

# 東京エリア デビアン 勉強会



Debian勉強会幹事 上川純一

2008年3月15日

# 1 Introduction

上川 純一

今月の Debian 勉強会へようこそ。これから Debian の世界にあしを踏み入れるという方も、すでにどっぷりとつかっているという方も、月に一回 Debian について語りませんか？

Debian 勉強会の目的は下記です。

- Debian Developer (開発者) の育成。
- 日本語での「開発に関する情報」を整理してまとめ、アップデートする。
- 場 の提供。
  - 普段ばらばらな場所にいる人々が face-to-face で出会える場を提供する。
  - Debian のためになることを語る場を提供する。
  - Debian について語る場を提供する。

Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりとするスーパーハッカーになった姿を妄想しています。情報の共有・活用を通して Debian の今後の能動的な展開への土台として、「場」としての空間を提供するのが目的です。

以上を目的とした、2008 年アジェンダです：

1. 新年会「気合を入れる」
2. Open Source Conference Tokyo (3/1)
3. データだけのパッケージを作成してみる、ライセンスの考え方 (David Smith)
4. バイナリーつのパッケージを作成してみる (吉田@板橋)  
バージョン管理ツールを使い Debian パッケージを管理する (git)  
アップストリームの扱い (svn/git/cvs)(岩松 信洋さん)
5. バイナリーの分けたパッケージの作成。(前田さん)  
バイナリーの分け方の考え方、アップグレードなどの運用とか。
6. パッケージ作成 (dpatch/debhelper で作成するパッケージ)(小林儀匡さん)  
man の書き方 (roff or docbook)(でんさん)
7. パッケージ作成 (kernel patch、kernel module)、Debconf 発表練習
8. Debconf アルゼンチン、共有ライブラリパッケージ作成
9. Open Source Conference Tokyo/Fall、デーモン系のパッケージの作成、latex、 emacs-lisp、フォントパッケージ
10. パッケージの cross-compile の方法、amd64 上で i386 のパッケージとか、OSC-Fall 報告会、Debconf 報告会
11. 国際化 po-debconf / po 化 / DDTP
12. 忘年会

# 会 勉 強 会 ト ビ ア ン ト

---

---

## 目次

1	Introduction	1
2	事前課題	3
3	Debian Trivia Quiz	8
4	最近の Debian 関連のミーティング報告	9
5	Open Source Conference 2008 Spring 報告	10
6	Debian パッケージ作成：データだけのパッケージ	12
7	Debian パッケージにおけるライセンスの取扱い	16

---

## 2 事前課題

上川 純一



今回の事前課題は以下から自由に選択して 200-800 文字で回答するというものでした。

1. データだけのパッケージでできること
2. ソフトウェアのライセンスで不自由したこと
3. 好きなソフトウェアライセンスとその理由

この課題に対して提出いただいた内容は以下です。

### 2.1 堀内寛己

僕が好きなソフトウェアライセンスは GPL です。理由は、ストールマンが好きだからです。

僕がストールマンのことを知ったのは、20 年以上前、東大の和田英一先生の講義をもぐって聞いたときのことです。先生は、MIT ハッカーズの墮落と、その中で孤軍奮闘しているストールマンの話を読みました。「でもストールマンは変わりませんねえ。グニューって知ってますか？ みんなタダなんですよ」先生が英語の微妙なニュアンスを聞き分けられないわけがないので、当時のストールマン自身が、free の 2 つの意味を混同していたらしいことがわかります。当然そのころ、GPL なんてありませんでした。僕は幸い、入社まもなく GNU Emacs を使う幸運に恵まれ、ストールマンへの敬意は確固たるものになり、そのまま GPL のことなど知らずに gcc-2 の make をやったものです。僕が GPL とその恐るべき意図について知ったのは、だいが後のことです。Microsoft がこれをウィルス呼ばわりしたのほ、さらにその後のことです。僕は、ウィルスという表現は、GPL についての最高の褒め言葉だと思っています。

### 2.2 山本ひろゆき

「データだけのパッケージでできること」インタープリタ依存のスク립トパッケージなんかができますかね？今抱えている ITP を片付けたら、kita2 (ruby スクリプト) を ITP しようかと迷っています。パッケージング自体はできますが、ruby を知らないものでメンテナンスが心配。誰か教えて！

「ソフトウェアのライセンスで不自由したこと」某巨大掲示板発祥の開発プロジェクトあたりだと、有用なソフトウェアだが、ライセンスは明記されていなく開発者も匿名で確認しようがないということもあります。

「好きなソフトウェアライセンスとその理由」GPL とか BSD ライセンスとか。DFSG 万歳！

### 2.3 Suguru Sekine

「好きなソフトウェアライセンスとその理由」  
[BSD ライセンス]

自由であり続けるために「自由でなければならない」という制約がない、かつソースコードの公開を拒む自由があるためである。私はソフトウェアは自由であるべきだと考えている。また、自由であるための協力もできる限りの協力を行うつもりである。しかし、自分が考えている事を他の人に強要したり、または強要されたりすることがどうしても我慢できない。以上のことから自由であることを強要されない点から BSD ライセンスを評価しているという結論に至る。

### 2.4 Yamane Toshiaki

「データだけのパッケージでできること」

データだけのパッケージというものが例えばフォントパッケージ等を指すという理解で文章を書いてみます。できること、ではないかもしれませんが、自分が使いたいものを選択する自由があると考えます。あるいは X と xfont なパッケージのようにアプリケーションとデータがそれぞれ互いに依存していない関係にする事ができるのではないかと考えます。データだけのパッケージとそのデータを使うパッケージ間を疎結合な関係にしておくことによって、保守とか拡張などの対応を柔軟に行なう事ができるのではないのでしょうか。

### 2.5 沖中

ライセンス的に配布が難しいデータは、自分でパッケージすることになると思いますが、データだけのパッケージと言えばフォントが代表的だと思います。辞書データもフリーの物が少ないので自分でパッケージ化できたら便利ですね。それ以外の使い方としては、Web サイトのドキュメントをパッケージ化して管理できないか考えています。そうするとデータだけとはいきませんが、スクリプトだとビルド不要なのでデータと同じように扱えるのではないかと考えています。

## 2.6 山根秀樹

ソフトウェアのライセンスで不自由したこと

今まさに不自由しています。いわゆる「フリーフォント」が巷にはたくさんあるのですが、みんな「自分で規定しましたライセンス」が多くて、二次利用などがづらいですね。作者の方々は善意で策定したんでしょうけど、「悪いことに使っちゃダメ」ライセンスみたいなのは勘弁して欲しいです（線引きが曖昧だし、そもそも悪いことしようとする人がライセンスを気にすることなんて無いわけで、そんな規定作ること自体が意味ないんですが...）。

あとは最近あったのが PHP プロダクトで PHP ライセンスのものとか。PHP の名称を使うことをそもそも想定していないんだから、別のライセンスの方が扱いやすいんですが、「PHP 使ってるから PHP ライセンス」なんじゃないかな。

## 2.7 satoken

好きなソフトウェアライセンスとその理由

GPL。やはり、ソースが有るっちゃうのは良いもんです。頑張れば移植もできるし。でも、Debian に浸ってからあんまりコンパイルしなくなりました。apt-get install で済んでしまうところが楽なだけで墮落の始まりかもしれない。

## 2.8 Hisashi MORITA

「データだけのパッケージでできること」

すぐに思いつく例はドキュメント、辞書、フォント、アイコンなど。あと XML の DTD などローカルにあると役立つ（ネットワークアクセスを避けられる）。

「ソフトウェアのライセンスで不自由したこと」

プロプライエタリなソフトウェアが不自由というのはよくある話ですが、GNU GPL なソフトウェアで、CD-ROM に収まらなかったのバイナリとソースを別々に配布しようとしたら結構大変だったことがあります（3-b 参照）。一緒に配布するのが楽ですね。

「好きなソフトウェアライセンスとその理由」

今は X/MIT スタイルが好きです。短いので。

## 2.9 Kobayashi Noritada

「ソフトウェアのライセンスで不自由したこと」

開発者が FLOSS の世界に通じており、開発環境として Debian のようにライセンスに頼りものを用いていると、広く採用されているライセンスを使用してもらえるので楽なのですが、FLOSS の世界に通じていない人が作成したソフトウェアはいい加減なオレオレライセンスになりがちで、パッケージ化しようとしても困るときがあります。特にフォントなどは必ずしも FLOSS の世界に通じていない人でも作ってしまうので、このような傾向が顕著な気がします。また、FLOSS の世界に通じていない人だと、改変・再配布の自由まではあまり認めたくないという人が多く、パッケージ化の点からは不自由なライセンスを採用しがちな気がします。そのような意味では、DFSG-free なライセンスの認知度を高めること、そして FLOSS の開発の理念を広めることは大切で、その点で Debian の果たす役割は重要だと考えています。

「好きなソフトウェアライセンスとその理由」

基本的に、広く採用されていて DFSG-free だと分かっているライセンスが好きです。その理由としては、もちろん Debian の公式パッケージにしやすいこともありますが、自分で細かくレビューする必要がないというのが大きいです。自分で物を公開する場合にも他人の物を使用する場合にも、その物のライセンスのコンセプトを把握しておかなければなりません、ライセンス文の法的な用語や法的な思考は慣れていないと難しいものです。そんなときに、広くレビューされ吟

味されているライセンスだと、コンセプトが分かっているので、自分で細かい点まで理解しなくても安心して使えます。

## 2.10 Noriaki Sato

「好きなソフトウェアライセンスとその理由」

好きなソフトウェアライセンス: GPL

その理由: RMS 信者だから

その昔、大学の研究室に配属されて初めて Unix を触って、emacs (当時は mule2 でした) も初めて使ったのですが、先輩に借りた emacs の本で GPL とか copyleft の事、そして RMS の名前を知りました。「何て素晴らしいんだ！」

そう思っていた時期が私にもありました。

#いや、別に今は素晴らしいと思ってないわけじゃないのですが。

昨年の秋に専修大学で RMS の講演があった時も、有休を取って行ってきました。英語は半分くらいしか聞き取れませんでした orz が、質疑応答で、ある学生さんが《私は Open Source Software の研究をしていて》と（英語で）言ってしまった所、すかさず《Open Source じゃない、Free Software だ》と口を挟んで、その後学生さんに口を挟む余地を与えずに 5 分くらい喋り続ける、とゆーのが生で見ただけでも良かったです。

というわけで、私は GPL が好きなのです。

## 2.11 鈴木

「データだけのパッケージでできること」

フォントとか設定ファイルだけを提供するパッケージがな。今日の勉強会で学びます。

「ソフトウェアのライセンスで不自由したこと」

ライセンスを語る前にソフトを作ることから離れました。よかったのは、Tex のクヌース先生のパスカルを読んで、真似して C で作りました。あの時は、そんなこと意識せずにソースが見れたのがよかった。自分のソースを見られるのが嫌なソフト屋が多かったですね。ライセンスにあまり関係なくて、すみません。

## 2.12 藤崎 祥見

好きなソフトウェアライセンスとその理由

ソフトウェアライセンスはわかりやすいものが好きです。もし自分が適用するならば、一般に広く知られているものかわかりやすい条文のものにします。その方が、配布者・利用者ともにハッピーだと思うからです。たとえば、ライセンスを読んだ人が理解できずに質問をする・それに回答するという労力を、上記のソフトウェアライセンスを適用することによって避けられます。ソフトウェアライセンスは理解されてこそ最大の効力を発揮するので、理解しやすい・または理解されているものを選ぶというのは優先順位が高い条件ではないでしょうか。ということで、私は Ruby ライセンスが好きです。Ruby ライセンスは GPL と Artistic 類似の独自ライセンスのデュアルライセンスです。この独自ライセンスの条文はやさしく理解しやすいものになっています。利用者が GPL を理解しているなら GPL を適用すればよいし、GPL が難しくてもわからない場合でも理解しやすい独自ライセンスの元で利用すれば OK です。ですが、松本さんは Matz 日記で、

Ruby のライセンスがあれなのは

- 開発当初の regex.c が GPL だったので GPL を適用する必要があった
- でも、自分が書いたコードについてはより緩くしたかった

という理由ですから、そういう事情のないソフトウェアが Ruby ライセンスを適用する必然性はほとんどないと思います。ということで、自分のソフトウェアに安易に Ruby ライセンスを適用しないように。

と書いてらっしゃるので、今のところ自分が適用するなら X11 かなぁ。といいつつ 1 年後には変わっているかもしれません。

## 2.13 CCG

「好きなソフトウェアライセンスとその理由」

NYSL。「煮るなり焼くなり好きなようにしろ」ライセンスです。日本の法律では著作権を放棄できないため、public domain という概念を適用できないということを聞いたことがあるのですが、心気がとても伝わってくるどころがたまりません。NYSL 以外では、世に言う修正 BSDL が私の心にあっていて(と書くの大袈裟ですね...)。職業プログラマですが、プログラマとしての最大の喜びは沢山のの人に自分の書いたプログラムを使ってもらえる事、少しでも便利に生活を送ってもらえる事と考えています。GPL でもそれは達成できるかもしれませんが、書き換えたものの公開を強制することを理由に使うことをためられるのであれば、修正 BSDL にして(場合によっては LGPL にして)使ってもらえる事を望みます。好みとは関係ないですが GPL と MPL と CDDL の違いが分かりません...

## 2.14 野村

タイトル:「好きなソフトウェアライセンスとその理由」

私が好きなライセンスは BSD ライセンスである。はじめに断っておくが、決して GPL が嫌いなわけではない。

私の会社では、PC に無償のソフトウェアをインストールする際には、そのソフトウェアのライセンス、脆弱性について上司に報告し承認をもらう必要がある(今は面倒なので報告せずに勝手にインストールしているが、入社したてのころは真面目に報告していた)。

このとき、オープンソースに明るい上司であれば、BSD ライセンス/GPL/Apache Software License のソフトは、「オープンソース系のライセンスです」と報告すれば一発で許可をもらえるが、オープンソースに疎い上司だと、GPL 等に対して変な勘違いをしている場合があり、「業務で使っても問題ないのか? 個人の使用は無償でも、業務で使うとお金がかかるとかではないのか?」といった質問をされたことがある。恐らく、どこかで GPL のコピーレフトの仕組みの説明を見たものと読まずに早とちりしてしまい、間違った解釈をしていたものと思われるが、GPL の説明をゼロからやらされた私にとってはいい迷惑である。

その点、BSD ライセンスであれば、「基本的にどうやって使っても自由です。ほら、そう書いてあるでしょう」と言えば済むので、GPL より説明が楽である。

以上が、私が BSD ライセンスを好む理由である。

## 2.15 吉田@板橋

お題:好きなソフトウェアライセンスとその理由

昨年末に「世界初(多分)の GPLv3 本」を発行@コミケしたのもあって GPLv3 が好き... というわけでもありませんが、いままでは自分が作ったものは、修正 BSD ライセンスで放流することが多かったのですが、調べてみると GPLv3 でも悪くは無いと思いました。(以下、本一冊分略) さすがにあれだけ話題になっただけあってかなり考えられて作られていますね。特に、やり方によっては Apache License 2.0 等の修正 BSD 系のライセンスからコードを取り込むこともできるようになったのはいい点だと思います。ついでに、GPLv2 との互換が無いのが素敵です。

## 2.16 前田 耕平

「データだけのパッケージでできること」

1. 自分で管理しているサイトの構築や運用に必要なスクリプトの配布。

公式パッケージだけでは、やはりどうしても細かい部分までは行き

届かないので、構築時や運用改善でスクリプトを書きます。一台二台ならそれを scp など配布してしまえば良いですが、台数が増えるとそれが面倒なので、パッケージにしてしまえと。各種設定ファイルのパラメータ変更もスクリプト書いて、パッケージで配布してガツガツ書き換え、というのも、小規模なら良いかもしれませんが、そういうのは寧ろ Puppet などの導入を考えたほうが良いですね。

2. 複数のアーキテクチャが混在した環境への配布。

1も結局はアーキテクチャに依存しない、というメリットがあるので作ったスクリプトを違う環境(x86, ppc, arm などなど)でもそのまま使えるのが、データだけのパッケージの最大のメリットかと。

## 2.17 Seiji Kuroda

ソフトウェアのライセンスで不自由したこと

十数年前、仕事で半年ほどアメリカのボストンに滞在していました。当時、現地のレンタルショップで自宅用としてパソコンを借りていました。実際には plain text でメールの下書きや会社で読みきれない資料を読むくらいでしたので、もともと入っていた Windows を消して、日本から持っていった J-DOS や OS/2 の J に入れ替えて使っていました。しかし、ノーブランドに近いパソコンのためか、ハングアップやダウンはしょっちゅうでした。

やむを得ず、現地のパソコンショップで、MS-DOS のアップグレード版を買ったのですが、ライセンスがないと売れないというので、J-DOS のディスクとライセンス証書(日本語)を持って、UPGRADE 版を売れと迫りました。ライセンス証書は日本語だったはずで相手は読めるはずもなく、随分と僕のほうが無茶な要求をしたものです。ただ通い始めて三日目に店員が僕の粘りにあきれて、MS-DOS のアップグレード版を売ってくれました。

いまならこんなことにはならないと、でも必死になれない英語で粘ったのは、懐かしい思い出です。

## 2.18 原田 一範

「好きなソフトウェアライセンスとその理由」

好きなソフトウェアライセンスは GPL です。GPL の「すべてのソフトウェアは自由であるべき」と言う考え方がシンプルだから好きです。私はソフトウェアライセンスには、できるだけ変更の自由を保証してほしいと思っています。変更の自由が保証されていれば、便利な機能を付け加えたり、改良したりする際にライセンスによって邪魔されず、短時間で作業できると考えられるからです。また、ソースコードが開示されていれば、そのソースを参考に新たなソフトウェアを作ることが可能だし、初学者がそこから多くのことを学ぶことも可能です。GPL ならば、GPL が適用されたソフトウェアの派生ソフトウェアまで GPL が適用されるため、変更の自由が保証され続けるという意味で、良いライセンスだと思います。

GPL は自由に関して厳格であるため、もし自分がライセンスを使用するような状況になったら、もっと緩いライセンスを使用するかもしれません。しかし、GPL のような、ソフトウェアを使用するユーザーの自由を保証するライセンスは必ず必要だと思うので、好きなソフトウェアライセンスに GPL を挙げました。

## 2.19 日比野 啓

データだけのパッケージでできること

プログラムを含んだパッケージを作ることが多いので、データだけのパッケージといわれるとあんまりネタが無いんですが、作ったことがある中で、大きめなのは gnujdoc<sup>\*1</sup> でしょうか。

autoconf, binutils, bison, cvs, diff, emacs, flex, gdb, libtool といった、GNU のツール群のドキュメントの日本語訳が含まれています。全部ビルドすると 60 個以上のパッケージになったりとか、makeinfo 日本語まわりが変だったことがあったりとか、なかなか面倒だっ

<sup>\*1</sup> <http://openlab.ring.gr.jp/gnujdoc/snapshot/>

た記憶があります。

ライセンス関係のことをきちんと勉強できていないので、今回の発表を聞いて勉強させてもらいます。よろしくお祈りします。

## 2.20 濱野

データだけのパッケージでできること

主要な RFC や仕様書がパッケージになっているとばつと参照するのに便利でしょうか。辞書データ、Micropolis(SimCity)の街データ、そういえばオープンソースで開発されたファミコン ROM って在るのかな。

好きなソフトウェアライセンスとその理由

Beerware License とかカッコイイですね、いつかこのライセンスで当たり障りのないソフトウェアを書いてみたいです。

# debian の main カテゴリには入らないようですね、残念。派生ソフトウェアに関する記述が無いからでしょうか。

## 2.21 奥野 由紀

ソフトウェアのライセンスで不自由したこと

私がソフトウェアのライセンスで不自由したことといえば、他にも同じ事を述べられる方がいらっしゃるかもしれませんが、Microsoftをはじめとする、プロプラエタリなソフトのライセンスです。

Windows は XP からアクティベーションを導入したため、自作 PC 派にはやりづらいことになってます。Vista になって一段と再アクティベーションを要求される条件が厳しくなったそうです。

また、先日とあるソフトを業務で使用することになったのですが、一つ前のバージョンまでアクティベーションはありませんでしたが、最新バージョンになってからアクティベーションが導入されていました。

ソフトウェアメーカー側としては、利益の確保のためには止むを得ないことなのでしょうが、正規ライセンス所持者にも使い勝手が悪くなります。

また、メーカーによっては業務用のボリュームライセンス制度があるところもありますが、これがまたどのライセンスを選べば良いのが、一目見て分かりづらいものが多いです。

以上が、私がソフトウェアのライセンスで不自由したことです。

## 2.22 Osamu Matsumoto

ソフトウェアライセンスについては、正直あまり真面目に気をつけてコードを書いた事はありません。GPL2 で公開されていたツールキットをサブセットに分解して GPL2 で公開するといった経験ぐらいです。特に好きなライセンスはありません（正確には知りません...）が、Creative Commons Lisence はライセンスに精通していないユーザでも、目的に合ったライセンスを選択できる点が良いと思います。

「データだけのパッケージでできる事」ですが、現在の理解ではデータを置く事しかできないんじゃないかと思っちゃいました。font とか document とかなのかな。当日、勉強して帰ります。

## 2.23 ake

「データだけのパッケージでできること」

どういふものがあるのか考えてみました。とりあえず思いつくまでに

- ドキュメントが主体のもの、と言うかドキュメントそのもの
- フィルタデータ (chastity-list とか kr-filter )
- フォントデータ等

こうやって改めて考えるとデータだけのパッケージって結構重要というか、無いと困るものがありますね。で、「できること」というと設定に関するものであれば何かと重宝するものが作れそうです。

「ソフトウェアのライセンスで不自由したこと」

CAL の数がどうか、プロダクトキーがどうか、管理するのに手

間のかかるライセンスはお金に換算できない(しにくい)コストが掛かるので、よっぽど必要なもので無い限り関わりたくないです。

「好きなソフトウェアライセンスとその理由」

フリーなものは大抵大好きです。無料か有料かではなくて、自由ってことが重要だと考えますので Debian のライセンスは大好きなものの1つです。

## 2.24 キタハラ

データだけのパッケージでできること

昔は、今ほど通信回線が速くなく、費用も高かったので、クライアント・サーバ型のシステムでプログラムとは別に、データをクライアントに配置し、定期的に更新するシステムなんてのが結構ありました。当時のマシンでパッケージが使えたら、バージョン管理もできるし、便利だっでしょうね。(業務で使うには、別の問題が発生するかもしれない...)

今でも設定ファイル等、クライアントに配布したいデータは幾つかあるのですが、OS がほとんど某窓になってしまったので、今のところ出番がないですね。

あとはサーバかな。設定ファイル等をパッケージにしておくと、サーバが多数ある場合や、一から作り直す場合は楽になるかな? どうだろう?

## 2.25 Osamu Kimura

Yacc や Lex といったパーサジェネレータはソースコードを出力するという性質をもつため、パーサジェネレータを利用したソフトウェアは使用するパーサジェネレータのライセンスに影響される可能性がある。

世の中には Yacc の上位互換ソフトとして byacc と bison というものがある。実はこの2つは同じ作者 (Robert Corbett さん) によって作成されたものである。byacc はパブリックドメイン、bison は GPL である。(ちなみに byacc の方が後に作成された) だから、ライセンス問題を動機として (byacc は) 作成されたのではないと思われる。にも関わらずこの2つのソフトウェアは微妙に動きが違ったりする。オプションの数が違ったり、出力ファイル名が違ったりしている。

ソフトウェアの中にはある環境では byacc を使い、ある環境では bison を使うという事をしているものがある。これで何が問題かというと、自分の使ってる OS には Ports とか Package が無いソフトウェアを移植する時に問題が起きる。こういう問題はマイナーな環境のマイナーな処理系などでは起こったりするので、面倒くさい。

私がこの問題に遭遇したのは Cygwin に Fortran 用プリプロセッサ fpp をインストールしようとしたときだ。FreeBSD の Ports から対応するパッケージをダウンロードして、ビルドをかけた時に、Bison がエラーを出しまくったと言う事があった。この時は Makefile のパーサを Bison から byacc に変えて解決した。

## 2.26 Shi

「データだけのパッケージでできること」

自作の音楽や画像、小説の配布が可能だろうと思います。コミックマーケットの創作ジャンルなどでは、印刷費を払って本を制作し、それを無償配布しているような人が多数みられます。従来、こうしたデータの無償配布は、Web 上での公開が主ですが、Web の規模が大きくなった現在、必ずしも検索エンジンだけでは追い切れない現状があると思います。apt 等のパッケージングシステムは、データに特化したサーバを用意することで、極めて高効率でデータの再配布を可能にすると思います。最初は壁紙や、効果音など、GUI 環境を提供する一環として開始されたら面白いかなと思います。

## 2.27 市川 憲人

データだけのパッケージであっても、データの中にソースを埋めこむことを許可すれば、基本的にはプログラムをからなるパッケージができることならばなんでも可能たらしめることができるはずである。データの中のソースをコンパイラに食わせてプログラムを作成すれば

基本的になんでもできるからである。データとプログラムの境界はないといってよい。どのようなデータであってもそのデータを解釈する主体が必要であり、それはプログラムであったり解釈する機能の一部を人間が担う場合があったりするが、どのような場合であってもデータは何らかの目的があって解釈されるはずであり、その結果は何らかの形の実行となって帰ってくる。したがって、人間を含む実行環境の中において、およそ全てのデータは実行されるプログラムと考えることができる。



## 3 Debian Trivia Quiz

小林 儀匡



ところで、みなさん Debian 関連の話題においついていますか？ Debian 関連の話題はメーリングリストをよんでいると追跡できます。ただよんでいるだけでははいあがないので、理解度のテストをします。特に一人だけでは意味がわからないところもあるかも知れません。みんなで一緒に読んでみましょう。

今回の出題範囲は `debian-devel-announce@lists.debian.org` に投稿された内容からです。

問題 1. lenny でのリリースゴールの 1 つに GCC 4.3 でのビルドが通ることが挙げられているが、それに伴って変更する必要が最も多く生じるのは、どの言語で書かれたプログラムか？

- A C
- B C++
- C Ruby

問題 2. 今のところ lenny で完全に削除される予定なのは次のうちどれか？

- A GNOME 1.x
- B KDE 3.x
- C Linux 2.x

問題 3. Debian に宣伝力がないと感じた Debian プロジェクトリーダー (DPL) の Sam Hocevar が提案したのは？

- A イベント会場で Debian ロゴの入ったシャツを着て練り歩く Debian Tank Top & Brief Team
- B ロゴや T シャツのデザインなどマーケティング面の任務を一手に引き受ける Debian Marketing Team
- C Debian のマスコットキャラクター「でびにゃん」

問題 4. 3 月頭の時点で lenny のリリースにとって致命的なバグ (RC バグ) は 460 ほどあるが、それを減らす方法として推奨されているのは？

- A `debian/changelog` に「`closes: #xxxxxx`」と書いて片っ端からパッケージをアップロードする
- B バグ退治パーティー (BSP) を開いて片っ端からパッケージを非メンテナアップロード (NMU) する
- C RC バグを放置しているパッケージメンテナを片づ端から呼び出して説教する

問題 5. lenny はリリースされると何になる？

- A Debian 4.1
- B Debian 5.0
- C Ubuntu 8.0

問題 6. `lists.debian.org` の新しい検索エンジンには何が用いられているか？

- A Google Search
- B Hyper Estraier
- C Xapian Omega

問題 7. 新しい armel アーキテクチャに関する現況として適切でないものは？

- A lenny のリリースアーキテクチャに含まれている
- B 全パッケージの 90% がビルドされている
- C パッケージを `ftp.debian.org` からダウンロードできるようにしている

問題 8. 次の開発者のうち、今年の DPL 選挙に立候補したのは？

- A Lucas Nussbaum
- B Junichi Uekawa
- C Steve McIntyre



## 4 最近の Debian 関連のミーティング報告

上川 純一

### 4.1 東京エリア Debian 勉強会 36 回目報告

東京エリア Debian 勉強会報告。1 月の第 36 回東京エリア Debian 勉強会を実施しました。今回の参加者は前田さん、山本琢さん、山本浩之さん、日比野啓さん、森田尚さん、あけどさん、岩松さん、やまねさん、小室文さん、吉田@板橋さん、鈴木@LSS さん、小林さん、本庄さん、高橋昭之さん、noriaki Cato さん、橋本徹さん、キタハラさん、野首さん、デビッド スミスさん、でんさんとその友人 A、上川の 23 人でした。

まず、クイズを今回も実施しました。今回も、debian-devel-announce の内容から出題しました。起動スクリプトを依存関係ベースで動かしてくれる insserv の登場が一番気になる所です。

今回は 2008 年の予定を決めました。前半について担当者をきめたのでお願いします。また、次回は OSC で、その実施内容がまだ未定のため、岩松さんが「何をしようか」という話を出しました。

- 3 月 データだけのパッケージを作成してみる、  
ライセンスの考え方 (David)
- 4 月 バイナリーつのパッケージを作成してみる (吉田@板橋)  
バージョン管理ツールを使い Debian パッケージを管理する (git)  
アップストリームの扱い (svn/git/cvs)(岩松さん)
- 5 月 バイナリの分けたパッケージの作成。(前田さん)  
バイナリの分け方の考え方、アップグレードなどの運用とか。
- 6 月 パッケージ作成 (dpatch/debhelper で作成するパッケージ)(小林さん)、  
man の書き方 (roff or docbook)(でんさん)

2008 年はテーマとして DEB パッケージの開発・管理に関連した内容にしよう、ということにしました。今回は最初の一回目ということで、DEB パッケージの作成の中身の詳細ではなく、作業の流れとそれにまつわる各種ツールをならべてみました。Deb パッケージが開発者からユーザに届くまでの流れとか、開発の各種ツールをどういう順番で利用しているのか、新しいバージョンのパッケージを作るタイミングとは、というような話を上川がしました。

今回は宴会は荻窪 高原亭にて開催しました。

## 5 Open Source Conference 2008 Spring 報告

山本 浩之 / やまね ひでき / 岩松 信洋

2008 年 2 月 29 日と 3 月 1 日の両日に渡って東京新宿で開かれた「Open Source Conference 2008 Tokyo/Spring」に東京エリア Debian 勉強会有志が参加してきました。なお、関係者のスケジュールの都合上、東京エリア Debian 勉強会は 3 月 1 日のみの参加でした。今回は会場 4 階でのブース設営とセミナー、そして初の試みとしてハンズオンを開催しました。

### 5.1 セッション: Debian Overview

セミナーは「Debian Overview」と題して、Debian の始まりの説明から DFSG とオープンソースの関係の紹介、現状の開発の活発さを示す指標として移植 / 派生ディストリビューション / 開発者数と現状 / パッケージについてなどを説明し、活況あふれる Debian コミュニティへの参加を促しました。参加者数は 40 名教室で 36,7 名でしたので盛況と言って良いのではないかと思います。

### 5.2 セッション: Debian Package ハンズオン

今回の OSC では、Debian 勉強会としては初の試みである ハンズオンを行いました。OSC の会場として専門学校を利用させていただいています。専門学校には実習室があります。以前から多くの方から依頼があった Debian Package のハンズオンを行いました。

簡単且つパッケージにする際に難しくないプログラムということで、sl<sup>\*2</sup> をベースに、環境の構築、パッケージの作成方法、パッケージテスト方法、問題があったときの対処方法を実際の手順に合わせて、参加者に行ってもらいました。結果は岩松の不手際や会場での問題等でスムーズに進行させることができなかったのですが、おおむね好評だったようです。初の試みということと、今後につなげていくために、実際の参加者の方に意見を聞いて回りました。まとめたものを以下に示します。

1. 講師一人ではサポートしきれないので、参加人数にあわせたサポートチームを構成する。  
一人脱落者が出ると、一人では対応できない。人によってレベルが様々なので、対応しきれない。
2. キーボードを打つ人が遅い人のために印刷したものを用意してそれを配布する。ついていけない人はこの紙を読んで進めるようにサポート環境を作る。
3. 環境のレベルを低くする  
vi / emacs なんて使えないぜ！という人がいる。これはこれでいろんな意味で困るが、普段使い慣れたエディタが使えると一番よい。gedit などが使える環境なども必要。今回は手順を省くためにコンソールベースにしていますが、マウスがないとだめな人もいたので、このあたりも考慮した環境を用意する必要がある。
4. 参加者の情報が必要

<sup>\*2</sup> <http://packages.debian.org/sid/sl>

入り口を狭めるという意味で情報を集めるわけではなく、チームでやる場合の人の配置などに使うためにスキルの情報があると楽になる。環境変数ってなに？とかいう人を一箇所に集めてフォローしやすくしたり、など。または思い切って参加者スキル制限を行うと良い事もある。

#### 5. プレゼン用の画面とコンソール画面があるとよい

2つ同時に出せる環境が理想という意見がありました。会社で行う研修ではこのような方法を使う事が多いらしい。実際にはむずかしいので、紙と連携させてやるのがよい。

#### 6. 45分では足りない

簡単な Debian Package でもハンズオンだと45分以上かかることが分かった。実際には割り込みが入ってしまい、もっと長くなる可能性もある。円滑に進めるには上に書いたようなことを行う必要がある。

#### 7. ネットワーク制限がないところがいい

不安要素が減るので。

今回のハンズオンで、いろいろ課題が分かりました。課題の解決と今後につなげていくために、近い将来にハンズオンのリベンジを行いたいと考えています。ハンズオンするにはそれなりの設備が必要なので、良い場所とかあれば教えてください。

### 5.3 セッション: 展示ブース

ブースでの展示では Debian JP への寄付金、計 9,520 円が集まりました。リアル Debian 掲示板には次のようなことが寄せられました：

- ラヴ・Etch ちなみに alpha ユーザです
- 関西から来ました！ Debian 最高 Death
- 3月の勉強会行きます！
- Air を Sid にしたよ
- Debian の Asterisk パッケージのバージョンを最
- 新に！（誰か、よろ）
- 人脈作成中
- パッケージ作成中
- Debi りましょう！！

展示ブースではお金の管理の問題があり、寄付金と物販の売上の違いが分からなくなり、結局、一部寄付を強要するような形となりました。今後は、寄付金集めと物販は同時には行わないなどの対策が必要でしょう。

### 5.4 まとめ

次回に向けて、ブースの準備（配布用 LiveDVD 準備が間に合わなかった、寄付金管理）、セミナーの時間配分（15分オーバー）、ハンズオンの準備（段取りなど諸々）など課題も見えたイベントでしたが、検索エンジンなどでの Blog 投稿や mixi の日記などを見る限りでは、参加頂いた方々に楽しんでいただけたのではないかと思います。

参加者の皆様、お疲れさまでした。

## 6 Debian パッケージ作成：データだけのパッケージ

David Smith



Debian パッケージ作成によろこそ！dpkg のパッケージ、いわゆる Debian パッケージの作成を分かりやすく、実例の分析に基づいて説明する。<sup>\*3</sup>今回データだけのパッケージについてみてみましょう。

データだけのパッケージは何かと言えばフォントやドキュメンテーションや設定ファイルなどのパッケージを指すことだ。要するにプログラムとライブラリではないパッケージだと言える。今回、GNOME デスクトップの背景ファイルの集合、gnome-backgrounds の構成を解析する。Debian Popularity Contest<sup>\*4</sup>によると現在 gnome-backgrounds が殆ど全部の GNOME インストールに含まれていて GNOME を使用していれば gnome-backgrounds も入っているはず。背景ファイルだがパッケージの構成に複雑の部分もあると思うから詳細までみてみよう。

### 6.1 準備

まず、Debian New Maintainer's Guide<sup>\*5</sup> のページを開いておいたらいいでしょう。不明な点があった場合、New Maintainer's Guide に分かりやすく書いているだろうし、質問は何処まで送ればいいのか教えるのだ。

それから gnome-backgrounds のソースパッケージをダウンロードしよう。

```
$ apt-get source gnome-backgrounds
```

準備の最後にパッケージの構成に必要な依存パッケージをインストールしなければビルドが出来ないので以下のコマンドをインストールしよう。

```
$ apt-get build-dep gnome-backgrounds
```

### 6.2 基本の Debian パッケージファイル

今のところ、apt-get source がソースパッケージを gnome-backgrounds-2.20.0 というディレクトリに展開した。そのディレクトリの中には debian/ディレクトリがあり、そこにパッケージ作成の設定ファイルがある。パッケージによるファイル数が異なるが、gnome-backgrounds は最低限に近い。

<sup>\*3</sup> Debian パッケージが作りたいよね。一応作りたいと前提に思っているからよろしく。

<sup>\*4</sup> Debian Popularity Contest の URL:<http://popcon.debian.org/>

<sup>\*5</sup> Debian New Maintainer's Guide の URL: <http://www.debian.org/doc/devel-manuals#maint-guide>

```
piyo:/tmp/gnome-backgrounds-2.20.0> ls -l debian/
合計 36
-rw-r--r-- 1 dds dds 3755 2008-03-08 16:21 changelog
-rw-r--r-- 1 dds dds 2 2008-03-08 16:21 compat
-rw-r--r-- 1 dds dds 1022 2008-03-10 15:11 control
-rw-r--r-- 1 dds dds 771 2008-03-08 16:21 control.in
-rw-r--r-- 1 dds dds 1780 2008-03-08 16:21 copyright
-rwxr-xr-x 1 dds dds 368 2008-03-08 16:21 rules
-rw-r--r-- 1 dds dds 145 2008-03-08 16:21 watch
```

この中で changelog、control、copyright、rules はあらゆるパッケージに必要です。

### 6.2.1 debian/copyright

debian/copyright は上流ソースコード、データなどの著作権を認めるためのファイル。形式は自由。パッケージビルドツールに対して必須なので自分専用でも自分の会社内専用のパッケージでも一応書かなければならない。最低限として「All Rights Reserved」と書いてもいい。gnome-backgrounds の debian/copyright ファイルが長すぎるのでここには示さないけど見てみてください。パッケージをインストールすると /usr/share/doc/パッケージ名/copyright にインストールされる。

### 6.2.2 debian/changelog

debian/changelog にはパッケージのバージョンを記録する。必ずしもパッケージのバージョンが上流のバージョンに一致することではないので必須になっている。それにパッケージの最新バージョン番号はこのファイルの最初の1行目から直接に抽出される。

gnome-backgrounds の最新バージョンの changelog を見てみましょう。

```
gnome-backgrounds (2.20.0-1) unstable; urgency=low

* New upstream release.

-- Sebastian Dröge <sloMo@debian.org> Sat, 22 Sep 2007 09:49:38 +0200
```

かなり簡単。だけど、空白と日付の形式を厳守しなければビルドするときエラーが発生するから注意。<sup>\*6</sup>

### 6.2.3 debian/control

debian/control にはパッケージ名、Maintainer、依存のパッケージなどのメタ情報が記している。gnome-backgrounds の場合、独特として debian/control と共に debian/control.in ファイルもある。debian/control.in は Debian のパッケージツールに対して関係ないから無視しても良い。でも目的は何かと言えば説明したい。

Gnome プロジェクトは規模が大きいのメンテに関わっている人が多い。debian/control にアップロード出来る人達の名前が記載されているのだが人の名前が多すぎて直接管理しにくいから @GNOME\_TEAM@ という変数だけを書いている。そして別のファイルにその人達の名前を書いて、発行時にスクリプトによって変数を変換するわけだ。必要がなくて複雑かもしれないけど Debian といえば自由主義ですね。

では debian/control をみてみよう。

```
Source: gnome-backgrounds
Section: gnome
Priority: optional
Maintainer: Sebastien Bacher <seb128@debian.org>
Uploaders: @GNOME_TEAM@
Build-Depends: debhelper (>= 5), cdb, gnome-pkg-tools
Build-Depends-Indep: libxml-parser-perl
Standards-Version: 3.7.2

Package: gnome-backgrounds
Architecture: all
Depends: ${misc:Depends}
Description: a set of backgrounds packaged with the GNOME desktop
 This is a collection of desktop backgrounds created with GNOME users in mind.
 It contains the following backgrounds:

[後略]
```

<sup>\*6</sup> debchange(1) というツールで手軽に changelog 編集が出来る。devscripts というパッケージに入っている。

第一段落にソースパッケージを記述する。apt-get source と apt-get build-dep を実行する時、ソースパッケージの情報を使っている。9 行目以降に作成する gnome-background のバイナリパッケージを定義している。順番に関してはソースパッケージが必ず最初に来るそして Source 又は Package はそれぞれの段落の最初の 1 行に書く。それ以外の中身の項目は特に決まった順番がない。

Section と Priority との項目は少しややこしいかもしれない。決まった選択肢の中から 1 つしか選べないので合わない場合があるけど gnome-backgrounds の場合はぴったり、gnome で optional。選択肢の詳しい情報は Debian Policy Manual に書いてある。<sup>\*7</sup> 後は Maintainer と Uploaders はそれぞれ書いた通り。Uploaders は共同にメンテをしている人なのだ。つまり、Maintainer の項目は必ず一人しか書けないけど他の人もメンテしていれば Uploaders に書くということだ。

次にビルドのための依存関係パッケージが書いてある。殆どのパッケージにはビルドするための依存パッケージと利用するための依存パッケージが異なる。例えば普通のプログラムの場合に C/C++ のヘッダーがビルド時に必要けど実行するのに要らないとして、Debian パッケージの構成にも同じことがある。<sup>\*8</sup>gnome-backgrounds がビルド時に使ってる依存パッケージがより典型的なのでそれぞれ少し説明する。

cdbs “Common Debian Build System” の省略、共有されているビルドシステム。多くの人が使っていると思う。  
debhelper いろいろパッケージビルドに役に立つツール。例えば上流のドキュメンテーションを自動的にインストールするツールなど。実にこれが cdbs から依存しているので述べる必要がない。でも重複しても問題はない。  
gnome-pkg-tools cdbs に少し似ている感じで GNOME パッケージを作るのに役に立つツール。  
libxml-parser-perl Perl の XML::Parser モジュール。

ソースパッケージの段落の最後に Standards-Version という項目は Debian Policy Manual の何バージョンに準拠しているかを述べる。

続いてバイナリパッケージの段落になる。パッケージ名が最初に来る。それから対応するアーキテクチャーを書く。データだけのパッケージの場合に尤も all にしてもいい。今度のバイナリパッケージ作成発表に all の他を説明する。その他は Depends ラインに runtime の依存関係のパッケージ（この場合は自動的に抽出される）が書いていて、最後にパッケージの記述。詳しくは Debian Policy Manual を参照してください。

#### 6.2.4 debian/rules

debian/rules でパッケージのビルドがどう行われるかを定義する。中身の形式は完全自由だが外部のツールが Policy Manual に書いている呼び出し方を依存している。それで Makefile 形式を用いているパッケージの割合は 100% に近い。しかし Makefile なのに cdbs を使っているので普通の Makefile に余りにも似ていない。時間と動力を節約したい方は是非試してください。

gnome-backgrounds の debian/rules をみよう。

```
#!/usr/bin/make -f

include /usr/share/cdbs/1/rules/debhelper.mk
include /usr/share/cdbs/1/class/gnome.mk
include /usr/share/cdbs/1/rules/simple-patchsys.mk
include /usr/share/gnome-pkg-tools/1/rules/uploaders.mk
-include /usr/share/gnome-pkg-tools/1/rules/gnome-get-source.mk

binary-post-install/gnome-backgrounds::
    rm -rf debian/gnome-backgrounds/usr/share/locale
```

殆ど以前に定義しているルールを再利用しているしかありません。1 つだけのこのパッケージに特別の追加があり、それは binary-post-install/gnome-backgrounds::のところ。意味としては gnome-backgrounds というバイナリパッケージを作ってから、必要ないのでそのディレクトリを削除する。

<sup>\*7</sup> Debian Policy Manual URL:<http://www.debian.org/doc/debian-policy/ch-archive.html#s-subsections>

<sup>\*8</sup> Build-Depends と Build-Depends-Indep の違いがあまり分からなくてすみません！

### 6.2.5 debian/watch

debian/watch ファイルは不要だが使ったら自動的に新しい上流のソフトを取得してビルドする仕組みが使える。

## 6.3 パッケージビルド

上記を設定してから、パッケージのビルド自体は dpkg-buildpackage 又は debuild コマンドだけで出来る。設定以来の変更を debian/changelog ファイルに記録するとバージョン管理がで要る。dpkg-buildpackage の出力が長い  
ためここに写さないけど自分で試してみてください。



## 7 Debian パッケージにおけるライセンスの取扱い

上川 純一



Debian Project の作成するディストリビューション Debian GNU/Linux はフリーソフトウェアから構成されています。Debian Project における「フリー」の定義は DFSG というガイドラインで定められています。ここでは、それが実際にどういう使い方をされているのかを紹介します。

### 7.1 なぜ DFSG が必要なのか

フリーソフトウェアにもいくつかの考え方があります。どれが正しいというわけではないので、一つに統一することはできないでしょう。具体的には次の三種類くらいを考えればよいでしょう。

- 自由なものを自由でありつづけさせたいライセンス (GPL など)
- 自由なものを不自由にする自由も与えたいライセンス (MIT, BSD など)
- 自分の作ったものは自由につかってくれてよいが、自分の作成したものは完全なため、勝手にいじってほしくない。よって、変更したものについては明示的にしておいてほしいもの (LPPL 1.0 など)

複雑な例を挙げると、昔の  $\text{\LaTeX}$  関連のライセンス LPPL 1.0 では、変更したものについてはファイル名を変更することを義務としていました。また、現在でも gnuplot ではオリジナルのソースコードの配布を義務とし、オリジナルのソースコードとパッチという形での配布のみ認めています。

もともとの開発者と開発に携わろうとしている人たちにとっては大きな違いがありますが、エンドユーザにとってはフリーソフトウェアとして、利用できることに関してはあまり大きな違いはありません。

Debian はそのどちらの立場でもなく、ディストリビュータとして関わることになります。許諾された範囲で次のことができればよいことになります：

- Debian を有償・無償を問わず配布すること (DVD・公開ミラー・非公開ミラー)
- Debian を改変すること (Debian の改良、Debian 外への流用、Debian からの派生ディストリビューションの開発など)
- Debian をあらゆる用途で利用できること (商用・教育用・宗教用・軍用・医療用などを問わない)

これらを現実的に阻害しないようなライセンスの条件を取りまとめたのが DFSG です。

### 7.2 DFSG 全文

DFSG 全文を図 1 に掲載します。

#### 1. Free Redistribution

The license of a Debian component may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

#### 2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form.

#### 3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

#### 4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form **only** if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software. (This is a compromise. The Debian group encourages all authors not to restrict any files, source or binary, from being modified.)

#### 5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

#### 6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

#### 7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

#### 8. License Must Not Be Specific to Debian

The rights attached to the program must not depend on the program's being part of a Debian system. If the program is extracted from Debian and used or distributed without Debian but otherwise within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the Debian system.

#### 9. License Must Not Contaminate Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be free software.

#### 10. Example Licenses

The "GPL", "BSD", and "Artistic" licenses are examples of licenses that we consider "free".

図 1 The Debian Free Software Guidelines (DFSG)

## 7.3 ライセンス管理にかかわるプロセス

Debian GNU/Linux で、ライセンス管理に関連するプロセスや仕組みを整理しましょう。

### 7.3.1 ITP

Debian ディストリビューションにパッケージを追加するのなら、最初に「ITP」を行います。まずここでライセンスも付記することになり、特殊なライセンスであったら、議論になります。

### 7.3.2 debian-legal

複雑な条件や新規の状況が発生した場合には、debian-legal メーリングリストで議論します。メーリングリストでコンセンサスが醸成されたり、されなかったりします。フレームワークが展開されたりします。結論はなかなか出ませんが、いろいろな視点が展開されるので見ているとおもしろいです。

### 7.3.3 NEW キュー

ITP 後パッケージをアップロードすると、メーリングリストの議論とはあまり関係ない方向で、ftp-master が debian/copyright を確認し、パッケージを承認し、新しいパッケージがディストリビューションに入ります。この時点で Debian としてこのパッケージをこのライセンスで配布することは DFSG フリーであるという判断をしたことになります。

この判断をくつがえすには、パッケージに対して、ライセンスが間違っていると serious バグを報告することにな

ります。

### 7.3.4 再配布

ftp-master に一旦入ると、Debian をミラーしている各サーバで再配布されることになります。また、CDROM イメージなどでも再配布されることになります。

各国の著作権法では著作物の複製は基本的には許諾がなければ許可されないというルールです。そのため、Debian としてはライセンスの条件に同意して複製しているという考え方をとっています。各ミラーが到底許諾できないような条項は受け入れられません。

### 7.3.5 改変

ユーザはダウンロードした Debian の一部を改変します。ユーザは改変をする際には、各パッケージの許諾にしたがっています。

## 7.4 パッケージ作成における debian/copyright 作成の実際

Debian パッケージを作成する際に、ライセンス関連を確認します。著作権は原則として許諾がないと「利用」ができないという類の権利です。Debian パッケージの場合、その情報を debian/copyright ファイルに整理します。

では、ライセンスはどうやって確認するのでしょうか。原作者はフリーソフトウェアとして一旦公開することを決心したのですから、良心的にライセンス情報がある程度整理してくれているはずです。しかしながらその内容は監査の手続きが確立されているわけでもなく、標準が存在するわけでもないので一律ではありません。また、よくあることですが、本人以外でその点に注意して確認しようとしたのは今 Debian パッケージ作業をしようとしているあなたが初めてかもしれません。必要事項が記述されていないこともありうるため、その場合は原作者に確認をとりましょう。

ライセンス関連の情報は、通常はマニュアルやウェブページに記述されています。そこで大まかな状況を確認します。例えば、failmalloc パッケージを例にとってみてみましょう。failmalloc のソースディレクトリを見てみます。

```
$ ls
AUTHORS ChangeLog Makefile.am NEWS THANKS configure debian failmalloc.c ltmain.sh mkinstalldirs
COPYING INSTALL Makefile.in README aclocal.m4 configure.ac depcomp install-sh missing
```

まず COPYING ファイルがあるのがわかります。これは FSF の出しているライセンスに多く、GPL 全文が入っている場合が多いです。この場合は内容を確認してみると、LGPL v2.1 の全文が入っていました。

```
GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[略]
```

他にも、マニュアルなどにライセンスについて記述されている事が多いです。確認してみましょう。今回は README, AUTHORS, ChangeLog, THANKS を確認しても特にライセンス関連の記述はありませんでした。

あと、最終的にソースコードそれぞれのライセンスを確認しましょう。別のプロジェクトからコピーしてきたファイルなどが含まれることもあり、別のライセンスのものが混じっている可能性もあります。failmalloc.c を確認すると、ライセンスについて明言している文面があるのが分かり、COPYING ファイルと整合性がとれているのが確認できます。

```

/* failmalloc - force to fail in allocating memory sometimes */
/*
 * Copyright (C) 2006 Yoshinori K. Okuji <okuji@enbug.org>
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * This library 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
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 */

```

今回は特にありませんが、バイナリデータファイルなども確認しておくといでしょう。mp3 ファイルや、画像データなど、本当に作者に権利があるのか念のため確認しておくといでしょう。例えば調べた結果音楽データが Creative Commons ライセンスで許諾されているのであれば、それは別のライセンスのファイルであるとして debian/copyright に記述します。

また、インストールされるバイナリがソースコードから生成されていることを確認します。<sup>\*9</sup> ソースコードから全部のバイナリが作成できないのであれば、原作者に問い合わせるステップが必要になることもあります。

そうやって調査した一連の内容を debian/copyright に整理して記述します。複数の許諾者や許諾内容がある場合はそれがわかるように整理しましょう。この場合は、LGPL である旨を記録し、あと Debian パッケージ関連のスク립トについても他の部分と同じライセンスである LGPL2.1 に従うということを明記しています。

```

This package was debianized by Hideki Yamane (Debian-JP) <henrich@debian.or.jp> on
Sat, 26 Jan 2008 09:28:51 +0900.

It was downloaded from http://www.nongnu.org/failmalloc/

Upstream Author:

    Yoshinori K. Okuji <okuji@enbug.org>

Copyright:

    Copyright 2006 Yoshinori K. Okuji

License:

    This package is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License.

    This package 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
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this package; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

On Debian systems, the complete text of the GNU Lesser General
Public License can be found in '/usr/share/common-licenses/LGPL-2.1'.

The Debian packaging is (C) 2008, Hideki Yamane (Debian-JP) <henrich@debian.or.jp> and
is licensed under the LGPL-2.1, see '/usr/share/common-licenses/LGPL-2.1'.

```

なお、/usr/share/common-licenses に一般的に使われるライセンス文<sup>\*10</sup>が用意されており、一般的な文面の場合はそれを引用することでよいことになっています。そうでない場合はライセンス文を全文 debian/copyright ファイルの中に転記します。

メンテナがこうやって準備した debian/copyright ファイルは、Debian ユーザは /usr/share/doc/パッケージ名/copyright で確認できます。

<sup>\*9</sup> ROM から吸い出したバイナリファームウェアが入っている「フリーソフト」もあるので注意しましょう。

<sup>\*10</sup> 現在は Artistic, BSD, GFDL, GPL, LGPL





**Debian 勉強会資料**

2008年3月15日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）

---