



第7回 東京エリア Debian 勉強会 事前資料*

Debian 勉強会会場係 上川純一†

2005年8月6日

* 機密レベル public: 一般開示可能

† Debian Project Official Developer

目次

1	Introduction To Debian 勉強会	2
1.1	講師紹介	2
1.2	事前課題紹介	2
2	Debian Weekly News trivia quiz	6
2.1	2005 年 27 号	6
2.2	2005 年 28 号	7
2.3	2005 年 29 号	8
2.4	2005 年 30 号	8
2.5	2005 年 31 号	10
3	最近の Debian 関連のミーティング報告	11
3.1	東京エリア Debian 勉強会 6 回目報告	11
4	Debconf 参加報告	12
4.1	上川の見た debconf	12
5	ITP の仕方からアップロードまでの流れ	14
5.1	はじめに	14
5.2	ITP とは	14
5.3	ITP をする前に	14
5.4	ITP のしかた	15
5.5	ITP したら	18
6	個人提案課題	19
7	Keysigning Party	20
8	次回	21

1 Introduction To Debian 勉強会

上川純一



今月の Debian 勉強会へようこそ．これから Debian のあやしい世界に入るという方も，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？

目的として下記の二つを考えています．

- メールではよみとれない，もしくはよみとってられないような情報を情報共有する場をつくる
- まとまっていない Debian を利用する際の情報をまとめて，ある程度の塊として出してみる

また，東京には Linux の勉強会はたくさんありますので，Debian に限定した勉強会にします．Linux の基本的な利用方法などが知りたい方は，他でがんばってください．Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりを作りながらスーパーハッカーになれるような姿を妄想しています．

Debian をこれからどうするという能動的な展開への土台としての空間を提供し，情報の共有をしたい，というのが目的です．次回は違うこと言ってるかもしれませんが，御容赦を．

1.1 講師紹介

- 岩松さん SuperH をハックしています．
- gotom さん Debian の glibc とかをやってます．
- 上川純一 宴会の幹事です．Debian Developer です．元超並列計算機やっていて，今は音楽関係とか，気づいたら canna とか．あと，pbuilder や，libpkg-guide などを通して，Debian の品質向上を目指しています．

1.2 事前課題紹介

今回の事前課題は「Debian のここが不満だ!」というタイトルで 200-800 文字程度の文章を書いてください．というものでした．その課題に対して下記の内容を提出いただきました．

1.2.1 吉田@板橋さん

1. 安定版リリース後の対応デバイス追加が無い

新しいデバイスにインストールするのにだれかが iso イメージなりを公開してくれるのを待つしかない？

2. インストール前後の手順がわかりづらい

Debian users ML を見ないと情報が無いのに，誘導がない

一番、日本語でわかりやすくまとまっているのは「Debian GNU/Linux スレッドテンプレ

(<http://debian.fam.cx/>)」だと思いが当然誘導がない

OpenBSD の afterboot とか商用ディストリビューションのようにブラウザを立ち上げると初期誘導で上記のようなところに行く (OrLink) といいのでは? (最近 x は使ってないので実はある?)

3. DFSG に合わないものはパッケージングされない

当然ですしかし、仕方ないので、野良パッケージを作ろうとすると...

4. 野良パッケージを作りにくい

rpm に比べて内部的なブラックボックス (dh~とか) が多く作りにくい

普通に調べると Debian 新メンテナガイドにたどり着くが、あまりにわかりづらく感じる

<http://www.jp.debian.org/doc/manuals/maint-guide/>

多くの項目が盛り込まれていて省略できる点がわかりにくい

かつ、書かれていないが、暗黙の了解事項が多いと思う

1.2.2 中島清貴さん

まず筆頭にあげたいものは名前がよくない。デビというところがデビルマンとかデーモンみたいでかなりよくないと思う。これは Debian に限ったことではないけれど「なんとかかんとか Linux」という書き方もあまりかっこよくないと思う。しかしこの名前もこないだ買った「DebianGNU/Linux 徹底入門第3版 Sarge 対応」だとかかなり良い。けっこう分厚い書籍で長ったらしい名前なので難解そうに感じるけれど徹底入門と書いてあるので、これ読んだだけでマスターしたような気になる。不満というより自分が満足に使いこなしているわけでないで、そんなところだろうか。

1.2.3 松山さん

じっくり他の OS と比べたわけではないけれど、Debian は日常生活を送るにそれなりに使える OS だと思います。それなのに、あまり人々に知られていないところが不満です。

あまり知られていない。

いいものなのに相応の評価がされるどころか、多くの人にとって評価の対象にすらなっていないのは、1 ユーザとしてちょっと面白くない。

大衆に向けてのアピールが弱いところをもっとなんとかすると、それによっていろんな人が使い出すのに伴っていろんな声が聞こえるようになってくると、今までとは一味違った「流れ」のようなものがまきおこるような気がしました。

1.2.4 小林さん

Debian では、綺麗なものは綺麗に配置され管理しやすくなっている一方で、余計なファイルが紙屑のようにあちこちに散らばってしまっているのが気になります。例えば/etc/apache2/ports.conf は、Apache 2 が監視するポート番号を指定するだけのために存在する設定ファイルで、当然ながら中身は 1 行です。また、/etc/default/apache2 も、実質的には、起動を制御する「NO_START」の行 1 行だけです。/etc/default の下には他にも同様のファイルが多くあります。/etc/environment も、環境に関する設定を意識した名前となっていますが、結局ほかのところではできない設定をやるために存在するだけで、場合によっては空ファイルとなっています。/etc/network の下にある ifstate など、`lo=lo` や `eth0=eth0` といった情報

しか書かれないファイルです。この/etc/network/ifstate の処置は、<http://www.debian.org/devel/todo/>の TODO リストに項目として挙げられていますが、特に作業は行われていないようです。

こういった余計なファイルが散らばる背景には、Debian 用パッケージ作成時に上流の設定ファイル以外の設定項目を作る必要が出てきた場合、その設定をどこですべきかというポリシーがないためだと思います。他方で Debian ではどの設定ファイルもいずれかのパッケージに所属するように定めるポリシーがあるため、個々のパッケージはそれぞれ余計なファイルを作って放ってしまうのではないのでしょうか。余計な設定用の処理システムを作り、個々のパッケージが余分な設定ファイルを作らずに済むようにするか、あるいはせめて、そういった余分な設定ファイルの置き場を定めるポリシーがあれば、と思います。

Debian についても一つ気に要らないものがあるとしたら、ウェブ上の文書にも古くなったまま保守されていないものが多いところだと思います。例えば、開発者向けの TODO リストである <http://www.debian.org/devel/todo/> も更新されていません。こういった古い文書は、おそらく片付けようという動きがないためにそのままになっているのですが、コンピュータは検索すれば何でも出てくる世界なので、わざわざ古いものを目につくところに置いておかなくても、と思います。文書などは時間が経てば膨大になっていく一方で、あまりその保守に時間を費やす人がいなく、身動きがとれなくなってしまったのが原因でしょうか。

Debian は文句なしに好きなおところは多いのですが、開発者がメンテナとしてパッケージの新しいバージョンの導入作業およびそれに伴うバグ退治に忙殺されてしまっており、古くからある変なおところがそのまま放置されているように感じられます。その背景にはマンパワー不足があることは誰もが認める事実でしょうし、開発者でさえなく、たとえ開発に参加できてまだまだ力量不足で作業できない自分が言うのはおこがましいのですが、ときには古いものを見直すのに時間を割くのも大切ではないかと思います。

1.2.5 えとーさん

書籍少ない！情報が少ない！人に勧めようとしても書籍ないからって逃げられる。更に言うとパッケージングとかよりメンドクサイところでの書籍や雑誌記事若しくはドキュメントなどがまったくない。更に幾つかやり方があることだと、口コミか全部で試す、などを行わないとなかなかどれを使うものか判断がし難いなどのなにかやろうとすると初期費用が結構高い。

政治面が不透明 FTP マスターや要職の人の離反が起きた時にリカバーする仕組みがないので混乱が長期化してしまう。

DebianJP 謎の組織になってる気がする。ML の運営などは.org の資産でやってもいいのなあと思っているのですが、あまり存在価値がわからない。H10 な状況に即した文章の作成、翻訳、イベント、人の発掘、などが Debian-JP の仕事だと思っていたのだが、そういうことは重視されていないようなので、いよいよ謎の組織になってる。高齢化しすぎ、というよりも新規の人が少ない気がするのですが、魅力的な部分をアピールできていないんでわなないかと思う。

パッケージ管理が辛いパッケージが多いのはよいのだけど、ハンドリングできる数を越えているので、パッケージ検索や、タグ、分類の詳細化などをもっと簡単に出来るようにして欲しい。

プロライエタリソフトウェアとの協調不足 main に入らないとかで Debian としては配布する必要はないと思うが、使えたほうが嬉しいこともあるので、プロライエタリソフトウェアのベンダーが Debian 対応をしたいと思った場合の検証手順のドキュメント化、窓口及び、プロライエタリソフトウェアのベンダーへの協力部隊などが欲しい。Debian のフリーの基準に満ちるソフトウェアの品質がプロライエタリに負けることはないと思うが、選択肢が狭くなるのが嫌なのでなんとかできたら嬉しい。

1.2.6 上川

ねえねえきいてよおくさん．最近 Debian ってさあ，あいかわらず dpkg って conffile が変更されていたらコマンドラインで質問してくるのよ．debconf でほとんど処理できるこの御時勢に，なんてださいのかしら．

ねえねえこれもちよときいてよ．dpkg ってさあ，/var/lib/dpkg/info 以下に大量にファイルを作成していてそこをアクセスするのが遅くってたまらないのよ．

ねえねえさらにこれもちよときいてよ．dpkg を実行する時の遅さ気にならない？もう遅すぎて遅すぎて我慢できないんだけど．

なにこの Packages ファイル．なんでこんなにでかいわけ？

もしかして毎回 description の文字列まで全部読み込もうとしているのかな？本当にやってられない...

あとさあ、聞いてよ．共有ライブラリのパッケージってさ、SONAME が変わったら毎回手作業でインストールするファイルを選んでたりするから空っぽのパッケージがアップロードされてたりするのよ．君らテストしているの？起動テストを少しでもしたらアプリケーションがまったく起動しないのに気づいてよねえ．

ほんとに．こまっちゃう．

2 Debian Weekly News trivia quiz

上川純一



ところで、Debian Weekly News (DWN) は読んでいますか？Debian 界隈でおきていることについて書いている Debian Weekly News. 毎回読んでいるといろいろと分かって来ますが、一人で読んでいても、解説が少ないので、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 DWN を読んでみましょう。

漫然と読むだけではおもしろくないので、DWN の記事から出題した以下の質問にこたえてみてください。後で内容は解説します。

2.1 2005 年 27 号

7 月 5 日版です。

問題 1. 共有ライブラリを作成するために必要な gcc のコンパイルオプションは

- A -fPIC だが i386 などでは-fPIC でなくても動作するという例外がある
- B -shared-library というのがある
- C -ansi

問題 2. sarge をリリース後の sid にて一番壊れている debian-installer の部品は

- A debootstrap
- B joeyh
- C cdrom

問題 3. Lars Wirzenius は RFP が多すぎる、と懸念していた。現在有効な RFP はいくつあるか

- A 1000 以上
- B 100 くらい
- C 500 くらい

問題 4. Firefox と Thunderbird についての商標ポリシーは何が問題か

- A 名前は特許です
- B 変更を加えるとその名前を名乗る事ができない
- C Debian が嫌い

問題 5. glade1 を glade2 に移行することで発生する問題はなにか

- A アプリが重くなる
- B glade2 への移行は gnome2 への移行が必要になるが、まだすべてのアプリケーションが gnome2 に移行できる準備ができていないわけではない

C カーネルのバージョンが 2.6 以上必須になる

問題 6. 教育上不適切だと思われる画像を表示するスクリーンセーバの利用についての質問への回答として Larz が勧めたのは

- A 眼鏡の度数を下げる
- B モニターのかわりにプロジェクターを使え
- C スクリーンセーバは画面を暗転するものだけをデフォルトとしてはいれよう

問題 7. DAK でバージョン文字列の中に `が入ったパッケージを扱う際に問題となるのは

- A 内部で別の目的でセパレータ文字列として利用しているため、それと衝突する
- B チルダは主義主張に反するので扱えない
- C 問題はない

2.2 2005 年 28 号

7月12日版です。

問題 8. Matt Taggart が提示した LSB 3.0 に準拠するのに必要な点はなにか

- A 新しい glibc と xorg パッケージが必要になる
- B 根性
- C 気合い

問題 9. gcc 4.0 を新規にデフォルトコンパイラとして導入することで、何が推奨されたか

- A C++ 関連のパッケージについてはアップロードを控える
- B C++ で書いてあるプログラムは C で書き直す
- C C++ で書いてあるプログラムをできるだけ java のプログラムでおきかえる

問題 10. Ludovic Brenta さんは ada について

- A 一年半関連したパッケージをごっそりとメンテナンスし、管理用ポリシーを書き上げた
- B 上流の開発をしていた
- C Debian Developer としてメンテナンスした

問題 11. パッケージの循環依存についてなにが言えるか

- A 循環依存はできるだけ使うべき
- B 循環依存はおいしい
- C 循環依存はアップグレード処理に際してうまくあつかえないため、直すべき。Pre-Depends を使えば問題を一部解消できる

問題 12. Frank Lichtenheld さんは何を計画したか

- A Debian から non-free なドキュメントを削除する計画
- B Debian から一定の人種を排除する計画
- C Debian から一部のライセンスを排除する計画

2.3 2005 年 29 号

7月19日版です。

問題 13. debian-cd でこれから改善するという 11 の項目にはいついたのは

- A 壊れた依存関係をうみださないようにチェックを追加したい
- B CD の素材はリサイクル可能なものに限定する
- C CD の大きさの規格を変更する

問題 14. GNU Hurd の現状について

- A buildd が稼働していて、40% 程度のパッケージがビルドできている。
- B 誰も使っていないので動くのかよくわからない
- C 今後カーネルは Linux カーネルを使うことになった

問題 15. g++4.0 への移行にともなって c++ で作成した共有ライブラリには何が必要となるか

- A ABI が変更となるので、SONAME がかわっていてもパッケージ名に 'c2' を追加して変更する
- B 特になにもないのでそのまましておく
- C ユーザへのいやがらせのために意味もない依存を負荷すること

問題 16. バグ追跡システムへ追加された機能は何か

- A 過去のバグレポートの傾向をみて重みをつけてくれるシステムが追加された
- B バグ報告の内容から類似のバグを検索して解決策候補を提示してくれるシステムが追加された
- C どのバージョンでバグが存在しているのかを調べるバージョントラッキングが追加された

問題 17. 印刷関連の問題を解決するべく発足したのは

- A debian-printing メーリングリスト
- B プリンタページ
- C 印刷会議

2.4 2005 年 30 号

7月26日版です。

問題 18. piuparts は何をやるツールか

- A Debian パッケージのアップグレードなどを実地検証するツール
- B π を計算するツール
- C ゲーム

問題 19. cpufrequtils を利用してデフォルトの CPU governor をシステム起動の早い時期に変更すること
に対する反論は

- A 起動シーケンスの初期には CPU はどうせ十分働く必要があるし、最初の起動の一分に必要な CPU による電力消費はたいした量ではない
- B CPU の周波数なんて関係ないんです
- C ノートパソコンなぞ使うな

問題 20. エディタ (jed) のデフォルトの外観を設定する debconf の質問を追加することに関しての反応はど
うだったか

- A エディターがデフォルトでどういう外観であるのかということはシステム管理において重要な設定項目
なのでシステム管理者に質問すべき
- B jed にデフォルトの外観なんて無い
- C そんな質問をシステムアドミニストレータにする必要はないだろう。

問題 21. デーモン管理用のパスワードをどうやって管理者に通知するべきかという質問に対する回答は

- A デーモン管理にパスワードは使うな
- B ユーザに設定させる
- C /etc/以下に適切な権限で配置したファイルに記述しておく

問題 22. Debian ロゴに使われている文字のフォントは

- A Poppl Laudatio Condensed という商用フォント
- B monafont
- C kochi

問題 23. BTS へのバグのサブスクリプションが可能になった。どうやったらバグレポートにサブスクライ
プできるか

- A owner@bugs にメールを送る
- B IRC で kamion をつつく
- C XXXX-subscribe@bugs.debian.org にメールを投げる

問題 24. texi2html の新しい挙動はなにか

- A セクション毎にファイルを生成するようにすると、サブディレクトリにファイルを生成するようになる。
- B 毎回エラーで終了する
- C html ではなく xhtml を生成する

問題 25. debtags はどうなったか

- A Debian パッケージ全部にタグをつけるのが完了した
- B Packages ファイルに統合された
- C この世の中から排除された

2.5 2005 年 31 号

8 月 2 日版です .

問題 26. パッケージの description についてレビューをしようというプロジェクトがはじまった . 何に関して議論が収束していないか

- A description は本当に必要なのか
- B どれくらいの技術的な内容を description に含めるべきか .
- C どのパッケージを対象にするか

問題 27. Popularity Contest のレポートの送信で現在圧縮して送信してくれるのはどのプロトコルを利用した場合か

- A SSTP
- B HTTP
- C メール

問題 28. hmh が主張している next generation initscripts はいつごろから始まったプロジェクトか

- A Debconf5
- B Debconf0
- C Debconf2

問題 29. メーリングリストアーカイブのウェブページに最近追加された機能は何か

- A 類似検索機能
- B SPAM 報告機能
- C メールを読んだ気にさせる機能

3 最近の Debian 関連のミーティング報告

上川純一



3.1 東京エリア Debian 勉強会 6 回目報告

前回開催した第 6 回目の勉強会の報告をします。

今月も Debian 勉強会を開催しました。Debian の GNU toolchain の話， dpatch の話でもりあがりました。

- DWN クイズを実施しました。DWN を日本語に翻訳している小林さんが全問正解していました。
- 上川が dpatch の説明をしました。dpatch-edit-patch というツールがあって、それを利用することが可能です。ただ、いろいろとつめがあまり部分もあるので、利用のしやすさについては今後かんがえていきたいな、と思っております。
- gotom さんが toolchain について語りました。glibc の locale 関連ではみんなしらなかったトリビアみたいなものがたくさんあり会場がもりあがりました。特に、`/usr/share/i18n/locales/ja_JP` ファイルの中身についての説明、`date +%Ec` が平成 17 年 07 月 03 日 11 時 32 分 15 秒という形でかえずぞ、という話しについてはもりあがりました。
- はなの舞で宴会，デニーズでデザート。debhelper の makeshlibs は共有ライブラリのデバッグシンボルの扱いなどをあまり楽にしてくれないので、問題ですね、という話題とかでもりあがりました。次回は岩松さんの superh ハック進捗について話しをしていただけるそうです。

4 Debconf 参加報告

後藤正徳, 上川純一



4.1 上川の見た debconf

4.1.1 IM BOF

Input Method BOF がありました。興味は、手の入力などでした。

SCIM が世界の主流になっているようで、canna/kinput2 は世界の主流からとりのこされているようです。UIM もどこかに置きさられているようです。

Debian 全体としては、IM に関しては一つに統一してサポートしていくというのが一番よいのですが、現状日本語環境についてはこれ一本でいこう、と言えるものがそろっていないというのが印象です。

現在の glibc の locale の実装は、環境変数で locale を起動時に設定してしまうため、実行中にアプリケーションの locale を切替えられないという制約があります。そこが今後どうなっていくか、utf-8 で全部やっていくのではないかと、いう考えでした。

IM の設定 HOWTO が必要だね、ということでした。

4.1.2 Altivec

powerpc の機能である、altivec はあまり活用されていないが、プロセッサとキャッシュの間のバスが altivec レジスタは 128bit であるため、l1/l2 cache にのる操作の場合、4 倍近いパフォーマンスの差が出る。libc を書き換えると良くなる関数がたくさんあるよ、という話でした。

聞いている側のコメントとしては、カーネル/ユーザ空間のコピーに利用したらパフォーマンスがでかくなるのではないかと、ということと、git のソースツリーに SHA-1 ハッシュの関数を手で最適化したものがあるよ、という点です。

4.1.3 Debconf を主催する

主催することについてはげしく議論をしていました。大学などのスポンサーがつかないと場所の確保や食糧の確保が難しい。計画と企画参加者*1が重要で、企画参加者が何にむいている人なのかを把握するのが重要だという話しが出ていました。

今後は500人規模になることが予想されるので、物価の高い国でホスティングするのは難しいのではないかと、ということでした。

*1 volunteer

4.1.4 BTS

google が master.debian.org を DoS していたので、現在 BTS は検索できるようになっていません。どうします？ということで、鶴飼さんが fabre を実装していました。

4.1.5 linda

linda の i18n の実装がださいぞーということでもりあがりました。msgid はタグであるべきだ、と StevenK が主張するんですが、それはどうか。

python の gettext は現在の locale を無視して、po ファイルのエンコーディングで文字列を返して来るらしいので、lgettext を利用するパッチを書いて送りました。

4.1.6 securing testing distribution

testing のセキュリティーチームを joeyh が構築したという話でした。協力者がもっと必要そうでした。

4.1.7 ハック

debconf は講義を聞いたり観光をしたりもできましたが、時間を惜しんでハックする場所でした。いろいろな部分に変更になったため、その対応などに追われました。特に、g++4.0 の移行については、debconf の最中に開始になったため、多くのパッケージが対応に追われていたはずです。

- pbuilder: su の変更の対応など
- dpatch: バグの修正と testsuite の追加
- libpkg-guide
- binfmtc
- ecasound: g++4.0 transition の対応

5 ITP の仕方からアップロードまでの流れ

岩松



5.1 はじめに

今回、「このソフトウェアを Debian パッケージにして Debian の一部として提供したい」と思ってるんだけどどうしたらよいか、Debian Developer ではない人の視点から、ITP をどのように行えばいいのかまとめてみました。

5.2 ITP とは

Intend To Package の略で新しく Debian にパッケージを提供したいときに BTS に登録します。これは Debian Developer でなくても 行うことができます。

5.3 ITP をする前に

ITP をする前に チェックしておかなければいけないことがあります。

1. パッケージ化したいソフトウェアのライセンスをチェックします。
Debian は Debian フリーソフトウェアガイドライン (DFSG) に準じているソフトウェアでないといけません。"GPL"、"BSD"、および "Artistic" がそのライセンスの例になります。
2. もうすでにパッケージがある可能性があるのでチェックする。
<http://www.debian.org/distrib/packages> や `apt-cache search` で確認することができます。
パッケージ化されていても Orphaned (みなしご化) されていることもあります。
<http://www.debian.org/devel/wnpp/orphaned> で確認することができます。
ITA (Intent To Adopt) をして引き取ってメンテナンスをしましょう。
3. ITP されていないか確認する。
もうすでに パッケージにしたいソフトウェアが ITP されている可能性があります。
http://www.debian.org/devel/wnpp/being_packaged で確認することができます。
4. RFP されていることをチェックする。
Request For Package の略でパッケージ化の要望が BTS されていることがあります。
<http://www.debian.org/devel/wnpp/requested> で確認することができます。
RFP されているときは バグの内容を RFP から ITP に変更する必要があります。
5. Debian パッケージにしてみる。
提供したいソフトウェアを Debian パッケージにしましょう。ITP してからパッケージ化してもいいと思いますが、先におこなっておいて不具合がないか調べておきましょう。

これらをチェックして ITP をしましょう。

5.4 ITP のしかた

実際にはどのように ITP をしたらよいのでしょうか。方法を以下にまとめてみました。

5.4.1 ITP

1. reportbug の起動

```
% reportbug --email hoge@example.org wnpp
```

として reportbug を起動します。

--email で e-mail アドレスを指定します。

wnpp というのは Work-Needing and Prospective Packages の略です。

2. バグリクエストタイプを聞かれるので ITP を選択します。

```
1 ITP This is an 'Intent To Package'. Please submit a package description
along with copyright and URL in such a report.
```

```
2 0 The package has been 'Orphaned'. It needs a new maintainer as soon as
possible.
```

```
3 RFA This is a 'Request for Adoption'. Due to lack of time, resources,
interest or something similar, the current maintainer is asking for
someone else to maintain this package. He/she will maintain it in the
meantime, but perhaps not in the best possible way. In short: the
package needs a new maintainer.
```

```
4 RFH This is a 'Request For Help'. The current maintainer wants to continue
to maintain this package, but he/she needs some help to do this, because
his/her time is limited or the package is quite big and needs several
maintainers.
```

```
5 RFP This is a 'Request For Package'. You have found an interesting piece of
software and would like someone else to maintain it for Debian. Please
submit a package description along with copyright and URL in such a
report.
```

Choose the request type: 1

3. パッケージ名を求められるので ITP したいパッケージ名を入力します。

```
Please enter the proposed package name: hoge
```

4. パッケージの簡単な説明を求められるので入力します。

Checking status database...

Please briefly describe this package; this should be an appropriate short description for the eventual package:

> hogehoge program

5. 以下のような画面になるので、ソフトウェアの内容 (バージョン、提供元 Web サイト、ライセンス、説明文) を入力します。

Subject: ITP: hoge -- hogehoge program

Package: wnpp

Owner: hoge <hoge@example.org>

Severity: wishlist

*** Please type your report below this line ***

```
* Package name      : hoge
  Version           : x.y.z
  Upstream Author   : Name <somebody@example.org>
* URL               : http://www.example.org/
* License           : (GPL, LGPL, BSD, MIT/X, etc.)
  Description       : hogehoge program
```

(Include the long description here.)

-- System Information:

Debian Release: testing/unstable

APT prefers unstable

APT policy: (500, 'unstable')

Architecture: i386 (i686)

Shell: /bin/sh linked to /bin/bash

Kernel: Linux 2.6.12

Locale: LANG=ja_JP.eucJP, LC_CTYPE=ja_JP.eucJP (charmap=EUC-JP)

6. 入力が終わると送信確認が行われます。

Your report will be carbon-copied to debian-devel, per Debian policy.

Spawning sensible-editor...

No changes were made in the editor.

Report will be sent to "Debian Bug Tracking System" <submit@bugs.debian.org>

Submit this report on wnpp (e to edit) [y|n|a|c|E|i|l|m|p|q|?]?

”y” を押すことによってサーバーに送信されます。これで ITP の作業は終わりです。

5.4.2 RFP ITP

パッケージ化を希望されているパッケージをパッケージ化し、Debian に提供する方法です。RFP のバグに対して BTS します。実際にはタイトルを RFP から ITP に変更します。

以下のようなメールをバグ番号に対して送信します。バグコントロールメールサーバにコマンドを送信しています。

```
owner 283119 ! /// address を #bugnumber の「所有者」に設定
retitle 283119 ITP: libflash -- GPL Flash (SWF) Library /// タイトルを変更する
thanks /// コントロールサーバーへのコマンド終了
```

Hi,

I am interested in this package.

I wish to adopt this package.

Thanks,

Iwamatsu

5.4.3 O ITA

みなしご化されているパッケージを引き取ってメンテナンスしたいときの方法です。Orphaned のバグに対してメールをします。実際にはタイトルを O から ITA に変更します。

以下のようなメールをバグ番号に対して送信します。バグコントロールメールサーバにコマンドを送信しています。

```
owner 283119 ! /// address を #bugnumber の「所有者」に設定
retitle 283119 ITA: libflash -- GPL Flash (SWF) Library /// タイトルを変更する
thanks /// コントロールサーバーへのコマンド終了
```

Hi,

I am interested in this package.

I wish to adopt this package.

Thanks,

Iwamatsu

5.5 ITP したら

1. スポンサーを探す ITP をしたら スポンサーを探す必要があります。スポンサーとは スポンサーは現存する Debian Developer で 自分の指導者をしてくれる方です。Debian Developer でないとパッケージを Debian にアップロードすることができません。スポンサーは作成したパッケージをチェックし、パッケージをアップロードしてくれます。また、間違いがあるときは指摘をしてくれたり、アドバイスをしてくれます。

スポンサーを探すには `debian-mentors` ML で聞いてみるとよいでしょう。

<http://lists.debian.org/debian-mentors/>

また、GPG キーサインでサインして頂いた Debian Developer の方に相談するのも方法のひとつです。

2. スポンサーが見つかったら スポンサーが見つかったらそのスポンサーとやり取りし、パッケージを改善します。

スポンサーがアップロードしてもいいと判断された場合、スポンサーの手によって Debian にアップロードされます。

6 個人提案課題



名前 _____

下記の空欄を埋めてください:

Debian の不満な、()
に注目し、今後の Debian は ()
します。

企画案の図：

7 Keysigning Party

上川純一



事前に必要なもの

- 自分の鍵の fingerprint を書いた紙 (gpg --fingerprint XXXX の出力.)
- 写真つきの公的機関の発行する身分証明書, fingerprint に書いてある名前が自分のものであると証明するもの

キーサインで確認する内容

- 相手が主張している名前の人物であることを信頼できる身分証明書で証明しているか^{*2}.
- 相手が fingerprint を自分のものだと主張しているか
- 相手の fingerprint に書いてあるメールアドレスにメールをおくって, その暗号鍵にて復号化することができるか

手順としては

- 相手の証明書を見て, 相手だと確認
- fingerprint の書いてある紙をうけとり, これが自分の fingerprint だということを説明してもらう
- (後日) gpg 署名をしたあと, 鍵のメールアドレスに対して暗号化して送付, 相手が復号化してキーサーバにアップロードする (gpg --sign-key XXXXX, gpg --export --armor XXXX)

^{*2} いままで見た事無い種類の身分証明書を見せられてもその身分証明書の妥当性は判断しにくいので, 学生証明書やなんとか技術者の証明書の利用範囲は制限される. 運転免許証明書やパスポートが妥当と上川は判断している

8 次回



次回は 9 月 10 日土曜日の夜を予定しています。内容は本日決定予定です。
参加者募集はまた後程。