFSFが命名したものではない。

Stallman氏は「コピーレフト」は普遍的な概念であって、これを実際に達成するための手段は、場合に応じて数多く存在していると述べる。それらの選択肢の中で、特にGNUプロジェクトにおいて、コピーレフトの概念を具現化するために作られたライセンス条項の中心が「GNU一般パブリック・ライセンス (GNU general public license)」であり、頭文字を取って「GPL」の略称で呼ばれる。

なお、コピーレフト概念を具現化するためにFSFが作成したライセンス条項として、ほかに「GNU劣等一般パブリック・ライセンス (LGPL: GNU lesser general public license)」, GNUソフトウェアのマニュアルなどドキュメント用の「GNUフリー文書ライセンス (FDL: GNU free documentation license)」がある。

以上を要約すると、Stallman氏が主宰するFSFのフリー・ソフトウェア運動「GNUプロジェクト」が守ろうとする運動対象を概念規定したものが「フリー・ソフトウェアの定義」であり、運動の普及を実現するために考案された概念が、著作権 (コピーライト) を逆利用した「コピーレフト」, この概念を具現化するために作られたライセンス条項の中心的存在が「GPL」, という位置付けとなる。

■ GPL上の諸概念の解釈

GPLの現行バージョンは1991年6月発表のVersion 2である。GPL英語版はGNUのWWWサイトに掲載され、さらに新旧2種類の邦語訳へのリンクが張られている^{注7)}。GPLには「英文文書 (GNU general public license) を正式文書とする」ことが明記されているので、以下では英文文書に本執筆者が独自に邦訳を加えたものを使用して説明する。

GPLにおいて、ソース・コード公開に関する諸規定と相まって「コピーレフト」を具体化する中心的な規定が第5条であり、同条第4文は「あなたが本プログラム (または本プログラムを基礎とした著作物 (any work based on the Program)) の改変または頒布を行えば、それ自体であなたは本ライセンスを受け入れ、かつ、本プログラムまたは本プログラム2次的著作物の複製、頒布、改変に関するこれらの条項と条件のすべてを受け入れたことを示します」と規定する。

「本プログラムを基礎とした著作物」の概念は、換言すると「コピーレフト」が及ぶ範囲を画する重要な概念であり、一般に「伝搬性の及ぶ範囲」と呼ばれることも多い。定義規定がGPL第0項に別途置かれており「本プログラム (the Program) または derivative work under copyright law」を意味することが明記されている。

GPLのいう「derivative work」には、前述した新旧2種類の邦訳では、共に「派生物」という語が充当されてきた。しかし、そこには「under copyright law (著作権法上の)」という言葉が付加されている。従って「derivative work」は著作権法上の用語であり、現にStallman氏の母国でありかつFSFの本拠が存在する米国の著作権法第101条にも「derivative work」という用語の定義規定が置かれている。さらに、ベルヌ条約自体も第2条第3項で「derivative work」に言及している。こうした著作権法上の概念である「derivative work」には、我が国では「2次的著作物」という邦訳が当てられることが一般的であるから、GPLにいう「derivative work under copyright law」の語も「派生物」ではなく「著作権法上の2次的著作物」と邦訳すべきことになる。そうすることにより「コピーレフト」が逆利用しようとした著作権法との関連性を、より明確化すること

注 4) Free Software Foundation, 「The Free Software Definition」, <http://www.gnu.org/philosophy/free-sw.html>

注 5) Free Software Foundation, 「What Is Copyleft?」, <http://www.gnu.org/philosophy/free-sw.html>

注 6) こうした放棄の理解は米国著作権法を前提としており、大陸法系諸国の著作権法では著作人格権は放棄不能とされる傾向がある。

注 7) Free Software Foundation, 「Unofficial Translations」, <http://www.gnu.org/licenses/translations.html>

ができる。

ところで一般に著作権法では、原著作物を改変して2次的著作物を作成するには原作者の許諾を要し、改変されて作られた2次的著作物にも原著作権の効力が及ぶ。また、前述した米国や日本の著作権法をはじめ諸国の著作権法制には、頒布の前提となる複製行為はもとより、頒布行為自体にも著作権が及ぶとしているものが多い。GPLには「derivative work」の定義規定が置かれているが、本定義規定の性格については、こうした著作権法の構造を前提として、著作権法上の2次的著作物に関する法解釈の指針を示そうとしたものと見るべきか、それとも契約法理にのっとり「コピーレフト」が及ぶ範囲を独自に定めたものととらえるべきか、疑問が生じ得る。

本定義規定は文言上、米国著作権法上の定義と対比しても内容が異なっているように思われることからすれば、後者の立場も成り立ち得ないではない。しかし、GPL第5条は、前記引用個所に先立ち第2文以下で「あなたに本プログラムまたはその2次的著作物を、改変または再頒布する許可を与えるものは本ライセンス以外にはありません。これらの行為は、あなたがもし本ライセンスを受け入れないのであれば、法律で禁止されます」と規定している。この点からも明らかな通り「コピーレフト」という手法において、それを具体化すべきGPLを伝播させる機能を営むものは、著作権侵害の回避という点にある。前述の通り「著作権法上の」という形容詞が付されていることと併せて検討すると、GPLにいう「derivative work」の概念は、ライセンス契約上の独自概念として考えるべきではなく、著作権法に従って解釈されるべき2次的著作物という法概念であり、前記定義規定は法解釈指針を示そうとしたものととどまると解することが論理的な帰結であろう。

GPLの適用されたソフトウェアの作成や流通が1つの国の内部にとどまる場合には、その国の著作権法に従って2次的著作物への該当性を判断すれば足るが、同一国内にとどまらない場合は、著作権の準拠法による。準拠法については「準拠法と解釈の基準」(pp. 68-70)で検討されているので、それを参照されたいが、国ごとに2次的著作物の法解釈は異なり得ることを考えると、GPL上に統一的な

独自の定義規定が置かれたことには、やや無理があることを付言しておきたい。

(8部 2.1 岡村委員)

2.2 契約と諸法の関係

「2.1 FSFの思想とGPL文言解釈の関係」(pp. 63-66)で見た通り、GPLにかかわる混乱の多くは「派生物」の定義をはじめとするフリー・ソフトウェア運動の活動背景そのものにある。しかし、実際に課題を解決する上では、そうした混乱を法的に整理していかなければならない。

その際重要なのは、個々の課題がGPLという契約に基づくものか、そもそも著作権法に基づくものか、どの国の何法を根拠に判断すればよいのか、正確に理解することである。このため、ここではその点を体系的に整理する。

■契約と著作権の論理的な関係

ここでは、特に焦点となる契約法と著作権法とのいずれが、それぞれの判断の局面で重要となるのか、その論理関係を明らかにし、GPLという契約の問題なのか、いずれにせよ著作権法に基づいて判じればよいのか、検討を行う。

契約と著作権の論理関係

(1) 著作物利用と契約の必然性

著作物を利用するのに契約や許諾は必然ではない。著作権侵害とならない使用利用形態であれば契約や許諾なく行える。

複製物に添付された契約条項がある場合、著作権侵害とならない使用利用を行うことについて、この契約条項が使用者/利用者との関係で成立しているのかという問題がある。使用者/利用者の契約締結意思の問題ほか、そもそもこのような使用利用について許諾は必然ではないからである。

次に、もし許諾がなければ著作権侵害となる使用利用を行うことについて、この契約条項すべてが使用者/利用者との関係で成立しているのかという問題がある。使用者/利用者の契約締結意思の問題が同様にあるが、このような使用利用については、許諾がなければ適法とならないので、使用利用の前提となる契約の成立を使用者/利用者は認め

ていると見なせるとの議論はあり得る。

著作物の複製物に添付された契約条項による契約成立の議論は「契約の成立」(pp. 70-73)を参照されたい。

(2) 著作物の使用利用と著作権の及ぶ行為

著作物としてのソフトウェアの使用利用に関しても、契約を行う必然性が生じるのは、ソフトウェアの使用利用が著作権の及ぶ行為(複製、頒布・譲渡・貸与、翻案など)となる場合、すなわち無断で行うと原則著作権侵害となる場合である。

(3) ライセンス契約と禁止権の解除

ライセンス契約には、①著作権に関係のある規定、②著作権に関係のない規定、の2つの規定が含まれる。

①は著作権の及ぶ行為に対する禁止権の解除であり、②は著作権に関係のある規定以外の規定となる。

ライセンシーは、①の禁止権の解除が欲しいという理由から、②の条件を受け入れることになる。

(4) 契約違反と著作権侵害

契約違反が債務不履行にとどまる場合と、債務不履行に加えて著作権侵害にも該当する場合がある。例えば、ライセンス料の不払いは債務不履行にとどまるが、許諾形態を超えた利用は債務不履行に加えて著作権侵害となる。

先の①の規定違反は債務不履行に加え、原則著作権侵害となる。他方②の規定違反は、債務不履行にとどまる(ただし、債務不履行が原因でライセンス契約が解除された後の利用は無許諾利用となり著作権侵害となる)。

(5) 法律が認める利用形態と契約による禁止

法が無許諾で利用することを認める形態(権利制限)を契約で禁止することが可能かとの議論がある。すなわち、法律が認める利用形態(権利制限)の規定は強行法規か任意法規かという問題である。例えば、複製物の所有者となる利用者にバックアップ用も含め一切の複製を禁止する契約条項は著作権法47の2の規定と抵触するのでこの問題が生じる。

(6) 著作権と無関係の条項の妥当性

また、著作権法以外の観点から契約条項の妥当性を見る必要性もある。例えば、ライセンス契約におけるさまざまな条件が独占禁止法、消費者保護法、公序良俗などの観点から問題がないかなどである。その他の法律との関係の議論は「その他、法との関係」(pp. 79-81)を参照されたい。

契約としてのGPLと著作権法

(1) GPLにおけるライセンサー/ライセンシーとライセンス対象プログラム

ライセンサーの数は、改変頒布があるたびに、ねずみ講的に広がる可能性がある。GPLの対象となる改変がなされ頒布されると、ライセンシーはライセンサーにもなる。このような改変・頒布のあるごとにライセンサーおよび対象プログラムが増殖していく。このような場合、自分以外の権利者との関係ではライセンシー、自分の権利についてはライセンサーとなり、双方の地位を併有することとなる。

個々のライセンサー(改変者を含む)は、当初自分が権利を持つプログラムがその後どのような具体的なライセンス対象のプログラム(例えば自分の改変部分がどのような態様で含まれているのか)として流通しているのか認識しないままライセンスしている事態も生じ得る。

(2) GPLにおける許諾対象となる行為

GPLでは複製、改変、頒布の許諾の規定を置いている(GPLはじめに、同第0条)。これらは著作権に関係する規定といえる。なお、契約の対象は上記のみであるとされる。また実行自体には制限がないとされる(GPL第0条)。

(3) GPLにおけるライセンスの条件

GPL第1条では複製、頒布の条件として①著作権表示、②無保証(保証の否認)表示、③GPL添付などを挙げる。

GPL第2条では改変・複製・頒布の条件として、上記条件に加え、①変更と変更日時の告示、②第三者への無償利用許諾、③著作権表示、無保証(保証の否認)表示、④頒布条件などの印刷、画面表示による表示、などを挙げる。

GPL第3条では派生物も含むGPLの対象となるプログラ

ムを、オブジェクト・コードないし実行形式で複製または頒布する場合、ソース・コードの頒布などを要求する。

そのほかにもライセンス条件があるが省略する。

(4) GPLにおける改変に関するライセンス適用条件

GPLでは改変に関するライセンス適用条件として以下の通り記載（GPL第2条）し、「全体としての改変された著作物に適用されるが、別の独立した著作物には適用されない」としている。

「以上の必要条件は全体としての改変された著作物に適用される。著作物の一部が『プログラム』から派生したものではないと確認でき、それら自身、別の独立した著作物であると合理的に考えられるならば、あなたがそれらを別の著作物として分けて頒布する場合、そういった部分にはこの契約書とその条件は適用されない。しかし、あなたが同じ部分を『プログラム』を基にした著作物全体の一部として頒布するならば、全体としての頒布物は、この契約書が課す条件に従わなければならない。というのは、この契約書が他の契約者に与える許可は『プログラム』丸ごと全体に及び、誰が書いたかは関係なく各部分のすべてを保護するからである。従って、すべてあなたによって書かれた著作物に対し、権利を主張したりあなたの権利に異議を申し立てることはこの節の意図するところではない。むしろ、その趣旨は『プログラム』を基にした派生物ないし集合著作物の頒布を管理する権利を行使するということにある。また、『プログラム』を基にしていないその他の著作物を『プログラム』（あるいは『プログラム』を基にした著作物）と一緒に集めただけのものを1巻の保管装置ないし頒布媒体に収めても、その他の著作物までこの契約書が保護する対象になるということにはならない」。

(5) GPLが改変・頒布条件に関して示す「別の独立した著作物」の基準について

ある著作物がGPLの対象となるプログラムの複製物か、2次的著作物か、それとも独立した別の著作物かどうかは、著作権法の評価により決定される。仮にGPLがその基準を示していたとしても、これらに関する著作権法上の評価と

は直接関係ないといえよう。

GPLの対象となるプログラムの複製物か、2次的著作物に該当するものについては、GPLの対象となるプログラムの著作権が及ぶので、「契約と著作権の論理的な関係」（pp. 66-67）で述べたように、GPLを適用させる前提条件を持っているといえる。すなわち、対象となるプログラムの改変・複製・頒布を伴う行為（著作権の及ぶ行為）については、GPLの条件の下に、改変・頒布を許諾するとの枠組みを当てはめることは可能と考えられる。

しかし、別の独立した著作物のみの複製頒布については、この枠組みは使えないことになろう。また、GPLの対象となるプログラムとともに使えるプログラム（あるいはGPLプログラムのためのプログラム）であっても、別の独立した著作物であれば、そのプログラムだけを（GPLの対象となるプログラムの改変・複製・頒布を伴わずに）頒布することについては、同様にGPLを適用させる枠組みで扱うことはできないと考えられる。なぜなら対象となるプログラムの改変・複製・頒布など、著作権の及ぶ行為が前提にないと考えられるからである。

（8部 2.2「契約と著作権の論理的な関係」 吉田委員）

■準拠法と解釈の基準

GPLに関し、何らかの法的問題が発生した場合、いずれの国の法律を準拠法とするかという厄介な問題が生じる。ここでは、準拠法についての基本的な考え方を整理する。

前提

- 訴訟が日本の裁判所に提起される場合を前提とする。
- 行為または結果の一部が日本で発生していることを前提とする。例えば、日本国内で複製・翻案・送信・受信などがなされているものとする。
- GPLでもそのFAQでも、準拠法の指定に関する規定も言及もなされていない。

参考となる判例など

①カード・リーダー事件

日本国内の行為が米国特許の侵害に当たるとして、差止

および廃棄請求、損害賠償が求められた事件である。

[東京地判平11.4.22]

- 差止及び廃棄請求：規定がないので正義及び合目的性の理念に基づき判断する。特許権に基づく差止及び廃棄請求に関しては、当該特許権が登録された国（米国）の法律を準拠法とする。米国法は積極教唆及び寄与侵害について域外適用を認めるが、法例33条（公序良俗違反）により適用しない。
- 損害賠償：不法行為であると性質決定し、法例11条1項により、「原因たる事実の発生した地」の法律として、日本法を適用する。しかし、米国特許権は、わが国の不法行為法によって保護される権利に該当しない。

[東京高判平12.1.27]

- 差止及び廃棄請求：属地主義の原則から、外国特許権に基づく差止及び廃棄を内国裁判所に求めることはできない。準拠法決定の問題は生じる余地がない。
 - 損害賠償：地裁と同じ。
- [最判平14.9.26]
- 特許権に基づく差止め及び廃棄請求の準拠法は、当該特許権が登録された国の法律である。
 - 米国特許権を適用して、米国特許権の侵害を積極的に誘導する我が国内での行為の差止め又は我が国内にある侵害品の廃棄を命ずることは、法例33条にいう「公の秩序」に反する。
 - 米国特許権の侵害を積極的に誘導する行為を我が国で行ったことを理由とする損害賠償請求について法例11条1項にいう「原因たる事実ノ発生シタル地」は米国である。
 - 米国特許権の侵害を積極的に誘導する行為を我が国で行ったことは、法例11条2項にいう「外国ニ於テ発生シタル事実カ日本ノ法律ニ依レバ不法ナラサルトキ」にあたる。

②ベルヌ条約5条(2)項

- (1) の権利の享有及び行使には、いかなる方式の履行をも要しない。その享有及び行使は、著作物の本国における保護の存在にかかわらない。したがって、保護

の範囲及び著作権者の権利を保全するため著作者に保障される救済の方法は、この条約の規定によるほか、専ら、保護が要求される同盟国の法令の定めるところによる。

- 道垣内正人「国境を越えた知的財産権の保護をめぐる諸問題」ジュリスト1227号52頁

「この規定は準拠法として保護国法（すなわち利用行為地法）を定めたものであると解するのが妥当であるように思われる（保護国法説）」

③法例11条

- 1 事務管理、不当利得又ハ不法行為ニ因リテ生スル債権ノ成立及ヒ効力ハ其原因タル事実ノ発生シタル地ノ法律ニ依ル
- 2 前項ノ規定ハ不法行為ニ付テハ外国ニ於テ発生シタル事実カ日本ノ法律ニ依レバ不法ナラサルトキハ之ヲ適用セス
- 3 外国ニ於テ発生シタル事実カ日本ノ法律ニ依リテ不法ナルトキト雖モ被害者ハ日本ノ法律カ認メタル損害賠償其他ノ処分ニ非サレハ之ヲ請求スルコトヲ得ス

④法例7条

- 1 法律行為ノ成立及ヒ効力ニ付テハ当事者ノ意思ニ從ヒ其何レノ国ノ法律ニ依ルヘキカヲ定ム
- 2 当事者ノ意思カ分明ナラサルトキハ行為地法ニ依ル

⑤法例9条

- 1 法律ヲ異ニスル地ニ在ル者ニ対シテ為シタル意思表示ニ付テハ其通知ヲ發シタル地ヲ行為地ト看做ス
- 2 契約ノ成立及ヒ効力ニ付テハ申込ノ通知ヲ發シタル地ヲ行為地ト看做ス若シ其申込ヲ受ケタル者カ承諾ヲ為シタル当時申込ノ発信地ヲ知ラサリシトキハ申込者ノ住所地ヲ行為地ト看做ス

GPLの準拠法

GPLに関する法律問題が日本の裁判所に提起された場合の準拠法については、その法律問題の内容により次のよう

になると考えられる。

① ライセンス契約の違反に伴う請求

法律問題が著作権侵害関連以外の点にある場合、（例えば、ソフトウェアに機能上の不具合があった場合の損害賠償請求に関し、損害不担保の規定の効力が問題となる場合）法例9条および7条から、申し込みの発信地が行為地ということになる。従って、ソフトウェアの著作権者がそのソフトウェアをGPLの対象とした場合、その意思を対外的に（ライセンシーとなる者に対し）表明した場所が申し込みの発信地となり、その国の法律が準拠法となろう。

② 著作権侵害に伴う差止請求

ベルヌ条約5条（2）項の解釈から、保護国法たる日本法となる。

③ 著作権侵害に伴う損害賠償請求

著作権侵害に基づく損害賠償請求は不法行為であると性質決定されるから、法例11条により「原因タル事実ノ発生シタル地」の法律が準拠法ということになる。日本で侵害行為が行われたものと評価できれば日本法が準拠法となる。

FAQから

なお、FAQには「GPLでカバーされたプログラムのソース・コードを使うに際して『フェア・ユース』の権利があるのでしょうか」という質問に対し「ある」としつつ、何が「フェア」かについては国によって違うということが述べられている。これは、同じ著作権の問題が発生しても、適用される著作権法によって違法かどうか影響されることを認めているものと解し得る。

Derivative work under copyright lawについて

GPLにおける「Derivative work under copyright law」の意義については、この語が当該紛争で問題となる論脈によって異なってくるように思われる。例えば、直接、著作権侵害の問題ではなくライセンス条件に関してこの語が問題となる場合であれば、前述のように発信地法国の法律に

従った意義が与えられるものであろう。米国内における当事者の意思であれば米著作権法によるということだろう。他国においては、これに対応する当該国の著作権法上の概念（日本法なら二次的著作物に関する規定と判例法）に従って、意味が与えられるものであろう。

著作権侵害が直接問題になる場合についても、準拠法たる著作権法のその国の「対応する語の」解釈によるということになると思われる。

（8部 2.2 「準拠法と解釈の基準」 梶山委員長）

■ 契約の成立

GPLの問題を議論する場合、その課題が契約の問題として議論される場合と、著作権の問題として議論される場合の双方があることは、これまで見てきた通りである。しかし「派生物」の範囲をはじめ、契約上の判断と著作権法上の判断が擦れ違う場合、そもそもその契約が有効に成立していたと言い切れるのか、その「契約の成立性」が厳しい論点として提示される。このため、ここではGPLにおける「契約の成立」について、整理を行う。

GPL (GNU general public license) Version 2について検討する。なお、日本語訳には初期に翻訳された「GNU一般公有使用許諾書」と最近翻訳された「GNU一般公衆利用許諾契約書」があるが、「GNU一般公有使用許諾書」では英語版と条項番号にずれがあるため、以下議論では英語版の条項番号に従う。

GPLの日本語訳が「一般公有使用許諾書」とされていたことから分かるように、GPLが契約書であるのかは必ずしも明確ではない（なお、FAQでは「許諾契約書」と訳されていて不統一である）。また、内容的にも「はじめに」の部分にフリー・ソフトウェアの概念、運動理念が宣言されており、その最後の部分に「改変・複製・頒布に対する正確な条項と条件を次に示します」と記載され、その後に「GNU一般公有使用許諾の下での改変・複製・頒布に関する条項と条件」とのサブ・タイトルがあり、以下に条件が記載されている。

一般に、GPLはフリー・ソフトウェアの改変・複製・頒布に関してのライセンス（使用許諾）契約であると解され

ている。しかし、この体裁および内容からは、GPLは対象となるソフトウェアの改変・複製・頒布に関しての著作権者の意向を宣言したものにとすぎず、GPLは契約ではないと解することも可能である。

GPLを宣言書であり契約は成立しないと解した場合、GPLはそこに記載されている条件を守る限り著作権者は著作権を行使しないと公に宣言したものであり、これに反する権利行使は権限濫用、禁反言などの一般法理によって許されないと解することができる。さらに、この場合の適用法は日本の著作権法および民法であり、仮に条件の中に日本法では著作権者に認められていない権利を認める趣旨の条項があったとしても、契約としては成立していない以上、その条項は条件としては法的拘束力を持たない。従って、当該条項違反を理由とする著作権行使も単に条項違反を理由とするだけでは原則として許されないことになる。その場合には、条項の持つ合理性が吟味され、当該条項がGPLの宣言の趣旨から合理的に要請されるものであり、それに反する行為を差し止める必要性が十分に認められる場合にのみ権利行使が許されることになるものと考えられる。例えば、GPLソフトウェアを改変して頒布した場合にソース・コードを引き渡す義務が条件とされているが、著作権法ではこのような要求をする権利は著作権者にはない。そこでこの条件に違反した場合に著作権者が複製・頒布を禁ずることができるかが問題となるが、GPLの趣旨によればソース・コードの引き渡しはGPLの基本的理念の実現のため不可欠なものであり、この条件は合理性を有すると考えざるを得ないため、権利行使は許されることになるものと思われる。他方、GPLでは明確でないプログラムの公衆送信については「複製や頒布、改変以外の活動はこの契約書ではカバーされない。それらはこの契約書の対象外である」(第0条後段)とされているものの、複製・頒布が許されて公衆送信が許されない合理的理由を見いだすことは困難である以上、公衆送信を理由とする権利行使は許されないものと考えられる。

なお、GPLを契約でないと考えた場合、対象となるプログラムが軋々と流通した場合であっても契約の成立を考える必要はないので、GPLであることを認識していた利用者、

認識していない利用者の区別なく、すべての利用者に対して同一の法律問題を考えればよいことになる。

上記の通り、GPLは契約ではないと考えることも可能である。しかし、GPLは随所にソフトウェアの改変・複製・頒布を許諾する旨の記載がなされていること、プログラムの実行については制約がないが、改変・複製・頒布以外の行為は許諾の対象としないとされていること(第0条後段)、利用者は一定の条件に従ってプログラムを複製・頒布できるとされていること(第3条)、GPLに従わない頒布などの行為があった場合には利用者に許諾している権利を自動的に消滅させるとされていること(第4条)、GPLを受け入れなければプログラムの改変・再頒布などは禁じられ、改変・再頒布を行えば条項と条件をすべて受け入れたものとされていること(第5条)、その場合には使用許諾を受けたものとされていること(第6条)、条項の一部が無効とされた場合にも他の条項は有効として適用されるように意図されていること(第7条2段)、さらにはFAQにおいて「GPLは著作権に基づくライセンスである」と回答されていることなどを考えた場合、GPL作成者の意図は利用者にGPLプログラムの改変・複製・頒布を許諾する許諾契約であると考えていたことは間違いなく、GPLは使用許諾契約であると考えざるを得ないであろう。

GPLが使用許諾契約であるとした場合、3つの問題が生じる。大前提として、そもそも契約はどこの国の法律によって認められるのかという問題と、仮に日本法によるとした場合、意思の合致としての契約の成立はあるのかという問題、著作権法に反する条項が存在するときに拘束力を有するのか(契約条項が著作権法をオーバーライドするか)、という問題である。

①GPL文書自体の作成者はFSFであり、米国の法人である。また、GPLには準拠法の記載はない。そこで、GNUプログラムにGPLが添付され、日本法人Aがそれ取得し利用しようとする単純なモデルを考えた場合、当事者の意思が準拠法を米国法(いずれかの州法)にするか日本法にするか判明しないときは、(GPLの添付を契約の申し込みと観念できれば)法例7条、9条により申し込みの発信地で

ある米国法が準拠法とされることになる。その場合には、契約が成立するか否かは当該米国法（州法）の解釈問題となり、契約の効力も当該米国法によって決せられることになる。

次に、日本法人Aは当該GNUプログラムを利用せずにGPLが添付されたまま別の日本法人Bに移転した場合、さらには転々流通して日本法人Xが取得した場合はどうか。これはいわゆるシュリンク・ラップ契約の場合と類似する。シュリンク・ラップ契約において最終ユーザーが封を切ったときにメーカーとユーザー間で契約が成立すると理解するのであれば、GPLも利用者が利用した段階で権利者との間で直接、契約が成立するということになる。ところが、最終ユーザーXの前の段階で、AなりBがGNUプログラムを改変した場合にはこのように考えることはできなくなる。日本法人Aの改変によって2次的著作物が作成された場合、当該著作物の著作権者はAである。そこで、改変物に添付されているGPLは日本法を前提とした契約の申し込みということになる。しかし、当該プログラムの原プログラムについてのGPLは米国法の適用となる。このように考えると1つのGPLについてある場面では米国法を準拠法とし、ある場面では日本法を準拠法として契約の成否や法的効力を考える必要が生じる。Linuxのように世界中のソフトウェア・エンジニアが開発に協力している場合にはあらゆる国の法律を準拠法として考えなければならなくなる。

果たして、このような事態が生じることを前提にGPLは作成されたのであろうか。GPLに違反した場合、著作権者は著作権に基づいて複製などの差止請求を行うことになるであろう。この差止請求は保護を求める国の著作権を根拠として行われることになる。GPLは改変・複製・頒布を許諾するものであり、契約違反により許諾が取り消されたときに差し止めなどの請求権の根拠となるのは保護を求める国の著作権だからである。日本において米国著作権を根拠に差し止めを認めることはできない。すなわち、救済を考えた場合に外国法を準拠法として契約の成立や効力を考えることは意味がない。

このように考えると、そもそもGPLについて当事者の意

思が判明しないとした前提を見直す必要がある。GPLに準拠法の規定がないことは、意思が判明しないことではなく、消極的に「GPLを承諾する利用者の意思に従う」という意思であると解することはできないだろうか。米国で起草された契約書に準拠法の規定がないということは、積極的に規定を入れなかったのであり、準拠法を米国法とする意思ではなかったと考えられないであろうか。他のGPL、例えばMOZILLA GPLでは準拠法をカリフォルニア州法と規定しており、準拠法の規定を加入することは容易だからである。なお、欧州委員会のレポート「POOLING OPEN SOURCE SOFTWARE」では契約に明記されていない場合には一定の要件の下にユーザー国の法律が準拠法になるとする。このように、当事者の意思が不明の場合に準拠法をどの国の法律にするかは国際的に統一された準則があるわけではないものと思われる。GPLに準拠法の定めがないことから、国によって準拠法の決定が異なることを容認していることになり、それは利用者の意思に従う趣旨であると解することに何の支障もない。

以上の通り、筆者は、GPLの準拠法については利用者の意思に任されていると解するが、これはあくまでも私見である。なお以下は、準拠法は日本法であるとしての検討である。

②GPLの契約成立問題に関しては、権利者と利用者とが明示に合意をするということではなく、条項についても交渉や変更の可能性がないという点で、いわゆるシュリンク・ラップ契約、クリックオン契約に類似するものと理解されることが多い。しかし、これらの契約とGPLは状況が異なる。シュリンク・ラップ契約やクリックオン契約はほとんどの場合、プログラムの使用に関する契約であるが、GPLはプログラムの使用について制約はないとし（第0条後段）、改変・再頒布するのでもなければGPLを受け入れる必要がないとされている（第5条）のであって、使用に関する許諾契約ではない。

またシュリンク・ラップ契約の場合には、購入し開封した後でなければ契約条項が判明しないことが問題とされ、クリックオン契約においては画面上の条項が細かく、かつ

分かりにくい上、現実に画面上の契約条項を読むユーザーはほとんどいない点が問題とされた。そしてこれらはいずれも消費者がユーザーとして登場する場面であった。これらに比べて、GPLは公開されていて十分に事前吟味は可能であり、改変し再頒布する利用者は消費者ではない。むしろ、ソフトウェア・メーカーが既に存在するプログラムを改変し再頒布するために、提示された契約条項を吟味する場面と近い。提示の場面で契約相手方が現実には現れず、提示された条件を交渉できない（諾否のいずれかの選択しかない）というものにすぎない、と考えることもできる。

なお、GPLにおいては条項中の一部が無効とされた場合にも他の部分が有効に適用されるよう意図されている（第7条2段）。これは交渉の余地がないことから条項の一部が法令や判決により無効とされた場合に、全体として無効になることを避けるためにさまざまな言い方で条件を繰り返し規定していることを意味する（そのため、条項は同じ意味の繰り返しが多く、分かりにくくなっているともいえる）。

このように考えると、GPLの契約についてはシュリンク・ラップ契約やクリックオン契約における法的構成、特に、条項を十分に理解せずとも意思の実現として契約は成立するののかという議論は、必ずしも当てはまらないことになる。GPLについて利用者は、契約条項を十分に吟味した上でプログラムを利用するか否かの決定がなされることができるところからである。従って、契約条件が公開されていてプログラムとともに明示されており、それらが入手できる状況にあることは利用者に対する申し込みであると解し、明示の意思の合致がなくとも「承諾の意思表示と認むべき事実」があったときに契約が成立する（意思の実現による契約の成立、民法526条2項）と考えることに支障はないものと思われる。具体的には、複製や改変、あるいは頒布に着手したときに契約が成立したことになる。

なお、準拠法を日本法とした場合には契約条項の解釈も日本法によることになる。従って、条項中の「派生物」という概念も日本の著作権法を前提として「二次的著作物」となるもの、ならないものを区別して把握した上で検討する必要が生じる。今後、GPLに準拠法として米国法などの外国法が記載された場合には別の議論が必要になる。先に

述べた通り準拠法が米国法とされると、そもそも契約の成立自体が米国法（州法）の解釈で成否が決まることになり、成立したとすれば、条項の解釈も米国法によることになる。従って「派生物」も米国著作権法の意味を前提として概念される。ただし前述の通り、差止請求などの著作権法に基づく請求は日本法によることになる。

③条項が著作権法に規定のない義務を利用者に負わせるものである場合、一般的には契約の内容として有効となる。従って、改変したプログラムを頒布する場合にソース・コードの添付を要するとした条項も有効である。著作権法の規定に反する場合の効力については、当該条項が法のいかなる条文に違反するものであるかによって異なるといわざるを得ない。一般論としては強行法規性を持つ条文に反する条項は無効となり、法的拘束力を持たないといえるが、強行法規か否かは条文自体には記載がないため、それぞれ個別に判断することになる。

（8部 2.2 「契約の成立」 小川委員）

■公衆送信権、著作者人格権

ここでは、課題が著作権侵害の問題に特定されている場合を念頭に置き、具体的にどのような行為が著作権法上禁止される行為に該当するか、GPLで定める文言との整合性を見つつ確認を行う。

公衆送信権

GPLにおいては「Activities other than copying, distribution and modification are not covered by this License ; they are outside its scope」（第0条2項）とあり、自動公衆送信（automatical public transmission）や送信可能化（making transmittable to the public）のような公への伝達（communication to the public）については、何ら言及されていない。米国を含むほとんどの国々では、自動公衆送信権および送信可能化権自体は著作（財産）権に含まれていないことから、これまで特段の問題が発生しなかった。

しかし日本の著作権法は、自動公衆送信権および送信可能化権自体を著作（財産）権の1つとして規定する異例の

法体系を有していることから、GPLの下で公開されたプログラムについて、日本法が適用される領域内で自動公衆送信ないし送信可能化された場合、著作権侵害に当たるのではないかという問題が生ずる。

これに対しては、GPLを文理解釈して日本法が適用されるような態様で自動公衆送信または送信可能化を行った場合には、著作権侵害に当たるとする考え方もあり得る。しかし今日では、特にGPLの下で公開されるようなプログラムについては、(著作権侵害に問われるという法的リスクを論理的には負うものの)日本においては、インターネットなどを介して公衆に提供されるのが通常であり、そのような方法によりGPLプログラムを改変して作成したプログラムの提供ができないという結論は妥当性を欠くと思われる。

他方、米国においては、著作物のネットワーク配信を「頒布 (distribution)」に含める旨の判例があり、GPLはこの判例を前提に起草されていることから、GPLにおける「distribution」にはネットワーク配信を構成する行為 (自動公衆送信および送信可能化) が含まれるのだとする考え方もあり得る。これに対しては、最初にGPLの下でソース・コードを公開した著作権者が、米国法を前提としていられる場合はそのような解釈でもよいが^{註8)}、日本において専ら日本語のみを解するプログラマーが、GPLの日本語訳のみを見て自ら作成したプログラムをGPLの下で公開したときに、そのプログラマーが米国の判例法により拡張された意味で「頒布」という語句を用いたのだと意思解釈することは擬制的すぎるのではないかという批判もあり得るところである。

もっともこれに対しては、日本語版のGPLにおいては英文文書こそが正式文書であって日本語版には法的効力はないと記載されており、GPLを使用する以上、「頒布」という語句はすべて米国法上の「distribution」の意味で用いられているとみることができるのだ、という反論もあり得る。これに対しては、確かに「頒布」という語句は英語の「distribution」の意味で用いられているとみることまでは許容されるとして、米国においては判例法上、意味範囲が拡張された「distribution」の意味で用いられているとまでみることには無理があるのではないかと、との再反論もあり得る。

著作者人格権

▶ 総論

日本の著作権法においては、著作 (財産) 権とは別に著作者人格権というものが著作者の権利として定められており、しかも、それは一身専属権であるから譲渡も放棄もできないものとされている^{註9)}。そのため、日本において作成される著作権譲渡ないし利用許諾契約書においては、少なくとも著作物を改変して使用することがあらかじめ想定される場合には、著作者人格権不行使特約を付帯させるのが通常である^{註10)}。しかし、基本的に著作者人格権という概念がない米国法を前提として起草されたGPLには、当然のことながら著作者人格権の行使に関する明文の規定はない。

この点について、GPLに従う限りにおいては自由な改変を認めるGPLの趣旨から当然に著作者人格権 (少なくとも同一性保持権) を行使しない旨を要約したのだとする考え方もあり得るわけではない。しかし、著作者人格権不行使特約は契約書上明文で記載されている場合であってもその有効性に疑問が持たれているので、明文の規定がない場合に意思解釈によって著作者人格権不行使特約の成立を認めてしまうのは大いに疑問である。

なお、仮にGPLの下にプログラムを公開したプログラマーについて著作者人格権不行使特約の成立を認めた場合、どのような法的効果が生ずるのかという点について若干触れる。著作者人格権不行使特約はあくまで債権的契約にすぎないので、プログラマーがこれに違反して著作者人格権を行使した場合には、不行使特約の相手方がこれにより生じた損害の賠償を請求できるにすぎないとするのが原則である。

すなわち、例えば、当該改変により同一性保持権を害されたとしてプログラマーがプログラムの翻案物の販売などの差し止めを請求してきた場合には、不行使特約の存在を主張して、上記販売の差し止めにより相手方が被った損害の賠償を請求することはできるが、販売などの差し止め自体は認められてしまうのが原則である。差し止め請求権に対する抗弁の1つとして不行使特約を位置付けるためには、特約の効力として差し止め請求権を主張させないことができると構成するよりは、不行使特約を結んだ経緯と行使する著作

者人格権の内容および目的などを総合的に勘案して禁反言の法理を適用するのだと構成する方が「人格権は放棄できない」というテーゼと矛盾する度合いは小さいように思われる。

▶ 氏名表示権

GPLでは「an appropriate copyright notice (適切な著作権表示)」を行うことが義務付けられている。しかし日本の著作権法の下では、著作権者が誰であるかではなく、著作者が誰であるのかを表示することが求められている(著作権法19条1項)。もちろんそのプログラムに著作者名が表示されていない場合は、原則としてそのまま著作者名を表示しなければいいのであるが(同条2項)、著作者が別段の意思表示、例えば特定のペンネームを著作者名として表示せよと主張した場合、それ以降はそのペンネームを著作者名として表示しなければならないということになる。もっとも、これはGPL上の義務ではないから、著作者の指示に従って著作者名を表示しなかったからといって、GPL違反になるわけではない。GPL上は、あくまで適切な著作権表示を行うことが義務付けられているのである。

▶ 同一性保持権

GPLでは「You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or works…」として、プログラムを改変する、または改変したプログラムを複製し、頒布することを許容するものとしている。従って、日本法の下においてもGPLに従う限りにおいては、GPLの下で公開されたプログラムを改変して、複製・頒布することは、翻案または複製、もしくは譲渡(自動公衆送信・送信可能化が含まれるかは「公衆送信権」(pp. 73-74) 参照)については著作権者の許諾がある以上、著作権侵害には当たらない。

しかし日本の著作権法においては、著作(財産)権としての翻案権のほかに、著作者人格権としての同一性保持権という権利が(著作権者の権利ではなく)著作者の権利と定められていることから、プログラムを改変して頒布する

ことが同一性保持権侵害に当たらないかどうかという点も吟味する必要がある。

なお、この点は、著作者＝著作権者の場合と、著作者≠著作権者の場合とで分けて論ずる必要がある。

もっとも、プログラムに関しては、同一性保持権侵害の成否が問題となるケースはあまり多くない。まず、バグの修正やバージョンアップのための改変は、そもそも同一性保持権侵害には当たらない(著作権法20条2項3号)。また、著作物を他の著作物などと併せて1つの著作物に組み入れてしまうことは同一性保持権が禁止しようとする「改変」には含まれない。当該プログラムのうちの特定のサブルーチンなどを取り出して他のプログラムに組み入れる場合は、形式的には著作物の「切除」を行っているともいえそうだが、「切除」という態様での「改変」は極めて限定的に解釈されており、サブルーチンなど、ある程度まとまった意味を独立して持ち得る部分を取り出すという行為が同一性保持権侵害と認定される可能性は低い。

① 著作者＝著作権者の場合

「You may modify your copy」という語句は「翻案してもよい」という意味と「改変してもよい」という意味の双方を兼ねており、翻案の許諾を意思表示であると同時に、改変についての事前の包括的同意の意思表示でもあるとの考えもあり得る。著作権情報センターが著作権法20条1項を英訳するに当たって「改変」の訳語として「modification」という語句をあてている(「翻案」には「adapt」の語があてられている)ことから分かる通り「You may modify your copy」を「改変してもよい」という意味に受け取るのは自然ではある。

ただし、これに対しては、GPLは著作者人格権としての同一性保持権というものが制度的に存在しない米国法を前提に作成されているので、GPLにいう「You may modify

注8) 派生著作物の著作者がGPLを遵守している場合には、派生著作物に関する許諾の範囲は、原著物に関する許諾の範囲と同じであると認定して差し支えないであろう。

注9) ただし、放棄については田村解説など異説あり。

注10) もっとも、著作者人格権不行使特約の有効性およびその効力については議論があるところである。

your copy」との語句は「翻案」という、著作物の利用態様を許諾するという意味しか持ち得ないのだという批判がある。さらに改変についての事前の包括的同意の意思表示だとすると、いつでも一方的に同意を将来的に撤回できることになり、かえって法的安定性を欠くのではないかとの批判もあり得る。もっとも前述した通り、プログラムの著作物については同一性保持権侵害が成立する余地がもともと小さい上、少なくとも翻案について許諾している以上、翻案に伴う必然的な改変は著作者の意に反しないものというべきであって、著作物の本質に触れるような重大な改変のみ同一性保持権侵害が問題となると解されている（加戸逐条講義p.164）。すると、オープンソースないしフリー・ソフトウェアを利用するという場合には、著作物の本質に触れるような重大な改変というのは、実際にはほとんど存在しないのではないかと考えられる。

②著作者≠著作権者の場合

日本の著作権法では、著作者人格権は一身専属権なので著作者に留保されるが、著作（財産）権は財産権なので転々と譲渡され得るのであり、譲渡された場合には著作者≠著作権者となる。GPLの下でプログラムを公開するに当たっては、少なくとも複製権、翻案権、譲渡権（および送信可能化権を含む自動公衆送信権）を有することが必要となるから、プログラムを公開することができるのは、実際の問題としていえばほぼ著作権者に限られるであろう。もっとも、翻案権が著作者＝譲渡人に留保されている場合（著作権法61条2項）には、著作者＝譲渡人の許諾なしには、プログラムをGPLの下で公開することはできないであろう。

この場合、GPLのライセンサーは著作者ではないから「You may modify your copy」との規定がGPLに記載されているからといって「著作者が」改変についての事前の包括的同意を行ったとみることはできない^{注11)}。もっとも、プログラムの改変については同一性保持権侵害が問題となる余地がもともと小さいことは既に述べた通りであるし、少なくとも翻案権も含めて著作権を譲渡した場合には、翻案を許諾した場合と同様に、翻案に伴う必然的な改変は著作

者の意思に反しないと解されることから、実際上の問題はあまり多くないとも予想される。

▶名誉・声望を害する方法で利用されない権利

日本の著作権法は「著作者の名誉又は声望を害する方法によりその著作物を利用する行為は、その著作者人格権を侵害する行為とみなす」（113条5項）と定め、著作者に名誉・声望を害する方法でその著作物を利用されない権利を認めている。日本の著作権実務は「名誉又は声望を害する方法」により利用されているかどうかということを比較的広く解釈する傾向がある^{注12)}。

GPLは、著作者の名誉・声望を害する方法でその著作物を利用されない権利については何ら触れていないから、GPLの下でプログラムを公開した者が、特定の事業者^{注13)}などがそのプログラムを業務システムに組み込むのを、名誉・声望を害する方法での利用に当たるといって差止ないし損害賠償を請求してくることも論理的には考えられないわけではない。これに対しては、プログラムは著作物とはいえ機能的な側面が強く、プログラムと利用者との間の思想的ないし芸術的な結び付きはもともと極めて希薄なので、社会的にネガティブな評価を受けがちな業者によって、あるいは社会的にネガティブな評価を受けがちな業務に利用されたからといって、そのプログラムの著作者（プログラマー）の名誉・声望が害されるということは、通常考えにくいとの批判を加え得るといえる。

（8部 2.2「公衆送信権、著作者人格権」 小倉委員）

■特許問題

GPLにかかわる特許の問題を概観する。というのも、GPL自体は主として著作権の扱いを明確化することにより、ソフトウェアの改変・流通が自由にできる環境を保証しようとしており、特許に関する扱いを特定することは目的としておらず、不明なケースが多いからである。

ここでは、ライセンシーが特許を有している場合と、第三者に特許権がある場合とに分けて、その対応の整理を試みる。

初めに

GNUのGPLは、コピーレフト (copyleft) の理念を、著作権 (copyright) の利用許諾という方法により実現することを目的とした、利用許諾条項のパッケージである。

換言すれば、GPLは、copyleftの理念の実現のために必要な条件を、copyrightの行使を媒介にした利用許諾条件のパッケージの設定を通じて実現しようとしているものである。

このため、GPLにおいてcopyleftの理念実現のための媒介手段は著作権である。

それでは、ソフトウェアにおいて、著作権とともに重要と考えられている特許権は、GPLにおいて、どのように位置付けられ、どのように取り扱われているのであろうか。

この点について結論から述べれば、GPLの作成者は自ら特許権を保有することにより、特許権の実施許諾を媒介にして、上記理念 (copyleft) を実現しようとはしていない。

すなわち、GPLにおいては、copyleftの理念の実現のための媒介手段とされているものは著作権であり、特許権ではない。

このためGPLにおいては、特許権はライセンス利用許諾条項中に取り入れられているものではなく、その外側に位置付けられているものである。従って、特許権は、GPLに基づくライセンス対象のソフトウェアにとっては、基本的には外部から攻撃を加えてくる可能性のある権利と位置付けられることになる。

そこで、以下においては、①ライセンシー以外の第三者が特許権を保有している場合、②ライセンシーが特許権を保有している場合、の各場合に分けて検討していくこととする。

①ライセンシー以外の第三者が特許権を保有している場合

ここでは、GPLに基づくライセンス契約の当事者とはなっていない第三者が、ソフトウェアに関する特許権を保有しており、この特許権をGPLに基づくライセンス対象のソフトウェアに対して行使してきた場合について検討をする。

この場合に当該第三者は、GPLによるライセンス契約の当事者ではないから、特許権行使を行う上で、GPLのライ

センス条項には何ら縛られるところがない。

従って当該第三者は、GPLによるライセンス許諾の対象となっているソフトウェアの開発者、頒布者、使用者に対して、一般の特許権行使の場合と全く同様に、特許権行使を行い得ることになると考えられる。

このためGPL中のライセンス条項も、このことを前提として上で、以下のように規定している。

「if, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.」(第7条第1パラグラフ)

〔裁判所の判決、または特許侵害の申し立て、または(特許問題に限らない) 何らかの理由の結果として、あなたに課せられた条件が本使用許諾と相入れないものであったとしても(裁判所の命令、契約、その他によるものであれ)、本使用許諾の条件が免除されるものではありません。本使用許諾による責務と、その他の何らかの関連責務を同時に満たす態様で頒布することができないならば、あなたは「プログラム」を全く頒布してはいけません。例えば特許権の

注11) 著作権を包括的に譲渡したことをもって、少なくとも譲受人の同意の下で行われた改変については事前に包括的に同意したものと見なすことは、一身専属権である同一性保持権の譲渡を事実上認めることになるから、許されないというべきであろう。

注12) 例えば、コマーシャルへの楽曲使用は特定の商品または企業イメージと密接に結び付くから著作人格権を侵害するものであるという前提で動いているといわれている(安藤和宏、「よくわかる音楽著作権ビジネス 実践編 2nd Edition」p. 75)。

注13) 特に、社会的にネガティブな評価を受けがちな業者が想定されるが、コマーシャルへの楽曲使用の場合から類推すると、それに限られない。

内容が、あなたから直接または間接に複製を受け取ったすべての人に使用料のないプログラムの再頒布を許さないものであれば、あなたがかかる特許上の要請と本使用許諾の両方を満足させる方法は「プログラム」の頒布を完全に断念することだけです。」(翻訳)

「If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.」(第8条)

「『プログラム』の頒布・使用が、ある国において特許または著作権で保護されたインタフェースのどちらかで制限される場合、『プログラム』を本使用許諾下においた著作権保持者は、その国を除外する旨の明示的な頒布地域制限を加え、それ以外の(除外されない)国に限定して頒布が許されるようにすることができます。そのような場合、その制限を本使用許諾の本文にあたかも書かれているかのように本使用許諾の中に組み入れられるものとします。」(翻訳)

この引用部分は、GPLに基づきライセンス許諾される対象となるソフトウェアのライセンシーは、第三者からの特許権の攻撃に対して、GPLに基づくライセンス許諾を受けていることを抗弁とすることができず、特許権の攻撃に対しては、非侵害の主張をすること以外には、反撃が困難、換言すれば、いわば「お手上げ」の状態にならざるを得ないことを前提としている。

このため、上記の帰結として、第三者からの特許権の攻撃が奏功すれば、GPLによるライセンス対象のソフトウェアの頒布も使用も不可能となってしまうことになる。

また、これとは別に、仮に、特許権を保有している第三者から特許ライセンスが得られ、ソフトウェアの頒布が可

能になった場合にも、GPL第7条第1パラグラフは、当該特許ライセンス条件が実施料の支払いを前提にしており、このためGPLに基づく利用許諾対象のソフトウェアの頒布においても、無償での頒布が困難となる場合には、当該ソフトウェアの頒布それ自体を禁止する旨を規定している。

同じくGPL第8条も、第三者からの特許権の攻撃によりソフトウェアの無償での頒布が困難という事態に立ち到った場合には、そのような事態が生じた特定国において、GPLに基づくライセンス対象のソフトウェアの頒布を中止することが可能であることを規定している。

上記の通り、GPLは第三者からの特許権の攻撃に対しては基本的に無力であり、上記で引用したGPL第7条、第8条に規定されている通り、消極的な対応を採ることを予定しているのみである。

以上に付言して述べれば、以下の通りである。

すなわち、GPLによるライセンス対象のソフトウェアは、オープンソース、すなわちソース・コードを開示することを前提にしているから、第三者において当該ソフトウェアの内容を調査することは、通常の実行ファイルの形態で提供されるソフトウェアの場合に比べると、より容易である。

従って、オープンソース・プログラムは、この点においても、第三者からの特許権行使に対して脆弱な構造を抱えていることを付言しておく。

以上の通りであるが、第三者からの特許権攻撃の対象が、Linuxのような社会のインフラになりつつあるソフトウェアの場合にも、上記がそのまま当てはまるのであろうか。

この場合には特許権行使のうちで、特に差止請求が問題になる。

すなわち、差し止めが命じられた場合の影響を考慮すると、差止請求については、権利乱用の法理その他の法理に基づき権利行使を制限することが、すべての場合ではないにしても、時として必要になることがあると考えられる。

もっとも他方で、損害賠償請求については、この点についてまで請求を制限することには、格別の根拠は見いだされない。

従って、この場合には、差止請求については、時として請求が制限されることがあり得るとしても、他方で損害賠

償請求については、これを制限すべきではなく、合理的なロイヤルティーの支払いが認められて然るべきものと考えられる。

②ライセンシーが特許権を保有している場合

GPLに基づきソフトウェアのライセンスを受けた者（ライセンシー）が、(a) もともと特許権を保有しており、GPLに基づくライセンス対象のソフトウェアが、当該特許発明の技術的範囲に属している場合、(b) ライセンスを受けたソフトウェアを翻案した場合に、この翻案を行った後の部分につき、当該ライセンシーがもともと特許権を保有しており、当該翻案部分が当該特許発明の技術的範囲に属している場合、(c) ライセンスを受けたソフトウェアを翻案し、この翻案時に新たに生まれた技術につき特許権を取得した場合、などの場合に、当該ライセンシーは、当該ソフトウェアの再頒布の際（この場合にはライセンサーとして再頒布することになる）に、有償にて再頒布することができるのか、あるいは当該ライセンシーと同様に当該ソフトウェアのライセンスを受けた別の第三者が、当該ソフトウェアを再頒布することに対して、特許権の行使を行うことが可能であるのかなどが問題になる。

GPLは、これらの点について、直接明示的に規律する条項は設けていないものと考えられる。

このことを前提とした場合には、GPLはこのような事態が生ずることを禁ずる旨を明示的には規定していないことになるから、上記ソフトウェアの有償での頒布などは、取りあえずは許されるのではないかと考えられる。

しかし、GPLの第7条第1パラグラフは、前記引用の通り、ライセンシーは、第三者から特許実施料の支払いを義務付けられた場合にも、当該実施料の支払いを、ソフトウェアの再頒布者に対して請求することができない旨を規定している。

従ってこのことからすると、ライセンシーにおいて、第三者に対して実施料の支払い義務を負っておらず、専ら自己の利益のためにのみ対価の支払いを請求するという上記の場合において、対価の支払い請求が当然に許されると解することは、GPL第7条第1パラグラフの場合と比べても、

バランスを失うことになる可能性があるとも考えられる。

よって、このことを前提にした場合には、上記の (a) ないし (c) のような場合にも、ライセンシーは、自らが特許権を保有していることを根拠にして、ソフトウェアの再頒布の際に対価を徴収することは許されないものとも考えられる。

また、同様に、別の第三者が当該ソフトウェアを再頒布することに対して特許権を行使することも、許されないものとも考えられる。

いずれにしても、この点に関しては、GPL上も一義的に明確とまではいえないと考えられ、この点に関しては、さらなる検討が必要と考えられる。

ここでのまとめ

以上述べた通り、GPLは、copyleftの理念の実現のために、特許権を積極的に活用することを予定しておらず、外部からの特許権行使に対しては、消極的な対応を採ることを予定するにとどまっている。

このことは、GPLが、著作権のライセンス契約を媒介にしてcopyleftの理念実現を目指しているために、ライセンス契約に縛られない外部の第三者からの特許権攻撃に対しては、有効な手段を持たないことを意味しているが、GPLが上記のような構造を採用している以上、やむを得ないものと考えられる。

(8部 2.2 「特許問題」 水谷委員)

■その他、法との関係

ここでは、前述以外の各関係法律とGPLとの関係を概観する。

民法(権利濫用)

GPLに準拠するソース・コードの比率が小さいソフトウェアの全体に対して、ソース・コードの開示などの制約がかかるのは民法の権利濫用の法理で無効になる可能性があるとの意見がある。

免責(民法・消費者契約法)

▶ ライセンサーの責任

ユーザーが消費者である場合でも、隠れた瑕疵については、無償契約であるために消費者契約法第8条第1項第5号（明文で有償契約に限定している）の適用はなく、GPLによる免責が有効と考えられる。しかし「知りて告げざりし瑕疵または欠陥」については、同項第1号、第2号および第10条の趣旨から、免責が無効になる可能性がある。

ユーザーが消費者でない場合は、GPLの免責条項は一応有効であろうが、故意にウイルスのような破壊的なプログラムを混入させた場合など故意・重過失の場合は、公序良俗違反（民法）により、GPLによる免責は無効とされる可能性がある。

▶ 販売者の責任

プログラムのバグなどの不具合については、目的たるプログラムが取引の通念に照らし、合理的に期待される通常有すべき機能・品質を有していない場合で、かつ通常予見可能な使用環境・使用方法の範囲内で動作上の不具合が発生した場合、瑕疵（民法第559条、第566条、第570条）に該当し、ベンダーの瑕疵担保責任が発生するとされている（経済産業省、「電子商取引等に関する準則」, p. 71, 2002年7月（以下、準則））。

GPL準拠のプログラムが販売された場合、販売者は売買契約に基づく瑕疵担保責任を負うことになる。販売者はライセンス契約の当事者でない（ライセンス契約はGPL準拠のプログラムのすべての著作権者とユーザーとの間で直接締結される）ので、GPLの無保証条項は、販売者には適用されない。販売者はライセンサーに転嫁できない責任を自己自身で負担しなければならない。

▶ 製造物責任法

製造物責任法は、無過失責任である製造物責任の対象となる「製造物」を「製造又は加工された動産」としている（製造物責任法第2条第1項）。ソフトウェア自体は動産ではないから、製造物責任法の対象ではないとすることに異論はない。

しかし、ハードウェアに組み込まれたソフトウェアの欠

陥によりハードウェアが誤作動するなどして、消費者に積極的損害が発生した場合は、製造物の欠陥による損害として製造物責任が発生するとの意見がある（経済企画庁国民生活局消費者行政第一課編、「逐条解説・製造物責任法」, 商事法務研究会, p. 59, 1994年（以下、逐条解説・製造物責任法））。

ソフトウェアがハードウェアに組み込まれてはいるが、書き換え可能であるような場合については、ハードウェアとソフトウェアのメーカーが同一である場合のみ製造物責任の対象となる、とする説やハードウェアの一部として製造物責任の対象となる、とする説など諸説が対立している（岡本佳世ほか、「企業のPL対策」, 商事法務研究会, p. 67, 1995年（以下、PL対策））。

製造物責任の免責条項については、一般に公序良俗違反（民法90条）として無効と解される場合が多いと考えられている（「逐条解説・製造物責任法」, p. 128）。GPLライセンスのソフトウェアが組み込まれたハードウェアについて、ソフトウェアの欠陥が原因で製造物責任が発生した場合、GPLの免責条項の有効性が問題となるが、ハードウェアに組み込まれている場合には、ハードウェアのユーザーとGPLソフトウェアのライセンサーとの間にライセンス契約が成立するわけではないので、GPLの免責条項がハードウェアなどに表示されていたとしても、免責の効力は発生しないものと解される可能性がある。

▶ 特許法（「特許問題」(pp. 76-79) 参照）

GPL準拠のソフトウェアは、特許のライセンス料を徴収することも禁止されている（GPL第7条）。特許ライセンスを必要とするソフトウェアについては、そのライセンスが誰でもプログラムを使用料無料で再頒布することを認めていない場合、プログラムの頒布を完全に中止しなくてはならない。そうした意味でGPLは特許と一緒に使用することが困難なライセンスといえる。

また、自己が有する特許をGPLソフトウェアと組み合わせた場合に、その特許についてまでライセンス料の徴収ができないのかの問題について明確な回答が存在しない。自己の特許についてもライセンス料を徴収できないとすると、

この意味でもGPLは特許との共存が困難なライセンスといえる。

▶ 商標権

GPLは商標権については、何も規定していない。「Linux」のロゴの商標については、平成14年（2002年）4月30日東京高裁平成13年（行ケ）435号商標権行政訴訟事件で争われ、裁判所はLinuxの商標登録について「その指定商品中『印刷物』について、商標法4条1項15号に規定する『他人の業務に係る商品又は役務と混同を生ずるおそれがある商標』に該当すると認められるものであり、これと同旨の認定判断のもとに、本件商標は、その指定商品中の『印刷物』について、商標法4条1項15号に違反して登録されたものであるから、同法46条1項の規定により、その登録を無効とするとした審決に原告主張の誤りはない」としている。また「オープンソース」のロゴについては2000年に商標の登録申請がなされ、2002年に登録番号第4553488号として商標登録がなされている。商標権は権利者に独占的権利を与えるものであって、商標権とGPLの関係は明確ではない。

▶ 不正競争防止法

不正競争防止法の「営業秘密」との関係でも疑問がある。甲がGPLに準拠する可能性のあるソフトウェアを改変し、乙に再頒布した場合を想定する。甲がかかると考えソース・コードを開示せず、かつかかるソフトウェアの情報が不正競争防止法上の営業秘密の要件を満たす場合に、乙がそのソフトウェアをGPLに準拠すると考えソース・コードを開示してしまった場合に、不正競争防止法に違反するのかが問題である（「第三者問題」（pp. 95-97）参照）。

▶ 契約条項の言語（民事訴訟法）

裁判所法第74条は、裁判所では日本語を用いると定めている。裁判実務では、英文契約書を当事者が書証として提出するには、日本語訳を付けることが要求されている。

GPLは、英文が正文であり、公式の日本語訳がない。ま

た、その文言も必ずしも一義的ではない。このため、日本での裁判となった場合、一方当事者が提出した日本語訳の正確性などについての争いが生じ、争点がいたずらに拡大する可能性も高い。このことは、GPLについて日本の裁判所がどのような判断を示すかの予測を妨げる事由でもある。

また、消費者契約法3条1項において「事業者は、消費者契約の条項を定めるに当たっては、消費者の権利義務その他の消費者契約の内容が消費者にとって明確かつ平易なものになるよう配慮するとともに、消費者契約の締結について勧誘をするに際しては、消費者の理解を深めるために、消費者の権利義務その他の消費者契約の内容についての必要な情報を提供しよう努めなければならない」とある。この点も英文が正文であるGPLとの関係で問題となる。

▶ 特定商取引法

特定商取引法において「プログラムを電子計算機に備えられたファイルに記録し、又は記録させること」が指定役務に加えられている（同法2条4項、政令3条3項、別表第3の13）。しかし、GPLによるソフトウェアの利用は無償なので、同ソフトウェアの提供は同法の「通信販売」には当たらず、同法の規制は原則として及ばない可能性が高い。もっとも、GPLでは必要な実費を要求することも許容されており、その場合は「通信販売」に当たると解釈される可能性もあり得ないわけではなく、そのような解釈の下では同法の規制（広告規制（同法第11条、第12条）、表示義務（同法第11条、施行規則第8条）など）が及ぶことになる。また、GPLのソフトウェアのライセンスに伴う有償の保証契約が、上記の指定役務に該当する場合も、同法の規制が及ぶことになる。

▶ 独占禁止法

GPLが、GPLと両立しないソフトウェアからのリンクなどの利用を認めないことや、2次著作物の頒布に際してソース・コードの開示を強制することについては、GPLのライセンサーが市場において独占的地位を有する場合などには、不公正な取引方法あるいは市場における公正な競争を阻害するものとして独占禁止法違反となり、GPLが無効

とされる場合もあり得ると考えられるとの意見もある。

(8部 2.2「その他、法との関係」 平野委員)

2.3 法律上問題となり得る技術の解説

前述までに、法的な判断を下す上での法的基盤の整理が整った。

ここでは、Linuxを例に、GPLを理解するために必要な技術用語とGPLとの関係について解説する。これにより、さらに詳細な実務的、個別的論点への検討に入ることができる。

■技術の解説

全体概要

図1はLinuxシステムの概要を示している。Linuxシステムは大きく分けて4つのコンポーネントより構成されている。第1にアプリケーションであり、ユーザーが直接起動することにより実行されるプログラムある。

アプリケーションは、ブラウザやエディタなどのクライアント・アプリケーションとWWWサーバなどのサーバ・アプリケーションなどが存在する。プラグインはブラウザなどのクライアント・アプリケーションを拡張するためのプログラムである。通常、プラグインの起動は新しいプロセスを生成し、プラグインとクライアント・アプリケーシ

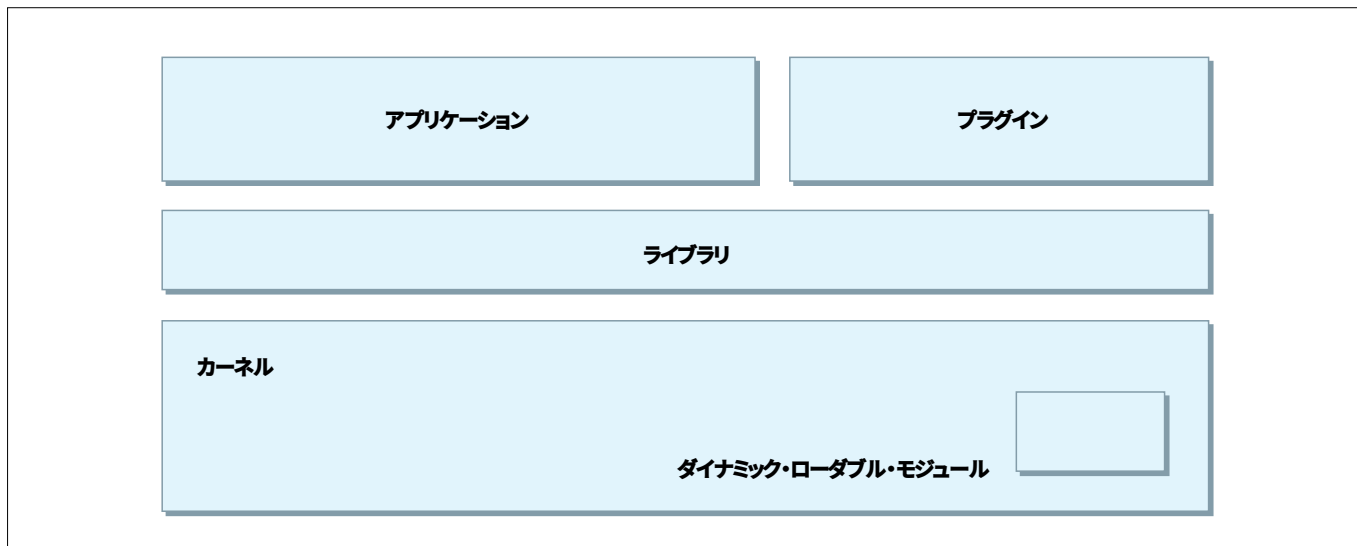
ョン間の通信はパイプなどのプロセス間通信機能を用いることにより実現される。ライブラリは、アプリケーションを実行するために必要となる補助的なプログラムである。ライブラリとアプリケーションの結合（リンク）は静的リンクと動的リンクの2つの方法が存在する（後述）。カーネルはCPU、メモリ、デバイスなどのハードウェア・リソースを管理し、それら进行操作するための高レベルの抽象化を提供する。ダイナミック・ロードブル・モジュールはカーネル機能を実行中に動的に拡張するために存在する。新しいデバイスを接続したときに動的にデバイス・ドライバをカーネル内に組み込むことも可能となる。

技術詳細

①ライブラリのリンク

ライブラリをアプリケーション・プログラムとリンクするための方法として静的リンクと動的リンクの2つの方式が存在する。静的リンクは実行ファイルを作るときにライブラリとアプリケーション・プログラムのリンクを行う。動的リンクは実行時にライブラリとアプリケーション・プログラムの結合を行う。ライブラリとアプリケーション・プログラムの生成は独立して行われるため、アプリケーション・プログラムとライブラリを別々にパッケージ化することが可能となる。つまり実行時に、どのライブラリを使

図1



用するかを決定することが可能となる。

② GPLとLGPL

GPLライブラリを使用する場合は、アプリケーション・プログラムのライセンスもGPLとして頒布する必要がある。この場合、静的リンクを用いるか動的リンクを用いるかは無関係である。しかし、動的リンクを用いる場合はライブラリとアプリケーション・プログラムを独立に実行プログラムとすることが可能であるため、アプリケーション・プログラムの開発者と独立したディストリビューターがGPLライセンスにより頒布されていないアプリケーションをGPLのライブラリと一緒にパッケージ化した場合は、そのディストリビューションを頒布することができなくなる。

③ カーネルとアプリケーション

アプリケーションからカーネルを呼び出す場合は、トラップといわれる機構が使用される。トラップを用いた結合は動的リンクによる方式と似ている。カーネルがGPLライセンスにより頒布されている場合にアプリケーションをどのようなライセンスにより頒布すべきかは、今後検討が必要である。Linuxの場合は、著作権を持った開発者の1人がアプリケーションをGPLライセンスにより頒布する必要がないことを明言している。

④ プラグインとGPL

プラグインはクライアント・プログラムと独立したプロセスとして生成される。つまり、クライアント・プログラムとプラグインは異なるプログラムと考えることができるため、お互いはライセンス的には無関係と一般的にいわれている。GNU FAQでもクライアント・プログラムがGPLによりライセンスされている場合でもプラグインはGPLとする必要がないことが明記されている。

⑤ コンパイルとライセンス

C言語やC++言語を用いる場合はプログラム中で用いられるデータ構造やマクロを定義するためにヘッダ・ファイルを使用する。GPLライセンスにより頒布されるヘッ

ダ・ファイルを使用する場合、プログラム自体をどのようなライセンスにより頒布するかは検討すべき課題である。特にヘッダ・ファイル内に複雑なマクロを含んでいる場合は、マクロの創造性などを検討する必要がある。

⑥ ダイナミック・ローダブル・モジュール

ダイナミック・ローダブル・モジュールは、カーネル内にプログラムを動的にリンクするためのメカニズムである。GPLライセンスにより頒布されるカーネルにダイナミック・ローダブル・モジュールをリンクする場合は、ダイナミック・ローダブル・モジュール自体もGPLライセンスで頒布する必要がある。ダイナミック・ローダブル・モジュールを使用する注意点は、以下の2点がある。1つは標準ドライバを使用する場合の扱いである。ドライバが一般に公開されている標準インタフェースを使用する場合は、一般にGPLライセンスに影響されないと考えられている。もう1つはカーネル内にフック（元のソース・コードからダイナミック・ローダブル・モジュールを呼び出す部分）を入れ、フックのみGPLライセンスとして公開することも考えられるが、この場合はダイナミック・ローダブル・モジュールをGPLライセンスとして公開する必要があると考えられている。

⑦ Linuxカーネルの拡張

Linuxカーネルを拡張する場合、拡張するためのソース・コードは基本的にすべてGPLライセンスとして頒布する必要がある。しかし、フックをダイナミック・ローダブル・モジュールから明確に分離し、フックの部分のみをGPLライセンスで頒布し、ダイナミック・ローダブル・モジュールを非GPLプログラムとして頒布することも考えられる。従来から、ダイナミック・ローダブル・モジュールはGPLライセンスで頒布すべきかの議論はグレーであり、Linuxの場合、著作権を持った開発者の1人が基本的にダイナミック・ローダブル・モジュールはGPLライセンスで頒布すべきであると主張しているが、法的な根拠が何であるかは議論する必要がある。しかし、フックを用いたカーネルの拡張がGPLに縛られず行えるとするとLinuxのオープン性

が失われていくことになりかねないので、勝手にカーネルを拡張できない根拠が法的に示されることは有意義と思われる。

(8部 2.3「技術の解説」 中島委員)

■技術的特性に対する法律的当てはめと検討

Linuxを前提として、GPLの下で提供されているオープンソース・ソフトウェアについて抽出された技術的特性を踏まえると、法的当てはめの結果としては、以下の点が論点となると考えられる。

(1) GPLライブラリとアプリケーション間でのリンクを前提とした場合

①リンクが形成されることによって、当該GPLライブラリなる著作物についての「改変または頒布 (modify or distribute)」がなされていると評価できるのか

②リンクされているアプリケーションなる著作物について「当該ライブラリを包含する」または「GPLの下での (ライブラリに対する) 2次的著作物 (派生物)」と評価され得るのか

(2) GPLの下で提供されているカーネルとアプリケーションが、トラップによって結合されている場合

①トラップを通じてアプリケーションと結合されることによって、当該カーネルなる著作物は「改変または頒布 (modify or distribute)」がなされていると評価できるのか

②当該アプリケーションなる著作物について、トラップを通じて当該カーネルと結合していることをもって「当該カーネルを包含する」または「GPLの下での (カーネルに対する) 2次的著作物 (派生物)」と評価され得るのか

(3) クライアント・プログラムがGPLの下で提供されている場合、当該クライアント・プログラムに対するプラグインについて

①当該プラグインが導入されることによって、当該クライアント・プログラムなる著作物について「改変または頒布 (modify or distribute)」がなされていると評価できるのか

②当該プラグインなるプログラム著作物について、当該クライアント・プログラムに対応することによって「当該クライアント・プログラムを包含する」または「GPLの下での (クライアント・プログラムに対する) 2次的著作物 (派生物)」と評価され得るのか

(4) GPLの下で提供されているヘッダ・ファイルを組み込んだプログラムについて

①当該プログラムヘッダ・ファイルが組み込まれていることによって、当該ヘッダ・ファイルなる著作物について「改変または頒布 (modify or distribute)」がなされていると評価できるのか。そもそもヘッダ・ファイルは著作物たり得るのか

②当該プログラムというプログラム著作物は、ヘッダ・ファイルを組み込むことによって「当該ヘッダ・ファイルを包含する」または「GPLの下での (ヘッダ・ファイルに対する) 2次的著作物 (派生物)」と評価され得るのか

(5) GPLの下で提供されているカーネルがダイナミック・ローダブル・モジュールにリンクしている場合

①当該ダイナミック・ローダブル・モジュールにリンクすることによって、当該カーネルなるプログラム著作物について「改変または頒布 (modify or distribute)」がなされていると評価できるのか

②当該ダイナミック・ローダブル・モジュールは、カーネルとリンクすることによって「当該カーネルを包含する」または「GPLの下での (カーネルに対する) 2次的著作物 (派生物)」と評価され得るのか

検討

まず大前提として、以下の点を踏まえるものとする。

- ここで検討対象となるオープンソース・ソフトウェアのプログラムについては有効な著作権 (プログラム著作物) が発生しているものとする。
- GPL自体の法的性質として、オープンソース・ソフトウェアの著作権に関するライセンス契約であるという、

いわば知的財産権ライセンス契約の一形態であるものとして解釈するものとする。

(1)～(5)の①については、ライセンス契約としてのGPLが成立し得るための要件であるといえる。

GPL第5条では、GPLによる提供対象となるプログラム、または対象となるプログラムを基礎とする任意の成果物の「改変または頒布 (modify or distribute)」によって、GPLのライセンス条件の承諾を行ったものとする規定されている。このような条項自体が元来、契約の解釈問題として有効であるのか否かという問題は別途存在し得るところである。そして、この点は、GPLについての準拠法が何であるのかという点にも左右されるところといえよう。

ともかく当該条項については一般的な有効性を前提とした上で、(1)～(5)の一連の技術的特性がかかわることによって、そもそもGPLなるライセンス契約の一形態が成立しているのか否かという点について個別的に考慮されなくてはならないだろう。そこで、リンク形成、トラップによる結合、プラグイン導入、ヘッダ・ファイルの組み込み、ダイナミック・ローダブル・モジュールのリンクといった一連の行為が、GPL上の「改変または頒布 (modify or distribute)」という文言の解釈として含まれると評価し得るのか問題となってくる。

もっとも、ここで再度注意すべきは、GPLを解釈する準拠法が何かにによって、その解釈基準にかなりの差異が生じ得るということである。

(1)～(5)の②については、ライセンス契約としてのGPLが有効に成立していることを前提として、当該GPLの契約の効果として、各対象がGPLの下での提供が要求され得る範囲に含まれるか否かについてのメルクマールとなっているといえる。

GPL第2条b)において、GPLの対象となっているプログラムまたは任意の一部について、その全体または一部を包含するか、あるいはプログラムまたはその任意の一部から派生された成果物については、GPLのライセンス条件の下で任意の第三者へ提供しなくてはならないと規定されている。ここでは「contain」および「derive」なる文言の解

釈が問題となる。ここでも、GPLの準拠法が何であるのかによって、その解釈基準に差異が生じ得ることは無視し得ないところではある。

とりわけ「derive」については、GPL第0条において、著作権法の下での「derivative work」なる記述があることから、著作権法における2次的著作物概念に対応するものとして解釈することに相当の合理性があるものと考えられる。そこで次の問題としては、何国の著作権法がここでのGPLの解釈の前提となるのかという点が生じてくる。例えば、日本著作権法を前提とするのであれば、リンク形成、トラップによる結合、プラグイン導入、ヘッダ・ファイルの組み込み、ダイナミック・ローダブル・モジュールのリンクといった一連の行為が、日本著作権法における「翻案」概念の解釈として含まれ得ると評価できるのかが論点となってくる。

次に、日本著作権法の下で付与される著作権を前提として、さらにGPLを解釈とする準拠法についても日本法であることを前提として、GPLを解釈するという状況設定（端的には日本国内の法人同士がGPLの下でオープンソース・ソフトウェアを取引し、契約準拠法については日本法とすることで合意しているような場合）の下で、個別論点について考察してみる。

▶ (1) について

①「改変または頒布 (modify or distribute)」という文言が契約解釈の問題として、具体的にいかなる行為までを含むものとされ、ライブラリとアプリケーション間のリンクがそこに該当すると解し得るのか、ということによる。解釈の1つの考え方としては、著作権法における「翻案」および「頒布」なる文言を参考にすることが考えられ、以下、そのような考え方に沿って検討してみる。

まずGPLライブラリとアプリケーション間にリンクが形成されていることをもって、GPLライブラリという原著物の表現についての「翻案」がなされているとまでの評価を行うことは一般的には困難であるように考えられるところであるが、「翻案」概念より若干広い意味として「改変 (modify)」を解するのであれば含まれると解し得る余地は

あろう。

一方、リンクが形成されたことだけをもって、その後アプリケーションの頒布によって、リンクを介して当該GPLライブラリが用いられることで直ちに「頒布」されると評価することは困難であるように考えられる。リンクが形成されていてもGPLライブラリ自体が併せて「頒布」されていなければ、リンクによる実際のアクセスは実現されないからである。逆に言えば「頒布」されたアプリケーションが現にGPLライブラリにリンク可能である場合については、当該GPLライブラリについても「頒布」されているという解釈もし得るだろう。もっとも、この点についてはリンクの形式が静的なのか動的なのかという差異についての考慮も無視し得ないように考えられる。

②リンクによって、アプリケーションがGPLライブラリを包含すると評価し得るのかという点も、具体的にアプリケーションと当該ライブラリの間がいかなる依存関係にあったのか、いかなるリンク形式で結び付いているのかといった点も考慮して、個別具体的に判断されることになろう。

一方、ライブラリという著作物を基礎として考えた場合に、アプリケーション自体がライブラリの2次的著作物であると評価することは多くの場合困難であるように考えられる。ライブラリとアプリケーションのそれぞれは独立性は高く、一方の翻案の結果得られるという関係にあるものではないだろう。

▶ (2) について

①「頒布 (distribute)」として解することは困難であろう。「改変 (modify)」と解することについても、カーネルというプログラムの著作物がアプリケーションとトラップを解して結合していることをもって直ちにカーネル自体の「翻案」が生じているとは解しにくいように考えられる。

②アプリケーションがカーネルを包含すると解したり、アプリケーション自体がカーネルの翻案生成物であると解することは困難であろう。

▶ (3) について

①「頒布 (distribute)」として解することは一般には困難

であろう。もちろん、当該クライアントとプラグインが常に併せて提供されているという状態を前提とされているのであれば「頒布」と解し得るものといえようが、単にあるクライアントに対応しているプラグインがあるからといって、直ちに「頒布」とは評価し得ないだろう。また、プラグインが導入されている状態をもってクライアントが「改変 (modify)」されていると解する余地は否定できないものの、一般的にはクライアントとプラグインはそれぞれ独立性をある程度有しているであろう。

②クライアントがプラグインを包含すると解するのはともかく、プラグインがクライアントを包含すると解することは困難であり、プラグイン自体をクライアントの2次的著作物と解することも難しいように考えられる。

▶ (4) について

①ヘッダ・ファイルに対する「改変 (modify)」と解される余地はあるものと考えられる。

もっともヘッダ・ファイル自体が極めて短いものや技術的に特定されるプログラムのようなものである場合、そもそも著作物性がないと評価されるものもあり得るかもしれない。

②元のプログラムがヘッダ・ファイルを包含すると解する余地があることは考えられる。

一方、ヘッダ・ファイルの翻案生成物として、プログラム自体をとらえることは困難であるように考えられる。

▶ (5) について

① (1) と同様に、カーネルとダイナミック・ローダブル・モジュールの間にリンクが形成されていることをもって、GPLライブラリという原著作物の表現についての「翻案」がなされているとまでの評価を直ちに行うことは困難であるように考えられるが、「翻案」概念より若干広い意味として「改変 (modify)」を解するのであれば、含まれると解し得る余地はあろう。

カーネル自体の表現形式については何ら変化が加えられていないものの、ローダブル・モジュールとのリンクによって実質的にはプログラムとして変化しているということ

を著作権法の下でどのように評価すべきかという点である。また、(1)のようなライブラリとアプリケーション間のリンクと実質的な意味での違いをどのように評価するのか、という点でも違いが生じてくるであろう。「頒布」については、(2)と同様のことが議論できるものと考えられる。②この点も(1)と同様、ダイナミック・ローダブル・モジュールがカーネルを包含すると評価し得るのかという点も、具体的に両者がいかなる依存関係にあったのか、いかなるリンク形式で結び付いているのかといった点を考慮して、個別に具体的に判断されることになる。またダイナミック・ローダブル・モジュールをカーネルの翻案生成物として解することは、一般的には困難であるように考えられる。

(8部 2.3 「技術的特性に対する法律的当てはめと検討」 平嶋委員)

2.4 個別的論点

ここでは「2.3 法律上問題となり得る技術の解説」(pp. 82-87)のような技術的検討を踏まえた上で、特にオープンソース・ソフトウェアを組み込んだソフトウェアやシステムを開発する者に頻繁に課題となる個別的論点を整理する。

■免責

GPLの第11条は、GPLに基づく許諾の対象となるプログラム(以下、GPLソフト)について、一切の保証がなされない旨を定めている。このため、GPLソフトを組み込んだシステムやソフトウェアを提供する事業者が、ユーザーとの関係でどこまで、いかなる責任を負うのかが頻繁に課題となる。ここでは、このGPL第11条の有効性、適用範囲およびその実務的な問題点について検討する。

GPL第11条の意義と問題点

▶ 品質保証(機能保証および性能保証)について

コンピュータ・プログラムには、不可避免的にバグが含まれることなどの理由から、その品質についての保証(機能保証と性能保証を含む)が限定されるのが一般的である。

しかし、商用ソフトウェアの使用許諾契約においては、記録媒体に物理的な瑕疵があった場合に良品と取り替える義務が規定されたり、一定期間内に瑕疵の修補を行うことが規定されたりする場合も多い。GPLにおいては、ライセンサーは、そのような限定的な保証義務を負わないものとされている点に注意が必要である。特に、GPLソフトに構造上の欠陥があったり、GPLソフトがウイルスに感染していたりしたために、GPLソフトをインストールしたことによってシステムが復旧不可能の障害を被った場合であっても、通常、GPLソフトの利用者はライセンサーに対して一切の責任を追及できないものと考えられる。従って、GPLソフトを使用または頒布する場合には、そのようなリスクを勘案した上で、十分な検証を行う必要がある。

▶ 権利不侵害の保証について

GPL第11条を文字通り解釈すれば、GPLソフトのライセンサーは、当該GPLソフトが第三者の著作権、特許権その他の人格的または経済的な権利を侵害したとしても、一切の責任を負わないことになる。商用ソフトウェアの場合にも、常に権利不侵害についての保証がなされるわけではないが、多くの場合には、少なくとも、第三者の権利侵害が問題となった場合に、ライセンサーがライセンシーを補償(indemnification)すべき旨が規定されている。GPLソフトの機能および性能をあらかじめ十分検証すれば、品質保証が一切行われていないことに伴うリスクを軽減することができるが、権利不侵害の保証についてはGPLソフト自体をいくら念入りに検証しても、そのリスクを把握できない。特に著作権侵害の有無に関していえば、GPLソフトの開発者が、他者の著作物に依拠して当該GPLソフトを開発したかどうかによって著作権侵害が問題となるか否かが決まることになるが、その事実関係については、開発者自身に表明、保証してもらう以外には、ほとんどの場合確認する方法がない。開発者が一切の表明、保証をせずに頒布されているGPLソフトを使用している以上、GPLソフトの利用者は第三者の権利を侵害するリスクをある程度は承知していたはずである、と評価される余地がある。利用者が善意無過失であれば、真の権利者に対する損害賠償義務

を免れる余地があるが、GPLソフトの使用者については、少なくとも権利侵害の可能性を認識していた、との理由から、過失があると認定されて損害賠償を余儀なくされる可能性がある。

GPL第11条の有効性

日本の消費者契約法第8条1項によると「消費者契約が有償契約である場合において、当該消費者契約の目的物に隠れた瑕疵があるとき…に、当該瑕疵により消費者に生じた損害を賠償する事業者の責任の全部を免除する条項」（同条同項5号）は、無効とされている。GPL第11条は「損害を賠償する事業者の責任の全部を免除する条項」であるので、消費者契約法に基づいて、GPL第11条が無効とされる可能性はないか、が一応問題となり得る。しかし、消費者契約法第8条1項5号は、消費者契約が有償契約である場合に限定して適用されるものであるところ、GPLは一般的には無償のライセンスと理解されているので「有償契約」ではない、との理由で、適用が否定される可能性が高い。もっとも、GPLはGPLソフトの複製物を頒布するというサービスに対する対価を徴収することを制限していないので、そのようなサービスの対価が徴収されている場合には、消費者契約法上の「有償契約」に該当するものと評価される可能性があるものと思われる。

GPL第11条は、米国の統一商法典の規定を意識したものであり、黙示の保証を限定するための要件を満たしているので、GPLが契約として有効に成立している場合には、特別の事情がない限りGPL第11条は、米国法上有効と思われる。日本法上は、担保責任免除の特約に関する民法第572条の適用のいかんが問題となる。民法第572条は「知りて告げさりし事実及び自ら第三者の為に設定し又は之に譲渡したる権利については其の責を免るることを得す」と規定しているので、瑕疵があることを知りながら、それを告げずにGPLソフトを頒布した場合には、同条の規定に基づいて、GPL第11条に基づく無保証の規定の適用が否定される可能性がある。もっとも民法第572条は売買に関する規定であり、売買に関する規定は、民法第559条によって他の有償契約に準用されるものとされているが、前述の

通り、GPLが一般的には無償のライセンスであると認められることからすれば、民法第572条の規定の適用が認められない場合が多いと思われる。従って、日本法上GPL第11条の有効性を争うためには、公序良俗違反その他の一般規定を根拠とせざるを得ないが、これらの一般規定に基づいてGPL第11条が無効であると主張し得るのは、使用者に損害を被らせることを意図してGPLソフトを頒布した場合のように、相当まれなケースに限定されると思われる。

GPL第11条の適用範囲

GPL第11条は、GPLの条件に従ってGPLソフトを使用する者との間の契約上の保証責任に関する規定であり、GPLに従ってGPLソフトを使用する者以外の者に対しては、効力が及ばない。例えば、GPLソフトを組み込んだ機器を使用する者は、GPLに従ってGPLソフトを使用する者ではないので、GPL第11条に規定した無保証の規定の拘束を受けず、機器の欠陥を理由として損害賠償請求ができる。現に、高機能のハイエンドのマルチメディア編集ソフトウェア「Broadcast 2000」というGPLソフトの開発者は、Broadcast 2000が、ハイエンドのビデオ制作に使用されているため、損害が発生した場合に、莫大な損害賠償を受ける可能性があることなどを理由として、その頒布を中止している。

また、GPLソフト自体は、製造物責任法上の「製造物」、すなわち「製造又は加工された動産」でないと考えられるが、GPLソフトが格納されたASICや当該ASICを組み込んだ機器は「製造物」とであると解される。その場合、GPLソフトの不具合が当該ASICまたは当該機器の欠陥であると評価される可能性は否定できず、そのため、GPLソフトの開発者またはその使用者が製造物責任法上の責任を追及されることが考えられる。

(8部 2.4「免責」 宮下委員)

■内部利用

GPL問題で頻繁に焦点となるソース・コードの開示義務は、プログラムを開示した際に課せられる。このため、実務的には何が開示には該当しない内部利用となるのかが頻

繁に課題となる。ここでは、この「内部利用」の範囲に焦点を当て、整理を行う。

論点

GPLでのソース開示義務は、プログラムを「distribution (redistribution)」した場合に発生する。これを企業活動に当てはめて言い換えれば、その企業内で「内部利用」している場合には「distribution (redistribution)」が行われていないことになり、GPLによるソース・コードの開示義務は発生しないといえる。

従って、企業においてはどのような利用の形態がソース・コードの開示義務を伴わない「内部利用」となるのか、すなわち「distribution」にならないのかを検討し理解しておくことが、ソース・コード開示に関しての無用なGPL違反を指摘されないためにも必要なことといえる。

初めに、一般的に「内部利用」に当たると考えられる場合と当たらないと考えられる場合とを以下に列挙する。

▶「内部利用」と考えられる場合

- 自社で用いる社内システム（出退勤管理システムなど）に利用する場合
 - 自社の他部門へプログラムを提供する場合
 - 製品開発の際のツール・プログラムとして利用する場合（ただし、ツール・プログラムのコードの一部が出力結果である開発された製品（プログラム）にコピーされている場合は、この限りではない）
 - 製品の性能テスト用のプログラムとして利用する場合
- 以上の例は、そのプログラムの利用や提供が社内で行われており、一般的には社外に出ないものといえるので、この場合は「内部利用」と判断できることが多いといえる。

▶「内部利用」と考えられない場合

- 販売する自社製品の組み込みOSとして使用する場合
 - 自社で利用している社内システムを外販する場合
 - 製品のドライバ・プログラムなどを無償でユーザーに提供する場合
- 以上の例は、そのプログラムの利用・提供が社内だけで

なく、製品などの何らかの形で社外に出ているものといえるので、この場合は「内部利用」ではないと判断できることが多いといえる。

上述のように、プログラムのやりとりが自社内で行われており外部に出ていないものは、一般的に「内部利用」とあると判断できる。しかしながら、個々のケースによっては「内部利用」となるか否かの判断に迷う場合もある。例えば、①関連会社や子会社へのプログラムの提供、②機密保持契約の下でのプログラムの頒布、③ASPサービス、などである。

解釈

上述の各ケースに関しては、それぞれ以下のように解釈することができる。

①関連会社や子会社へのプログラムの提供

FSFのFAQでは「organization」や「company」での使用や複製は「distribution（頒布）」に当たらないと説明されている。

これに従えば、関連会社や子会社は自社とは別の「company」であり、このような関連会社や子会社へのプログラムの提供は「distribution（頒布）」といえ「内部利用」ではないと考えられるので、ソース・コードの開示が義務付けられることになるとする意見がある。

一方、関連会社や子会社の場合、資本関係があり、全くの他人であるとはいえず、考え方によっては「organization」、すなわち、企業グループの内部と解釈され、ソース・コードの開示が義務付けられないとする意見がある。

この場合は、FAQでいうところの「organization」という概念が明確に定義できないことから、開示義務があり/なし、のいずれとも断定できないことになる。

②機密保持契約の下でのプログラムの頒布

GPLは開示義務を規定する一方、機密保持義務は非開示義務を定めるものであるため、GPLと機密保持義務の関係について考慮することは必要である（図2）。

まず、B社がGPLコードを改変してA社へ頒布する場合、A社とB社との関係は互いに別の会社であり内部利用の範囲ではないとなると、B社からA社に対してプログラム（オブジェクト・コード）が頒布された時点で、B社はA社に対するソース・コード開示義務を負うことになる。

B社がオブジェクト・コードと一緒にソース・コードをA社に提供した場合には、GPL第3条a)の態様であり、B社のソース・コード開示義務はA社に対してであり、A社以外の第三者に対してはその義務は負わないと考えられる。なお、オブジェクト・コードおよびソース・コードを受領したA社については、受領したプログラムをさらに外部に頒布しない限り、第三者に対してのソース・コード開示義務は負わないと解釈することができる。

本来は以上のような開示義務がGPLにより課されるということを前提とした上で、仮にA社とB社との間に機密保持契約が存在したとして、それが内部利用といい得るかという、通常は内部利用といえないことが多いのではないと思われる。むしろこの場合、機密保持義務の内容によっては、GPL違反の問題が生じ得ることに注意する必要がある。

▶ A社がB社に対して機密保持義務を課している場合

B社がA社に対して頒布したプログラム（オブジェクト・コード）およびソース・コードをA社以外に対して秘密に

しておく義務をA社がB社に課すことは、GPL違反ではないと解釈できる。

この場合B社は、A社へのオブジェクト・コード頒布の際に、受領者であるA社に対してソース・コード開示義務を履行してさえいれば、A社以外の第三者からのソース・コードの開示要求に応えなくとも、何らGPLに違反することはないといえる。

このことは、後述の通りFSFのFAQにも記載されており、A社がB社に機密保持義務を負わせてプログラムを開発させることはGPL違反でないとしている。

▶ B社がA社に対して機密保持義務を課している場合

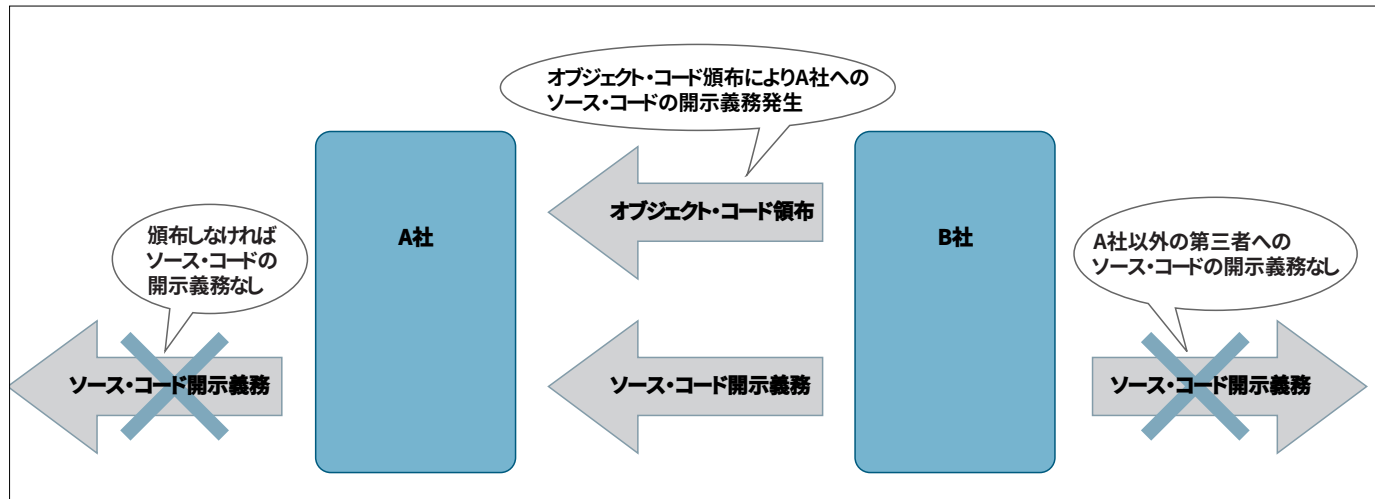
B社が開発した成果であるプログラムおよびソース・コードをA社に頒布する際、B社がA社にプログラムの第三者への開示を制限することは、GPLに対し異なる別の条件を追加したことといえる。

このことは、後述の通りFSFのFAQにも記載されており、B社がA社に機密保持義務を課してプログラムを提供することはGPL違反であるとしている。

③ ASPサービス

ASPサービスの場合、一般的に、サービスを行うサーバは自社内にあり、外部ユーザーはサーバに対してデータを送信し結果を受信する形態であり、プログラム自体は社内

図2



で閉じており、内部利用の範囲といえる。

しかしながら、クライアント側にJavaスクリプトなどを提供している場合は、スクリプトの部分は「distribution (頒布)」となる可能性がある。

理由(実態・ファクト)

上記解釈の根拠は、FSFのFAQの以下の記載による。

▶ 頒布に関する一般的な説明

Q1：GPLは、改変されたバージョンのソース・コードを公に発表することを要求しますか？

A1：GPLでは、あなたが改変したバージョンを公開することは要求してはいません。改変を加えて、公開せずに個人的に使うのはあなたの自由です。

しかし、もしあなたが改変されたバージョンを何らかの形で公にするならば、GPLはあなたが改変したソース・コードをユーザーがGPLの下で入手できるようにすることを要求します。

すなわち、GPLは改変されたプログラムを特定のやり方で公開する許可を与えていますが、別の形での公開は許可していないのです。しかし、公開するかどうかはあなた次第です。

Q2：ある企業がGPLの適用されたプログラムの複製を所有しており、それを入手するためにはお金が掛かります。インターネット上で入手できるようにしないという点で、彼らはGPLに違反しているのではないのでしょうか？

A2：いいえ。GPLはインターネットを使って頒布することを要件とはしていません。また、GPLはプログラムを誰か特定の人が再頒布することも要求しません。そして(ある特別な場合を除いて)、誰かがプログラムを何回かにわたって再頒布する際にも、GPLはその人が特定個人としてのあなたやほかの誰か特定の人に頒布しなければならないということを要求していません。

GPLが要求するのは、もし頒布する人がそうしたければ、あなたに複製を頒布できる自由を有しなければならないということです。いったん著作権者が誰かにプログラムの複

製を頒布したら、今度はその誰かが、自分にできる範囲でプログラムをあなたや誰かほかの人に再頒布すればよいわけです。

▶ 関連会社や子会社へのプログラムの提供

Q1：Is making and using multiple copies within one organization or company “distribution”?

A1：No, in that case the organization is just making the copies for itself. When the organization transfers copies to other organizations or individuals, that is distribution. In particular, distributing copies to contractors for use off-site is distribution.

▶ 機密保持契約の下でのプログラムの頒布

Q1：GPLは、改変されたバージョン、あるいはベータ版を機密保持契約(NDA)の下で頒布することを許可していますか？

A1：いいえ。GPLは、あなたからあなたのバージョンの複製を入手した人は皆、そのバージョンの複製を(改変の有無を問わず)再頒布する権利を有していると規定しています。あなたには、著作物の頒布に関してGPLより厳しい制限をかける権限はありません(前述の「B社がA社に対して機密保持義務を課している場合」(p.90)に対応)。

Q2：GPLは、私が機密保持契約の下で改変されたバージョンを開発することを許可していますか？

A2：はい。例えば、あなたはあるプログラムに改変を加え、そのあなたの改変した点をクライアントがOKを出すまで公開しないことに同意するという契約を受諾することができます。この場合、GPLによって保護されているコードがNDAの下で頒布されているということにはなりませんので、こういったことが可能になるのです。

また、あなたはクライアントに対して自分の改変した点をGPLの下で公開し、しかしそれらをクライアントがOKと言うまで他の人には公開しないということに合意することもできます。この場合も、GPLで保護されたコードがNDAやほかの追加的な制限の下で頒布されるということに

はなりません。

GPLはクライアントがあなたのバージョンを再頒布する権利を与えます。しかしこの筋書きでは、クライアントがその権利を行使しないと決定しているわけです（前述の「A社がB社に対して機密保持義務を課している場合」（p.90）に対応）。

(8部 2.4 「内部利用」 JEITA)

■ソース・コードの開示

ここでは、ソース・コードを開示しなければならない対象者（相手方）はいかなる範囲であるのかという点を検討し、併せて、ソース・コードの開示やライセンス条件の提示などの方法の例を紹介する。

まずソース・コード開示対象者に関し、Xが頒布したGPLコードを受領したYが、受領したコードをそのまま、または受領したコードに改変を加えGPLという「derivative work」となったコードを再頒布する場合、Yは、再頒布コードのソース・コードを開示しなければならないとされる（図3）。ここで、Yがソース・コード開示の義務を負うのは、Yから直接コードを受領したZだけであるのか、Zから（すなわちYから間接的に）コードを受領したWも含むのか、それとも、Yから直接・間接にコードを受領していない全くの第三者をも含むのか、という開示対象者の範囲が問題とされる。

また、GPLでは実際にソース・コードを開示することとなる場合に、ソース・コードの開示に加えてGPLライセンス条件の提示方法などの手続きも規定している。どのよう

に開示・提示をすべきかはGPLに一応の記載はされているが、実際に行うとなると具体的な手続きに迷うこともあると思われることから、具体例も紹介する。

ソース・コード開示の義務を負う対象者の範囲

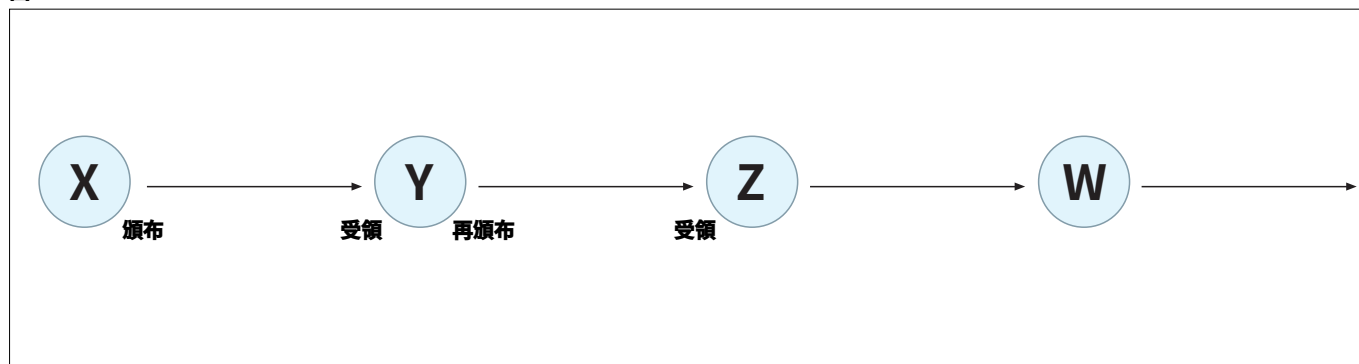
▶ 解釈

Yがソース・コード開示の義務を負うのは、①Z、Wに限らず、全くの第三者も含めてあらゆる者に対して、または、②GPL第3条a)の態様でソース・コードを引き渡す場合にはZに対してのみ、③GPL第3条b)の態様でソース・コードを引き渡す場合にはZとW（W以降のコード受領者も含む）に対してのみ、という解釈が成立し得ると思われる（②と③は両立し得る）。

▶ 理由

GPL第2条b)では、改変して頒布するコードについて「この契約書の条件に従って第三者（原文では、all third parties）へ無償で利用許諾しなければならない」と記載され、そして、FAQにおいては「すべての第三者とは、誰にでも無条件で、ということ」と記載されている。誰にでもGPLに従って利用許諾をする義務を負う以上、誰に対してもソース・コード開示の義務を何らかの形（例えば要求されればソース・コードを送付するなど）で負っていると考えることができなくもない。そうすると、Z以外の者すべて（Wなどや全くの第三者）に対してソース・コード開示の義務を負うと解釈できる余地もある。

図3



▶ 参考

● 第2条

あなたは自分の「プログラム」の複製物かその一部を改変して「プログラム」を基にした著作物を形成し、そのような改変点や著作物を上記第1節の定める条件の下で複製または頒布することができる。ただし、そのためには以下の条件すべてを満たしていなければならない：

b) 「プログラム」またはその一部を含む著作物、あるいは「プログラム」かその一部から派生した著作物を頒布あるいは発表する場合には、その全体をこの契約書の条件に従って第三者へ無償で利用許諾しなければならない。

● FAQ

Q：GPLによると、改変されたバージョンは、公開された場合「すべての第三者に…ライセンスされ」なければならないとされています。この場合第三者とは誰のことですか？
A：GPLの第2節では、改変されたバージョンを頒布する場合にはすべての第三者に対しGPLの下でライセンスされなければならないと述べています。「すべての第三者」とは、誰にでも無条件で、ということです。しかしこれは、あなたが彼らのために何か物質的に*してあげる*ことを要求するものではありません。これは、彼らがあなたのバージョンに関して、あなたからGPLの下でライセンスされているということだけを意味します。

一方、GPL第3条の記載を検討すると、GPL第3条a) の場合は、Zのみに開示すれば足りるとも解釈できる。GPL第3条では「オブジェクトコードないし実行形式」の「プログラム」に、a) 「『プログラム』に対応した完全かつ機械で読み取り可能なソースコード」を「添付する」ことさえすれば、GPL第3条に定められたソース・コード開示についての義務は果たされたと読める。ここで、GPL第3条からは「以下のうちどれか一つを実施」すれば足りると考えられることにも着目すべきである。そうすると、Zにのみソース・コードを開示すれば足りる、と解釈することも自然である。なお、GPL第3条b) では「いかなる第三者に対しても」と明示的に示されているのに対し、a) では第三者に何ら言及していない点に着目すると、a) の要件を

満たすためにはZにのみ開示すれば足り、Z以外の第三者に開示する義務はないとの解釈も受け入れやすい。

▶ 参考

● 第3条

あなたは上記第1節 [第1条] および第2節 [第2条] の条件に従い、「プログラム」(あるいは第2節 [第2条] における派生物) をオブジェクトコードないし実行形式で複製または頒布することができる。ただし、その場合あなたは以下のうちどれか一つを実施しなければならない：

a) 著作物に、「プログラム」に対応した完全かつ機械で読み取り可能なソースコードを添付する。ただし、ソースコードは上記第1節 [第1条] および第2節 [第2条] の条件に従いソフトウェアの交換で習慣的に使われる媒体で頒布しなければならない。あるいは、

b) 著作物に、いかなる第三者に対しても、「プログラム」に対応した完全かつ機械で読み取り可能なソースコードを、頒布に要する物理的コストを上回らない程度の手数料と引き換えに提供する旨述べた少なくとも3年間は有効な書面になった申し出を添える。ただし、ソースコードは上記第1節 [第1条] および第2節 [第2条] の条件に従いソフトウェアの交換で習慣的に使われる媒体で頒布しなければならない。

さらにFAQの記載を参考にすると、GPL第3条b) の場合には、Zに加えてWにも開示すれば足りるようにも読める。FAQでは「第三者」とは「そのオファーを持つ誰も」であり、また「バイナリを間接的に受け取った人々 [Wおよびそれ以降の受領者] もソースコードをあなた [Y] に注文することができる」と解説されている。従って、ソース・コード入手の手段が確保されているべきであるのは、オブジェクト・コード/オファーを入手した者だけに限定されていると解釈することも不自然ではないと思われる。

▶ 参考

● FAQ

Q：この「いかなる第三者に対しても法的に有効な書面によるオファー (written offer valid for any third party)」と

は何のことですか？ これは、世界中の誰もが、GPLが適用されているほぼどんなプログラムのソースでも手に入れられるということなののでしょうか？ [GPL第3条b)を引用すると、「いかなる第三者に対しても…提供する旨述べた少なくとも3年間は有効な書面になった申し出」(written offer, valid for at least three years, to give any third party)]

A：「いかなる第三者に対しても法的に有効」とは、そのオファーを持つ誰もが、あなたにオファーの内容に応じるよう要求する権利があるということです。

GPLには、バイナリをソースコード抜きで商業的に頒布する場合、あなたが後にソースコードを頒布する旨書かれた書面によるオファーを提供しなければならないとあります。ユーザがあなたから受け取ったバイナリを非商業的に再頒布するときには、この書面によるオファーの複製と一緒に渡さなければなりません。これは、バイナリを直接あなたから入手しなかった人々も、書面によるオファーと一緒にソースコードの複製を受け取ることができるということを意味します。

私たちが、オファーがいかなる第三者にとっても法的に有効であることを要求するのは、そうすることによって、バイナリを間接的に受け取った人々もソースコードをあなたに注文することができるからです。

ソース・コードの開示、ライセンス条件の表示の例

GPLに従ってコードを頒布するためには、上述のようにGPL第3条のa), 同b), 同c)のいずれかの態様でソース・コードを開示するとともに、GPL第1条および第2条に定める方法に従うことが要求されている。

▶ 参考

● 第1条

それぞれの複製物において適切な著作権表示と保証の否認声明(disclaimer of warranty)を目立つよう適切に掲載し、またこの契約書および一切の保証の不在に触れた告知すべてをそのまま残し、そしてこの契約書の複製物を『プログラム』のいかなる受領者にも『プログラム』と共に頒布する限り、あなたは『プログラム』のソースコードの複

製物を、あなたが受け取った通りの形で複製または頒布することができる。媒体は問わない。

あなたは、物理的に複製物を譲渡するという行為に関して手数料を課しても良いし、希望によっては手数料を取って交換における保護の保証を提供しても良い。

● 第2条

あなたは自分の『プログラム』の複製物かその一部を改変して『プログラム』を基にした著作物を形成し、そのような改変点や著作物を上記第1節の定める条件の下で複製または頒布することができる。ただし、そのためには以下の条件すべてを満たしていなければならない：

a) あなたがそれらのファイルを変更したということと変更した日時が良く分かるよう、改変されたファイルに告示しなければならない。

b) 『プログラム』またはその一部を含む著作物、あるいは『プログラム』かその一部から派生した著作物を頒布あるいは発表する場合には、その全体をこの契約書の条件に従って第三者へ無償で利用許諾しなければならない。

c) 改変されたプログラムが、通常実行する際に対話的にコマンドを読むようになっているならば、そのプログラムを最も一般的な方法で対話的に実行する際、適切な著作権表示、無保証であること（あるいはあなたが保証を提供するという）、ユーザがプログラムをこの契約書で述べた条件の下で頒布することができるということ、そしてこの契約書の複製物を閲覧するにはどうしたらよいかというユーザへの説明を含む告知が印刷されるか、あるいは画面に表示されるようにしなければならない（例外として、『プログラム』そのものは対話的であっても通常そのような告知を印刷しない場合には、『プログラム』を基にしたあなたの著作物にそのような告知を印刷させる必要はない）。

コードの頒布におけるソース・コードの開示、ライセンス条件の提示などの実例「WWWサイトから不特定多数の第三者に対しソフトウェアをダウンロードさせる」を以下に示す。

④WWWサイトにおいて、ダウンロード可能なソフトウェアのリストを表示する。

- 当該ソフトウェアが自社製品でないことを表示する
 - 当該ソフトウェアは無保証，無サポート製品であることを表示する
 - 当該ソフトウェアをダウンロードする際に表示する，「注意事項」「GNU GENERAL PUBLIC LICENSE」のページを読むよう表示する
- ②ダウンロードを希望するソフトウェアが選択されると，新たなページが開き「注意事項」を表示する（GPLの中で特に重要と考えられる事項を表示）。
- 当該ソフトウェアをダウンロードするためには，「注意事項」「GNU GENERAL PUBLIC LICENSE」の内容に同意する必要があることを表示する
 - 「注意事項」として「当該ソフトウェアが自社製品でないこと」「当該ソフトウェアは無保証，無サポート製品であること」「自社は当該ソフトウェアおよびその使用に関し，損害賠償責任を含め一切の責任を負わないこと」を表示する
 - 上記「注意事項」の内容への同意の意思表示の確認を求める
- ③「注意事項」の内容への同意の意思表示が確認されれば「GNU GENERAL PUBLIC LICENSE」を表示する。
- 当該ソフトウェアをダウンロードするためには，「GNU GENERAL PUBLIC LICENSE」への同意が必要であることを表示する
 - 「GNU GENERAL PUBLIC LICENSE」の内容を表示する
 - 上記「GNU GENERAL PUBLIC LICENSE」の内容への同意の意思表示の確認を求める
- ④「GNU GENERAL PUBLIC LICENSE」への同意の意思表示が確認されれば，ダウンロード可能画面を表示し，ダウンロードに関する情報（ディスク・フォルダの指定など）が入力されると，ソース・コード（およびオブジェクト・コード）のダウンロードを開始する。

（8部 2.4「ソース・コードの開示」 JEITA）

■第三者問題

ここでは，GPLの対象となるソフトウェアと，GPLの対

象とはならない独自のソフトウェアが混在する場合について整理を行う。これまで検討してきた通り，特に改変・加工を行わず，ただ単にGPLの対象となるソフトウェアそのものを取り扱う場合でも，免責，内部利用の範囲，ソース・コードの公開など難しい課題を含む。

これに加えて，固有の要素を含む独自のソフトウェアと組み合わせたソフトウェアを流通させようとする場合，法的な処理はさらに複雑な整理を必要とする。ここでは，特にこうした第三者のソフトウェアが混在する場合のライセンス上の課題を，それらを含むソフトウェアの利用者の立場に立って整理をする。

理論的考察

GPLに基づく許諾の対象となるプログラム（GPLソフト）は，GPLに規定された使用許諾条件に従って頒布することができるが，その際，当該使用許諾条件を頒布先に対してそのまま提示し，頒布先が，当該使用許諾条件に従って使用許諾を受けるようにしなければならないものとされている。そしてGPL第6条によると，GPLソフトを受領した者は，原ライセンサーから自動的にGPLに従った使用許諾を受けるものとされている。

すなわちGPLソフトの開発者Aが，当該GPLソフトをGPLに従って頒布し（このソフトウェアを「GPLソフトA」とする），Bが当該GPLソフトを受領して，一定の変更を加えて，再頒布し（このソフトウェアを「GPLソフトB」とする），CがGPLソフトBを受領した場合，Cは，GPLソフトAに関して，開発者Aから，直接使用許諾を受けることになる。従って，仮に，AのBに対する使用許諾が何らかの事情で失効したとしても，AのCに対する使用許諾は，有効に存続することになる。

一方，GPLソフトBに関しては，Bが2次著作物の著作者としての権利を取得する可能性がある。また，Bが別個独立の創作物をGPLソフトAに付加したにすぎない場合には，Bは，当該創作物に対して独自の権利を取得し，Aは，当該創作物に対して何らの権利を取得しないものと考えられる。このような場合，Bが保有する上記権利に関する許諾は，BがCに対して行う必要がある。その場合，B

がCに対してどのような使用許諾条件を提示できるのかが問題となる。

この点について、GPL第2条は、変更版の一部が別個独立のものであり、もともとのGPLソフトとは別のものとして頒布されるのであれば、GPLの適用がないものと定めている。従って、GPLソフトBがGPLソフトAの2次的著作物と解される場合には、Bは、GPLソフトBをGPLに従って頒布すべきことになるが、Bが新たに創作した創作物が、GPLソフトAとともにGPLソフトAとは別のものとして頒布される場合には、Bは、自由にその当該独自の創作物についての使用許諾条件を定めることもできるし、使用許諾の対価も徴収できるものと解される。

仮に、GPLソフトAを、BがGPLに従わずに別の使用許諾条件を定めて頒布した場合、GPL第4条に基づき、AのBに対する使用許諾は自動的に失効する。この場合、Cが、GPLソフトAに対する使用権を取得すると解されるか否かが問題となる。GPL第4項によれば、GPLに従ってGPLソフトを取得した者は、GPLを厳格に遵守している限りGPLソフトに対するライセンスを失わないものとされるが、そもそもCが「GPLに従ってGPLソフトを取得した者」と解し得るか否か疑問もあるからである。

なおGPLに違反があった場合には、自動的にライセンスが失効することになるので、違反者に対してGPLソフトの開発者は、著作権侵害または契約違反に基づく差し止め請求が可能となるが、ソース・コードの開示請求が認められるわけではないと考えられる。ソース・コードは、ライセンスの条件でしかなく、ソース・コードの開示請求が、GPLのライセンサーに認められているわけではないからである。しかし、プロジグループが発売したソフトウェア「DVDコンバータwith DivX PRO」がGPLソフトを使用していながらこれを明記せず、ソース・コードも開示していないと指摘された事案において、その解決策として、「DVDコンバータwith DivX PRO」のうち、「DVDConverter.exe」のソース・コードが開示された事例がある。また、エプソンコーワが、GPL違反の指摘を受けて、Linux向けドライバソフトの頒布をいったん中止したものの、ソース・コードを開示することによって、頒布を再開したという事例も

公表されている。

以下ではGPL違反で頒布された場合の効果に関して、具体的な設例を挙げて検討する。

設例による考察

▶ 設例

Aによって作成され、GPLの下で頒布されたプログラム「GPLソフトa」とリンクする「プログラムb」を企業Bが作成したが、その新たなプログラムは派生物（derivative work）ではない、とBは解釈し、GPLに基づかない条件で他の企業Cにそのプログラムbのライセンスを与えたとする。原著作者AがBをGPL違反で訴え、GPL違反の事実を確認したとすると、Cは、Aとの関係においてbを内部使用することができなくなるのか。Cが内部使用のほか、再改変したプログラムcをDに頒布していたらどうなるか（BとCとの間の契約上、CからDへの非GPL改変・再頒布は許されているものとする）。

▶ 解釈

- AはBによるプログラムbの頒布を差し止められる。
- AはBに対してプログラムbのソース・コード開示請求権を持つわけではないが「ソース公開を条件に差し止めを取り下げる」要求をすることは現実により得る。
- Cは、もしAからbについて直接GPLでライセンスを受けていたと仮定した場合には許される範囲でなら、bについて内部使用をすることができる。ただし、もともとCはBとの契約関係に基づいてbを改変・再頒布しているので、Cによる「プログラムc」の頒布がAとの関係で認められるかには問題がある。
- Dは、もしAから直接GPLでライセンスを受けていたと仮定した場合には許される範囲でなら、b、cについて内部使用をすることができる。

▶ 考察

GPL違反の頒布の効果は、GPL第4条に規定するところによると、複製・変更・サブライセンス・頒布が「失効す

る(void)」(我が国の民法上の無効と同じではない)とされている。ただし、GPLを遵守(in full compliance)する限り、その瑕疵のある頒布を受けた第三者はなおプログラム使用を認められる、とも規定される。この遵守が、GPLに従ってソースをすべて公開することまで意味するのかは明文で述べられていない。さらにはソースをすべて公開すれば改変・再頒布が可能となるかどうかは明文で述べられていない。

これを当てはめると、まずBによるbの頒布をAは差し止められる。Aは恐らくbのソースの公開を要求することになるが、これは法律上の開示請求権があるからではなく、差し止めによる現実の不利益がBにある場合に事実上要求できる場合もある、という意味である。Cによるbの内部使用については、GPLに反していない限りGPL第4項に沿って、Aはそれを認めることになる。非GPLのライセンスにより、ソースを見ることなく使用しているユーザーに対してソース公開を要求できるはずはないので、bの内部使用を認めるに際してAがソース公開をCに求めることもないはずである。しかし、改変プログラムcの作成が可能だという場合、Cはソースを持っていることになるので、Cに対してもAはソース公開を求める可能性はある。

さらに、Cによるcの改変・再頒布に際しては明文の救済規定がないので、Dはcについての使用を続けられるかが問題となる。BからCへの非GPLライセンスに基づくCからDへの改変プログラムのライセンスまで、GPL4条で救うことができるか不明確なためである。この問題に対しては、第4条の救済規定の目的が何かを踏まえ、どのように類推適用が可能かを考える必要がある。第4条の存在自体から示されるように、GPLは契約違反のプログラム使用を差し止める可能性をできるだけ広く確保しようとはしていない。GPLはコミュニティーによるプログラム改良の可能性を最大限に確保することを目的の1つとしており、その目的を達する手段として改変物を公開することを義務付けていると考えられる。従って、BとC、CとDの間にGPLによるライセンスという認識が当初なかったとしても、Dがプログラムを内部使用することをAは当初から認めていると解すべきであろう。

このように解釈すれば、未認識の商業プログラムが混入していたときのように使えなくなるわけではないという点で、最終ユーザーとしてのCやDの立場は、Aとの関係においては比較的守られているといってよいのではないか。ただ、CやDとBとの契約関係は、以上では議論されていない。B,C,Dが、当初bに関して結んでいた契約関係が当然にすべて無に帰するわけではないので、例えばCがBに対して錯誤を主張できるか、などの問題が残ることになる。

おわりに

本報告では、オープンソース・ソフトウェアに関する不要な誤解や不安を取り除き、開発者、利用者共にフェアな立場でオープンソース・ソフトウェアの活用に積極的に取り組める環境を提供することを目的に、現状および課題を網羅的に整理してきた。

最終的には、ソフトウェアは、その内容の質の高低によって採否が決められるべきものであり、オープンソース・ソフトウェアであるか否かによってその優劣が決まるものではない。しかし、ソフトウェアが、開発者、利用者などのさまざまな立場を超えて知見を結集する必要があるものであることを考えると、オープンソースという開発スタイルは、今後、何らかの形で市場で大きな流れをつくっていくものと考えられる。

そうした流れをつくっていくに当たって「8部 GPLに関する法的問題の整理」(pp. 61-97)で詳細に検討した通り、いわゆるGPLの不明確さは今後大きな課題となることが懸念される。オープンソースという新たな流れに我が国が積極的に取り組んでいくためには、このGPLを正確に理解し的確に使いこなす必要がある。政府としても、今後、GPLの正確な理解と利用者へのサポートをさらに強化することが期待される。本報告は、その良き第一歩となるであろう。

また、本報告でも強調した通り、オープンソース・ソフトウェアはGPLの対象となるソフトウェアに限定されるわけではない。今後の課題としては、我が国のソフトウェア開発の実態に即したさまざまなオープンソース・ライセンスについて、政府も含め官民一体となって検討していく必要があるであろう。

「オープンソース・ソフトウェア＝アンチ Windows OS」といったような単純な誤解を排し、今後、我が国でも、さまざまな形でオープンソース・ソフトウェアの開発や利用が積極化されることを期待したい。

委員会メンバー一覧

<委員長>

梶山 敬士 弁護士（虎ノ門南法律事務所）／ソフトウェア情報センター主任研究員

<委員>

上野 剛史 日本IBM株式会社 法務・知的所有権 弁理士
（社団法人電子情報技術産業協会より）
大熊 但由 オモイカネ株式会社 代表取締役社長 技術コンサルタント
太田 博之 株式会社東芝 知的財産部 デジタル著作権 担当課長
（社団法人電子情報技術産業協会より）
岡村 久道 弁護士（岡村・堀・中道法律事務所）／
近畿大学（産業法律情報研究所）兼任講師／奈良先端科学技術大学院大学講師
小川 憲久 弁護士（紀尾井坂法律特許事務所）／ソフトウェア情報センター特別研究員
小倉 秀夫 弁護士（東京平河法律事務所）／中央大学法学部兼任講師
片瀬 成識 東京ガス株式会社 ビジネスサービス本部 情報通信部
ITリノベーション推進グループ 主幹（技術担当係長）
（社団法人日本情報システムユーザー協会より）
古賀 慎二 株式会社アプリックス 技術部
（社団法人日本パーソナルソフトウェア協会より）
斎藤美登里 株式会社SRA 総務人事部 法務担当課長
（社団法人情報サービス産業協会より）
中島 達夫 早稲田大学理工学部情報学科教授／
日本エンベデッドリナックスコンソーシアム会長
比屋根一雄 三菱総合研究所 情報環境研究本部 情報技術研究部
数理解析技術チーム チームリーダー 主任研究員
平嶋 竜太 筑波大学大学院ビジネス科学研究科助教授
平野 高志 マイクロソフトアジアリミテッド 法務本部部長／弁護士
（社団法人日本パーソナルソフトウェア協会より）
水谷 直樹 弁護士（水谷法律特許事務所）／ソフトウェア情報センター特別研究員
宮下 佳之 弁護士（あさひ・狛法律事務所）
吉田 正夫 弁護士（三木・吉田法律特許事務所）／ソフトウェア情報センター主任研究員
和田 哲夫 学習院大学経済学部助教授

<オブザーバー>

新原 浩朗	経済産業省	商務情報政策局	情報経済課	課長
村上 敬亮	経済産業省	商務情報政策局	情報経済課	課長補佐
西江 昭博	経済産業省	商務情報政策局	情報経済課	課長補佐
池谷香次郎	経済産業省	商務情報政策局	情報経済課	権利保護係長
久米 孝	経済産業省	商務情報政策局	情報処理振興課	課長補佐
住谷 安史	経済産業省	商務情報政策局	情報処理振興課	課長補佐
権藤 龍光	弁護士（権藤法律特許事務所）			

<事務局>

柳沢 茂樹	ソフトウェア情報センター	調査研究部	部長
上村 哲弘	ソフトウェア情報センター	調査研究部調査研究課	主任

*所属および肩書は委員会が終了した2003年3月末日のもの

付録

GNU GENERAL PUBLIC LICENSE (Ver.2)

<http://www.gnu.org/licenses/gpl.html>

GNU General Public License

What to do if you see a possible GPL violation

Translations of the GPL

GPL Frequently Asked Questions

The GNU General Public License (GPL) in plain text format

The GNU General Public License (GPL) in Texinfo format

Table of Contents

GNU GENERAL PUBLIC LICENSE

Preamble

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

How to Apply These Terms to Your New Programs

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies

of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and

the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However,

nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in

the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Return to GNU's home page.

FSF & GNU inquiries & questions to gnu@gnu.org. Other ways to contact the FSF.

Comments on these web pages to webmasters@www.gnu.org, send other questions to gnu@gnu.org.

Copyright notice above.

Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA

Updated: Last modified: Sun Jul 15 13:13:30 CEST 2001

参考サイト

GPLの日本語訳については、以下のWebサイトが利用者の便宜を図るため仮訳の提供を行っています。

- GNU 一般公衆利用許諾契約書
<http://www.opensource.jp/gpl/gpl.ja.html>
- GNU 一般公有使用許諾書（圧縮ファイル）
<ftp://ftp.sra.co.jp/pub/gnu/local-fix/GPL2-j/>