



第 9 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 10 月 15 日

* 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	3
2.1	2005 年 37 号	3
2.2	2005 年 38 号	4
2.3	2005 年 39 号	5
2.4	2005 年 40 号	6
2.5	2005 年 41 号	8
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 8 回目報告	11
4	apt-listbugs	12
4.1	はじめに	12
4.2	仕組み	12
5	debbugs の内部構造	13
5.1	はじめに	13
5.2	データ形式	13
5.3	コード形式	16
5.4	そして何がおきたか	17
6	次回	18

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ふ方も，すでにどっぷりとつかっているといふ方も，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 をがりがりと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ふ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

- 謎の講師 apt-listbugs を開発した人です．
- 上川純一 宴会の幹事です．

1.2 事前課題紹介

今回の事前課題は「Debian のバグシステムに物申す」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

1.2.2 上川

あとで埋める予定．

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 37 号

2005 年 9 月 13 日です．

問題 1. バグトラッキングシステムの見栄えで最近かわったのは何か

- A CSS を利用するようになった
- B DHTML になった
- C XHTML になった

問題 2. Debian UK で問題になったのは何か

- A メンバーが活動的でないこと
- B UK の経済状況がよろしくないこと
- C 商用利用をしようとした場合の Debian という名前の商標の利用の許可をする基準が不明確だったこと

問題 3. ソフトウェアを計測する，という論文で発表されたのは何か

- A Debian sarge には 2 億 3000 万行のソースコードが含まれている
- B Debian sarge の品質を計測した
- C Debian sarge の利用しやすさを計測した

問題 4. Joey Hess は testing に対して security 対応をすることを発表した．それに利用しているサーバはどれか

- A secure-testing.debian.net
- B security.debian.org
- C security.debuan.org

問題 5. /usr/doc をいまだにつかっているパッケージ数はどれくらいか

- A 100
- B 200
- C 500

問題 6. planet.debian.org をメーリングリスト経由で配布しようという意見に対して出た反論は

- A blog の内容は機密事項なので、メーリングリストで配布してはいけない
- B メーリングリストとして配布するとサーバの負荷が高くなる
- C blog の内容を永続的にメーリングリストのアーカイブとして保存されたくない

問題 7. `/usr/share/doc/パッケージ名/examples/` にあるファイルに実行権限をつけることについてはどうすべきか

- A サンプルは実行できるものは実行権限をつけるべき
- B サンプルなんてかざりなので実行しなくてよい
- C `/usr/share` 以下について実行権限をつけるのはこのましくなく、実行ファイルは `bin` におくべきだ。

問題 8. `sponsors.debian.net` が提供するサービスは何か

- A 金銭的寄付をつのるフィッシングサイト
- B 広告を配信し、広告収入を Debian プロジェクトの発展のために利用するサイト
- C まだメンテナになっていない人が管理しているパッケージについてスポンサーが必要な状況をトラッキングするシステム

問題 9. `1.0beta3` のようなベータ版のバージョン番号が `1.0` のような最終版のバージョン番号より低い、と `dpkg` が判定してしまう。この状況に関してメンテナはどう対応すべきか

- A 優先度の低いチルダ記号 `~` を利用して、`1.0~beta3` のような名前にする。ただまだアーカイブシステムが対応していないので、今後の改善が必要。
- B あきらめる
- C ベータ版はパッケージ化しない

問題 10. ソースのみのパッケージのアップロードを可能にするという提案についての反論は何か

- A バイナリが必要でなくなると、メンテナがテストをしなくなるのではないだろうか、という懸念がある
- B ソースのみだとパッケージインフラが破綻する
- C `katie` を改変するのが面倒

問題 11. BTS に任意のタグを追加できる機能が追加された、なんという機能が

- A `tagtag`
- B `たぐるんです`
- C `usertag`

2.2 2005 年 38 号

2005 年 9 月 20 日です。

問題 12. David Moreno Garza が `wnpp` にて `close` したバグレポートの数は

- A 729 のバグレポート
- B 100 のバグレポート
- C 123 のバグレポート

問題 13. International Conference on Open Source Systems に投稿された論文の中で説明されていた結果は

- A 開発者は短期間でどんどん入れ替わる
- B メンテナは実は幻想で、そんな人は存在しない
- C 長いあいだアクティブに活動し、パッケージの数も多くメンテナンスする

問題 14. Frank Lichtenheld が発表したのは、`non-free` なドキュメントを削除する処理を開始するということだっ

た．状況をトラッキングするために彼が利用したインフラは．

- A BTS の usertags 機能で debian-release@lists.debian.org ユーザのタグとして管理
- B Wiki ページ
- C CVS 管理のテキストファイル

問題 15. Software freedom day 05 で Debian-women が行って，結果として良かったので今後も継続することになったのは

- A debian-women-new IRC チャンネルがよい結果をもたらしたので，今後は debian-women チャンネルに新人を歓迎する時間帯というのをもうける
- B CD をたくさん焼いたら人気だった
- C Katie や BTSなどをインストールしてユーザがいじれるように提供したら人気だったので，今後もやる

問題 16. init.d スクリプトは現在直列に実行されているが，今後，並列実行を実装する際に便利だろうと思われる LSB 規格の仕様は

- A なんとなく並列に実行しても壊れないようにする仕様
- B 気持ちの中だけでは並列な年頃
- C init スクリプトの中で依存関係を記述できる仕様

問題 17. 新しいバージョンのパッケージにて問題が解決した場合の，バグレポートをクローズする方法でないのは何か

- A changelog でバグ番号を記述しアップロードする
- B バージョンヘッダを付けて，リクエストを -done アドレスに投げる
- C btsclose コマンドを利用する

問題 18. Marc Brockschmidt が説明した，新規メンテナプロセスの Front Desk の変更とは

- A 今後はより厳しい思想チェックを行う
- B Debian にコントリビュートしていることが要件になり，何もしていない場合は，応募が取り消される
- C 年齢制限を設けます

問題 19. security.debian.org で問題になったのは何か

- A セキュリティーアップデートが遅い
- B セキュリティーアップデートが嘘だった
- C xfree86 のセキュリティアップデートがあまりにも高いネットワーク負荷を発生させてしまい，機能しなくなってしまった．

2.3 2005 年 39 号

2005 年 9 月 27 日です．

問題 20. Ben Hutching が Debconf について報告したのは

- A もう終ってしまった事は忘れる
- B 忘れ物がありました
- C DVD が入手可能になった

問題 21. wiki.debian.org への移行で労力が必要だったのはどこか

A すでに wiki.debian.net から wiki.debian.org に移行してしまっているページがいくつかあったのでそれに対する手動の対処

- B kwiki から moinmoin ヘデータ形式の変更
- C ドメイン名の登録

問題 22. init の時点では/が read-only でマウントされているが、その時点でデータを保存するにはどうしたらよいか。

- A メモリファイルシステムを/run にマウントする
- B /mnt 以下にメモリファイルシステムをマウントする
- C /を rw にマウントしなおす

問題 23. GLU のライブラリが複数ある理由はなぜか

- A 一部のコードが一部のハードウェアでしか動かないという状況が続いているから
- B 複数のパッケージをメンテナンスしているほうがかっこいいから
- C メンテナの仲が悪いから

問題 24. Jeroen van Wolffelaar が提案したのは

- A libc5 を消す
- B libc6 を消す
- C libc6.1 を消す

問題 25. piuparts であきらかになる問題は

- A purge する際に、essential ではないパッケージに依存して、動作しないパッケージ
- B インストールしても動かないパッケージ
- C 使ってみて使いにくいパッケージ

2.4 2005 年 40 号

2005 年 10 月 4 日です。

問題 26.

- A
- B
- C

問題 27.

- A
- B
- C

問題 28.

- A
- B
- C

問題 29.

- A
- B
- C

問題 30.

- A
- B
- C

問題 31.

- A
- B
- C

問題 32.

- A
- B
- C

問題 33.

- A
- B
- C

問題 34.

- A
- B
- C

問題 35.

- A
- B
- C

問題 36.

- A
- B
- C

問題 37.

- A
- B
- C

問題 38.

- A
- B
- C

問題 39.

- A
- B
- C

問題 40.

- A
- B
- C

問題 41.

- A
- B
- C

問題 42.

- A
- B
- C

問題 43.

- A
- B
- C

2.5 2005 年 41 号

2005 年 10 月 11 日です .

問題 44.

- A
- B
- C

問題 45.

- A
- B
- C

問題 46.

- A
- B
- C

問題 47.

- A
- B
- C

問題 48.

- A
- B
- C

問題 49.

- A
- B
- C

問題 50.

- A
- B
- C

問題 51.

- A
- B
- C

問題 52.

- A
- B
- C

問題 53.

- A
- B
- C

問題 54.

- A
- B
- C

問題 55.

- A
- B
- C

問題 56.

- A
- B
- C

問題 57.

- A
- B
- C

問題 58.

- A
- B
- C

問題 59.

- A
- B
- C

問題 60.

- A
- B
- C

問題 61.

- A
- B
- C

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 8 回目報告

前回開催した第 8 回目の勉強会の報告をします。

4 apt-listbugs

たるさん?



4.1 はじめに

この文書は 2005 年 3 月に北京で開催された Asia Debian Mini Conf in Beijin で発表した資料の日本語訳です。

Debian は 24 時間 365 日、多くの開発者によって日夜開発されている自由なオペレーティングシステムです。わたしたちはこのような最新のスナップショットを Debian のアップグレードフレームワークを使うことで簡単に共有することができ、実際に、このフレームワークを利用して多くの人が最新の Debian スナップショットを利用しています。

一方で、この最新スナップショットはときどき壊れることがあります。apt-listbugs は、このような壊れたスナップショットからあなたのコンピュータを守るための仕組みです。

この短い文書では、apt-listbugs がどのようにコンピュータを守るのか、apt-listbugs の生い立ち、実装について簡単に説明します。また apt-listbugs の抱えている現在の問題点について紹介します。

4.2 仕組み

Debian には Debian バグ追跡システム (BTS) として知られる、Debian に関する全ての問題が利用可能な中央管理型バグ追跡システムがあります。Debian のパッケージインストールフレームワークである APT は apt-get や aptitude 等を利用してインストールやアップグレードを行う直前に BTS からこれらの問題を取得するために apt-listbugs を呼び出します。apt-listbugs はどのパッケージがインストールされるのか、またアップグレードされるのかを自動認識し、問題となるバグを見付けた場合はその事をユーザに通知し、インストールを継続するか警告を出します。

このアプローチは、主にふたつの問題、ひとつはユーザの視点、もうひとつは開発者の視点を解決します。

5 debbugs の内部構造

上川



5.1 はじめに

この文書は Anthony Towns がフィンランドの debconf 5 で発表した内容を日本にて展開するための資料です。Anthony Towns の作成した英語の資料を省略して抜粋しています。また、それ以降に変更した事項について追記しています。

Debian Bug Tracking System (BTS) は、ほぼ Debian に特化したバグ報告の管理のためのシステムです。他のプロジェクトでも利用されていることもありますが、Debian でバグがパッケージベースで厳格に分類できることなどの特性が反映されているため、Debian プロジェクトでいちばんつかいやすいように作られています。

Debian のインフラと統合されており、changelog にバグ番号を記述してパッケージをアップロードしたらバグが修正されたと認識されるようになっていたりします。

規模としては、55000 以上の現在アクティブなバグ報告、231000 アーカイブされたバグ報告を現在保持していて、毎週 1000 以上の新規のバグ報告が追加されています。ウェブインタフェースはすぐに追加された報告を反映しており、過去、ダウンタイムもほとんど発生していません。

Anthony Towns によると下記が要件です。

- インタフェース：開発者がメールで操作できるようになっており、誰でもウェブで閲覧できるようになっている。
- パッケージベース：バグ報告をパッケージ別に高速に管理する必要がある
- スケーラビリティ：大量のバグ報告に対応できる必要がある
- 即時性：現在のバグの状態をすぐに報告してくれる必要があり、バグの状態が変更されたらすぐに反映される必要がある
- 安定性：継続して動作する必要がある。新規の機能がどんどん追加されたとしても。
- 公開：議論の内容に Debian コミュニティ全体として参加できるように、永続的な公開記録として保存される必要がある。

5.2 データ形式

バグデータベースのスプールの形式は下記です。リレーショナルデータベースなどは利用していません、スプールディレクトリ以下にほとんどのデータが格納されています。

各バグについて、ファイルはそれぞれ 4 個あります。サマリーファイルはメタデータを保存します。ログファイルは、そのバグに対して流れたメールを全て保存します。

status ファイルは互換性のためだけに存在しています。report ファイルは、最初のバグ報告のメールで、バグが close されるときに送信されるものです。

- /org/bugs.debian.org/spool
 - incoming/
 - * T.*
 - * S[BMQF DU RC] *.*

```

    * R[BMQFDU RC] *.*
    * I[BMQFDU RC] *.*
    * G[BMQFDU RC] *.*
    * P[BMQFDU RC] *.*
- db-h/
    * 00/
        . ..
        . 314200.log
        . 314200.report
        . 314200.status
        . 314200.summary
    * ..
    * 99/
- archive/
    * 00/
        * ..
        * 99/
- index.db - index.db.realtime
- index.archive - index.archive.realtime
- nextnumber

```

5.2.1 incoming

incoming に来たメールは処理中、名前を変えます。

- T receive によってうけとられた
- S SPAM 確認待ち
- R SPAM 確認中
- I SPAM チェック通った
- G service か process スクリプトを通った
- P process 中

また、ファイル名の二つ目の文字はどこのメールアドレスにメールが送信されてきたものなのかということを示します。ファイル名ののこりは、バグ番号と、一意な ID です。一意な ID を決定するのに現在は時間とプロセス番号を利用しています。

- B: 通常のバグ報告 submit@ 1234@
- M: -maintonly メーリングリストに投げない
- Q: BTS に登録しない -quiet
- F: アップストリームにフォワード -forwarded
- D: バグ終了 -done
- U: サブミッターにメール -submitter
- R: ユーザのリクエスト用インタフェース request@
- C: デベロッパーの制御用インタフェース control@

5.2.2 Status と Summary

status ファイルの中身は行ベースです．無い行については空行とみなします．このファイルは今後なくしていこうとしています．

- バグ報告者のメールアドレス
- 時間 (秒)
- サブジェクト
- 元のメールのメッセージ ID
- バグがアサインされているパッケージ
- タグ
- close した人のメールアドレス
- 上流のメールアドレスか URL(forward されたばあい)
- マージされているバグ番号
- severity

summary ファイルは RFC822 形式で，拡張可能になっています．現在 Format-Version: 2 と 3 の二つの形式があります．3 は，ヘッダについては RFC1522(MIME) のデコードされた形式になっています．

- Format-Version: このファイル形式のバージョン
- Submitter: バグ報告者のメールアドレス
- Date: 時間 (秒)
- Subject: サブジェクト
- Message-ID: 元のメールのメッセージ ID
- Package: バグがアサインされているパッケージ
- Tags: タグ
- Done: close した人のメールアドレス
- Forwarded-To: 上流のメールアドレスか URL(forward されたばあい)
- Merged-With: マージされているバグ番号
- Severity: severity
- Owner: バグの所有者

5.2.3 log ファイル

あらゆるメールが log ファイルには追記されていきます．また，メタデータも追記されていきます．残念ながら，メタデータは生の HTML で書かれており，またバージョンによって記述の仕方が変わっており，さらに悪いことに，古いバグの中にあるテキストは更新されていないため，機械的に処理することは難しくなっています．

また，コントロール情報は，行頭のエスケープコードにより切り替わります．メールの中にエスケープコードのような文字列が出て来たら，それは 030 を追加してエスケープします．

詳細は Debbugs::Log を見てください．

- kill-init: まだ一行も処理していません
- incoming-recv: 07: あとに go がくる，Received:行
- autocheck: 01: X-Debian-Bugs-...: までの無視されている行，autowait が次に来る
- html: 06: 生で表示すべき HTML
- recips: 02: メールを受取人，04 で分割されている
- go: 05: メール本文

- go-nox: X: メールの文書, X ではじまる行
- kill-end: 03: メッセージの終り.
- autowait: go-nox があとにくる, 空行まで無視されるその他の情報.

5.2.4 Index ファイル

index ファイルは, pkgreport.cgi がどのパッケージにどのバグがわりあてられているかを確認するための情報です. 以前は, by-package.idx と by-severity.idx というのがあり, 高速化に貢献するはずだった. 一年以上長い間生成されていなかった., 誰も気づかなかったので必要ないんじゃないだろう.

データ形式としては下記のようになります. パッケージ, バグ番号, 時間, ステータス, メールアドレス, severity の順に書いた行が全てのバグに対して作成されています.

```
pbuilder 317998 1121196782 open [Junichi Uekawa <dancer@netfort.gr.jp>] normal
```

5.3 コード形式

debbugs は特に設計もされずに長い間パッチを累積してきました. ただ, 明確にわかれている部分はあって, メールを処理するコアのインタフェースのスクリプトと, ウェブを表示するための CGI 部分とで分離できます.

設定ファイルは全て/etc/debbugs にあります.

5.3.1 コアのスクリプト

メールを処理する部分があります.

- errorlib: ライブラリ
- receive: MTA からメールを受信する
- spamscan: 受信メールを SPAM チェックする
- processall: process と service にメールを分配する
- process: バグメールを処理する
- service: control@ と report@ メールを処理
- expire: close されてから 28 日過ぎたバグをエキスパイア処理する
- rebuild: index ファイルをリビルド

receive と rebuild 以外は cron から起動しています. 15 分に一回しか動作しません.

5.3.2 CGI スクリプト

CGI 関連は, errorlib 関数を活用している部分もありますが, ほぼ独立しています.

- bugreport.cgi: バグレポートを一つ表示
- pkgreport.cgi: パッケージやサブミッタなどでサマリを作成する
- pkgindex.cgi: パッケージや severity に対して数を表示
- common.pl: ライブラリとして利用

pkgreport.cgi は特に速度が重要視される部分なので, 注意.

5.3.3 ハックするには

debbugs のソースは CVS にあります. また, Debian Developer であれば, ミラーが merkel.debian.org の /org/bugs.debian.org 以下にあります.

5.4 そして何がおきたか

Google が master.debian.org にとって DoS になるような検索の仕方をしていたので、現在 BTS は google の検索対象にははいっていないので検索サービスが必要だろう。

バージョントラッキングは重要な機能だろう。

ユーザタグってどうやって実装されてるのよ。

request@bugs.debian.org にて: user debian-release@lists.debian.org usertag 123456 + nonfree-doc gfdl

6 次回



関西出張会議を 10 月 29 日に実施する予定です。

また、東京での次回は 11 月 12 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。