

A large, stylized pink brushstroke graphic that forms a circular shape, resembling a smile or a large letter 'D', is positioned in the background of the slide.

東京エリア Debian 勉強会

資料

上川 純一 dancer@debian.org
IRC nick: dancerj

2008年3月15日



設営準備に
ご協力くだ
さい

会場までの道のり



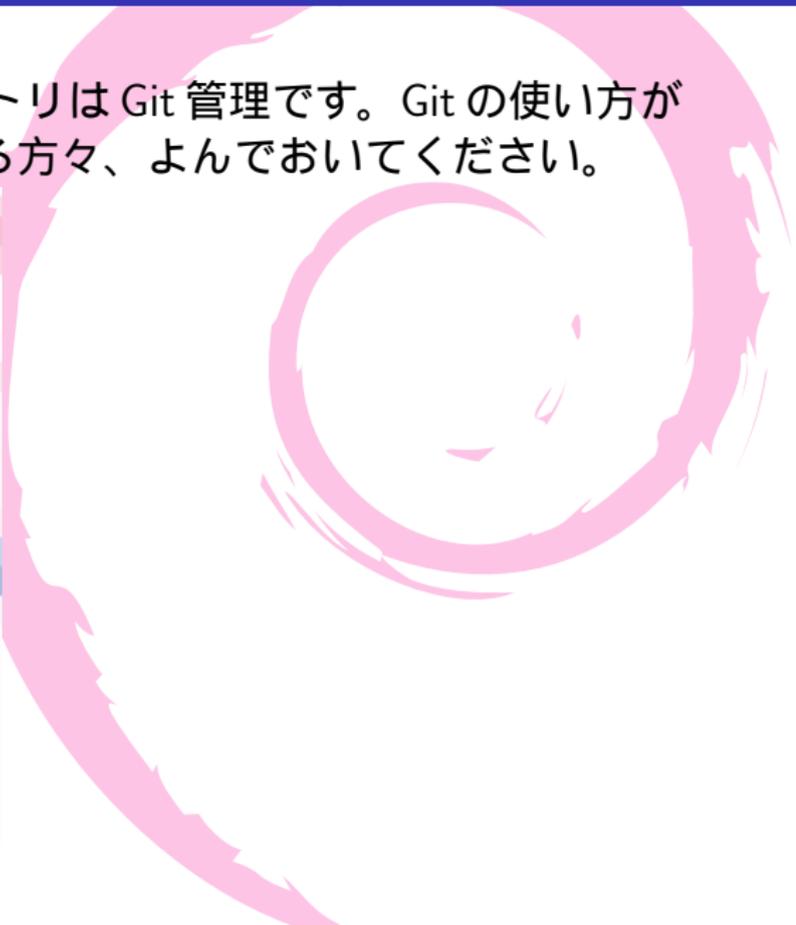
Agenda

- 注意事項
 - 撮影禁止
 - この部屋で話されたことはこの部屋の外には一步も出ません
 - 無線 LAN は ESSID google-guest
- quiz
- 最近の Debian 関連のイベント
 - 前々回
 - 前回 (OSC)
- データだけの Debian パッケージ
- Debian でのライセンスの考え方

前々回の Agenda

- 注意事項
 - 飲食禁止
 - 政治/宗教/営利活動禁止
- quiz
- 最近の Debian 関連のイベント
 - 前回
 - 2008 年度 Debian 勉強会企画
 - Debian Package 管理の流れ

Debian 勉強会のリポジトリは Git 管理です。Git の使い方がわからないといっている方々、よんでおいてください。





Debian 勉強 会とは

Debian 勉強会とは

語ってみる。



Debian 常識クイズ

Debian の常識、もちろん知ってますよね？ 知らないなんて
恥ずかしくて、知らないとは言えないあんなことやこんな
こと、みんなで確認してみましよう。

今回の出題範囲は、

<http://lists.debian.org/debian-devel-announce/> に
ある最近のアナウンス文書です。

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?

答えは:

-  A C
-  B C++
-  C Ruby



B

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

答えは:



A

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?

答えは:



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 Team



B ロゴやボタンなどマーケティングを一手にMarketing



C Debian Cクター「こしにやろ」



B

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っ端から呼び出して説教する

問題4. 3月頭の時点で lenny のリリースによって致命的なバグ (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?

答えは:



A debian/changelog に
「closes: #xxxxxx」と書いて
片っ端からバグを
プロードで



B バグ退治
を開いて片
ジを非メン
(NMU) す



C RCバグ
ケージメン

呼び出して説教する



B

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5. lenny はリリースされると何になる?

-  A Debian 4.1
-  B Debian
-  C Ubuntu

答えは:



B

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 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?

答えは:

-  A Google Search
-  B Hyper E
-  C Xapian



C

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](ftp://ftp.debian.org) からダウンロードできるようになっている

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 全パッケージされている



C パッケージからダウンになっている

答えは:



A

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

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

•  C Steve M

答えは:



C



OSC 參加報告



事前課題の 紹介

事前課題問題

- 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

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されました。「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、だいぶ後のことです。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。

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リプ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！

「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 ライセンスをを評価しているという結論に至る。

「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。

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。そうにな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と思っています。

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、二次利用などがしづらいですね。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ぞ無いわけで、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」なんでしょうかねえ。

好きなソフトウェアライセンスとその理由

GPL。やはり、ソースが有るっちゅ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んだけど墮落の始まりかもしれない。

「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などもローカルにあると役立ちます（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が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スタイルが好みです。短いので。

「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い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れてしまうので、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。

「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。

去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 の研究をしています》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。

「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かな。
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脱しました。
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。

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 かなあ。といいつつ 1 年後には変わっているかもしれません。

「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、心意気がとても伝わってくると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ます (と書くと大袈裟ですね …)。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) 使ってもらう事を望みます。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 …

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際に、きちん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 よりは説明が楽である。

以上が、私が BSD ライセンスを好む理由である。

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、やりようによっては 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。

「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。

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 - D O S や O S / 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を読めるはずもなく、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。

「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」とい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、できるだけ改変の自由を保証してほしいと思っています。改変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、改変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。

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、大きめのものは
gnujdoc¹ 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。よろしく願
いします。

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

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ぱっ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。

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ようになったのですが、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かが、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。

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。

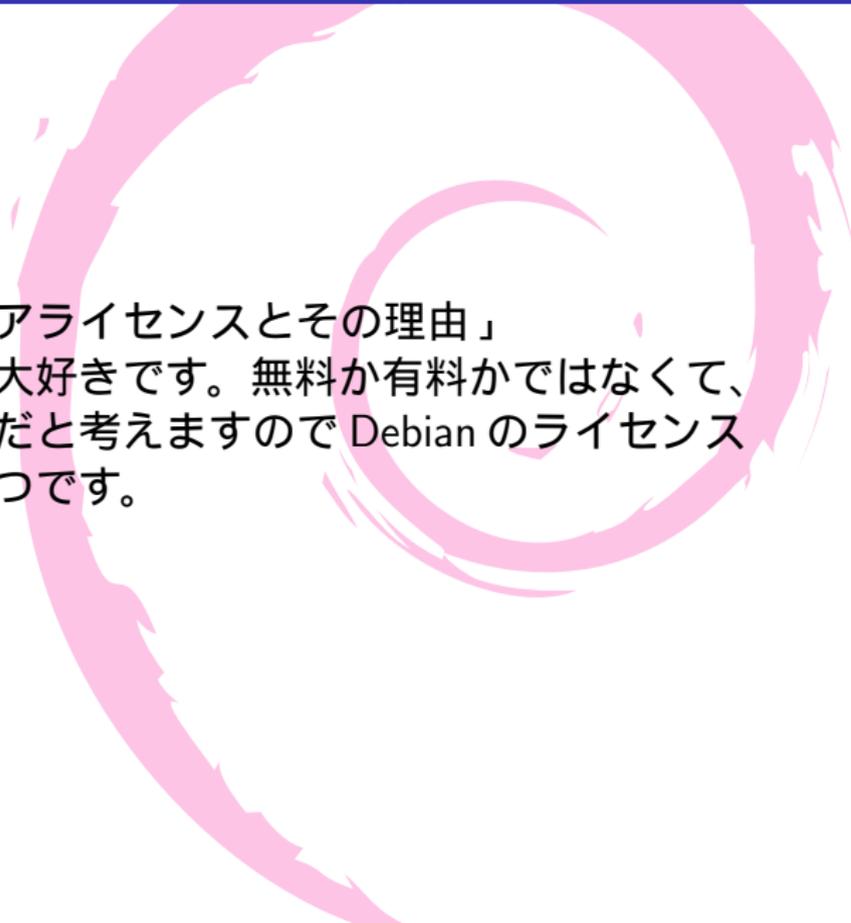
「データだけのパッケージでできる事」ですが、現在の理解ではデータを置く事しかできないんじゃないかと思っちゃいました。fontとかdocumentとかなのかな。当日、勉強して帰ります。

「データだけのパッケージでできること」
どういうものがあるのか考えてみました。とりあえず思い
つくままに

- 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。

A large, stylized pink swirl graphic that starts from the top right and curves around the text, ending at the bottom left. It has a hand-drawn, brush-stroke-like appearance.

「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。

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
もしれませんが…。）

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？

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かと思われる。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、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。

「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と思います。

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。



2008
計画

年



2008年計画

- 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 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

A large, stylized pink circular brushstroke graphic that frames the text on the right side of the image.

データだけ
のパッケージ
作成



Debian での
ソフトウェア
インストールの
考え方

ライセンスの分類?

現実的しがらみ

理想主義

自由風

一部自由

自由
自由でありつづけることには
こだわらない

自由でありつづける

Debian Projectで扱う範囲

大雑把に自由なライセンスの類型でわけてみる

- 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)

Debian Project での使われ方

- 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・軍事用・医療用などを問わない）

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".

Debian パッケージのライセンス管理

発生するイベント

- ITP
- New queue
- 再配布
- 改変



- README
- SOURCE
- BINARY



宴会場所

- 宴会場所
本日の宴会は「庵 GuRi (あぐり) 5566」です。直前に人数を確定して 03-3780-5566 に電話してくださいと依頼されているので電話します。
参加者は B1F に集合し、全員で移動しましょう。
- 片付け
部屋を片付けるのにご協力ください。