

## GPL の派生物に関する「検討事項」

モンタビスタ・ソフトウェア社

Jason B. Wacha

1. あるプログラム（例えばアプリケーションプログラム）と GPL 対象プログラム（例えばライブラリプログラム）を静的リンクした場合、実行形式プログラム全体が GPL 対象となるか？

私は、このようなプログラムが「派生著作物」であるかどうかを分析するべきであると考えます。米国の法律には、派生著作物とは何かについて、いくつかの判断基準があります。しかし、GPL に関連するこの問題は裁判所で審理されたことがないため、プログラミング・コミュニティの慣行と、そのリーダーの見解に頼らざるをえません。プログラムが（静的または動的に）リンクされ、読み込まれる方法は、検討が必要な要素の 1 つです。しかし、その他の要素も同様に検討する必要があります。一般に、プログラムが（GPL 以外の）オリジナルのコードで記述され、他の GPL 著作物とは別のプログラムとして区別できる場合は、そのプログラムは任意の方法でライセンスできると考えられます。LGPL ライブラリにリンクするだけで、Linux 上で実行されるアプリケーションは、明らかに独立した、非派生著作物として受け入れられます。GPL ライブラリにリンクするアプリケーションは、GPL に基づいてライセンスされるアプリケーションと見なされます。この分野では、静的リンクと動的リンクは区別しません。ただし、カーネル・モジュールについて検討する場合には、リンクおよび読み込みの方法によって区別が生じると考えます。この点については、以下に詳しく説明します。

実行形式プログラム全体が GPL 対象プログラムの、米国著作権法にいう derivative work に相当し、GPL 対象プログラムの著作権者の権利が及ぶために使用許諾を得なければならないのか？

上の回答を参照してください。

それとも、GPL の当該条項を承諾したために、契約を遵守するために GPL 対象としなければならないのか？

GPL の第 2 条は、(1) すでに GPL の対象であるプログラムの再配布と、(2) 派生著作物と判断されたプログラムの配布のみに関係します。したがって、第 2 条の実際の趣旨は、「複製、変更および配布を管理する権利を著作権者に与えている米国著作権法に従わなければならない」ということです。第 2 条は、派生著作物が何であるかを定義していません。（実際、GPL のいかなる規定も派生著作物を定義していません。GPL は米国著作権法に完全に従っています。）

## 2 . あるプログラムと GPL 対象プログラムを動的リンクする場合、そのプログラムは GPL 対象となるか？

上の回答#1 を参照してください。アプリケーション分野では、リンクの方法によって判断されることはないと考えます。カーネル・モジュール/ドライブの分野では、動的リンクによって専用ライセンスが利用可能になることがあります。

## 3 . Linux カーネルのダイナミックロードブルモジュールは GPL 対象となるか？

実行時に動的に読み込まれるモジュールは、GPL 以外の方式でライセンスできると考えます。その代わりに、カーネルに静的にバインドされていて、起動時に読み込まれるモジュールは、実際上、別個の著作物としては区別されず、カーネル全体と共に GPL に基づいてライセンスする必要があると考えます。GPL/Linux コミュニティーは、一般にこの立場を支持しています。

この慣行 / 要件は、公開フォーラムで十分に議論されています。おそらく最も強い影響を持つ発言者であるところの Linus Torvalds 氏は、次のように記述しています。「実行時に読み込まれる部分が、その時点で GPL の派生著作物であるモジュールである場合には、GPL の対象となるでしょうか。(別のオペレーティング・システム用に) ドライバーを開発した人たちが、Linux 用のバイナリー・コードを提供するために進んで再コンパイルを行ったことがあります。この時点で、モラル上の理由から、このような状況に GPL は適用できないと判断しました。最終的に、システム・コールはカーネルに対するリンクとは見なされないと私たちは判断しました(あるいは、私が命令しました)。つまり、Linux で実行されているすべてのプログラムが、GPL の対象となるとは見なされないということです。このため、この商業ベンダーは、GPL について心配することなく、Linux 用のプログラムを開発できます。結局、読み込みのために通常のインターフェイスのみを使用する場合、モジュール・メーカーは専用モジュールを開発することができます。」

後に、Linus 氏は次のように記述しています。「カーネル・ライセンスには、モジュールに GPL を適用しないことを許可する規定はありません。モジュールに GPL を適用しないことを許可する唯一の規定は著作権法であり、特に『派生著作物』の問題です」。もちろん、反対も真実です。カーネル・ライセンスには、モジュールを GPL でライセンスすることを要求する規定はありません。

もう 1 人の著名な人物、Eric Raymond 氏は、さらに一歩進めて、Linux カーネル・ライセンスに次の一節を含む新しい文言を追加するよう Torvalds 氏に正式に求めました。「カーネルがビルドされた後に、実行時に読み込まれるカーネル・モジュールは、派生著作物とは『見なされません』」。

この件に関してはまだ議論の余地がありますが、Linux コミュニティーは一般に、読み込み可能なモジュールに対して専用ライセンスを使用するアイデアを支持しています(また、明

らかに実施しています)。ただし、そのモジュールがその他の点で派生著作物と見なされないことが前提です。

#### 4 . Linux のデバイスドライバはダイナミックロードブルモジュールの形態をとることが多いが、GPL 対象となるか？

私は、ドライバーについてモジュールと同様に考えています。一般にカーネル・モジュールは、カーネルの機能を拡張するプログラムであると見なしています。ドライバーは、カーネル・モジュールの一種にすぎないと考えます(その他の種類には、ファイル・システムなどが考えられます)。

#### 5 . Linux のアプリケーションプログラムは GPL 対象とならないか？

Linux 用のアプリケーション・プログラムは、そのプログラムが GPL プログラムの派生著作物である場合や、そのプログラムの作者が GPL に基づいてプログラムをライセンスすることを決定した場合にのみ、GPL の対象とするべきです。プログラマーは、Microsoft に対して権利を放棄せずに、Windows<sup>®</sup>で実行されるアプリケーションを開発できます。プログラマーは、Apple に対して権利を放棄せずに、Macintosh<sup>®</sup>で実行されるアプリケーションを開発できます。同様に、プログラマーが Linux で実行されるアプリケーションを開発しても、GPL の対象とはなりません。Linux のカーネル・ライセンスを規定している GPL の前文で、Linus 氏自身もこの立場を取っています。この「コピーされた」文言は、Linux カーネル・システム呼び出しインターフェイス(ユーザー・アプリケーションの主要なエントリー・ポイント)を使用することによって、そのプログラムが派生著作物となることはないと規定しています。

もちろん、Windows<sup>®</sup>用のプログラムを開発し、そのプログラムが Microsoft 所有のコードの派生著作物である場合は、Microsoft がそのライセンスを管理します。これは、ソフトウェアの世界では新しいことではありません。GPL プログラムの派生著作物であるプログラムを開発した場合は、GPL の規定が適用されます。ただし、FSF(フリー・ソフトウェア財団)でさえも Linux 開発者の商業上のニーズを認識しており、LGPL(劣等一般公共使用許諾契約書)の存在は、Linux 用アプリケーションに専用ライセンスを許可する意図を示しています。

Linus Torvalds が Linux のアプリケーションプログラムは GPL 対象とならないと宣言しているが、これがないと、GPL 対象になるのか？

Torvalds 氏は GPL を管理せず、米国著作権法(またはその他の著作権法)も管理していません。彼の発言は重要ですが、検討すべき指針にすぎません。米国著作権法は、著作物の複製、変更、再配布を管理する権利を Torvalds 氏に与えています。彼はその権利を渡すための特定の著作権ライセンスとして、GPL を選択しました。ただし、いかなる著作権者も、自ら何が派生著作物であるかを(また、法律により、何が GPL の対象となるか

を)判断することはできません。しかし、GPL に基づいてプログラムをライセンスするという選択によって、著作権者は次のことを表明します。「プログラムが派生著作物である場合、そのプログラムはこのライセンスの条件に基づいて配布しなければなりません」。

## 6 . Linux を使用する組込み機器の場合はアプリケーションを含めてひとつの実行形式プログラムとなると思われるが、全体が GPL 対象となるのか？

私の知るかぎり、Linux カーネル、モジュール、ライブラリー、アプリケーション・コードから構成される配置済みの組み込み Linux アプリケーションを 1 つのプログラムと解釈することは、Linux カーネル・ライセンスまたは GPL で受け入れられてはいません。また、MontaVista は、このようなコンポーネントが 1 つの著作物を構成するという立場を取っていません。この立場は、法律と技術の両方の議論によって防御できます。法律に関して、このような集合体のさまざまなコンポーネントは、それぞれ独立した存在および派生元を有し、エンド・アプリケーション・システムに同時に配置されているにすぎません。この配置は、個々の著作物としての独立した状態に影響を与えません。技術面では、このような集合体内のカーネルは、他のコンポーネントなしで実行および機能することができ、多くの場合、アプリケーション・コード、ライブラリー、およびデバイス・ドライバさえも、独立した存在であり、他の環境で実行できることがあります。このように、同時に連係して実行されることによって、それぞれの独立した技術的存在が否定されることはありません。

## 7 . 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 により頒布されるヘッダ・ファイルを使用する場合、そのプログラムは GPL 対象となるか？

ヘッダー・ファイルはインターフェイスを定義し、そのファイルが属するプログラムについて、ライセンス・ステータスなどのその他の情報を宣言します。ヘッダー・ファイル自体はプログラムではなく、一般に知的所有権 (IP) のステータスを与えられません。さらに、インターフェイス自体も IP ステータスを持たず、ライセンスまたは著作権の対象となりません。ただし、このようなインターフェイスを実装するコードは、著作権やライセンスの対象となることがあります。このように、ヘッダーはそれを記述するコードとは異なるライセンス・ステータスを取得することはできず、実際上、独自のライセンス・ステータスは有しません。

実際のコード (および知的所有権) が含まれるようにヘッダー・ファイルが「読み込まれる」場合、このようなファイルは、一般に受け入れられないプログラミング慣行によって作成された例外的なものです (または、そのヘッダー・ファイルをインクルードするコードを故意に「汚染」しようとしている可能性もあります)。

有効な例外としては、C++を使用して、または gcc コンパイラーの C の拡張機能を通じて、マクロとインライン関数を実装するヘッダーなどがあります。ただし、このような場合、プログラムの断片（マクロとインライン関数）の組み込みは、実装上の選択肢であり、そのインターフェイスを（派生またはその他の）知的所有権と見なすべきことを示すものではありません。