

プログラムの権利と オープンソースソフトウェア

ソフトウェアの権利

- ソフトウェア(プログラム)の表現(ソースコード)は、文学作品と同様、**著作権**で保護される。
- またそのアイデアは、青色LEDやオブジーボ(癌の薬)と同様に、**特許**で保護される。
- 一方、これらの権利を放棄したように見える**オープンソースソフトウェア**という物もある。
- この講義ではこれらについて学習する。

著作権と特許、保護対象の違い

著作権の対象 ⇨

ソースコードの場合、文字列そのもの

```
int main() {  
...  
...  
}
```

特許の対象 ⇨

ソフトウェアやプログラムのアイデア

Step1: 最初に...を行う。

Step2: 次に...を行う。

Step3: ...と...の結果を比較して、...を行う

ただし、単に、アルゴリズムや数式として説明したのでは特許にならない。コンピュータで実行されるプログラムとして説明する必要がある。

著作権と特許、保護内容の違い

著作権

特許

作者の死後70年まで

出願から20年

登録の手続きなく
権利化される(無方式主義)

登録が必要で
維持に費用を要する

表現が保護される
(狭い)

アイデアが保護される
(広い)

複製・頒布・実行・改変などが
保護されるという意味では同じ

当初、ソフトウェアに権利はなかった

- 1970年代まで
 - 大学や研究機関で開発されるソフトウェアは、**パブリックドメイン**^[1]で共有することが普通であった。
 - 米国のハッカーたちが作った**"LISP"**や**"Emacs"**は共有されていた。
 - 民間では、ソフトウェアをハードウェアと一体で販売していた。納入先ごとに、ソフトウェアを開発していたとしても、それらはハードウェアの一部であって、それだけでは価値を認めていなかった。
- その後、ソフトウェアを商品とする人が登場し、権利を主張するようになる。

[1] パブリックドメイン: 著作物や発明などについて知的財産権が発生していない状態、または、権利が消滅した状態のこと。

ビル・ゲイツのOpen Letter

- 1975年頃に始まった個人向けコンピュータ市場（当時、Altair, Appleなどパソコンが販売されていた）では、有償のソフトウェアも存在した。しかし、それらは勝手にコピーされていた（やりたい放題であった）。
 - **ビル・ゲイツ**は自分のBASICインタプリタ（有償, \$30から\$60）がコピーされることに対してOpen Letterを出して抗議している^[1]。
 - 「ホビー向けコンピュータ市場で最も欠けていることは、素晴らしいソフトウェアである。そこで私はAltair BASICというものを作った。ところが驚くことに、ほとんどのAltairユーザはわれわれの製品を購入していない。購入しているのは約10パーセントだ。ハードウェアにはお金を払うのに、ソフトウェアには払わない。それはフェアか？誰が無料のためにプロフェッショナルな仕事をするのか？君たちがやっていることは窃盗だ。」^[2]

[1] ビル・ゲイツの「ホビーストへの公開書簡」(https://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists)

[2] 「ハッカー精神とは何か」、ネットを支えるオープンソースソフトウェアの進化、p.159、角川インターネット講座、平成26年

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds \$40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than \$2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates
Bill Gates
General Partner, Micro-Soft

ソフトウェアの権利化

- 1980年頃からコンピュータ・プログラムを著作権で保護したり、ライセンス契約という形で利用許諾できるようになった。
 - **IBM産業スパイ事件** (1982年) : IBMのOSを盗んだとして日立と三菱の社員が逮捕された事件。
 - マイクロソフト社がIBM PC用のOS **MS-DOS**を**IBM**や**IBM互換機**に**ライセンス**した(1981年以降)。

ソフトウェアライセンス

- ソフトウェアの権利者(ライセンサ)が、一般ユーザや他のソフトウェア開発者(ライセンシ)に与える許可やそれに伴う条項。
- ソフトウェアの**実行に関するライセンス**
 - 実行許可、インストール可能な台数、**リバースエンジニアリング**を禁止する、などが規定される。
- ソフトウェアの**開発に関するライセンス**
 - ソースコードの提供かバイナリの提供か、
 - 完成したソフトウェアを頒布する時の扱い、などが規定される。

ライセンス契約の成立

- 他者が権利を持つソフトウェアを、許諾を受けることなく利用・使用することはできない。
- ライセンス契約は、次のように実行されることが多い。
 - 契約書への署名
 - シュリンクラップを破るや、梱包・封筒を開ける
 - ダウンロード時、ライセンス条項を確認するチェックボックスにチェックを入れる
- 後で述べるオープンソースソフトウェアの場合、ソフトウェアを受け取るだけで、そのライセンスに合意したとみなすことが一般的。

商用ソフトウェアの普及

- 権利が保護された商用ソフトウェアを**プロプライエタリ(proprietary)・ソフトウェア**^[1]という
 - 商用UNIX、MS-DOS、Mac OS、Windows、などはプロプライエタリ・ソフトウェアである。
 - 通常、ソースコードは提供されない。実行だけができる。コピーは必要最小限のみ可能。
- 商用ソフトウェアが成功するによって、ソースコードを共有しお互いに利用し合う文化は、大学などでのマイナーな存在になった。

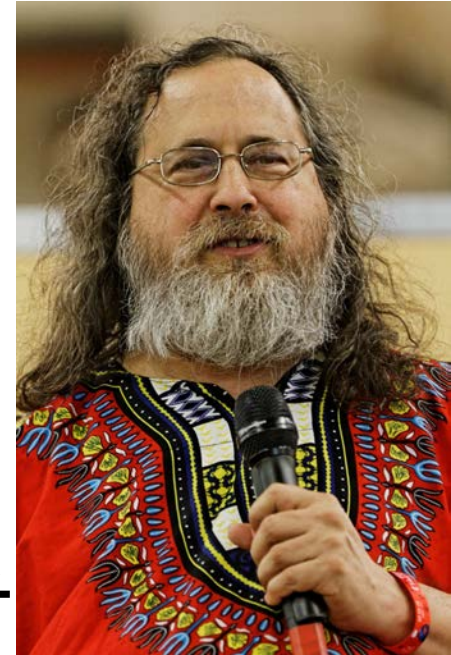
[1] 「クローズドソフトウェア」という呼び方もある。

リチャード・ストールマン(1953年～) の登場

- もともとMIT AIラボのプログラマ
- Emacs, gccなどの開発者
- GNUプロジェクト(1984年～)やフリーソフトウェア活動の創始者
- ストールマンによる、GNUプロジェクト最初の声明が

<http://www.gnu.org/gnu/initial-announcement.html>

にあるので、読んでみてください。



ストールマンのフリーソフトウェア財団

- ソフトウェアを作成、頒布、改変する自由をユーザに広く遍く推し進めることを目指す団体。
 - 商用ソフトウェアはオブジェクトコードを実行できるだけであり、開発者から見ると不便である。
 - この財団は**フリーソフトウェア**の開発を推進する。これは、ソースコードを公開し変更・再配布可能なソフトウェアである。ただし、改変し再配布するソフトウェアもフリーソフトウェアにする必要がある。このようなライセンスを**コピーレフト**とよぶ。
 - **GNUプロジェクト**: Unix互換OSをフリーソフトウェアとして開発することを目指した。gcc, emacsなどが完成した。OS本体は未完成であった。
 - **Linux**: **リーナス・トーバルズ**が開発を始めたUnix-likeなOS。GNUのツールを多く採用している。Linuxはフリーソフトウェアである。

フリーソフトウェア活動の影響

- 一般企業にとって、フリーソフトウェアという名称やコピーレフトのライセンスは不都合で受け入れがたいものであった。
- フリーソフトウェア活動の影響は限定的であった。そのような状況で、時代は1990年代に移っていく。

Linux

- リーナス・トーバルズが、1991年から開発を始めたUnix系OS。GNUのソフトウェアを取り込むことで完成した。
- 多くのハッカーが自由意志で開発に参加した。
バザールモデルと呼ばれるオープンソースソフト(OSS)の開発モデルになった。
 - エリック・レイモンドが「伽藍とバザール」という論文で、ソフトウェア開発の2つのモデルを「伽藍」モデルと「バザール」モデルとして説明している。
 - バザールはオープンモデル、伽藍はクローズドモデル

Linux(続き)

- 現在、LinuxはパソコンやデジタルTV、スマホなどで広く使われているOSである。
- Kernelだけで5,000万行(ほとんどC言語)、1ヶ月あたり1,000人以上が10,000箇所に変更を加えている^[1]。
- Linuxカーネルは2ヶ月半程度でメジャーリリースを繰り返す。バグ修正は現在と直前のメジャーリリースだけに対応。ただし、LTS(Long Term Support)版については長期サポートする。

[1] OSSコミュニティの活性度を測るサイトBlack Duck Open Hub(<https://www.openhub.net>)で、Linux Kernelを検索した結果(2019年9月6日)。

(参考) 伽藍とバザール

- エリック・レイモンドがソフトウェア開発手法を、**伽藍(カテドラル)モデル**と**バザールモデル**という2種類に分けて説明した。
- 伽藍モデルは、**トップダウン設計**であり、全体を緻密に設計した上で、細部の設計を進める方法。
- バザールモデルは、**ボトムアップ設計**である。ソースコードをインターネットで公開し、不特定多数の利用者・開発者により設計・実装を進める。
- Linuxカーネルがバザールモデルで成功したことにより、オープンソースの開発手法としてバザールモデルが定着する。
- <https://ja.wikipedia.org/wiki/伽藍とバザール>を参照のこと。

Netscape

- **Netscape**は最初期（1994年頃から）のWebブラウザ。一時期は80%のシェアを得た。1998年ごろにはマイクロソフトのIEにシェアを奪われ、低迷していた。
 - 苦肉の策として、1998年にソースコードを公開し、企業として初めてOSS^[1]の開発モデルを提案した。
 - Netscapeという製品は2007年で終了する。OSSとして開発されたブラウザはFirefoxに受け継がれる。
 - 2018年現在、最もシェアが高いブラウザはGoogle Chromeである。これはChromiumというOSSプロジェクトの成果物である。

[1] OSS: Open Source Software, オープンソースソフトウェア、詳細は後述。

オープンソース・イニシアティブ (OSI)

- オープンソースソフトウェア (OSS) の発展・推進を目的に、1998年に設立された団体。フリーソフトウェア財団とともに、ソフトウェアの権利擁護団体である。
 - オープンソースは、フリーソフトウェアの概念も包括するように定義されている。
 - 企業での商用利用に配慮したものであった。
 - フリーソフトウェアの考え方が普及しなかったのに対し、オープンソースは社会に広く受け入れられた。

オープンソースイニシアティブによる OSSの定義

オープンソースの定義 八田 真行訳、2004年2月21日バージョン 1.9

はじめに

「オープンソース」とは、単にソースコードが入手できるということだけを意味するものではありません。「オープンソース」であるプログラムの頒布条件は、以下の基準を満たしていなければなりません。

1. 再頒布の自由

「オープンソース」であるライセンス(以下「ライセンス」と略)は、出自の様々なプログラムを集めたソフトウェア頒布物(ディストリビューション)の一部として、ソフトウェアを販売あるいは無料で頒布することを制限してはなりません。ライセンスは、このような販売に関して印税その他の報酬を要求してはなりません。

2. ソースコード

「オープンソース」であるプログラムはソースコードを含んでいなければならず、コンパイル済形式と同様にソースコードでの頒布も許可されていなければなりません。何らかの事情でソースコードと共に頒布しない場合には、ソースコードを複製に要するコストとして妥当な額程度の費用で入手できる方法を用意し、それをはっきりと公表しなければなりません。方法として好ましいのはインターネットを通じての無料ダウンロードです。ソースコードは、プログラマがプログラムを変更しやすい形態でなければなりません。意図的にソースコードを分かりにくくすることは許されませんし、プリプロセッサや変換プログラムの出力のような中間形式は認められません。

3. 派生ソフトウェア

ライセンスは、ソフトウェアの変更と派生ソフトウェアの作成、並びに派生ソフトウェアを元のソフトウェアと同じライセンスの下で頒布することを許可しなければなりません。

(次ページに続く)

4. 作者のソースコードの完全性(integrity)

バイナリ構築の際にプログラムを変更するため、ソースコードと一緒に「パッチファイル」を頒布することを認める場合 **に限り**、ライセンスによって変更されたソースコードの頒布を制限することができます。ライセンスは、変更されたソースコードから構築されたソフトウェアの頒布を明確に許可していなければなりません、派生ソフトウェアに元のソフトウェアとは異なる名前やバージョン番号をつけるよう義務付けるのは構いません。

5. 個人やグループに対する差別の禁止

ライセンスは特定の個人やグループを差別してはなりません。

6. 利用する分野(fields of endeavor)に対する差別の禁止

ライセンスはある特定の分野でプログラムを使うことを制限してはなりません。例えば、プログラムの企業での使用や、遺伝子研究の分野での使用を制限してはなりません。

7. ライセンスの分配(distribution)

プログラムに付随する権利はそのプログラムが再頒布された者全てに等しく認められなければならない、彼らは何らかの追加的ライセンスに同意することを必要としてはなりません。

8. 特定製品でのみ有効なライセンスの禁止

プログラムに付与された権利は、それがあある特定のソフトウェア頒布物の一部であるということに依存するものであってはなりません。プログラムをその頒布物から取り出したとしても、そのプログラム自身のライセンスの範囲内で使用あるいは頒布される限り、プログラムが再頒布される全ての人々が、元のソフトウェア頒布物において与えられていた権利と同等の権利を有することを保証しなければなりません。

9. 他のソフトウェアを制限するライセンスの禁止

ライセンスはそのソフトウェアと共に頒布される他のソフトウェアに制限を設けてはなりません。例えば、ライセンスは同じ媒体で頒布される他のプログラムが全てオープンソースソフトウェアであることを要求してはなりません。

10. ライセンスは技術中立的でなければならない

ライセンス中に、特定の技術やインターフェースの様式に強く依存するような規定があつてはなりません。

<http://www.opensource.jp/osd/osd-Japanese.html>

文献^[1]によるOSSの定義

- 改変する自由がある。
- 頒布^[2]する自由がある。
- 著作権者が提示する条件を満たす必要がある。
- 著作権者に連絡し許諾を得る必要はない。
- 対価を支払う必要はない。

[1] 「OSSライセンス教科書」、上田理、技術評論社、2018年。

[2] 頒布はdistributionの訳語で、この資料では配布と同意。

OSS開発モデル普及の要因

- ソフトウェア開発が大規模になる中で、全てを自社開発することが難しくなった。そこで、自社の差別化技術以外の部分で高度な技術を要するものを、OSSとして導入したい。そのようなものには、OS、Webブラウザ、言語処理系などの、汎用ソフトウェアが多い。
- OSSは、過去の資産を有効に利用することで、低コストで高機能なソフトウェアの開発を可能にする。特に、インターネット・ウェブサービスのベンチャ企業がOSSを利用した開発に熱心である。
- ソースコードは無償、または、廉価である。
- ソースコードが提供されるので、カスタマイズが容易である。
- OSSの開発に参加することで、プログラマーとして実績を残すチャンスである。またOSSコミュニティに貢献することで、そのOSSをより良いものにできる可能性がある。
- OSSは、GitHubのような開発基盤(プログラマのためのSNS)を使うことで効率的な開発が可能になっている。

代表的なOSS

- OS: Linux, FreeBSD, Android
- プログラミング言語: シェル, Perl, Ruby, Python, PHP
- コンパイラ: Gcc、LLVM
- Web: Apache, Samba, Firefox, WordPress
- メール: Sendmail, Postfix, Thunderbird
- データベース: PostgreSQL, MySQL,
- 統合開発環境: Eclipse, NetBeans
- 仮想基盤: Xen, OpenStack
- データ処理: Hadoop

OSSに対する誤解と真実

- ソースコードを公開すればOSSだ。
 - それに加えて、誰でも自由に利用(複製、頒布、改良)できるということがある。
- OSSを使って商品を開発すると、自社開発したソースコードを公開しなければならない。
 - 使うOSSのライセンス条件による。
 - ソースコードの公開が必要なライセンスもある。
- OSSは無料で無条件で使えるソフトウェアだ。
 - 概ね無料で使えるが、無条件ではない。ライセンス条件に従わねばならない。
- OSSの利用責任は利用者にある
 - そのとおり。

OSSのライセンス

- ソフトウェアのライセンスは「利用許諾契約」とも呼ばれるものである。通常、あまり注意することもなく受け入れているものである。しかし、OSSを組込む場合には、注意が必要。
- OSSのライセンスは実際には非常に多数の種類がある。
 - それらはOSSの定義に合致するが、細部は異なっている。
 - ここでは、参考文献[3]に従って、**寛容型ライセンス**と**互惠型ライセンス**に分類して説明する。

寛容型ライセンス

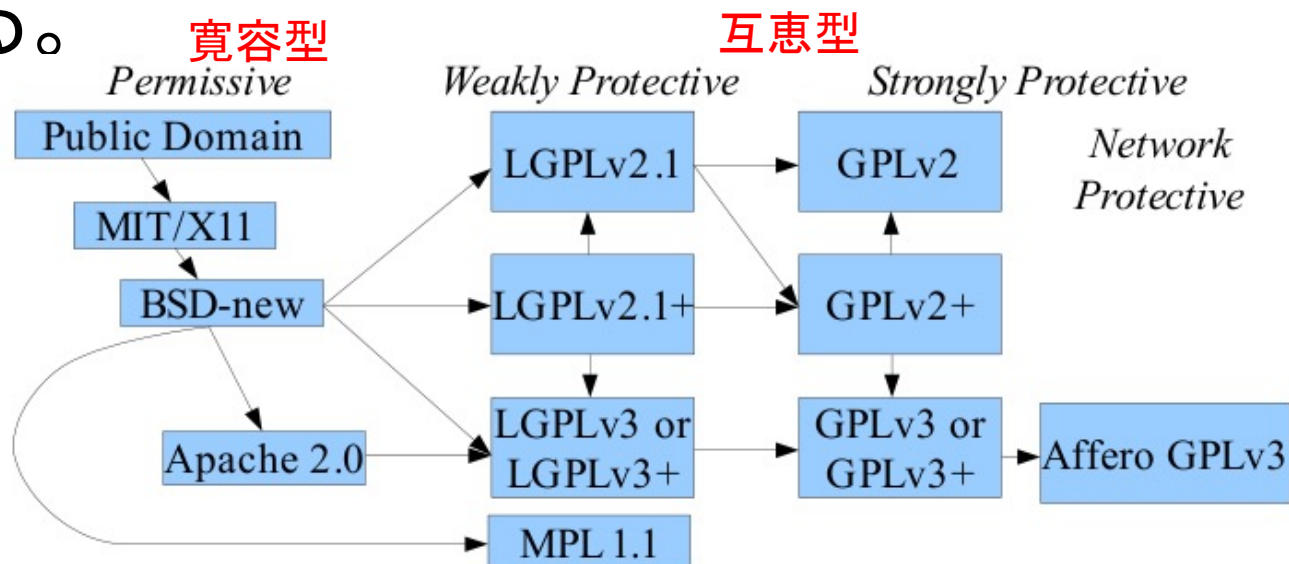
- 単純な事項を守ればOSSの自由な改変や頒布を認めるライセンス。概ね、著作権表記とライセンス表記をすることが必要。ソースコード開示は必要ない(してもよい)。
 - MITライセンス、BSDライセンス、Apacheライセンスなどがある
 - 著作権表記は“Copyright 2019 Hiroshi Kano”のようなもの。
 - Apacheライセンスには、コントリビュートした部分に利用者の特許があれば、無条件で許諾しなければならない、という条項がある。

互惠型ライセンス(コピーレフト型)

- 利用者(派生ソフトウェアの開発者)にもOSSの開発者と同等の貢献を要請するもの。GNU GPL/LGPLを例に説明する。
 - LinuxはGPLバージョン2で利用許諾されている。
 - ソースコードの開示(利用者の改変部分も含めて)が義務づけられている。
 - LGPLには、開示の必要がない条件もある。
 - 派生ソフトウェアとは、OSS自体を修正・追加した物や、静的/動的リンクで呼び出すことで一体化したものを指す。
 - 派生ソフトウェアの範囲は、ライセンスによって異なる。
 - 訴訟が発生しており、利用を回避する傾向が見られる。

異なるライセンスのOSSを同時に利用できるか？

- ライセンスに互換性がないOSSを同時に利用することはできない。
 - 例えばGPL2と4項型BSDは両立しない。
- ライセンスが矛盾していなければ同時に利用できる。



OSSの文化^[1]

- 開発方針や機能追加についての発言は、誰でもあれ、平等に扱われる。
- 情報を一人で囲い込まず、横展開する人が尊ばれる。
- 社会的な地位や経験より、積極的に参加する人が尊ばれる。
- 他人が進めている開発を進んでサポートする協業も尊ばれる。
- 開発したソフトウェアの成果は「参加者全員の成果」とみなされる。

OSSの開発に参加する理由

- 非金銭的なもの
 - OSSのコミュニティが住みやすい
 - 勉強する機会を得ることができる
 - 名声を得る
- 金銭的なもの
 - 一企業では収益を上げることが難しい言語系ソフトや、共通基盤になるインターネット系ソフトの開発が安価にできる。
 - 転職に有利に利用できる

OSSとハッカー

- 「ハッカー」という言葉はコンピュータ技術(ソフト／ハード)の知識を極める人、という程度の意味である。OSSの開発者はハッカーの中のハッカーである。
- エリック・レイモンドによる「ハッカーになろう(How to become a hacker)」という小文の訳が<https://cruel.org/freeware/hacker.html>にある。その中の「ハッカー的心構え」と「基本的なハッキング技術」のタイトル部分を示す
 - この世界は解決を待っている魅力的な問題でいっぱいだ。
 - 同じ問題を二度解くような無駄は嫌だ。
 - 退屈と単純作業は悪。
 - 自由は善。
 - 心構えは技能の代用にならない。
 - プログラミングを身につけること。
 - OSSのUNIX類のひとつを入手し、使い方と動かしかたをおぼえること。
 - WWWの使い方を学び、HTMLを書くこと。
 - まともな英語ができないならば、身につけること。
- この小文によると、勉強すべき言語はPython, Java, C/C++, Perl, LISP。

OSSでのソースコード管理

- ソースコード管理システム
 - CVS、Subversion、Gitなどのバージョン管理システムを利用する。
- コミッターとコミット権
- コミュニティで実力が認められること

オープンソースハードウェア

- 設計図や部品リスト、ソースコードなど再現に必要な情報一式をインターネット上に公開しているハードウェアを「オープンソースハードウェア」という。
 - ArduinoというCPUと、それに接続できる周辺機器
 - オープンソース3Dプリンタ
 - など、他にもたくさんある。

オープンソース3Dプリンタ

- 3Dプリンタは、機械部品を少量自作する機械として、オープンソースハードウェアの開発に必須のものである。
- 熱溶融積層式の3Dプリンタが、2010年台になって、急激に普及した背景は2つある。
 - 熱溶融積層3Dプリンタの基本特許が切れ、技術がオープンになった。
 - その3Dプリンタ自身がオープンソースハードウェア化され、誰でも自作できるようになった。

参考文献

- [1] 「ネットを支えるオープンソース」、まつもとゆきひろ監修、角川インターネット講座2、平成26年。
- [2] 「ネットで進化する人類」、伊藤穰一監修、角川インターネット講座15、平成27年。
- [3] 「OSSライセンス教科書」、上田理、技術評論社、2018年。
- [4] 「知る、読む、使う！ オープンソースライセンス」、可知 豊、達人出版会、2013年。