



東京エリア Debian 勉強会

Debian 勉強会幹事 上川 純一
2007 年 5 月 19 日

1 Introduction

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？

目的として次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について情報共有する場をつくる
- Debian を利用する際の情報をまとめて、ある程度の塊として整理するための場をつくる

Debian の勉強会といふことで究極的には参加者全員が Debian Package をがりがりと作るスーパーハッカーになった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、といふのが目的です。

目次

1	Introduction	1
2	事前課題	3
2.1	kinneko さん	3
2.2	前田 耕平さん	3
2.3	出井さん	4
2.4	濱野さん	4
2.5	鈴木崇文さん	4
2.6	小室 文さん	5
2.7	yamashita at cclx.com さん	5
2.8	山本浩之さん	6
2.9	Noriaki Sato さん	6
2.10	yamabe さん	7
2.11	Honjo Hironori さん	7
2.12	キタハラさん	8
2.13	鈴木さん	8
2.14	上川	8
3	Debian Weekly News trivia quiz	9
3.1	2007 年 5 号	9
4	最近の Debian 関連のミーティング報告	10
4.1	東京エリア Debian 勉強会 27 回目報告	10
5	サーバをエッチにしてみました	12
5.1	intro	12
5.2	やること～リリースノートを読んでみる～	12
5.3	やってみた	12
5.4	怒られた!	13
5.5	ついで: エッチを stable にした	14
5.6	まとめ	15
6	最近 pbuilder ってどうよ?	16
6.1	pbuilder の利用コンセプト	16
6.2	pbuilder 自身の開発の仕組み	17
6.3	派生物とその状況	18
6.4	さらなる開発のアイデア	18
6.5	References	19
7	Debian on SuperH	20
7.1	SuperH とは	20
7.2	歴史	20
7.3	現状	22
7.4	今後の課題	25
7.5	リンク	25
8	Debian の情報フロー ²	26



2 事前課題

上川 純一

今回の事前課題は「エッチになって困った事」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。

2.1 kinneko さん

Debian 勉強会の宿題なので、ちと考えてみる。

なんだろう。実はあまり使っていないので、困っていなかったりする。sources.list はだいぶ前から sarge に書き換える習慣だし。

- VMware のバージョンの古いのを使っているので、動かなくなりそうで移行できていないこと。
- 移行ノウハウがまだ十分に出揃っていない感じなので、さらでインストールする以外はまだ怖いな。
- のときには、d-i 試したり、GLANTANK でインストールしてみたり、ARM(XScale) での自動ビルドをやってみたり、LiveCD の MAKAI のベース変更テストしてみたりしたけど、リリースされてからのほうがまったくさわっていないかも。疲れたというか、飽きたというか。
- パッケージ配布サイトの正当性を確認するようになったけど、キーの取り込み手順をもっと自動化してほしいかも。ついつい、ぶつぶつ言われても放置しちゃう。
- SH アーキテクチャでは、Etch ベースで最新は海老原さんところと、岩松さんところかな。どっちも、どういう方向に進むのか外からあまり見えないところが難。それと、早くから進めていた kogiidena さんが低調なのが気がかりです。
- ARM では、やっぱり DMA まわりが遅いままなので、Debian ではちょっと使い物にならないです。EABI への対応も、はじまったばかりですし、ちょっとどうなることやら。
- udev や initramfs に移行しきれていないので、勉強しなきゃなとか。
- Etch ってどんなキャラだったか思い浮かばないこと。あ、Etch A Sketch か。絵は自分で書けてことね。それも UI は結構不自由な感じで。
- 逆に apt-build が Sarge ではうまく動かないのです。SH や ARM でしか試してないですけど。Etch では快調です。

2.2 前田 耕平さん

1. mod_security がなくなったこと。sarge だと、1.8.7 ですが、開発元の最新版は 2.1 で、設定ファイルも少し変わっているので、自分で Etch 用にバイナリ作っても設定をそのままは使えません。
2. Webmin と Usermin もパッケージから外れてました…。社内では使っているサーバでは、他の Linux を使えない人へ一部作業を移管するために Webmin, Usermin を導入してたのですけどね。

余談。逆に良かったのは、APT が NTLM 認証に対応したこと。これで、NTLM 認証している Proxy サーバ経由でアップデートできるようになりました。

2.3 出井さん

Debian の入門者で、今回初参加です。宜しく申し上げます。日経 Linux6 月号付属のネットワーク・インストール用 CD, ISO イメージを CD に焼き、Dynabook Tecra8000, PCMCIA(Corega CG-LAPCCTXD) にインストールしようとしたのですが、PCMCIA LAN Card を認識してくれません。ドライバーは pcnet_cs のようですがうまくゆきません。どなたか御教示戴けませんでしょうか。

2.4 濱野さん

サーバー用途で使用していた debian を etch にした際には特に困ったことは無かったように記憶しているのですが、デスクトップ用途で使用している debian を etch にした際にいくつか困ったことがあったので挙げさせていただきます。

xlock が無い etch には xlock, xlockmore が見あたらなかったのが会社などで離席が出来なくて困っています(ウソ、sid から持ってきました)

X.Org で戸惑いました 突然プレゼンを行う機会があり、とっさにマルチディスプレイで出力できず困りました。

udev の仕組みを理解出来なくて困った

2.5 鈴木崇文さん

課題についてですが、まず今回の課題の「エッチになって困った事」について書くため、サーバを Etch にしなければいけないという「困った事」が発生しました。とはいえ、このような事態は意外とみなさんの身にも発生しているのでは、と思いつつ実際に先程私が体験した「困った事」を以下に記述していきたいと思います。課題が出て初めて Etch にしようと思ったため、アップグレード作業時の話がメインになります。アップグレード後の使用感等についてはそこまで触ってないので、記述できないことを御了承ください。

まず始めにこちらの環境と実施した手順について記述し、その過程で発生した困ったことについて書いていきたいと思えます。

2.5.1 環境

(ここから Etch にアップグレードしました。)

CPU Pen4 1.6Ghz

Mem 768MB

OS debian sarge

主な用途 web server (勉強用)

主なサービス Apache (php とか動かしてます)

2.5.2 手順

Debian GNU/Linux 4.0 ("etch") リリースノート (Intel x86 用) 第 4 章 - 以前のリリースからアップグレードする <http://www.debian.org/releases/stable/i386/release-notes/ch-upgrading.ja.html> に従い作業を進めました。

2.5.3 発生した困ったこと

1. non-US が無くなり、apt-line が変わってしまったとこのことで、しばらく apt-line の書き方に戸惑いました。最終的には、non-US の部分だけを除外することで問題ありませんでした。この件に関しては、Etch のせいで困ったというより、apt に関する知識不足の自分に困ったという感じでした。

2. 実は今回は回避していますが、以前デフォルトで apt-line に「stable」と書いてあるため、自分の知らないうちに次期バージョンに「apt upgrade」してしまっていたことがあります。今回は「sarge」や「etch」と指定してアップグレードしています。(みなさんは通常どうされていますか? デフォルトの設定で特に困ることはないのでしょうか?)

結果的に上記 2 点以外、アップグレード作業自体においてほとんど困ることは発生しませんでした。全作業 ssh 経由で完了でき、web サーバも問題無く動作してしまいました。自分に関していえば、apt やパッケージに関する理解が足りないために困った事態になったということです。

2.6 小室 文さん

- 会社で上司とサーバーの話をする時。同じオフィスの総務・経理には啓蒙活動をして誤解を解く必要あり
例:「このサーバ、エッチにしといてくれない?」
- Lenny をまだよく理解していない事
- サーバを upgrade しないといけない(=休日出勤?) ので仕事が増えた
- 最近淘汰されましたが Mailbox が debian-usersML で溢れている事

上記以外はエッチをまだまだ使いこなしていないのか、前から使っていたからかそんなに不便な事はありません。

2.7 yamashita at ccclx.com さん

sarge からのアップグレードを行いました。

出力

```
# sources.list generated by apt-spy v3.1
deb http://www.ring.gr.jp/archives/linux/debian/debian/ stable main
deb-src http://www.ring.gr.jp/archives/linux/debian/debian/ stable main
deb http://security.debian.org/ stable/updates main

# aptitude update
# aptitude dist-upgrade
```

カーネルも最新版にアップグレード

出力

```
# aptitude install linux-image-2.6-686
# uname -a
# Linux debian 2.6.18-4-686 #1 SMP Wed May 9 23:03:12 UTC 2007 i686
GNU/Linux
```

問題は、その後パッケージのアップグレードが正常に行えない事です。

出力

```
# aptitude -f dist-upgrade
```

```

linux-image-2.6.18-4-686 (2.6.18.dfsg.1-12etch2) を設定しています ...
Running depmod.
Finding valid ramdisk creators.
Using mkinitramfs-kpkg to build the ramdisk.
initrd.img(/boot/initrd.img-2.6.18-4-686
) points to /boot/initrd.img-2.6.18-4-686
(/boot/initrd.img-2.6.18-4-686) -- doing nothing at
/var/lib/dpkg/info/linux-image-2.6.18-4-686.postinst line 583.
vmlinuz(/boot/vmlinuz-2.6.18-4-686
) points to /boot/vmlinuz-2.6.18-4-686
(/boot/vmlinuz-2.6.18-4-686) -- doing nothing at
/var/lib/dpkg/info/linux-image-2.6.18-4-686.postinst line 583.
The provided postinst hook script [/sbin/update-grub] could not be run.
dpkg: linux-image-2.6.18-4-686 の処理中にエラーが発生しました (--configure):
サブプロセス post-installation script はエラー終了ステータス 2 を返しました
dpkg: 依存関係の問題により linux-image-2.6-686 の設定ができません:
linux-image-2.6-686 は以下に依存 (depends) します:
linux-image-2.6.18-4-686 ... しかし:
パッケージ linux-image-2.6.18-4-686 はまだ設定されていません。
dpkg: linux-image-2.6-686 の処理中にエラーが発生しました (--configure):
依存関係の問題 - 設定を見送ります
mzscheme (352-6) を設定しています ...
Cataloging SLIB routines...
reference to undefined identifier: slib:features
dpkg: mzscheme の処理中にエラーが発生しました (--configure):
サブプロセス post-installation script はエラー終了ステータス 1 を返しました
以下のパッケージの処理中にエラーが発生しました:
linux-image-2.6.18-4-686
linux-image-2.6-686
mzscheme
E: Sub-process /usr/bin/dpkg returned an error code (1)
パッケージをインストールできませんでした。復旧を試みています:

```

今のところ自力で解決できませんが、普通にデスクトップ環境で使う分には問題ないようです。

2.8 山本浩之さん

エッチになって困った事

基本的に「困った事」より「良くなった事」のほうが圧倒的に多いと思いますが、強いて言うなら、各パッケージのリソース使用量が微妙に増えたことが挙げられます。

ハードディスク使用量も少し増えましたが、特にメモリ使用量関係には顕著に現れていると思います。一番実感したのが、iceweasel の体感速度の低下です。sarge の頃から kde 上で mozilla-firefox パッケージを使用してましたが、CPU : powerpc 300 MHz、メモリ : 192 MB という非力なマシンでも実用レベルの速度で動いていました。しかし etch になって、kde 3.5.5 + iceweasel 2.0.0.3 という構成ですと swap を使ってしまう、ほとんどフリーズに近いと言って過言では無いような状態となっております。kde のような統合デスクトップ環境ではなく、もっとメモリ使用量の少ないウィンドウマネージャも検討しましたが、iceweasel 自体のリソース使用量が圧倒的に多いため、仕方なく、そのマシンでは特別に理由がない限り、iceweasel は使わなくしています。

2.9 Noriaki Sato さん

今週の月曜と火曜の夜に時間を使って、自宅の sarge のサーバを etch に upgrade しました。まず月曜日、リリースノートに従って sarge のままで kernel を 2.6 に upgrade したのですが、kacpid が暴走するという現象が起きて、ここでかなり時間を浪費しました。これは結局 boot parameter で acpi=off にして凌ぎました。翌日、引き続きリリースノート通りに作業を進めて upgrade は無事、終了しました。apache, postfix など問題なく動いているようで、今の所 etch になって困った事は特にありません。以下、余談。実は今回リリースノートを読むまで devfs/udev の事を全く知りませんでした。リリースノート 4.7.1 に、再起動前の作業として devfs からのコンバートを手作業でやるように書いてありますが、何をしたら良いのか分からなくて、結局 (devfs は使っていないので) 何もなくて良いという事が水曜日に確信出来るまで、再起動を保留にしていました。余談の余談ですが、devfs/udev についてぐぐっていたら、@IT の Linux Kernel Watch の記事がヒットして、上川さんがこんな所に連載を持ってると言う事を初めて知ってました。

2.10 yamabe さん

2.10.1 前置き

sarge をサーバ用途 (DNS,DHCP,MAIL,Samba) で使っています。離れた部屋に置いてあるので、手元の WindowsXP から TeratermSSH を使ってログインしてログのチェックやセキュリティアップデートを行っている。

2.10.2 解決した?事

etch がリリースされてから、手動で時々、「# apt-get update; apt-get upgrade」と確認するたびに、アップデートされているパッケージ数が大きく増えているのにも関わらず、何もアップグレードされないという不思議な現象に悩んでいました。原因は、/etc/apt/sources.list にありました。入手先が”stable”になっていたのです。 ”stable”を”sarge”に置き換えて、現在も sarge を使いつづけています。

2.10.3 困った事 1

今の悩み：困っていることは、

- etch に切り替えるべきなのか。
- いつ、etch に切り替えるべきか。
- 安全に etch に切り替えたいが、何を注意すべきか。まったく注意不要か。

2.10.4 困った事 2

今、運用中の sarge は、サーバとして不要な deb パッケージが入っています。etch 移行前に、なるべく不要なものを削除整理しようと考えていますが、効率よく不要なものをチェックアウトする方法はあるだろうか。

- 常時必要なサーバプログラム関係以外はバツサリ削除したい。 (X (GUI) 関係もバツサリ削除するつもり)
- サーバとしての最小パッケージ構成のシステムにしたい。

2.10.5 少し困った事

通常は X を使っていないが、X を使いたい場合には、tightvncserver 経由：

出力

```
$ startx -- /usr/bin/Xvnc -geometry 1024x768 -depth 16
```

で、WindowsXP 上の tightvncserver を使っている。現在、日本語入力が行っていない。

- 上記方法でも日本語入力は可能だろうか。
- システムの文字コードは UTF にすべきだろうか。

以上 のんびり困っています。

2.11 Honjo Hironori さん

Etch で困ったというわけではありませんが、Sarge から移行する際に引っかかった点を列挙します。

インストール後の再起動に失敗する 再起動中に Begin: Waiting for root file system... の状態で止まることがあった。2 台のマシンで遭遇。

snmpd が localhost 以外からのリクエストを受け付けない /etc/default/snmpd でオプションに 127.0.0.1 が追加された模様。

apache で文字化け SJIS や euc-jp が文字化けしてしまう。/etc/apache2/conf.d/charset に AddDefaultCharset UTF-8 と設定されていた。
/var/lock/apache2/の権限が root WebDAV で書き込みが出来ない。
emacs で utf8 mule-ucs を入れると起動が重い。
trac を Sarge から移行するのが大変だった 大変です。

2.12 キタハラさん

実は、Debian をメインで使用しているマシンは、まだ Sarge だったりします。 したがって、「困る」以前の状況です。(笑)

しかし、ぷち Etch 体験を会社の Windows2000 上の coLinux でしています。(「apt-get update」しようとして、「apt-get upgrade」してしまった！)

それで困ったことですが、最初に漢字コードが変更になった影響で文字化けした事 (LANG 設定を修正)、同じ理由で適当に作成したスクリプトが挙動不審になった事 (真面目に修正する必要あり? 今は Sarge に戻している) くらいでしょうか?

ただまだ本格的に使っていないので、これから一杯困った事が出てくる気がします。

2.13 鈴木さん

apt-setup がなかったことかな。隣の人が sarge から etch に upgrade した後、apt line を変更したいと言うので、apt-setup で選べばと言ったがなかった。不具合のようなので待っていればいいかなと。SUSE はどこを選べばいいか Web のサイトを見てもよくわからないので、事情により変更したいとき等は便利だと思う。

2.14 上川

Sid 常用しています。気づいたら Lenny 相当になっていました。仮想マシン内部でしか、etch をつかってません。

3 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。

3.1 2007 年 5 号

<http://www.debian.org/News/weekly/2007/05/> にある 4 月 24 日版です。

問題 1. 3 月 12 日 Alioth で新規に使えるようになったバージョンコントロールシステムはどれか

- A Mercurial
- B RCS
- C git

問題 2. Robert Milan が goodbye-microsoft 0.4.0 の機能として発表したのは何か

- A Ubuntu 対応
- B etch 対応
- C Windows Vista 対応

問題 3. Aurelien Jarno が kFreeBSD の新しいインストール CD を発表したけど、対象アーキテクチャは何か

- A i386
- B i386 amd64
- C ppc hppa arm

問題 4. teTeX と TeXLive で何がおきたか？

- A TeXLive はもう古いので teTeX でおきかえる
- B teTeX はもう古いので TeXLive でおきかえる
- C TeX のコンセプトが古いのもう両方ともやめる

問題 5. Debian etch の CD/DVD イメージは何枚分あるか

- A 666 枚の CD と 13 枚の DVD
- B 292 枚の CD と 39 枚の DVD
- C 1 枚の DVD に全部おさまる

4 最近の Debian 関連のミーティング報告

上川 純一

4.1 東京エリア Debian 勉強会 27 回目報告

東京エリア Debian 勉強会参加報告。4 月の第 27 回東京エリア Debian 勉強会を実施しました。

今回の参加者はやまねさん、青木さん、小室さん、溝口さん、あけどさん、小林さん、David Smith さん、脇さん、でんさん、鈴木邦男さん、北原さん、noriaki sato さん、橋本さん、本庄さん、中原健吾さん、森田尚さん、Charles Plessy さん、えとーさん、上川の 19 人でした。

最近のイベントの紹介として、最初に前回の報告を行いました。仮想化友の会との共催です。また、最近のイベントとして etch のリリースがあり、etch のリリース宴会の開催の報告を行いました。

次に事前課題の紹介を行いました。各種 SCM をすでに活用している人が多いのがわかりました。subversion や cvs や VSS が主流で、分散 SCM はあまり利用されていないようでした。そもそも分散バージョン管理のツールを企業内の協同作業者に浸透させるのに悩んでいることや、部署が VSS しかつかわせてくれないとか、VSS すらつかわせてくれないので tar で管理しているなどの悩みがでてきました。また、タスクトラッキングとして、trac の話題が出てきました。バージョンの管理するだけでなくそういうシステムを活用して総合的に運用することの悩みなどをみなさま感じているようです。

DWN クイズを今回も実施しました。全員に起立してもらい、グー・チョキ・パーで選択してもらいました。1 問目から多くの方が間違っしまい、熾烈なたたかいになりました。5 問全問正解したのは、David Smith さんただ一人でした。おめでとうございます。あけどさんから mozilla.party グッズが贈呈されました。あけどさん、ありがとうございます。

「時間がなくて準備が全くできていないのですが」、とのセリフとともに小林さんが quilt の紹介をはじめました。1 月に実施する予定だったものが諸事情で 4 月まで延期になっていたものです。

内容は、コマンドラインで quilt の各種命令を実際に入力してデモをするというものでした。説明しきれなかった部分はホワイトボードに図を書いて説明しました。

dpatch との差分として、quilt には「現在のパッチ」(top) という概念があります。quilt push / pop で編集するパッチを選べます。また、dpatch は debian/patches/00list を利用者が編集することが前提ですが、新規のパッチを追加する quilt new は debian/patches/series ファイルを変更してくれるということもわかりました。dpatch と違い、編集対象のファイルは quilt add で明示的に宣言してから編集する必要がある、もしくは quilt edit コマンドでエディターを起動して編集する必要があることもわかりました。dpatch-edit-patch ではシェルが起動してそれを終了したらパッチに反映されるのですが、quilt では、quilt refresh でパッチに反映するそうです。また quilt header でパッチのヘッダ情報を編集する、というような詳細なデモが約 1 時間みっちり続きました。

個人的には、パッチのコンフリクトが起きる場合のワークフローが気になっていたのですが、quilt push -a でフェールしたパッチを quilt push -f で強制的にプッシュし、rejects を確認して、編集し、quilt refresh することでパッチに反映されるということをデモして説明してくれました。

プレゼンテーションの手法について、会場から 2 つコメントをいただきました。全体の流れがわからないので、アジェンダみたいなのがあったとよかったですね。またこのツールが何をやるものか、そもそもどういう課題があって、

それをどう解決してくれるものなのか、ということの説明するとわかりやすかったかもしれないですね、というものです。今後ツールの紹介をする際にそれを念頭においているとよいかもしれません。

「darcs がパッチをマージしようとする際に無限ループ状態におちいつてしまって資料が完成しませんでした」、という言い訳を最初にしてから、デモをしたのは最近 darcs に我慢ならなくなり Mercurial に移行してしまった David Smith さんです。darcs の紹介です。emacs でバッファを二つ開いて、上半分にアジェンダの書いてあるテキストファイルを表示し、下半分を shell-mode で動かしていました。なかなかストイックな感じですが、ありかもしれません。

プレゼンテーション手法もさることながら、発表の題材である darcs がストイックな思想をもったツールだということがひしひしと伝わってきました。どうも目を疑ったのですが、説明どおりにパッチ管理システムで、リビジョン管理システムではないようなのです。darcs も init や add コマンドがあり、whatsnew コマンドで確認する、などというフローになりますが、コミットの管理のしかたの発想が cvs / svn / git などの他のツールよりもむしろ quilt / dpatch などのツールのほうに近いことがわかりました。

darcs record でパッチを記録することができるのですが、darcs unrecord で任意のパッチをはずすことができます。依存されていないパッチをはずすことができます。つまり、darcs で管理されている最新版は、最新版のツリーとしてではなく、darcs の管理している独立したパッチを全部適用した状態、として存在しているのです。

おかげさまでマージ処理のためのパッチ代数を処理するために CPU 時間が使われ過ぎるという弊害があるそうです。

darcs はパッチの送信や受信のためのプロトコルやコマンドが組み込まれた quilt のような印象を受けました。

最後のネタとして上川が git-buildpackage の紹介を行いました。git の簡単な使い方紹介と、git-buildpackage の考え方の紹介するとどめ、具体的な操作方法についてはあまり説明していません。

git について出た質問として、どれくらいのディスク容量を使っているのか、というものがありました。仕組みとしては、ファイルをそのまま圧縮して保存しつつける仕組みなのですが、pack 方式というものがあり、効率よく保存するようになっています (git-repack / git-gc コマンドなどで実施)。既存の事例を調査すると、Debian 勉強会や Linux Kernel などのツリーでは、ワーキングツリーとレポジトリが同じくらいの大きさであることがわかりました。両方とも 2 年くらい運用しているツリーなので、そこまで大きくないですね、という結論になりました。

今回は宴会は「遊喜酒場 ~ 想作串 ~」にて開催しました。おいしかったです。ただ、18 人で二階を満員占拠したのですが、料理の単品の量がすくなかったのと出てくるペースが遅かったのは問題でした。この店にこの人数で入るのは無理があったかもしれません。

trac の話題で盛り上がりました。trac-ja-resource がうまく動かないということと、etch の trac にアップグレードする際には、sqlite から sqlite3 にデータベースを移行する必要があると手動操作が必要なのだそうです。今回は SCM の話題だったので、trac ユーザが多くいたようですね。

次回のネタとして、エッチにアップグレードして苦労した話を交換できればよいですね、というところで 4 月は終了でした。

5 サーバをエッチにしてみました

小室 文

5.1 intro

エッチがリリースされてから早一ヵ月たちました。debian-users ml では助けを求める亡者の投稿で ML はかつて無い程の盛り上がりを見せていました (います? 現在進行形?). woody から sarge へのアップグレード時の盛り上がりを知らないのであれですが、初めてのリリースにちょっとした祭を感じました。と言う訳でちょうど GW だったので、サーバ 2 台を sarge からエッチにアップグレードしてみました。

5.2 やること ~ リリースノートを読んでみる ~

GW 前後に日本語のリリースノートが出たので、読んでみました。

Debian GNU/Linux 4.0 - リリースノート

<http://www.debian.org/releases/stable/releasenotes>

5.2.1 リリースノートのポイント

1. データや設定情報のバックアップを取り、パッケージ状態のチェック (hold 状態の物は解決をしておく)
2. sarge で最新状態にする
3. エッチで update と upgrade
4. いろいろパッケージをいれる
5. dist-upgrade
6. 最終チェック・再起動・動作確認

5.2.2 エッチの事 (個人的に)

- 全体のパッケージ数 18200 ~ / エッチでの新しいパッケージ 6500 ~ / sarge から約 65 % のパッケージが更新された
- default encoding が UTF-8 に
- installer が結構変わったらしい
- debian-volatile 公式リリース
- Exim4.5 から Exim4.63 に / Apache2 から Apache2.2 に / PHP5 / Bind9 から Bind9.3 に

5.3 やってみた

1. バックアップを取る

/etc/以下、/var/lib/dpkg の中身、dpkg --get-selections '*' の出力、/var/backups など。後あまり影響はないが/home/以下の隠れファイルとかも X など使用している場合はバックアップ対象としていたほうがよい

2. パッケージ状態のチェック & セッションの記録
3. (sarge での) パッケージの更新
`/etc/apt/source.list` は sarge のままで `aptitude update`
 更新候補: `libapache-mod-php4 libapache2-mod-php4 libc6 libkrb53 libmagic1 locales man-db php4 php4-comm`
4. (エッチでの) パッケージ情報の更新/`/etc/apt/source.list` をエッチ (stable) にして
`aptitude update`
`aptitude upgrade`
 削除されたパッケージ: `libruby1.6 ruby1.6`
5. パッケージの確認・インストール
 - (a) `aptitude install initrd-tools` (すでに入っている場合もある)
 新規に入ったパッケージ: `libdevmapper1.02 libselinux1 libsepol1 tzdata`
 削除されたパッケージ: `base-config`
 - (b) デスクトップ環境があれば
`aptitude install libfam0 xlibmesa-glu x11-common`(入っていたら)
 新規に入るパッケージ: `libfontenc1 libfs6 libx11-data libxau6 libxdmcp6 libxfont1 xbitmaps xcursor-themes xfonts-encodings xfonts-utils xutils-dev`
 削除されたパッケージ: `xfree86-common`
 - (c) カーネルのアップグレード
`aptitude install linux-image-2.6-` (すでに 2.6 系であれば今すぐに更新をする必要は無い)
6. `aptitude dist-upgrade`
 アップグレード中に (主に Exim4 と Apache2 設定ファイル) 各種変更部分の選択を迫られる
 使われてないから削除されたパッケージ: `libfam0c102 libreadline4 ntp ntp-simple`
 新規インストールされたパッケージ: `apache2.2-common courier-authlib courier-authlib-userdb cpp-4.1 debian-archive-keyring dmidecode gnupg gpgv laptop-detect libapr1 libaprutil1 libbind9-0 libdb4.3 libdb4.4 libdns22 libedit2 libfribidi0 libgnutls13 libisc11 libiscfg1 libltdl3 liblwres9 libncursesw5 libnewt0.52 libpcap0.8 libpci2 libpq4 libreadline5 libsigc++-2.0-0c2a libslang2 libsp1c2 libsqlite3-0 libssl0.9.8 libtasn1-3 mktemp modconf openbsd-inetd openssl-client openssl-server readline-common sysvinit-utils tasksel-data update-inetd`
 削除されたパッケージ: `libnewt0.51 libsp1 netkit-inetd ntp-server apache2-common exim4-daemon-heavy`
7. 確認・再起動
 - `/proc/mounts` に 'devfs' があれば devfs スタイルのデバイス名の変更をする。
 - lilo をブートローダーとして使用している場合、もう一度実行させておく
`/etc/kernel-img.conf` の内容を調べ、`do_bootloader = Yes` と書かれていることを確認
 - grub の場合、`/etc/kernel-img.conf` を編集、`update-grub=/sbin/update-grub` を `/usr/sbin/update-grub` に変更する

5.4 怒られた!

以下のパッケージの処理中にエラーが発生しました:

`exim4-config postgresql-7.4 postgresql-contrib-7.4 tcsh-kanji postgresql exim4-base postgresql-contrib exim4-daemon-heavy mailagent at exim4 qpopper amavisd-new mutt spamassassin`

怒られた原因と解決方法

- debconf: unable to initialize frontend: Gnome
dpkg-reconfigure debconf にして設定を Dialog に変更。なんで gnome を選んだのかは自分でも不明
- tcsh conflicts with tcsh-kanji
tcsh-kanji: Depends: tcsh (≥ 6.14.00-6) but it is not installable
一番分からずはまった。tcsh-kanji depends on tcsh だと思い込んだのとちゃんとエラーを読んでいなかったのが原因。
aptitude show tcsh
パッケージ: tcsh
バージョン: 6.14.00-7
依存: libc6 (≥ 2.3.6-6), libncurses5 (≥ 5.4-5)
競合: **tcsh-kanji (< 6.14.00-6)**
置換: tcsh-kanji (< 6.14.00-6)
提供: c-shell, tcsh-kanji
上記を見る限り tcsh と tcsh-kanji が conflicts しているようなので、tcsh-kanji を remove/purge する
- /etc/exim4/update-exim4.conf.conf: line 32: acl __check __helo:: command not found
昔追加した文を削除して update-config.conf をして/etc/init.d/exim4 reload をする (今は使ってないはず)
- Errors were encountered while processing: exim4-config
dpkg -l exim4-config では Failed-config になっているので dpkg --configure exim4-config をする
- exim4-daemon-heavy が exim4-daemon-light に replace
exim4-daemon-heavy が入っているサーバをアップグレードしたら heavy から light に置き換わった。なぜ。
以下の新しいパッケージがインストールされます:
apache2.2-common courier-authlib courier-authlib-userdb cpp-4.1 debian-archive-keyring dmidecode
exim4-daemon-light gnupg gpgv laptop-detect libapr1 libaprutil1 libbind9-0 libdb4.3 libdb4.4
libdns22 libedit2 libfribidi0 libgnutls13 libisc11 libiscfg1 libltdl3 liblwres9 libncursesw5 libnewt0.52
libpcap0.8 libpci2 libpq4 libreadline5 libsigc++-2.0-0c2a libslang2 libsp1c2 libsqlite3-0 libssl0.9.8
libtasn1-3 mktemp modconf openbsd-inetd openssh-client openssh-server readline-common sysvinit-utils
tasksel-data update-inetd
以下のパッケージが削除されます:
apache2-common **exim4-daemon-heavy** libnewt0.51 libsp1 netkit-inetd ntp-server

5.5 ついで：エッチを stable にした

もともと testing の時から etch にしていた PC の/etc/apt/source.list を testing から stable に直して aptitude update したらこんなエラーが。

出力

```
パッケージリストを読み込んでいます... 完了
W: GPG error: http://security.debian.org stable/updates Release:
公開鍵を利用できないため、以下の署名は検証できませんでした:
NO PUBKEY A70DAF536070D3A1
W: GPG error: http://cdn.debian.or.jp stable Release:
公開鍵を利用できないため、以下の署名は検証できませんでした:
NO PUBKEY A70DAF536070D3A1 NO PUBKEY B5D0C804ADB11277
W: これらの問題を解決するためには apt-get update を実行する必要があるかもしれません
```

なので

出力

```
gpg --keyserver wwwkeys.eu.gpg.net --recv-keys A70DAF536070D3A1
gpg --keyserver wwwkeys.eu.gpg.net --recv-keys B5D0C804ADB11277
gpg --armor --export A70DAF536070D3A1 | apt-key add -
gpg --armor --export B5D0C804ADB11277 | apt-key add -
```

とするとエラーが出なくなった。`/etc/apt/source.list` で指定している名前が違う = パッケージも違う?のだろうか。

5.6 まとめ

エッチにアップグレードはそんなに難しくない、という事で。リリースノートと `aptitude show` パッケージ名があれば大方のトラブルには対応出来るかと思えます。



6 最近 pbuilder ってどうよ?

上川

この文書は pbuilder とは何か、そして、最近は何がおきたのか、そしてこれから近い将来になにがおきることが予測されるのかということを紹介する記事です。

Debconf7 で紹介する予定の内容です。

6.1 pbuilder の利用コンセプト

pbuilder は chroot 内部で利用するベースファイルシステムイメージを管理し、ビルドのたびに新しいベースファイルシステムイメージを展開することを通して、クリーンルーム環境で Debian パッケージの試験をするのを簡便にします。

基本操作のためのコマンドがいくつかあります。pbuilder create、pbuilder update、pbuilder build^{*1}命令がよく利用される例です。詳細な情報が必要であれば、pbuilder のマニュアルを参照してください。/usr/share/doc/pbuilder/pbuilder-doc.html にあります。

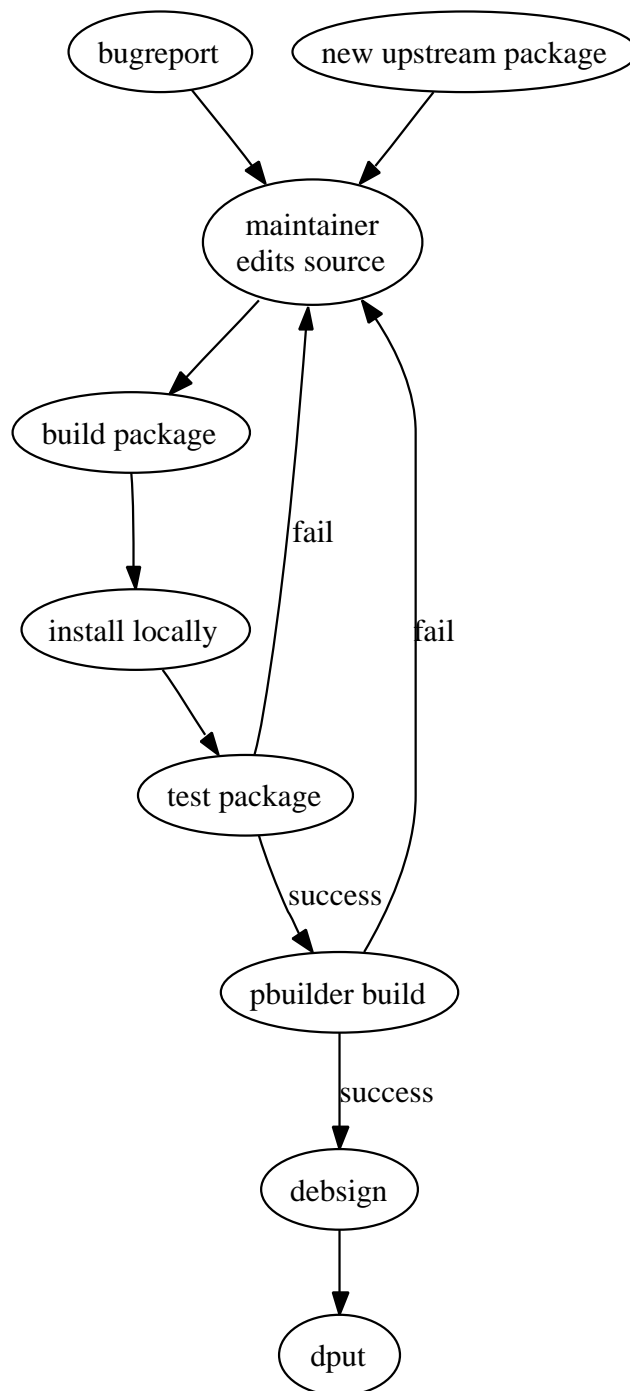
設定が適切に行われ初期化が終了していれば、pbuilder build コマンドは .dsc ファイル (Debian のソースパッケージ) をあたえられると chroot 内部でパッケージをビルドします。

表 1 コマンドの意味

操作	操作頻度	意味
create	最初に base.tgz を作成するときに一度	ベースファイルシステムの作成
update	一日二回 (unstable のアップデートに伴う)	ベースファイルシステムの更新
build	パッケージビルドのたび	Debian パッケージを chroot 内部でビルドする

Debian Developer の通常のある一日に発生するイベントを検討してみましょう。

^{*1} pdebuild 命令のほうが便利な場合があります



pbuilder はパッケージのビルドのプロセスの一部、「確認プロセス」に組み込まれています。^{*2} これは、Build-Depends を正しく設定できているのか試験するのに便利です。また簡単な回帰テストのフレームワークとして活用できます。

6.2 pbuilder 自身の開発の仕組み

pbuilder 自身がどう開発されているのか解説します。現在、alioth で提供されているリソースを活用して co-maintain(共同メンテナンス) されています。最近の主要な開発メンバーは Loïc Minier と上川です。たまに Matt

^{*2} これは作業フローの一例で、全員がこういう作業フローになっているというわけではありません。例えば、一部の開発者はローカルでビルドするということをせず、chroot 内部で全部の作業を完結している場合もあり、その場合はローカル環境でのビルドとテストのステップが省略されます。利点の一つとしてはローカル環境に sid の環境をもたなくてすむことがあげられますが、その点については sid を自分自身でも利用しないことになるため、賛否両論です。

Kraai と Mattia Dongili がコミットします。

プロジェクトページは <http://alioth.debian.org/projects/pbuilder> にあり、ホームページは <http://pbuilder.alioth.debian.org/> にあります。ホームページは pbuilder マニュアルになっています。

ソースコードの管理には git を利用しています。レポジトリは下記のコマンドのいずれかを利用してチェックアウトできます。^{*3}

```
git-clone git://git.debian.org/git/pbuilder/pbuilder.git
git-clone http://git.debian.org/git/pbuilder/pbuilder.git
git-clone ssh://git.debian.org/git/pbuilder/pbuilder.git
```

6.3 派生物とその状況

pbuilder にはいくつかの派生物があり、異なるバックエンドをサポートしています。それらは別の方法をクリーンルーム試験環境を提供するのに利用しています。それらを簡単に紹介します。

6.3.1 LVM スナップショット版

誰かが LVM スナップショットを base.tgz の管理用に利用する仕組みを提案しました。どっかにメールで投稿されています。ただ、誰も採用して開発を継続しようとはしていないようです。環境の分離方法は chroot を利用しています。LVM スナップショットの利点としては、tar アーカイブの展開より格段に高速だという点があげられます。

6.3.2 user-mode-linux 版

pbuilder-uml が存在します。どうやらほとんどの人が利用できているようです。Mattia Dongili たちがこの移植版の開発に携わっています。

base.tgz の展開のかわりに UML cow デバイスでクリーンルーム環境の維持が実現されています。また、chroot のかわりに user-mode-linux を活用しており、結果として各種システムコールが遅くなり全体としては実行オーバーヘッドがあります。

6.3.3 cowdancer 版

上川が cowdancer 版の作業を 2005 年くらいから行っています。安定しているようです。base.tgz の展開が cp -1a におきかわっており、高速です。

ただし、cowdancer が libc のコールをフックして実現しているため、一部のパッケージのビルドに影響が出る可能性があります。^{*4}

6.3.4 qemu 版

上川が qemu/kqemu/kvm 版の開発を 2007 年初頭から始めました。QEMU の COW ブロックデバイス機能を活用しているため、base.tgz の展開が不要になります。

qemu 版は別アーキテクチャ向けのビルド (クロスビルド) 機能を提供するという特徴があります。たとえば i386 マシン上で ARM 用のパッケージをビルドすることだってできるはずです。

6.4 さらなる開発のアイデア

6.4.1 インストールテスト

インストールテストについては、いくつかの piuparts のようなプロジェクトがあり、pbuilder でも応用できそうです。コンセプトを実装する簡単な例としてのスクリプトは pbuilder で提供しています。/usr/share/doc/

^{*3} ssh アクセスには alioth のアカウントが必要です

^{*4} Etch のリリースの際には、残念ながら Bug 413912 のような問題が発生しました。

pbuilder/examples/execute_installtest.sh です。

```
pbuilder execute \  
  /usr/share/doc/pbuilder/examples/execute_installtest.sh \  
pbuilder
```

このコマンドは 指定したパッケージを chroot 内部で apt-get でインストールしようとし、成功するかしないかを確認してくれます。

6.4.2 パッケージのテスト

パッケージのテストの機能は重要です。とくに、開発者の時間は限られており、手動でテストを繰り返すというのは楽しいことではないからです。pbuilder は例としてフックスクリプトを提供しています。/usr/share/doc/pbuilder/examples/B92test-pkg はパッケージのビルドが成功した場合に、テストを実行するようになっています。

テストファイルは debian/pbuilder-test/NN_name (NN は数字です) におきます。ファイル名は run-parts の標準に従います。^{*5}。

6.4.3 aptitude

pbuilder は現在 apt-get コマンドを活用しています。しかしながら、aptitude の普及に伴い、aptitude を活用する方法を考える時期にきているかもしれません。

6.4.4 apt-key support

pbuilder はあいかわらず apt-key をサポートしていません。現在の stable リリースで apt-key が提供されているため、そろそろ apt-key をサポートしようかな。

6.4.5 build-dependency parser

Build-Depends を解析するパーサは古く、最適ではありません。それをうけて、Loïc Minier はいくつかの再実装を試行しています。

6.4.6 buildd.net のような仕組みのサポート

pbuilder はたくさんのログを提供するのですが、pbuilder 自体は歴史の概念をもちあわせていません。pbuilder 単体ではログを集めて活用するというようにはなっていません。^{*6} 過去のビルドログをローカルに集めて、各ビルド間での差分を確認することを通して、問題を検出できるかもしれません。debdiff などのツールと組み合わせて利用するとよいかもしれませんね。

6.5 References

- <http://pbuilder.alioth.debian.org/> か /usr/share/doc/pbuilder/pbuilder-doc.html: pbuilder マニュアル
- cowdancer パッケージ
- piuparts パッケージ
- autodebtest: Ubuntu の自動テストシステム
- schroot / dechroot
- buildd

^{*5} ファイル名に '.' が入っていたら無視されますよ!

^{*6} pbuildd を作成するというプロジェクトは存在しました。最近どうなってるのかについては把握していません。



7 Debian on SuperH

岩松 信洋

今年に入って、ちまちまと Debian の SH へのポーティングを再始動したのですが、Debconf7 の BOF で Debian porting for SH が通ってしまいました。Debconf7 で発表する内容を以下にまとめたいと思います。

7.1 SuperH とは

SuperH(以下、SH) は ルネサステクノロジ <http://www.renesas.com/> が販売している 組み込み向けの CPU です。特徴としては以下のものがあります。

- 日本国産
 - 低電圧
 - 種類が多い
- SH1/SH2/SH2A/SH3/SH3-DSP/SH4/SH4A/SH4AL/SH4AL-DSP

携帯電話、HDD コンボ、液晶テレビ、カーナビゲーションシステム等で採用され、Linux が動作しています。

7.2 歴史

2000 年前後から Debian に SuperH を移植しようとする活動が行われてきました。それらを紹介します。

7.2.1 第 0 次 SH ブーム

約 7 年前、情報処理推進機構 (旧 情報処理振興事業協会)、*⁷ 略称「IPA」の未踏ソフトウェア創造事業に採択され、SH が Linux に移植されました。*⁸ *⁹

そして、この時の移植チームのメンバの一人で、Debian Developer である八重樫 剛史 氏*¹⁰を中心に、X Hacker である石川 睦氏*¹¹が Debian に移植を試みました。このときの状況は、

- サポートアーキテクチャは sh(little endian) と sheb(big endian)
- base はできており、コンパイラも当時最新のもの
- ネイティブで作成するのはリソースが足りないため、DODES プロジェクトを立ち上げ、コンパイル

とすばらしいものでした。

過去に行われた Debian Conference 2001 Tokyo, Japan *¹² で石川氏 が Debian GNU Linux on SuperH *¹³ として発表されておられます。

*⁷ 英語名 Information-technology Promotion Agency

*⁸ <http://www.ipa.go.jp/NBP/12nendo/12mito/mdata/5-9gh/5-9gh.pdf>

*⁹ <http://lc.linux.or.jp/lc2001/papers/linux-superh-paper.pdf>

*¹⁰ yaegashi@debian.org

*¹¹ ishikawa@debian.org

*¹² <http://www.debian.or.jp/community/events/2001/1025-dcj2001/>

*¹³ <http://www.debian.or.jp/community/events/2001/1025-dcj2001/HANZUBON-debian-sh.html>

しかし、八重樫氏が Debian-superh ML に移植を行う旨

<http://lists.debian.org/debian-superh/2001/12/msg00013.html>

を伝えたところ、

- big endian 必要ないのでは？

<http://lists.debian.org/debian-superh/2001/12/msg00014.html>

- SH3/SH3eb , SH4/SH4eb の 4 つのアーキテクチャを入れると、サーバーの容量の問題が発生するため、入れること難しいと思う。

<http://lists.debian.org/debian-superh/2001/12/msg00020.html>

という問題提議があり、sh3/sh3eb/sh4/sh4eb にアーキテクチャを分けたまま、移植は止まってしまったのです。

<http://lists.debian.or.jp/debian-devel/200011/msg00044.html>

当時、ビルドに使われていたマシン以下の通りです。



図 1 Solution Engine

	Solution Engine
CPU	SH7709 (133Mhz)
memory	32MB
Flash	4MB
IDE	PCMCIA slot
Ethernet	stnic



図 2 CAT 709

	Solution Engine
CPU	SH7709 (133Mhz)
memory	32MB
Flash	8MB
IDE slot	CF Slot
Ethernet	なし

その他に Hewlett-Packard 社から販売されていた、Jornada6xx シリーズも使われていたとのこと。

7.2.2 第 1 次 SH ブーム

約 2 年前、SH を採用した NAS、LANDISK / LANTANK が I/O データさんから販売されました。これらは I/O データさんに在籍しておられる、kinneko さん^{*14} および iohack project <http://iohack.sourceforge.net> の元、開発が行われ Debian パッケージでシステムが構築されていました。安値であり、自由に触ることができるということで、人気があり、各 Linux 雑誌でも取り扱われ、ブームを築きました。これが 第 1 次 SH ブームです。

以下に LANDISK / LANTANK のスペックを示します。

しかし、SH の販売元であるルネサステクノロジはこのブームをうまく活用することができず、そのまま消えていこうとしていたのです。

7.2.3 歴史は繰り返さないために

終了してしまったように見えた、SH の Debian への移植ですが、私がパッケージを再ビルドし、再移植を行うことにしました。理由としては、

^{*14} <http://d.hatena.ne.jp/kinneko/>



図 3 LANDISK

	LANDISK
CPU	SH7751R (266Mhz)
SDRAM	64MB
Flash	ROM
IDE	UDMA133 PATA (ACARD ATP865)
Ethernet	10/100Base-T (RTL8139CL + EEPROM 93C46)
USB	USB2.0 TypeA Conn x2 (NEC D720101GJ)
値段	約 35000 円



図 4 LANTANK

	LANTANK
CPU	SH7751R (266Mhz)
SDRAM	64MB
Flash	ROM
IDE	UDMA133 PATA (ACARD ATP865) 2disk support
Ethernet	10/100Base-T (RTL8139CL + EEPROM 93C46)
USB	USB2.0 TypeA Conn x2 (NEC D720101GJ)
値段	約 19800 円

- SH で 容易に使用できるディストリビューションがない。
gentoo でサポートされているが、ビルドが面倒。
- 仕事でも使えるようにしたい :)
個人的理由。
- Debian User だから。

私は今回の移植では、以下のポリシーで作成することにしました。

- Debian でサポートするアーキテクチャ
SH4 のみをサポートします。SH4A 等もすべて SH4 として扱います。
- SH3 サポート
SH3 と SH4 の大きな違いは FPU があるか、ないかです。SH3 は SH の Linux カーネルでサポートされた math-emu を使って、math をエミュレーションする方法を取ります。もちろん、math-emu を使うと、遅くなりますが、SH3 を採用した CPU がほとんど存在しないため、コストにはならないと考えています。
他には、cache の違いもあるので、これをうまくトラップできる仕組みを入れる必要があります。
- big endian サポート
big endian もサポートするすることも考えています。

7.3 現状

7.3.1 開発メンバー

現在、基本的に開発を一人でやっていますが、以下の方々にサポートをしてもらっています。

- 小島先生 binutils SH メンテナ
SH の分からないところは質問させていただいています。
- 武藤さん
buildd の構築で相談に乗ってもらっています。
- kinnneko さん
いろいろ手伝ってもらっています。
- gotom さん
Debian glibc メンテナ. 開発用のボードは渡したんだけど.....。
- 上川さん
LANTANK 買わせたんですが.....。

7.3.2 現在使用しているビルドマシン

現在、ビルドに使用しているマシンは以下の通りです。

1. LANTANK x 3



図 5 LANTANK x 3

2. RTS7751R2D

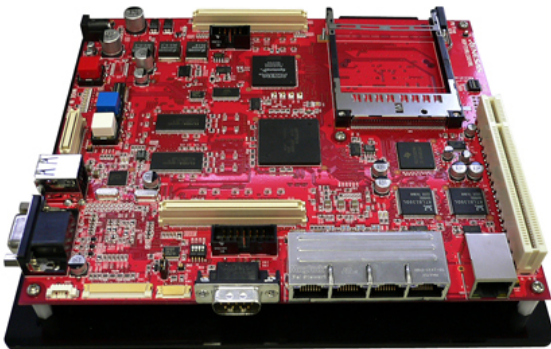


図 6 RTS7751R2D

	RTS7751R2D
CPU	SH7751R (266Mhz)
SDRAM	64MB
Flash	ROM
IDE	CF slot and PCMCIA
Ethernet	10/100Base-T (RTL8139CL + EEPROM 93C46)
USB	SM501 USB1.1 (Silicom Motion)

ルネサスソリューション様から提供していただきました。

7.3.3 パッケージ状況

現在のパッケージ状況は以下の通りです。

- SH4 のみをサポート
ビッグエンディアンのマシンを持っていないため。
- build-essential ビルド完了

gcc-4.1.2 / binutils / glibc-2.5

- sid debootstrap サポート

debootstrap できるパッケージができています。

- SH4 buildd 稼働中

公開はされていませんが、SH4 向けの buildd が稼働中です。今後、buildd.net www.buildd.net に登録する予定です。

これらの成果物を <http://www.nigauri.org/~iwamatsu/debian/debian-sh4/> で公開中です。

7.4 今後の課題

- Debian の正式なサポートアーキテクチャにする
- buildd.net への登録
- buildd のメンテナが行える体制を作る
共同メンテナを募集しています。
- 回線の保持
buildd 用マシンネットワークの確保

7.5 リンク

- SuperH <http://www.renesas.com>
- IRC #debian-superh @ oftc.net
- ML debian-superh@debian.org
- Debian Packages repository <http://www.nigauri.org/~iwamatsu/debian/debian-sh4/>

- iohack project <http://iohack.sourceforge.jp>



8 Debian の情報フロー

上川

Debian etch を活用するためには、何をしたらよいですか？ 今日のようにみんなあつまって議論すると、いろいろな新しい発見や情報が出てくるけど、それって普段みんなどうしているの？ その疑問を追求するためにワークショップをしてみました。

まず建前から確認してみましょう。

- 基本はリリースノート
- BTS で生きた情報を得る
- debian-users メーリングリスト

しかし実際の運用は少し違うこともあります。理由としては例えば次のようなものがそれぞれあります。

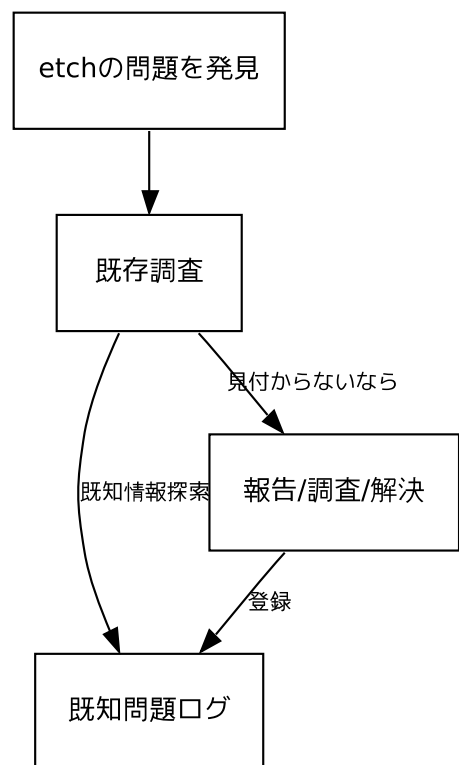
- リリースノート：一度リリースされてしまうとそう簡単に項目が追加される類のドキュメントではない。
- BTS：英語なので日本では活用しているメンバーに限られる。また英語が問題とならなかったとしても、BTS の仕組みとして、情報がパッケージ単位で管理されるため複合的に発生する問題は登録しにくい

現状の運用はどうかというと、ワークショップで出てきた話を総合すると、実際は次のような流れになっているようです。

- IRC でまずきく
- google で検索 (blog とか スレッドテンプレで発見)
- 報告は blog に記述
- 2ch で質問が出たりしたら blog を参照してスレッドテンプレに

登録

では、現状を把握したところで、この仕組みのまま改善するか、違う仕組みを考えるか、検討していけばよいはずですが、続きはまた。



下ビアノ勉強会



Debian 勉強会資料

2007年5月19日 初版第1刷発行
東京エリア Debian 勉強会（編集・印刷・発行）
