

# 東京エリア Debian 勉強会

第 75 回 2011 年 5 月度

岩松 信洋 iwamatsu@debian.org  
IRC nick: iwamatsu

2011 年 5 月 21 日

# Agenda

- 注意事項
  - 飲酒禁止
  - 宗教禁止
  - 営利活動禁止
- 最近あった Debian 関連のイベント報告
  - 第 75 回 東京エリア Debian 勉強会
  - 第 46 回関西 Debian 勉強会 @OSC2011 Kobe
- Apache2 のモジュールをつくってみた
- Debian on NiftyCloud
- Debian/m68k 開発
- 月刊 PPC64 パーティンク

- 注意事項
  - 飲食禁止
  - 宗教禁止
  - 営利活動禁止
- 最近あった Debian 関連のイベント報告
  - 会長就任挨拶
  - [backports.debian.org](http://backports.debian.org) の話
  - initramfs-tools の話
  - 月刊 PPC64 ポーティング
  - 僕が DD 目指すの手伝ってください



# イベント報告



DWN quiz




# 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. HPPA と alpha の移転先はどこでしょうか？

-  A [buildd.debian.or.jp](http://buildd.debian.or.jp)
-  B [buildd.debian-ports.org](http://buildd.debian-ports.org)
-  C [www.buildd.net](http://www.buildd.net)

# 問題1. HPPA と alpha の移転先はどこでしょうか？

-  A `buildd.debian.or.jp`
-  B `buildd.d`
-  C `www.bu`

答えは:






B

解説：



## 問題2. linux カーネル 2.6.39 が Debian に入ることによって起きる変更は？

-  A i386-bigmem が i386-pae になった
-  B amd64 が i386 になった
-  C i386 は amd64 のマルチバイナリになった

## 問題2. linux カーネル 2.6.39 が Debian に入ることによって起きる変更は？

答えは:

-  A i386-bigmem が i386-pae になった
-  B amd64 7
-  C i386 は a  
ナリにな:






A

解説：






## 問題3. Qt3パッケージが削除されない理由は？

-  A Qt3 ユーザによる哀願のため
-  B LSB 4.1 が Qt3 を必要としているため
-  C 削除の仕方がわからない

# 問題3. Qt3パッケージが削除されない理由は？

答えは:

-  A Qt3 ユーザによる哀願のため
-  B LSB 4.1 ているため
-  C 削除のため



B

解説：



## 問題 4. Debian のサーバに追加された機能は？

-  A ログインしているユーザを IRC に流す機能
-  B RFC1149 の実装
-  C DNSSEC

## 問題 4. Debian のサーバに追加された機能は？

答えは:

-  A ログインしているユーザを IRC に流す機能
-  B RFC114
-  C DNSSEC



C

解説：






prework

Debian 限定だと思いつかない…。(お題の意図を読み間違えているのかも) apt-get を http で実行するとウェブサービスと言える?

Debian 使いとしてウェブサービスに期待すること．最近は少なくなりましたが，IE 必須のサービス等の環境依存のサービスをやめて欲しいです．最近だと Silverlight 必須のサービスで Moonlight で動きそうで動かないといったことがありました．

[http://live6.channel.ne.jp/world\\_ipv6/](http://live6.channel.ne.jp/world_ipv6/)

情報に対する課金がなくなること。



- jslinux という強力なエミュレータも出たので、ブラウザで動く Debian experimental 環境とかブラウザで動く Gnome のお試し環境とかを提供するウェブサービスとか素敵かも。こもきっとウェブサービス!(なんか空気読めてない回答な気もするけど...)
- USB に書き込めば debian 環境がそのままブートできるようなイメージをつくってくれるウェブサービスが良さそうな気も... 例えば、パッケージ一覧にチェック入れて、sid とかにチェック入れると、USB メモリにそのまま書き込めばその仕様で debian sid がブートできるようなカスタムイメージを作ってくれるとか。
- チェックボックスとセレクトタだけで、preceed ファイル生成してくれるウェブサービスもいいかも... 大量のインストール時とかよさそう。(もう言いたい放題ですね...)

- 全世界の Web サーバを提供する OS が Debian になること。
- 分散コンパイルサーバとか欲しい。

Web サービスもできれば機械処理しやすいものが良い。あと、クラウド上での API を提供しているようなサービスに、関数型言語に対するサポートが増えてほしい。

CPU とある deb パッケージを選択すると、その CPU 向けに最大限の最適化したパッケージと依存するパッケージを再ビルドしてくれるサービス。

翻訳をたまにしているので、ディストリビューション間横どおしでの翻訳関連情報を提供するサイトがあればいいなと思うことがあります。  
#課題とは少しズレているかも知れませんが、#個人向けのウェブサービスには食傷気味というところもあるので。

# まえたこうへい


Debian システムで作った環境との相互互換性。例えば、最近 GAE/Python をよく使うので、作ったシステムを GAE/Python じゃ Debian システムのどちらでも (ほとんど変更なしで) 動かせると便利ですね。すぐ始めるのにクラウドサービスを利用して作ったけど将来は Debian で動かしたい、逆に今は政治的な理由で外に出せない Debian システムを将来は自分の管理から外れるので手離れをよくするためにクラウドサービスに簡単に移行できる、など。

そうですね。今の所導入を検討しているのは、パーソナルストレージサービスぐらいですかね。あらゆる所で自分のデータが自分で共有できれば、それで十分な感じです。



Apache2 の  
モジュール  
をつくって  
みた

# Apache2 モジュール入門

- apache httpd で動くモジュール
  - C 言語で実装
  - Debian の流儀
- 

## apxs2: テンプレート作成

```
$ apxs2 -g -n dancergps
$ cd dancergps
$ ls
$ ls
Makefile  mod_dancergps.c  modules.mk
```

# コードを書く

適当にフックを記述



# apxs: インストール

コンパイルしてインストール

```
$ sudo apxs2 -c -i mod_dancerqps.c
```

4 種類方法があります。

- Debian way 1 a2enmod
- Debian way 2 手動で設定
- Apache を適当な httpd.conf で起動
- Apache を自前でインストールしなおす

# 適当な httpd.conf

```
Listen 8080

LockFile /home/test/tmp/apache.1.lock
PidFile /home/test/tmp/apache.1.pid

# log configuration.
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog "/home/test/log/access_log" common
ErrorLog "/home/test/log/error_log"

# Order, Allow.
LoadModule authz_host_module /usr/lib/apache2/modules/mod_authz_host.so
# map from / -> /index.html
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
# .html -> content-type: text/html
LoadModule mime_module /usr/lib/apache2/modules/mod_mime.so
TypesConfig /etc/mime.types

# Document root
DocumentRoot "/home/test/hoge"
<Directory "/home/test/hoge">
    Options Indexes FollowSymLinks

    AllowOverride None

    Order allow,deny
    Allow from all
```

# apache 実行

```
APACHE_RUN_USER=dancer \  
APACHE_RUN_GROUP=dancer \  
/usr/sbin/apache2 -f $(readlink -f ./httpd.conf) -k restart
```

# apachebench 使ってみる

```
$ /usr/sbin/ab -c 100 -n 100 http://localhost:8080/
```

# apache 実行

```
$ /usr/sbin/ab -c 100 -n 100 http://localhost:8080/
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done

Server Software:      Apache/2.2.9
Server Hostname:     localhost
Server Port:         8080

Document Path:       /
Document Length:     44 bytes

Concurrency Level:   100
Time taken for tests: 0.056 seconds
Complete requests:   100
Failed requests:     0
Write errors:        0
Total transferred:   29600 bytes
HTML transferred:    4400 bytes
Requests per second: 1796.17 [#/sec] (mean)
Time per request:    55.674 [ms] (mean)
Time per request:    0.557 [ms] (mean, across all concurrent requests)
Transfer rate:       519.21 [Kbytes/sec] received

Connection Times (ms)
```



Debian on  
NiftyCloud



Debian/m68k  
開発

# m68k とは？

- Motorola  
680x0/m68000/68000 の事。省略して m68k。
- 32bit で CISC。エンディアンはビッグ。
- 今はフリースケール・セミコンダクタによって開発および販売。
- Debian に最初にポーティング (hamm) され、最初に脱落した (etch) アーキテクチャ。

メーカ	ハードウェア
Apple	Macintosh SE
シャープ	X68000
Palm	Palm Pilot
ATARI	Atari Falcon
HP	HP 9000 Series 200
SUN	Sun-1
DEC	VAXstation 100
SGI	RIS 1000
SEGA	メガドライブ
SNK	ネオジオ

# Debian/m68k の現状

- etch から脱落した後、Thorsten Glaser 氏が拾い上げ debian-ports.org 上で開発継続中。
- ハードウェア (ATARI 社の Amiga など) は入手が難しくなっているので主にエミュレータを使っている。
- Debian の bootstrap が行える程度のパッケージはメンテナンスされている。
- ちなみに、Debian に再度取り込むことは目標にしていない。Linux/m68k の開発ベースとして生きるみたい。
- 開発議論は ML (<http://lists.debian.org/debian-68k/>) と IRC ([debian-68k@oftc](mailto:debian-68k@oftc)) で行われている。



# なぜ m68k に手を出してしまったのか

- Ruby1.9.1 パッケージのバグ #611691 ( m68k が FTBFS ) を見つけた。
- Ruby コミッタになったのでなんかできないかなあと。

# 開発環境設定方法

- 実機での開発は行われておらず、エミュレータを使って開発。
- qemu の 68k は不具合が多いので、Debian では ARAnyM という 68k エミュレータを使って開発。

# ARAnyM とは

- ARAnyM は Atari Running on Any Machine の略。
- 68040 + MMU + FPU(68882) を実装したエミュレータ。
- グラフィックス、ディスクドライブ、CDROM、ネットワークのサポート。
- OpenGL を使った高速なグラフィックと 4GB のメモリを扱える。



# ARAnyM のインストール

```
$ sudo apt-get install aranym p7zip
```

# カーネルとユーザランドイメージのダウンロード

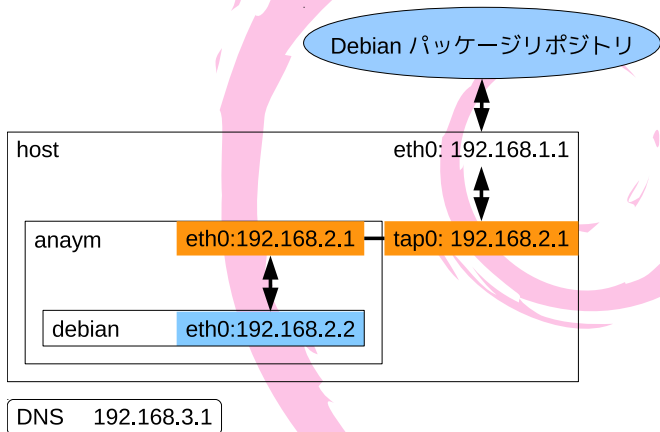
Debian m68k の開発に必要なカーネル、ユーザランドイメージのダウンロードします。

```
$ wget http://debian.nctu.edu.tw/debian-ports/pool-m68k/main \
    /l/linux-2.6/linux-image-2.6.38-2-atari_2.6.38-5_m68k.deb
$ ar -x linux-image-2.6.38-2-atari_2.6.38-5_m68k.deb
$ tar -xzf data.tar.gz
$ ls boot/vmlinuz-2.6.38-2-atari
-rw-r--r-- 1 iwamatsu iwamatsu 1767311 2011-05-12 00:48 boot/vmlinuz-2.6.38-2
```

build-essential がインストールされたイメージが既にある。

```
$ wget http://people.debian.org/~smarenka/aranym/sid/disk.tar.7z
$ 7zr x -so disk.tar.7z | tar xvf -
$ ls -l disk.img
-rw-r--r-- 1 iwamatsu iwamatsu 10737377280 2011-05-18 00:37 disk.img
```

# ネットワーク構成



# uml-utilities パッケージのインストール

ARAnyM では tun を使うので uml-utilities パッケージをインストールする。

```
$ sudo apt-get install uml-utilities
```

## uml-net グループへの追加

tun および ARAnyM を使うユーザを uml-net に追加する。

```
$ sudo gpasswd -a iwamatsu uml-net
```

# ネットワークの設定

ホスト側の ネットワークを以下のように設定する。

```
$ cat /etc/network/interfaces
auto tap0
iface tap0 inet static
address 192.168.2.1
pointopoint 192.168.2.2
netmask 255.255.255.255
tunctl_user iwamatsu
up iptables -t nat -A POSTROUTING -s 192.168.2.2 -j MASQUERADE
down iptables -t nat -D POSTROUTING -s 192.168.2.2 -j MASQUERADE
```

# フォワーディングを有効

フォワーディングを有効にして、tap0 ネットワークデバイスを上げる。

```
$ sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'  
$ sudo ifup tap0
```

# AranyM の設定

```
$ cat aranyM.config
[GLOBAL]
FastRAM = 768 # メモリサイズ。単位は MB。
Floppy =
TOS =
EmuTOS =
AutoGrabMouse = No
GMTIME = Yes

[LILO]
# Linux カーネルイメージ
Kernel = vmlinuz-2.6.38-2-atari
# these Args for normal X operation
# カーネルコマンドライン
Args = root=/dev/hda1 console=tty debug=par

# these Args for headless
#Args = root=/dev/hda1 console=nfcon

# ネットワーク設定
[ETH0]
Type = bridge
Tunnel = tap0
# エミュレータで使う仮想ネットワークデバイスの Mac アドレス
Mac = XX:XX:XX:XX:XX:XX

[STARTUP]
GrabMouse = No
```

# Aranyam の起動

```
$ aranyam-mmu -l -c aranyam.config
```

uname と /proc/cpuinfo:

```
$ uname -a
Linux aranyam 2.6.38-2-atari #1 Mon May 9 16:39:31 UTC
2011 m68k GNU/Linux
$ cat /proc/cpuinfo
CPU:68040
MMU:68040
FPU:68040
Clocking:73.5MHz
BogoMips:49.04
Calibration:245248 loops
```

# ターゲットでの設定

- Debian OS が立ち上がったら、root ユーザでログイン（パスワードは無し）し、ネットワーク設定を行う。
- 起動時に ARAnyM の仮想ネットワークデバイス（nfeth:nat-feature）を eth0 として認識する。
- 認識されている場合には、ARAnyM で設定した MAC アドレスが eth0 が認識されている。

```
# dmesg | grep eth0  
eth0: nfeth addr:192.168.0.1 (192.168.0.2) HWaddr:XX:XX:XX:XX:XX:XX
```

- もしホスト側の設定が間違っている場合、eth0 が存在しない状態になる。このような場合には、ホスト側の設定を見直す。

eth0 が認識されているのなら、`/etc/network/interfaces` と `/etc/resolv.conf` を以下のように変更する。

```
# cat /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.2.2
netmask 255.255.255.0
gateway 192.168.2.1
# cat /etc/resolv.conf
nameserver 192.168.3.1
```

# ネットワークのチェックと確認

```
# ifup lo
# ifup eth0
# ping 192.168.2.1 # gateway へのチェック
# ping 192.168.3.1 # DNS へのチェック
# apt-get update # apt-get update
# apt-get install debian-ports-archive-keyring
# apt-get update
# apt-get dist-upgrade
```

## その他開発環境

エミュレータを使って開発できるのはすごく良いことなのですが、エミュレータだけでは遅いのでクロスツールチェーンが欲しくなります。Debian でのクロス toolchain は emdebian プロジェクトが提供していますが、m68k のものは提供されていません。しかし、amd64 バイナリは Thorsten Glaser 氏が以下の apt-line で提供しています。

```
deb http://www.freewrt.org/~tg/debs68k/ cross main
```

# ARAnyM 上での開発

- 動作しているのが エミュレータ上というだけで通常の開発と変わらない。
- cowbuilder も使えるので、遅いという以外には問題はないだろう。
- 開発速度を上げたい場合には、distcc/icecc/ccache など使うとよい（このあたりの話はまた今度）。

# Ruby の FTBFS バグはどうなったのか？

Debian/m68k の開発環境は構築できましたが、Ruby のバグはどうなったのかというと、  
<http://redmine.ruby-lang.org/issues/4745> としてバグレポートし、r31646 でコミットしておきました。



月刊 PPC64  
ポータイン  
グ

# 今後のイベント

- 5月 第47回関西 Debian 勉強会 (5月22日)
- 6月 OSC2011 Hokkaido 出張勉強会 (6月11日), 第77回東京エリア Debian 勉強会 (6月18日)
- 7月 Debian 勉強会 & Debconf11 in ボスニア

未実施ネタ : PS Move ネタ? デジタル放送取り込み? Debian Pod cast? 100台 Squeeze アップグレード (吐血) 体験記?

# 今日の宴会場所

新宿のどこか。

