

A large, stylized pink brushstroke graphic that forms a partial circle or swirl, framing the central text.

東京エリア Debian 勉強会

資料

上川 純一 dancer@debian.or.jp
IRC nick: dancerj

2009年4月18日



設営準備に
ご協力くだ
さい

今日の参加の目標

- 前田: ocaml 勉強しに来ました
- きたはら: ocamlって何ですか
- 中尾: java policy 勉強しにきました
- 山本: 飲みにきました
- 吉田@板橋: poken 個人情報漏洩しにきた
- まとはら: 誰か Debian で動く outliner のよいソフトをおしえてください
- あけど: emacs を使えるようになりたい。なんか紹介してください。MacOS の消し方を知りたい。emacs -nw
- 高橋(仮): poken を受けとりに着た
- やまだ: DDTP: 作業効率よくなるネタがあるとうれしい。omegat?
- 小川: poken を渡しに来た。
- 日比野: ocaml の入門ぷりをチェックしたい。

前回考えていた今回のテーマ


- 事前課題: 「やまねさんの開発効率を上げる方法を提案してください。」
- 藤澤さん: 連載 GXP: ITP してみました。
- John: I made a clozurecl package!
- やまね: 俺の開発効率をあげてください session。

Agenda

- 注意事項
 - 飲食禁止
 - 政治/宗教/営利活動禁止
- 最近あった Debian 関連のイベント報告
 - 前回の勉強会
 - OSC
 - Ubuntu オフラインミーティング@秋葉原
 - Linux Consortium 10 years event
 - Hack Cafe
- Debian Java policy について
- Debian 開発のワークフローについて語る会
- ocaml 勉強はじめました

- 注意事項
 - ?
- 最近あった Debian 関連のイベント報告
 - 前回の勉強会
 - OSC
 - Ubuntu オフラインミーティング@秋葉原
 - Linux Consortium 10 years event
 - Hack Cafe
- Debian クイズ
- 研究室のソフトウェアを Debian パッケージにしてみる
- Debian での Common Lisp プログラミング環境

毎週水曜日、週に一回東京のどっかのカフェでハック。






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

`debian-devel-announce@lists.debian.org` に投稿された
内容と Debian Project News からです。

問題1. 4月9日にアップデートされた etch のバージョンは?

-  A 4.0r8
-  B 4.0beta8
-  C etch-a-sketch

問題1. 4月9日にアップデートされた etch のバージョンは?




-  A 4.0r8
-  B 4.0beta8
-  C etch-a-s

答えは:






A

問題2. 4月11日にアップデートされた lenny のバージョンは?

-  A 5.0r1
-  B 5.0.1
-  C lenny++



問題2. 4月11日にアップデートされた lenny のバージョンは?




-  A 5.0r1
-  B 5.0.1
-  C lenny++

答えは:



B

問題3. Debian.org DPL になったのは？

-  A Stefano Zacchiroli
-  B Steve McIntyre
-  C Nobuhiro Iwamatsu

問題3. Debian.org DPL になったのは？



A Stefano Zacchiroli



B Steve M



C Nobuhir



答えは:

B

問題4. Debian JP Leader になったのは?

-  A Kei Hibino
-  B Hiroyuki Yamamoto
-  C Nobuhiro Iwamatsu

問題4. Debian JP Leader になったのは?




-  A Kei Hibino
-  B Hiroyuki
-  C Nobuhiko

答えは:



C

問題5. Debian で新しく追加されたアーキテクチャは？

-  A GNU/kFreeBSD
i386/amd64
-  B GNU/kMinix-3.0
i386/amd64
-  C GNU/kOpenDarwin
i386/amd64

問題5. Debian で新しく追加されたアーキテクチャは？



A
i386/amd64



B
i386/amd64



C
i386/amd64

GNU/kFreeBSD



答えは:

A

2009年計画

- 1 新年の企画 (アンサンブル荻窪開催)
- 2 OSC Tokyo
- 3 VAIO P インストール記録、カーネル読書会 ディストリビューション大集合 (小林さん)(東京大学?)
- 4 Git Handson (岩松)(あんさんぶる荻窪?)
- 5 家 Debian サーバ vs 職場のネットワーク (千代田区都立図書館?¹)
- 6 Asterisk (東京大学?)
- 7 スペインにて開催
- 8 Debconf 報告会
- 9 OSC Fall?
- 10 udev + HAL
- 11 3D graphics 開発
- 12 Debian サーバ+VMware + 各種 OS、他の仮想化ツール (vserver etc.)、忘年会

¹<http://www.library.chiyoda.tokyo.jp/>



Debian
Java



Debian で
Ocaml
使ってみた

Ocaml はじめてみました

ocaml 使われている。advi とか unison とか重要なパッケージが ocaml でかかっているんだけど、何がかかっているのかまったく理解できない。

ocaml がまったく理解できないのはまずいので何冊か図書館から本借りてきて読むことにした。読むだけではやるきが出ないので今日発表することにしたぜ。

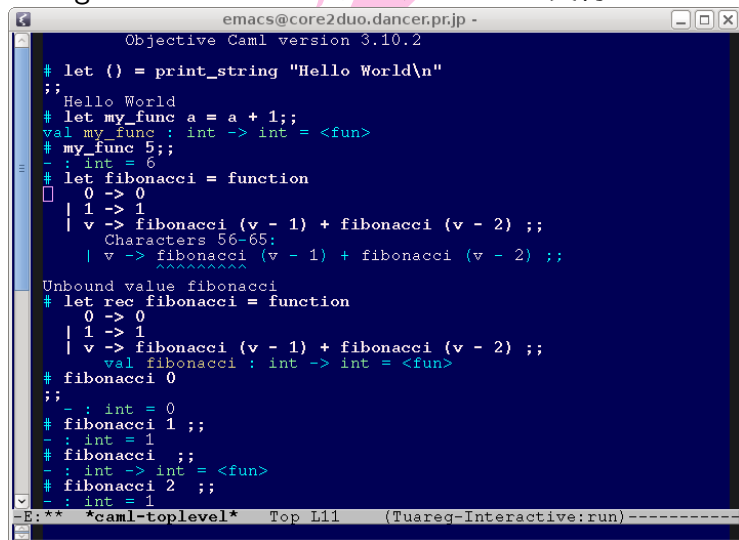
ocaml の普及度

```
$ apt-cache search ocaml | wc -l  
224
```

emacs での利用方法

tuareg-mode というのがあるよ。

tuareg-run-ocaml でインタラクティブに実行

The image shows a screenshot of the Emacs editor window titled 'emacs@core2duo.dancer.pr.jp'. The window contains Objective Caml code and its execution output. The code defines a 'Hello World' function, a simple increment function 'my_func', and a recursive Fibonacci function 'fibonacci'. The output shows the results of evaluating these functions: 'Hello World', '6', and the first three Fibonacci numbers (0, 1, 1). The status bar at the bottom indicates the current mode is 'caml-toplevel' and the cursor is at 'Top L11'.

```
Objective Caml version 3.10.2

# let () = print_string "Hello World\n"
;;
Hello World
# let my_func a = a + 1;;
val my_func : int -> int = <fun>
# my_func 5;;
- : int = 6
# let fibonacci = function
| 0 -> 0
| 1 -> 1
| v -> fibonacci (v - 1) + fibonacci (v - 2) ;;
Characters 56-65:
| v -> fibonacci (v - 1) + fibonacci (v - 2) ;;
^^^^^^^^^^
Unbound value fibonacci
# let rec fibonacci = function
| 0 -> 0
| 1 -> 1
| v -> fibonacci (v - 1) + fibonacci (v - 2) ;;
val fibonacci : int -> int = <fun>
# fibonacci 0
;;
- : int = 0
# fibonacci 1 ;;
- : int = 1
# fibonacci ;;
- : int -> int = <fun>
# fibonacci 2 ;;
- : int = 1

-E: ** *caml-toplevel* Top L11 (Tuareg-Interactive:run)-----
```


インストール

```
# apt-get install tuareg-mode ocaml-native-compilers ocaml-interp ocaml
```

とりあえずコード書いてみた

```
# 1 + 2 ;;  
- : int = 3
```

とりあえずコード書いてみた

```
let rec fibonacci = function
  | 0 -> 0
  | 1 -> 1
  | v -> fibonacci (v - 1) + fibonacci (v - 2)

let () = print_int (fibonacci 40); print_newline ()
```

コンパイラでコンパイルしてみる

	ネイティブコンパイルされた	中間言語にコンパイルされている
ネイティブコンパイル結果を出力	ocamlopt.opt	ocamlopt
中間言語を出力	ocamlc.opt	ocamlc


fibonacci でとりあえず処理能力を調べてみた

fibonacci(40) を求めてみた。再帰アルゴリズムで、C、ocamlpt、ocaml を比較。

- C: 2.325s
- ocamlpt: 2.933s
- ocaml: 14.544s

先はまだ長そう



- 
- ① 各自の「私の Debian 開発ワークフロー」を紹介してください。

私の Debian ワークフロー

- メールでバグレポートを受け取る
- コードを直す・パッチを git am で適用
- pbuild-normal スクリプトを実行、cowbuilder -update, cowbuilder -build が実行され、一連のインストール・実行テストスクリプトが実行される。成功したら pending ディレクトリにパッケージが移動される。
- pending ディレクトリを確認、debsign で署名、dput でアップロード

こう改善したい

全アーキテクチャでのビルドとテストを自動化したい。

まえだこうへい

私の **Debian** ワークフロー

ganttproject を初めて ITP してから止まったまま。

こう改善したい

家庭と仕事に影響されずにパッケージメンテナンスできるようにしていきたい。

小川伸一郎

私の作業環境について

会社では Ubuntu 8.04.1 Desktop をインストールしたデスクトップ PC で、家では Ubuntu 8.10 をインストールした Thinkpad X61 を使って、開発や日々の業務などをこなしています。全然 Debian じゃないのですが、Ruby on Rails なので、Ruby の Version があわないので、Ubuntu 使っています。

GW 中に Thinkpad に Lenny 入れる予定です。会社のサーバ群も、Debian にしたいなど、いろいろ模索中です。

私の Debian ワークフロー

パッケージ化したいソフトウェアを見つけたら、まず自分自身用の野良パッケージを作り、試します。次に大雑把にライセンスを確認し、良さげなら、自分に喝を入れるため ITP します (笑)。それからコードなど、技術的な検討に入ります (ここで挫折したのもいくつかあるのやら...)。さらにコピーライトやライセンスの精査をし、debian/copyright を完成させます。次に私にとってとても難関の (笑) 英語のドキュメントをつけて、pbuilder でビルドします。最後に mentors.debian.net へのアップロードと mentors@org ML、および debian-develop@jp ML にメールを投げてスポンサー探しをします。以上。

こう改善したい

みんなが使っている文字コードや locale を UTF-8 に統一したい。

やまだたくま

私の Debian ワークフロー

- ① DDTSS (ja) で Pending review の項目を順番に選びます。
 - ② doc/(パッケージ名) フォルダを作成し、原文 (英語) と日本語訳のコピーのテキストファイルを作成します。
 - ③ 翻訳ソフトで英日翻訳を実行します。
 - ④ 用語とその日本語訳を確認して、対訳リストを作ります。
 - ⑤ 文章の内容を確認するため、オンラインマニュアルやパッケージ関連ファイルを参照します。
 - ⑥ 使用例や用語 (訳語) の使用頻度を調査して、訳語を選びます。
 - ⑦ 原文を先頭から順番に手動で再翻訳します。
 - ⑧ Debian JP の文書作成 / 翻訳ルールを守っているか確認します。
 - ⑨ debian-doc ML へ査読依頼します。
 - ⑩ 査読完了後に DDTSS (ja) へ登録します。
- 作業は、複数の場所、複数の PC で行なっています。
 - 作業ファイルは、Mercurial で同期管理しています。

私の Debian ワークフロー

私は Debian に貢献しているわけではないで、Debian 開発の開発工程はもっていませんので、普段の作業工程を記述します。

- まず、必要とされている機能を見付けます。見つけ方は、ボスから指示があったり、手作業でやっていたらいついたとき、簡単な作業でも毎日やっていることに気付いたとき等によく見付かります。
- どうやったら、その機能が実現できるかを考えます。個人的にこの段階が一番楽しいです。
- この機能が本当に必要か考えます。
- 必要ならば、本当に実装して良いか考えます。私がいる職場は、放射線が出たり、100kV の高電圧がかかっていたりするので、放射線管理上問題がないか安全上問題がないか検討します。
- 一番楽しい段階が終わったこともあり、本当に私が実装すべきか考えます。
- 私が実装すべきという結論が出た場合、ぶつぶつ文句を言いながら、実装します。
- 時々 Debug します。テストファーストとか、自動化はできてい

あけど

私の Debian ワークフロー

Debian 上で作業することが少ないなと思っています。せいぜい管理しているサーバのファイアウォールルールを手直しするくらいなので、手元のメインマシンが Mac OS X(10.5.6)ということもあり、Debian なデスクトップ環境を殆ど使ってません。Debian のデスクトップ環境は Debian 勉強会の事前課題に使う程度なので (いろんな環境に慣れるという意味で) もっと使う様にするには Debian 勉強で標準的な環境の emacs を使うのがいいかなと思います。

こう改善したい

上記を踏まえて、勉強がてら emacs を使う様にしてみます。

私の Debian ワークフロー

パッケージをメンテナンスしたり、といった Debian への貢献はまったくできていないのですが、Debian 上でスクリプトを書くことは結構あります。

フローにすると、

- ① 仕事とかしてて、こういうのがあったらなあ、と思う
- ② 思いついたことを実現する仕組みを考えてみる
- ③ 実現できそうなら、実際に使用する環境やユーザの範囲を考える
- ④ 具体的に作るもののイメージができれば、エディタを起動する
- ⑤ 適当に書いて、とりあえず動くものを作る
- ⑥ 規模が小さければ、ホワイトボックステストをする
- ⑦ 自分以外に使う人がいれば、とりあえず試してもらう
- ⑧ 動くものを見て、新たに出てきた要望に答える
- ⑨ 飽きたら開発終了

Debian である必然性のないワークフローになりました。この情報は一体何の役に立つんだろう、と自問自答。

こゝに改善したい

私の Debian ワークフロー

Debian のワークフローかどうか分かりませんが、Debian も利用している私の会社でのワークフローを紹介します。

- ① 自動化できそうな機能や Debian 化することでインストールが楽になりそうな機能を見つける - たとえば
 - バッチジョブのログを取りながら経過や結果を監視し、問題があるようならアラームをいろんな方法で投げる
perl script
 - 複数のアーキテクチャや Debian のバージョンに対して内製のパッケージを build する cowbuilder や pbuilder の wrapper
 - 社内の Debian で運用するサーバーのインストーラースクリプト
- ② 利用者になりうる人や他の開発者の意見を取り入れながら、機能をプログラム化する
- ③ まとまった規模になったら Debian 化を行なう
- ④ 適当な区切りで履歴管理システムにタグを打ち、内製の package

- スペイン参加
- 飛行機代 20 万弱
- 7/23-7/30
- 優待つき事前参加登録締切りは 4 月 15 日まで、それ以降の参加登録は 300 euro 以上必要。
- <http://debconf9.debconf.org/register.xhtml>

宴会場所

- 宴会場所
本日の宴会は「はなの舞」です。
21:00 開始です。参加者は1Fに集合して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。