

東京エリア デビアン 勉強会



Debian勉強会幹事 上川純一

2008 年 7 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 の世界にあしを踏み入れるという方も、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？

Debian 勉強会の目的は下記です。

- Debian Developer (開発者) の育成。
- 日本語での「開発に関する情報」を整理してまとめ、アップデートする。
- 場 の提供。
 - 普段ばらばらな場所にいる人々が face-to-face で出会える場を提供する。
 - Debian のためになることを語る場を提供する。
 - Debian について語る場を提供する。

Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりと作るスーパーハッカーになった姿を妄想しています。情報の共有・活用を通して Debian の今後の能動的な展開への土台として、「場」としての空間を提供するのが目的です。

以上を目的とした、2008 年アジェンダです：

1. 新年会「気合を入れる」
2. Open Source Conference Tokyo (3/1)
3. データだけのパッケージを作成してみる、ライセンスの考え方 (David Smith)
4. バイナリーつのパッケージを作成してみる (吉田@板橋)
バージョン管理ツールを使い Debian パッケージを管理する (git)
アップストリームの扱い (svn/git/cvs)(岩松 信洋さん)
5. バイナリの分けたパッケージの作成。(前田さん)
バイナリの分け方の考え方、アップグレードなどの運用とか。
6. パッケージ作成 (dpatch/debhelper で作成するパッケージ)(小林儀匡さん)
man の書き方 (roff or docbook)(でんさん) OSC 2008 Hokkaido
7. パッケージ作成 (kernel patch、kernel module)(岩松 信洋) Debconf 発表練習 (上川さん)
8. Debconf アルゼンチン、共有ライブラリパッケージ作成コミックマーケット 74
9. Open Source Conference Tokyo/Fall、デーモン系のパッケージの作成、latex、emacs-lisp、フォントパッケージ
10. パッケージの cross-compile の方法、amd64 上で i386 のパッケージとか、OSC-Fall 報告会、Debconf 報告会
11. 国際化 po-debconf / po 化 / DDTP
12. 忘年会

目次

1	Introduction	1
2	事前課題	3
2.1	鈴木 崇文 さん	3
2.2	前田 耕平さん	3
2.3	やまねさん	4
2.4	あけどさん	4
2.5	伊藤 弘和さん	4
2.6	大波 誠さん	4
2.7	市川 憲人さん	5
2.8	本庄さん	5
2.9	濱野さん	5
2.10	青木 修	5
2.11	藤沢 理聡 さん	6
2.12	岩松 信洋	6
3	Debian Trivia Quiz	7
3.1	debian-devel-announce	7
3.2	Debian Project News 2008 年 05 号	7
3.3	Debian Project News 2008 年 06 号	8
4	最近の Debian 関連のミーティング報告	9
4.1	東京エリア Debian 勉強会 41 回目報告	9
4.2	eeePC Developers Conference day 1	9
4.3	Introduction to eeePC SDK	11
4.4	eeePC Developers Conference day 2	12
4.5	OSC 2008 Hokkaido	14
5	Linux カーネルパッチ の Debian パッケージ作成	15
5.1	はじめに	15
5.2	Linux カーネルパッチ の Debian パッケージの仕組み	15
5.3	パッケージ作成前の準備	15
5.4	dh_make を使った雛形の作成	17
5.5	debian ディレクトリ内の変更	17
5.6	パッケージのビルドおよびインストール	20
5.7	パッケージのテスト	20
5.8	パッケージのアップデート方法	20
5.9	まとめ	20
6	Linux カーネルモジュールの Debian パッケージ作成	21
6.1	はじめに	21
6.2	なぜカーネルモジュールソースコードをパッケージ化するのか	21
6.3	カーネルモジュールパッケージの仕組み	21
6.4	ドライバソースコードの取得と展開	21
6.5	dh_make を使った雛形の作成	22
6.6	debian ディレクトリ内の変更	22
6.7	パッケージのビルドおよびインストール	26
6.8	モジュールパッケージの作成	27
6.9	まとめ	27

2 事前課題

岩松 信洋

今回の事前課題は

1. Debian に取り込んでほしい Linux カーネルパッチ、Linux ドライバを教えてください。
2. unstable でアップデートされなくて困ってる Debian パッケージ、BTS に登録されているバグのうち、早く直してほしいバグを挙げてその理由を教えてください。
3. ここで開催してくれないかなあ という勉強会の開催地をその理由と共に挙げてください（近いから、だけは却下！）

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 鈴木 崇文 さん

たいして需要もありそうでもなく、結構不安定だという話なので、個人的な趣味の問題ですが、vlookback というカーネルモジュールが面白いので追加でインストールできるとうれしいです。vloopback を使用すると、ビデオ出力に手を加えて仮想的なビデオ入力デバイスを通してループバックしたりでき、つまり dv 入力を v4l に変換したり、uvcvideo デバイス (v4l2) を v4l に変換できたりして ustream.tv でのストリーミングが使えるようになったりします。追加モジュールとして deb パッケージで追加インストールできるのならば、今回の勉強会で学んで試しに自分用のパッケージを作成してみたいです。

2.2 前田 耕平さん

- Debian に取り込んでほしい Linux カーネルパッチ、Linux ドライバを教えてください。Debian というよりは、Linux 自体に対してなんですが、bcm4328 がサポートされるとうれしいですね。Broadcom の製品概要を見ると Linux はサポート OS にはなっているものの、Linux Wireless のサイトを見るとまだ対象外^{*1} のようです。先日、Ndiswrapper を使ってようやく MacBook Air で無線 LAN を利用できるようになったものの、黒 MacBook に比べるとどうも IP アドレスの割り当てが不安定です（一度割り当てされると安定するんですが）。BB モバイルポイントで WEP で接続する分にはまったく問題無かったので、無線 LAN ルータとの相性な気もしなくはないですが。^{*2}

Debian に、という観点では、特定のパッチやドライバというよりは、最新のパッチやドライバ、ソフトウェアの Sid への取り込みがもっと早ければなあ、と思うことはあります。Upstream に比べると遅くなるのは仕方無いのですが、あまりにかけ離れてしまうと、自分で Upstream の最新版を持ってきちゃえばいいかと、Debian パッケージは使わずに自分でビルドしてしまうので。

^{*1} <http://ja.broadcom.com/collateral/pb/4328-PB02-R.pdf>

^{*2} <http://wireless.kernel.org/en/users/Drivers/b43#unsupported>

- ここで開催してくれないかなあ という勉強会の開催地をその理由と共に挙げてください（近いから、だけは却下！）。

割と手頃な値段で借りられそうな候補を上げておきます。

- － 晴海区民館 <http://www.city.chuo.lg.jp/sisetugaido/syukaisisetu/syukaisisetu17/> インターネット接続環境はなし。プロジェクターも無いのがネックです。
- － ITS 健保の会議室半日借りられる、という点では割と安いです。プロジェクターは借りられます。また、ITS の健保組合員がいないと借りられません。
- － 市ヶ谷会議室 http://www.its-kenpo.or.jp/restaurant/itigaya_kaigisitu/index.html 少人数（10 人程度）の場合はまあまあかも。
- － 山王会議室 http://www.its-kenpo.or.jp/restaurant/sannou_kaigisitu/index.html ある程度人数がいないとペイできません。（50 人以上）
- － 大久保会議室 http://www.its-kenpo.or.jp/restaurant/okubo_kaigisitu/index.html 部屋の種類（定員）が一番多いです。

2.3 やまねさん

- BTS に登録されているバグのうち、早く直ってほしいバグを挙げてその理由を教えてください。
最近バグウォッチしてないのでなんともむずかしいですが、自分が登録してる ITP バグをクローズしたいなーとは日々思っています。思ってるだけで何もしてないんですが。あとは l10n な ja.po は登録して 1 年とか放置されると悲しい気分になるので早めをお願いしたい所です（最近 i18n な NMU で fix されますが）。つーかね、動かねーとか何か変とかいうんだったら、BTS に追加情報だせよゴルァ！とか思ったりしたりしなかったりする今日この頃。

2.4 あけどさん

- unstable でアップデートされなくて困ってる Debian パッケージ、BTS に登録されているバグのうち、早く直ってほしいバグを挙げてその理由を教えてください。
net-tools パッケージのいろんなバグ特に netstat コマンドの IPv6 アドレスが切り落とされるバグは lenny 以降での対応が IPv4 アドレスを表示する形になっていてぱっと見ただけで如何にもダメな印象になってしまうので upstream を含めて何とかして欲しいところです。
ちなみに他の某ディストリビューションでは IPv6 アドレスが正常に表示されるようになっていてとても悲しかったりします。
どうやったらプロパーな状況になるんでしょうか？

2.5 伊藤 弘和さん

- unstable でアップデートされなくて困ってる Debian パッケージ、BTS に登録されているバグのうち、早く直ってほしいバグを挙げてその理由を教えてください。
バグというよりは、open-vm-tools パッケージの修正が一段落ついて欲しいなと思っております。熱い状態ということもありますが自分も常用している物こそ、貢献できるようになりたいとパッケージの熟成をウォッチしています。

2.6 大波 誠さん

- Debian に取り込んでほしい Linux カーネルパッチ、Linux ドライバを教えてください。

正直に申し上げます。パツと思いつくものがありませんでした。ただ、この機会に「Linux ドライバ」とか「debian ドライバ」とかでぐぐってみました。そしたら The Linux Foundation がデバイスドライバのオープンソース化を声明として発表していたり グラフィックボードのドライバなどのプロプライエタリなソフトウェアに関する問題が載っていたりして勉強になりました。前田 耕平さんから勉強会を紹介していただきました。初参加させていただきます。

2.7 市川 憲人さん

- Debian に取り込んで欲しい Linux カーネルパッチ
個人的な事情なのですが、自宅サーバが MacBook でそのネットワークインターフェイスが Marvell Yukon 88E805x で、kernel 標準の sky2 ドライバはやや挙動が怪しいので、Marvell 提供のドライバを debian のパッケージ化してあると便利です。自分の技術力では make bzImage すら通らなかったのどなたか技術力のある方がパッケージ化して下されると大変嬉しいです。

2.8 本庄さん

- Debian に取り込んでほしい Linux カーネルパッチ、Linux ドライバを教えてください。
特に思いつかないです。
- unstable でアップデートされなくて困ってる Debian パッケージ、BTS に登録されているバグのうち、早く直ってほしいバグを挙げてその理由を教えてください。
アップデートされなくて困っているというわけではありませんが、最近 webmin を使いたいと思うことがあり、ちょっと残念でした。
- ここで開催してくれないかなあ という勉強会の開催地をその理由と共に挙げてください（近いから、だけは却下！）
以前にも話に上がっていた、温泉宿はどうでしょう。夜を徹して語り合う先になにかが見えるかもしれません。

2.9 濱野さん

- Debian に取り込んでほしい Linux カーネルパッチ、Linux ドライバを教えてください。
netfilter の P-O-M patch に含まれる ipset というモジュールを使用しています。なかなか mainstream に入ってくれない様なので package で用意しておくとう便利かな、と思いました。
もうひとつ La Fonera のイメージを触るのに squashfs もあると嬉しいかな、mksquashfs というユーザランドツールでこと足りてたりもするのですが^^;

2.10 青木 修

まあ、かなり利用者の増えている SCIM ですがパッケージングに参加されている人がすくないので、なかなかアップデートが追いつきません。

メインの MING さんは忙しいので、協力してくれるかたいませんか?私は単なる UPLOADER で技術的にこれは難しいので、いい人探しています。

あー、開催場所ですが、東工大の長津田キャンパスなどだれか学生さんがいそうなのになぜないのかなーといつも思っています。

2.11 藤沢 理聡 さん

- ここで開催してくれないかなあ という勉強会の開催地とその理由
開催地：学校（高校・大学など）理由：一言で言うと、学生に興味を持って欲しいからです。Debian に触れるのに年齢は関係ないと思うけれど、若いうちに Debian に触れるのも悪くないと思います。学校という場所や学生という立場では、今のところ Linux のようなフリーな OS に触れる機会は多くないのが残念です（社会に出ても触れる機会は多くないけど）。とにかく、まだ知らない人が興味を持ってくれそうな、そういう場所で開催するのもいいんじゃないかな、と思いました。

2.12 岩松 信洋

- Debian に取り込んでほしい Linux カーネルパッチ、Linux ドライバを教えてください。
今回の資料作成のついでに作りました。内容は発表を参照。
- unstable でアップデートされなくて困ってる Debian パッケージ、BTS に登録されているバグのうち、早く直ってほしいバグを挙げてその理由を教えてください。
kernel-package 関係のバグなど。メンテナ (Manoj) サボリ気味ですね。困りましたね。libflash の不具合が直ってないのをなんとかしてほしい。
- ここで開催してくれないかなあ という勉強会の開催地をその理由と共に挙げてください（近いから、だけは却下！）。
先日、東京電機大学さんに行って、コラボできないか相談してきました。学生さんに興味を持たせるにはどうしたらいいですかね。あとは温泉を 8 月に企画中です。

3 Debian Trivia Quiz

小林 儀匡

ところで、みなさん Debian 関連の話題においついていますか？ Debian 関連の話題はメーリングリストをよんでいると追跡できます。ただよんでいるだけでははいあがないので、理解度のテストをします。特に一人だけでは意味がわからないところもあるかも知れません。みんなで一緒に読んでみましょう。

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。

3.1 debian-devel-announce

`debian-devel-announce@lists.debian.org` への投稿内容からです。

問題 1. Perl 5.10 のバグでどのような問題が発生した？

A Ruby と Python で書かれたアプリケーションやライブラリがインストールできない

B インストールされたファイルのパーミッションが 0777 になる

C 特定の名前のファイルがインストールできない

問題 2. Debian プロジェクト内のチームに関する調査で判明した「予想外のチーム」でないものは？

A 実際に動いているのは 1 人だけというチーム

B 誰が作業するか毎回じゃんけんで決めているチーム

C お願いやありがとうではなく脅迫で動いているチーム

問題 3. wxwidgets2.8 がアップロードされたが、長いこと wxwidgets2.6 の時代が続いていた。その理由は？

A パッケージメンテナが保守的で、2.8 の使用に対して非積極的だった

B アップロードしたところでどうせ誰も使ってくれないとパッケージメンテナが思った

C パッケージメンテナが多忙で作業時間がとれなかった

問題 4. Frans Pop の辞任によって、新たな執筆者が求められるようになったものとは？

A リリースノート

B Debian Project Blog

C DEB NOTE

3.2 Debian Project News 2008 年 05 号

<http://www.debian.org/News/weekly/2008/05/>

にある 6 月 23 日版です。

問題 5. `debian/rules` の `get-orig-source` ターゲットは何を記述するためのものか？

A 「オリジン 弁当」で弁当にソースをつけてもらう方法

B upstream からネットワーク経由でソースコードを取得して現在のソースコードと置き換える方法

C upstream からネットワーク経由で最新の `.orig.tar.gz` ファイルを取得する方法

問題 6. リリースゴールに関する Peter Eisentraut の意見は？

A リリースゴールなんて所詮一部の開発者の楽しみに過ぎない

B Debian の機能の実装に関するリリースゴールはリリース後にポリシーへと変えるべきだ

C リリースなんて飾りです。偉い人にはそれがわからないのですよ

問題 7. William Pitcock が削除を提案したブートローダパッケージは?

- A grub
- B lilo
- C yaboot

問題 8. Debian weather とはどんなサービスか?

- A 特定アーキテクチャのアーカイブの状態を要約して表示する
- B Debian 関連ホストが置かれている世界各地の天気を表示する
- C メーリングリストの流量から世界各地の天気を推測して表示する

問題 10. Debian のメニュー (.menu ファイル) とデスクトップ環境のメニュー (.desktop ファイル) に関する議論はどのような結論に落ち着いたか?

- A freedesktop.org の.desktop ファイルを Debian に合うよう拡張して使っていこう
- B freedesktop.org の.desktop ファイルには不便な点があるので、働きかけて修正してもらおう
- C freedesktop.org の.desktop ファイルは使えないので Debian の.menu ファイルを使わせよう

問題 11. 6 月末に初めて誕生した Debian 開発者同士の夫婦とは?

- A Junichi Uekawa と Kenshi Muto
- B Meike Reichle と Alexander Schmehl
- C Debra Murdock と Ian Murdock

3.3 Debian Project News 2008 年 06 号

<http://www.debian.org/News/weekly/2008/06/>
にある 7 月 7 日版です。

問題 9. Debian 15 周年はいつか?

- A 次回の東京エリア Debian 勉強会開催予定日である 8 月 16 日
- B 本日 7 月 19 日
- C 泣く子も黙る 7 月 9 日

問題 12. 結婚した二人について述べた以下の事項のうち、正しいものは?

- A 最初の贈り物: DebConf5 の土産
- B 秘密の愛の交換手段: wiki.debian.org
- C 婚約の公式発表手段: lists.debian.org

4 最近の Debian 関連のミーティング報告

岩松 信洋

4.1 東京エリア Debian 勉強会 41 回目報告

6 月の第 41 回東京エリア Debian 勉強会を実施しました。今回の参加者は あけどさん、前田さん、小林儀匡さん、吉田@板橋さん、山本 浩之さん、山本琢さん、藤沢理聡さん、日比野 啓さん、鈴木 邦男さん、岩松の 10 人でした。

- まず最初に最近のミーティングの報告を行いました。
- 今回のクイズは岩松が出題しました。全問正解者はいませんでした。似たような答えが多く、回答者は困っていました。
- 事前課題を紹介しました。AIX のログシステムと提供されている `errpt` コマンドがかなりいけているようです。実際に実機を触ってどんな感じなのか見てみたいです。あと、hinemos の話が出ました。Debian で使いたい方が多いようですが、Java で実装されており、要求するマシンスペックも高いため、パッケージングとテストの道のりは激しく遠い気がしました。
- 2008 年のテーマは DEB パッケージの開発・管理に関連した内容ですが、4 回目のテーマとして小林さんが `debhelper` を使ったパッケージング方法について紹介しました。`debhelper` を使わないパッケージング方法から、`debhelper` を使った方法、`cdb` を使った方法をいう流れで話してくれたので、`debhelper` と `cdb` のありがたさが分かりました。
- 最近の kFreeBSD, nexenta, Hurd, SuperH, eeepc について紹介して終了しました。
- 隠し球だった山本さんの Hurd ユーザーを増やす会の発表ができませんでした。
- 今回も勉強会の間違えてしまった人がおられました。開催場所は毎回チェックするようにしましょう。
- 勉強会のあと、居酒屋さつきで宴会をしました。Debian ぐるぐるカッティングステッカーを販売してみたところ、なかなかの好評ぶりでした。さっそく PC に貼っている人もおり、大満足のようです。ステッカーが欲しい人は私に連絡をください。

4.2 eeepc Developers Conference day 1

経済部工業局。登録したら無料で参加できるカンファレンス。講演者が全員スーツをきていますが、来場している方々は普段着です。カメラのフラッシュの量からは注目度が伺えます。携帯電話の着信音がなりほうだいです。

開始のオープニングは多少おくれましたがほぼ満員で開始しました。後半になると半分くらいの埋まり方になりました。

みなさま eeepc でプレゼンしていますが、プロジェクターが同期とれていない感じで、全体がうつっていません。外は暑く、会場が寒いんだけど、みんななれているものでジャンパー着てます。

5 月 8 日に 1 日目が開催されました。

09 : 00	09 : 30	Register	
09 : 30	09 : 45	Opening Remarks	
09 : 45	10 : 20	Keynote Speech Keynote Speech – EeePC Software Platform Business Model	ASUS — Ellis Wang
10 : 20	10 : 50	EeePC Application Demo Show	ASUS Patrick Chou
10 : 50	11 : 00	Tea Break	
11 : 00	12 : 20	EeePC SDK Announcement and Demo	ASUS
12 : 20	13 : 40	Lunch	
13 : 40	14 : 30	The rule of game to stand on the shoulders of giant	Florence T.M. Ko, OSSF
14 : 30	15 : 10	Developing with pyGTK in EeePC	TsungWei Hu, OSSF
15 : 10	15 : 30	Tea Break	
15 : 30	16 : 10	Developing with pyQT in EeePC	Gary Lee
16 : 10	16 : 50	Firefox Platform Application Programming	Thinker
16 : 50	17 : 30	Google Gears Programming	Tzeng Chien-Ming
17 : 30	17 : 40	Software Vendor Panel Discussion	ASUS

4.2.1 開会のご挨拶

eeePC SDK の発表

Open Source Development

4.2.2 開会のご挨拶 2

何かの話、まったくわからず。

4.2.3 Announcement of eeePC SDK, Dev tools and Community

ASUS の Ellis Wang 氏が発表しました。eeePC でプレゼンしています。オープンソースのデベロッパーの人たちにできるだけ参加してほしいという気持ちが伝わってきます。英語のプレゼンで、中国語で発表しています。

最初に開会の挨拶をしたらいひとになぜか eeePC を贈呈しています。

Linux で、コミュニティについてなんたら。

ISV とコミュニティ

SDK の話。

ASUS と eeePC のソースコードがツールダウンロードできるよね。

オープンソースコミュニティが eeePC を支援している。

Linux にかぎらず、Debian, KDE などがある。

Microsoft の UI に比べて、UI の変更が簡単と言っているような気がする。

Linux の素晴らしさについて。EeePC:

- easy – to learn, work and play
- excellent – on the go
- excellent – internet experience

インテグレーションすることの大切さ。contribution を行う。

汎用のツールのソースを組み合わせて作り上げる。

オープンソースは共有することが大切であり。IP の話。[share]

warranty の話?

reuse の話。

EeePC SDK とは？

マニュアルの目次を画面にうつす。EeePC のパッケージは Debian パッケージだよ。

パッケージのテストは？VMware をつかおう。

eeePC は開発者・ASUS・ユーザ・CC デジタルコンテンツをつなぐためのブリッジだ。

無線とか CPU とかカーネルとかは ASUS がやるだろ？

GPL だろ、とか。

Debian のパッケージとかを提供する。

パーティカルで No1 を目指す。テレコムと教育目的に。

partner のレベルがある。

システムと OS のパートナーは一番少ないがインテグレーションの度合いが高い。

SDK 作ったのでオープンソースデベロッパーたちにデベロップメントコミュニティをつくることができるだろ？
コマーシャルベンダーもツールが充実していて満足するにちがいない。オープンソースの技術者もよく知っている技術でうれしいに違いない。

Eclipse と Qt4 の開発ツールキットがはいっているぜ。

開発プロセスの Phase は二種類ある。実機と VMware だ。フェーズイーとフェーズアル。

テストの小技巧。画面サイズとか全部試しておくのがよいです。

デスクトップと比べると一部違います。ASUS desktop で、ユーザアカウントが一つしかないよ。また、init はミニマルです。

howto

画面小さい。Flash は 10 万回書き込むと wear out するのでできるだけ書き込みしない。

eeePC アプリケーションの移植は簡単よ。Java も動く。Picasa は Wine で動いている。(Linux の説明をしている気がする。)

Adobe flash はサポートするけど AIR はダメだね (?)。

と、終わったら人が騒然としはじめた。休憩時間開始。スナックとコーヒーが提供されました。

4.2.4 教育用のアプリケーション？

株式取引ツール

発音の出る辞書

カラオケみたいな、音楽が流れながら歌詞と訳語の出るツール。

ASUS MeBook

MeReader MeSong

本を買うことができる。

ラップ風の中国語会話。IQChinese。PinYin の勉強させてくれる。

音声制御のツール。Skype を声で起動できる？

一ページめくると音声で指示を出してプレゼンを制御している。

4.3 Introduction to eeePC SDK

Chih Wei Huang

赤ちゃんでもつかえるくらいシンプルだよ、と赤ちゃんの写真を見せて ice-break。

eeePC900 は 2008 年 4 月、8.9" シリーズとして登場。

eeePC は Debian ベースです。オープンソースですよ。

Xandros open circulation edition というのをつかってます。開発環境は Eclipse です。

Qt4 は今は Nokia の Trolltech が開発。

標準の .deb パッケージを使っている。SDK として、debian policy manual, new maintainer guide とかが紹介。エキスパートがたくさんいるよ、と。

デスクトップにアイコンを追加する方法は簡単だよ。XML ファイルを作成すればよい。

<http://sourceforge.net/projects/eeecomunity/> に VMware のイメージはあるから見てね。ISO を vmware-convert コマンドで変換したら利用できるよ。eeePC 701 のイメージなどがある。

i18n/l10n は、Qt と gettext があるよ。Qt は tr(). lupdate とか使う。gettext は _() で。

Eclipse の Qt 用の開発ツールがあるよ。Qt Designer が Eclipse 内部で動作します。

4.3.1 The rule of game to stand on the shoulders of giant

GPL の呪縛を回避する方法について。

FOSS の裁判案件の紹介。

Busybox 裁判など。

Creative Commons のすすめとか。

質問は、GPL の継承を回避する方法について。dynamic linker と static link の話をしていたような気がする。

4.3.2 Developing with pyGTK in EeePC

?? Open Source Software Foundry Introduction かも?

自由とは。

OSSF の話

Python を使うことについて。python の文法について。インデント重要とか

Gtk とか。

4.3.3 Developing with pyQT in EeePC

PyQt の使用方法についての紹介。

4.3.4 Firefox Platform Application Programming

ハッカー然とした「thinker」による firefox プラットフォームアプリケーションプログラミングの紹介。

「OK」としか表示しない mozilla アプリケーションの例の紹介。XUL。

時間オーバーだということで途中でできあげられてしまいました。

4.3.5 Google Gears Programming

Google Gears を利用して開発する方法について紹介しています。

4.4 eeePC Developers Conference day 2

5月9日に2日目が開催されました。

当初の発表されていた予定から大幅に変更されて実施されました。

09 : 00	09 : 30	Register	
09 : 30	10 : 00	Eclipse,Gambas,Lazarus Development Environment Introduction	ASUS & III Free Software Team
10 : 00	10 : 50	Adobe Air Platform Programming	Anistar Sung
10 : 50	11 : 00	Tea Break	
11 : 00	12 : 00	Eee PC Hacker Show	ASUS SAM
12 : 00	13 : 30	Lunch	
13 : 30	14 : 00	EeePC HotKey、WebCAM、Audio Control	ASUS & III Free Software Team
14 : 00	14 : 15	Introduction of Open Source Development Resource and Discussion Forum	III Free Software Team
14 : 15	14 : 30	Open Source Software Foundry Introduction	Tim Wu, OSSF
14 : 30	15 : 30	Debian Project and the Development Process, and how to co-work with it	Junichi Uekawa
15 : 30	15 : 40	Q & A	
15 : 40	16 : 00	Tea Break	
16 : 00	17 : 00	Linux Driver Project	Brandon D. Philips
17 : 00	17 : 10	Q & A	

4.4.1 GTK?

10 時ころについたら何か違う人が話をしていた。

4.4.2 Eclipse,Gambas,Lazarus Development Environment Introduction

「IDEAS III」

MonoDevelop

Lazarus は PASCAL のツール

なんか途中かなと思っていたら話が終わっていた。

4.4.3 LXDE の紹介

pcman による LXDE の紹介?

起動がすごく高速、リソースも全然使わないというデモ。ファイルマネージャをクリックしまくって起動してもスムーズだぜ!

計算機とかも便利!

タスクバーとか、パネルとか。

4.4.4 Deb?

4.4.5 Array 30 入力メソッド

新しい input method を作った。アレーベースの方法。

GCIN OXIM という古いもの、SCIM に。現在 SCIM は日本語とコリアンの入力に必須。

で、頑張っているいろと入力してみる。

4.4.6 EeePC HotKey

ホットキーのプログラムの書き方について紹介。GDK で hk のコールバックがあります、という紹介。

4.4.7 eeePC audio, webcam

オーディオは ALSA です、Wiki を見てください。

webcam の制御は、unicapGTK でやります。ucil はビデオに文字とかを載せるための API です。

unicapGTK の開発の話。

4.4.8 Open Source Software Foundry Introduction

フリーソフトウェアを支援する環境の紹介。

フリーソフトウェアを工業的統合を経たら製品になる素材と位置づけて推進していく。

4.4.9 Adobe Air Platform Programming

AIR プラットフォームの紹介。RIA のための AIR。

4.5 OSC 2008 Hokkaido

OSC 2008 Hokkaido に Git の話をしてください、と呼ばれたので Debian 勉強会として行ってきました。最近流行っている SCM Git ってどうよ？というネタで、Git の魅力と簡単な使い方についてお話をさせていただきました。15 人の登録でしたが、実際は 30 人ぐらいの参加でした。45 分では基本的なところしか話せないで、うまく Git の良さを伝えることができなかったかもしれません。けど、基本的な使い方は分かるように説明したつもりなので、感想とかあれば教えてください。RubySapporo の方に ustream と録画をしてもらいましたが、うまく撮れてなかったとのこと。残念。北海道の方や Ruby の高橋さんとお話する機会があり、勉強会をどのように盛り上げていくのか、意見交換をしました。が、酔っ払っていたのであまり覚えていません。

5 Linux カーネルパッチ の Debian パッケージ作成

岩松 信洋



5.1 はじめに

テスト段階の機能が Linux カーネルへのパッチとして公開されている場合がありますが、これらの パッチを Debian で利用したい場合には、Debian で配布している カーネルパッケージや vanilla カーネルにパッチを当ててパッケージ化されている方も多いのではないのでしょうか。Debian の特徴の一つとしてパッケージ管理が挙げられますが、実は Linux カーネルパッチも Debian パッケージとして管理することができ、既にいくつかのパッケージが利用可能になっています。また、この Linux カーネルパッチパッケージを作成するためのサポートパッケージ `dh-kpatches` が提供されており、このパッケージを利用することによりパッチ用パッケージの作成や、カーネルへのパッチ適用およびコンパイルが可能になっています。今回は Evgeniy Polyakov (えぶじえにー ぼるやこふ) が作成している新しい Network Filesystem POHMEIFS^{*3} を題材にして、Linux カーネル向けパッチのパッケージ作成方法とテスト方法について説明します。

5.2 Linux カーネルパッチ の Debian パッケージの仕組み

Linux カーネルパッチパッケージは、カーネルパッチと パッチをコントロールするためのファイル `kpatches` を提供します (図 1)。このファイルでは、アーキテクチャ、カーネルバージョンの指定が可能になっています。そして、パッケージ作成時に `kpatches` ファイルから `patch/unpatch` 時に使用される パッチコントロールスクリプトが生成されます。パッケージをインストールすると、`/usr/src/kernel-patches` ディレクトリに各ファイルが展開されます。Debian カーネルパッケージを作成するためのコマンド `make-kpkg` の `-added-patches` オプションで 指定されたパッチパッケージ名を このディレクトリから検索し、パッチを適用してから、カーネルをコンパイルします。

5.3 パッケージ作成前の準備

5.3.1 dh-kpatches パッケージのインストール

`make-kpkg` が利用できる Linux カーネルパッチパッケージを作成するには、`dh-kpatches` パッケージが必要です。`dh-kpatches` を使って、パッケージ化を行うことにより、`kernel-package` で提供されている `make-kpkg` コマンドでカーネルパッチパッケージで提供しているパッチを適用した Linux カーネルパッケージが作成できるようになります。パッケージ作成サポート用として、`dh-make` をインストールしておくとい良いでしょう。`dh-make` を使うことによって、パッケージの雛形を容易に作成することができます。

^{*3} <http://tservice.net.ru/~s0mbre/old/?section=projects&item=pohmelfs>

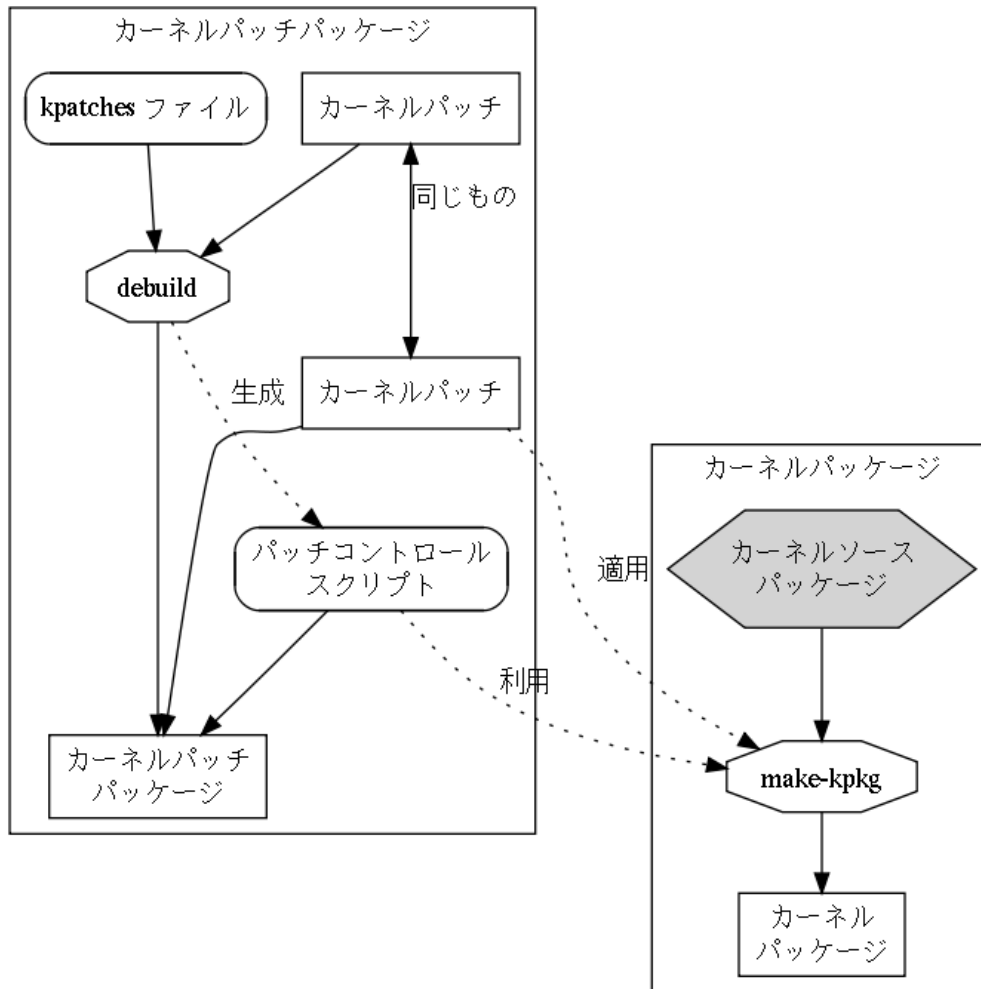


図 1 Linux カーネルパッチ の Debian パッケージの仕組み

```
$ sudo apt-get update
$ sudo apt-get install dh-kpatches dh-make build-essential
```

5.3.2 パッチの用意

まず、カーネル向けのパッチを作成する必要があります。開発者によっては、既にパッチがカーネルバージョン毎に用意されている事もありますが、最近では、Linux のツリーに容易に追従できるように、Git を使ってソースコードが管理されている場合が多いです。POHMELFS でも ソースコードは Git で管理されており、安定版のカーネル（この原稿を書いている時点では、Linux 2.6.25）に常に追従されています。この差分を取得するには、Git リポジトリを取得し、git diff コマンド等で取り出せばよいでしょう。

```
$ git clone http://tsservice.net.ru/~sOmbre/archive/pohmelfs/pohmelfs.git
Initialized empty Git repository in /tmp/pohmelfs/.git/
got 37e1b82c0535386cf09b3821dff5e8cb5f9e26b4
walk 37e1b82c0535386cf09b3821dff5e8cb5f9e26b4
<snip>
$ cd pohmelfs
$ git tag
<snip>
v2.6.24-rc8
v2.6.25
v2.6.25-rc1
v2.6.25-rc2
<snip>
$ git diff v2.6.25 > ~/pohmelfs.diff
```

5.3.3 ディレクトリの作成

パッチが作成できたら、パッケージ作成用のディレクトリを作成し、その中にパッチファイルをコピーします。ディレクトリ名は Debian のポリシーに合わせたもの (ソフトウェアまたは機能名-バージョン) にしておき、パッチファイル名はパッチ機能名-カーネルバージョン にしておきます。これは、このようなファイル名を `dh-kpatches` が要求するためです。

```
$ mkdir linux-patch-pohmelfs-20080707
$ cd linux-patch-pohmelfs-20080707
$ cp ../pohmelfs.diff pohmelfs-2.6.x
```

5.4 dh_make を使った雛形の作成

パッケージ作成の準備ができたなら、`dh_make` コマンドを使って雛形を作成します。`dh_make` にはカーネルパッチ用のサポートはないので、`single` バイナリの雛形を作成するオプションを指定して、作成します。`-r` オプションは `orig.tar.gz` ファイルを作成し、`-s` オプションはシングルバイナリ用の雛形を出力します。

```
$ dh_make -r -s
Maintainer name : Nobuhiro Iwamatsu
Email-Address   : iwamatsu@nigauri.org
Date            : Mon, 07 Jul 2008 01:00:33 +0900
Package Name    : linux-patch-pohmelfs
Version         : 20080707
License         : blank
Using dpatch    : no
Type of Package : Single
Hit <enter> to confirm:
Currently there is no top level Makefile. This may require additional tuning.
Done. Please edit the files in the debian/ subdirectory now. You should also
check that the linux-patch-pohmelfs Makefiles install into $DESTDIR and not in / .
```

5.5 debian ディレクトリ内の変更

`dh_make` コマンドでカーネルパッチを Debian パッケージにするための雛形が作成できました。これを元に内容を変更していきます。

5.5.1 サンプルスクリプト等の削除

`dh_make` で作成されるサンプルスクリプトは必要ないので全て削除します。また、`dirs` ファイルや `doc` ファイルも必要ないため削除します。

```
$ rm -rf ./debian/*.ex ./debian/*.EX ./debian/docs ./debian/dirs
```

5.5.2 debian/copyright の修正

パッチの情報に合わせて、`debiab/copyright` の修正を行います。`pohmelfs` の場合は以下のように修正します。

```

This package was debianized by Nobuhiro Iwamatsu <iwamatsu@nigauri.org> on
Mon, 07 Jul 2008 01:00:33 +0900.

It was downloaded from
  http://tsservice.net.ru/~s0mbre/archive/pohmelfs/pohmelfs.git
This is Git repository. I made the difference of the latest committing a
patch from v2.6.25 tag.

Upstream Author(s):

  Evgeniy Polyakov <johnpol@2ka.mipt.ru>

Copyright:

  Copyright (C) 2008 Evgeniy Polyakov <johnpol@2ka.mipt.ru>

License:

  GPLv2

The Debian packaging is (C) 2008, Nobuhiro Iwamatsu <iwamatsu@nigauri.org> and
is licensed under the GPL, see '/usr/share/common-licenses/GPL'.

# Please also look if there are files or directories which have a
# different copyright/license attached and list them here.

```

5.5.3 debian/README.Debian の修正

debian/README.Debian ファイルに Debian で使うための HowTo などを書いておきましょう。サポートしているカーネルバージョンや、カーネルイメージの作成方法などを書いておくのが一般的なようです。

```

linux-patch-pohmelfs for Debian
-----

This patch is pohmelfs support patch for Debian Linux kernel.
You can try with linux-source-2.6.25 package.

- How to use
$ sudo apt-get install linux-source-2.6.25 libncurses-dev kernel-package
$ cd /usr/src ; tar -xjf linux-source-2.6.25.tar.bz2 ; cd linux-source-2.6.25
$ make-kpkg clean
$ cp /boot/config-2.6.25-2-686 .config
$ make menuconfig
$ make-kpkg --rootcmd fakeroot --append-to-version -pohmelfs --revision 0.1 -added_patches=pohmelfs kernel-image

-- Nobuhiro Iwamatsu <iwamatsu@nigauri.org> Mon, 07 Jul 2008 01:00:33 +0900

```

5.5.4 debian/control の修正

次に debian/control ファイルを修正します。注目すべきところは、作成されるパッケージで指定してある、**Depends: \${kpatch:Depends}** です。ここで kpatch:Depends を指定することによって、カーネルパッチパッケージを使ったカーネル作成に必要な依存パッケージを置き換えてくれます。また、ソースパッケージの **Build-Depends** に **dh-kpatches** パッケージを追加する事を忘れないようにしましょう。

```

Source: linux-patch-pohmelfs
Section: devel
Priority: extra
Maintainer: Nobuhiro Iwamatsu <iwamatsu@nigauri.org>
Build-Depends: debhelper (>= 6), dh-kpatches
Standards-Version: 3.8.0.1
Homepage: http://tsservice.net.ru/~s0mbre/old/?section=projects&item=pohmelfs

Package: linux-patch-pohmelfs
Architecture: all
Depends: ${kpatch:Depends}
Description: POHMELFS kernel patch
 POHMELFS stands for Parallel Optimized Host Message Exchange Layered File System.
 Development status can be tracked in filesystem section.
 This is a high performance network filesystem with local coherent cache of data
 and metadata.
 Its main goal is distributed parallel processing of data. Network filesystem is a
 client transport.
 POHMELFS protocol was proven to be superior to NFS in lots (if not all, then it
 is in a roadmap) operations.

```

5.5.5 debian/changelog の修正

dch コマンドを使って debian/changelog ファイルを修正します。

```
$ dch
linux-patch-pohmelfs (20080707-1) unstable; urgency=low

* Initial release

-- Nobuhiro Iwamatsu <iwamatsu@nigauri.org> Mon, 07 Jul 2008 01:25:37 +090
```

5.5.6 kpatches ファイル

どのパッケージ内に収録されているパッチをどのカーネルバージョンに当てればいいのかなどのコントロールするためのファイル `kpatches` を用意する必要があります。このファイルは以下のようなフォーマットになっており、カーネルバージョン毎に `Patch-file` と `Kernel-version` の項目を追加する必要があります。また、このファイルを パッケージ名 `.kpatches` として 名前を変更する必要があります。

```
Patch-name: POHMELFS patch
Patch-id: pohmelfs <-- make-kpkg の --added-patches で指定するパッチ名
Architecture: all <-- サポートするアーキテクチャ
Path-strip-level: 1

Patch-file: pohmelfs-2.6.x <-- パッチファイル名
Kernel-version: 2.6.25 <-- パッチがサポートするカーネルバージョン
```

5.5.7 debian/rules ファイルの修正

`debian/rules` ファイルで注意する点として、`install` ターゲットで、`dh_installkpatches` を指定する必要があります。`dh_installkpatches` でカーネルパッチが `kpatches` ファイルと共にパッケージ作成用の一時ディレクトリにコピーされます。

```
#!/usr/bin/make -f

build: build-stamp
build-stamp:
    dh_testdir
    touch build-stamp

clean:
    dh_testdir
    dh_testroot
    rm -f build-stamp
    dh_clean

install: build
    dh_testdir
    dh_testroot
    dh_clean -k
    dh_installdirs
    dh_installkpatches

# Build architecture-independent files here.
binary-indep: build install
    dh_testdir
    dh_testroot
    dh_installchangelogs
    dh_link
    dh_strip
    dh_compress
    dh_fixperms
    dh_installdeb
    dh_shlibdeps
    dh_gencontrol
    dh_md5sums
    dh_builddeb

# Build architecture-dependent files here.
binary-arch: binary-indep

# We have nothing to do by default.

binary: binary-indep binary-arch
```

5.5.8 ディレクトリ構成

最終的な `debian` ディレクトリの構成をチェックしてみます。以下のようになっているとよいでしょう。

```
linux-patch-pohmelfs-20080707/
|-- debian
|   |-- README.Debian
|   |-- changelog
|   |-- compat
|   |-- control
|   |-- copyright
|   |-- linux-patch-pohmelfs.kpatches
|   |-- rules
|-- pohmelfs-2.6.x
```

5.6 パッケージのビルドおよびインストール

パッケージのビルドを行う際には、通常のパッケージ作成と変わりません。debuild などを使ってパッケージの作成を行ってください。作成されたパッケージをインストールすると、`/usr/src/kernel-patches` にパッチと制御用のスクリプトがインストールされます。make-kpkg コマンドはこのディレクトリを参照し、パッチを適用します。

```
% debuild
% sudo dpkg -i ../linux-patch-pohmelfs_20080707-1_all.deb
```

5.7 パッケージのテスト

カーネルパッチパッケージのテストは、カーネルソースコードにパッチを当て、カーネルがコンパイルできるか、確認する必要があります。また、Debian の場合は、Debian と kernel.org で配布しているカーネルの 2 種類を考慮する必要がありますが、前者の方をテストしていれば十分でしょう。インストールしたカーネルパッチパッケージを使って、カーネルをコンパイルするには、make-kpkg コマンドの `-added_patches` のオプションを使って、パッチを指定します。パッチが当たっているかソースを確認し、カーネルのコンパイルが正常に行われているか確認しましょう。また、作成された カーネルパッケージを実際にインストールし、テストを行うことも重要です。

```
$ sudo apt-get install linux-source-2.6.25 libncurses-dev kernel-package
$ cd /usr/src ; tar -xjf linux-source-2.6.25.tar.bz2 ; cd linux-source-2.6.25
$ make-kpkg clean
$ cp /boot/config-2.6.25-2-686 .config
$ make menuconfig
$ make-kpkg --rootcmd fakeroot --append-to-version -pohmelfs --revision 0.1 -added_patches=pohmelfs kernel-image
```

5.8 パッケージのアップデート方法

いまのところ、カーネルパッチソースパッケージのアップデート方法が決まっていません。適当なディレクトリを作成して、uupdate コマンドを実行するのが一番容易と思われます。将来的には、パッチを指定することによって、ソースパッケージをアップデートできるようにしたいと考えています。

5.9 まとめ

以上の手順を使うと、Linux カーネルパッチの Debian パッケージを作成することができます。しかし、このままではかなり手間がかかりますので、今回の成果物をまとめ、dh-make で Linux カーネルパッチパッケージの雛形が作成できるように機能を追加しました (#304688)。このパッチが取り込まれると、dh-make を使うことにより、より簡単に Linux カーネルパッチパッケージが作成できるようになるでしょう。また、cdbs を使った方法をまだ調べていないので、調べようと思っています。ちなみに、pohmelfs は linux-2.6.27 or 2.6.28 に取り込まれるようです。よかったよかった。

6 Linux カーネルモジュールの Debian パッケージ作成

岩松 信洋



6.1 はじめに

今回は video データ loopback 用 Linux カーネルモジュール、vloopback を題材にして、Linux カーネルモジュールの Debian パッケージ作成方法を説明します。

6.2 なぜカーネルモジュールソースコードをパッケージ化するのか

Linux の カーネルモジュールソースコードをパッケージ化する理由の一つとして、カーネルバージョンに合わせたドライバを管理する手間が省ける事とモジュールドライバコンパイルフロントエンド module-assistant による 操作のしやすさが大きいでしょう。このソフトウェアを使うことによって、容易にモジュールのコンパイルおよびインストールを行う事ができます。現時点では、カーネルバージョンが上がる度にコンパイルする必要がありますが、パッケージ化しておく、ある程度まで自動化できるため、アップデートも容易になります。

6.3 カーネルモジュールパッケージの仕組み

パッケージの作り方の説明の前にカーネルモジュールパッケージの仕組みについて説明します (図 2)。カーネルモジュールパッケージは 内部に カーネルモジュールソースコードとコンパイルするための Makefile(debian/rules) と、control ファイル (debian/control に相当するもの) を固めたものである ドライバソースイメージを持っています。このドライバソースイメージは、実際は Debian ソースパッケージと構成は同じになっており、 bzip2 で圧縮したものになります。カーネルモジュールパッケージをインストールすると、/usr/src/に ドライバソースイメージが置かれ、module-assistant がこのファイルを/usr/src/modules ディレクトリ以下、ビルド毎に展開し、ドライバ用の Debian パッケージビルドを行います。作成された パッケージは /usr/src/に置かれます。モジュールソースパッケージは、このドライバソースイメージを作成し、module-assistant と連携できる機能を提供します。

6.4 ドライバソースコードの取得と展開

vloopback は flashcam という フリーソフトウェアと共に提供されています。今回は、vloopback のソースコードだけをターゲットにするので、ソースコードを取り出します。

```
$ wget http://www.swift-tools.net/Flashcam/flashcam-1.1.tgz
$ tar -xzf flashcam-1.1.tgz
$ cp -rf flashcam-1.1/vloopback-1.1.2 .
$ cd loopback-1.1.2
```

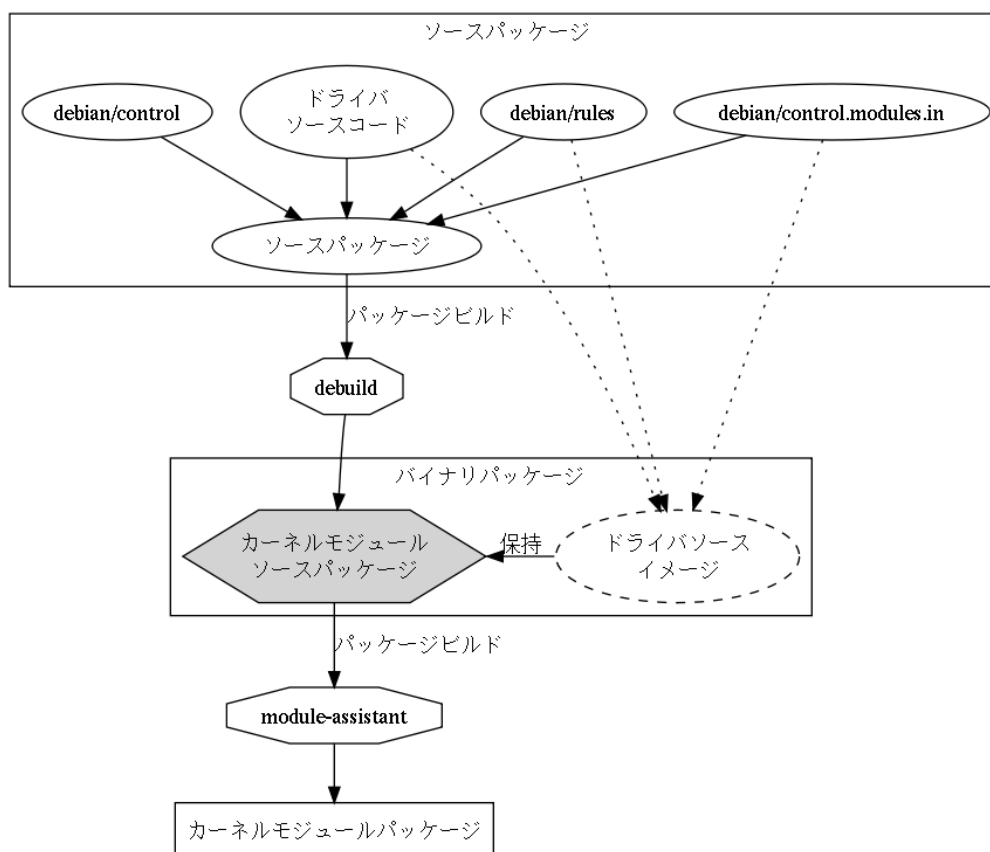


図 2 カーネルモジュールパッケージの仕組み

6.5 dh_make を使った雛形の作成

パッケージ作成の準備ができたなら、dh_make コマンドを使って雛形を作成します。dh_make には、カーネルモジュールの雛形を作成するオプション -k / -kmod が提供されています。

```
$ dh_make -k -r
Maintainer name : Nobuhiro Iwamatsu
Email-Address   : iwamatsu@nigauri.org
Date            : Tue, 15 Jul 2008 23:21:23 +0900
Package Name    : vloopback
Version         : 1.1.2
License         : blank
Using dpatch    : no
Type of Package : Kernel Module
Hit <enter> to confirm:
Done. Please edit the files in the debian/ subdirectory now. You should also
check that the vloopback Makefiles install into $DESTDIR and not in / .
```

6.6 debian ディレクトリ内の変更

Debian パッケージにするための雛形が作成できたので、これを元に内容を変更していきます。

6.6.1 サンプルスクリプト等の削除

dh_make で作成されるサンプルスクリプトは必要ないので全て削除します。

```
$ rm -rf ./debian/*.ex ./debian/*.EX
```

6.6.2 debian/copyright の修正

パッチの情報に合わせて、debian/copyright の修正を行います。vloopback の場合は以下のように修正します。

```
This package was debianized by Nobuhiro Iwamatsu <iwamatsu@nigauri.org> on
Tue, 15 Jul 2008 23:21:23 +0900.

It was downloaded from <http://www.swift-tools.net/Flashcam/flashcam-1.1.tgz>

Upstream Author(s):

    Olivier Debon <olivier@debon.net>

Copyright:

    Copyright (C) 2008 Olivier Debon <olivier@debon.net>

License:

    GPLv2

The Debian packaging is (C) 2008, Nobuhiro Iwamatsu <iwamatsu@nigauri.org> and
is licensed under the GPL, see '/usr/share/common-licenses/GPL'.

# Please also look if there are files or directories which have a
# different copyright/license attached and list them here.
```

6.6.3 debian/README.Debian の修正

debian/README.Debian ファイルを修正します。雛形がある程度書いておいてくれるので、あまり修正する必要はありません。確認した Linux カーネルバージョンなどを書いておくとも良いかもしれません。

```
vloopback for Debian
-----

Please see ./README for a description of the vloopback software.

The Debian vloopback source package provides two packages,

    vloopback-source, which provides the source for the kernel modules

The vloopback-source package can be used in several ways,

- Using the make-kpkg(1) command provided by the kernel-package Debian
  package. This will produce a corresponding vloopback-modules package for
  the Debian kernel-image package that you are using. This is "the Debian
  way". See the "modules_image" section of the make-kpkg(1) man page.

- Changing to the /usr/src/modules/vloopback/ directory and building as
  the README file instructs using "make; make install". This will build
  and install a module specific to the system you are building on and is
  not under control of the packaging system.

-- Nobuhiro Iwamatsu <iwamatsu@nigauri.org> Tue, 15 Jul 2008 23:21:23 +0900
```

6.6.4 debian/control の修正

次に debian/control ファイルを修正します。dh_make で作成した場合は、vloopback-utils といった、ドライバをサポートするためのパッケージを作成するエントリがあります。ソースコードにツールが含まれている場合は、このエントリを使って、ツール用のパッケージを作成できるようになっています。今回は必要ないので削除します。

```
Source: vloopback
Section: graphics
Priority: extra
Maintainer: Nobuhiro Iwamatsu <iwamatsu@nigauri.org>
Build-Depends: debhelper (>= 7), bzip2
Standards-Version: 3.8.0.1
Homepage: http://www.swift-tools.net/Flashcam/

Package: vloopback-source
Architecture: all
Depends: module-assistant, debhelper (>= 7), make, bzip2
Description: Source for the vloopback driver.
 This package provides the source code for the vloopback kernel modules.
 The vloopback package is also required in order to make use of these
 modules. Kernel source or headers are required to compile these modules.
```

6.6.5 debian/changelog の修正

dch コマンドを使って debian/changelog ファイルを修正します。

```
$ dch
vloopback (1.1.2-1) unstable; urgency=low

* Initial release

-- Nobuhiro Iwamatsu <iwamatsu@nigauri.org> Tue, 15 Jul 2008 23:21:23 +0900
```

6.6.6 control.modules.in ファイルの修正

カーネルモジュールソースパッケージは 2 つの control ファイルを持ちます。一つは、カーネルモジュールソースパッケージ用、もう一つは カーネルモジュールパッケージ用のカーネルモジュールソースパッケージ用の debian/control ファイルが control.modules.in です。ドライバパッケージ作成時にソースパッケージからカーネルモジュールソースパッケージが作成され、ユーザはカーネルモジュールソースパッケージをコンパイルし、カーネルモジュールパッケージを作成し、インストールして利用します。このファイルの中にはカーネルのバージョンによって置換される文字列が `_KVERS_` という文字列になっています。control.modules.in は module-assistant などにより処理され、各 Linux カーネルバージョン用の内容に変更されます。

```
Source: vloopback
Section: graphics
Priority: optional
Maintainer: Nobuhiro Iwamatsu <iwamatsu@nigauri.org>
Build-Depends: debhelper (>= 7)
Standards-Version: 3.8.0.1

Package: vloopback-modules-_KVERS_
Architecture: any
Provides: vloopback-modules
Description: vloopback modules for Linux (kernel _KVERS_).
 This package contains the set of loadable kernel modules for the
 <description>.
.
 This package contains the compiled kernel modules for _KVERS_
.
If you have compiled your own kernel, you will most likely need to build
your own vloopback-modules. The vloopback-source package has been
provided for use with the Debian's module-assistant or kernel-package
utilities to produce a version of vloopback-modules for your kernel.
```

6.6.7 debian/rules ファイルの修正

dh_make で作成された、debian/rules ファイルは パッケージ作成用のターゲット (build/install/clean) と、ドライバパッケージ作成用のターゲット binary-modules/kdist_clean が用意されています。これらのターゲットの中を修正します。

```

#!/usr/bin/make -f

psource:=vloopback-source
sname:=vloopback

# prefix of the target package name
PACKAGE=vloopback-modules
# modifieable for experiments or debugging m-a
MA_DIR ?= /usr/share/modass
# load generic variable handling
-include $(MA_DIR)/include/generic.make
# load default rules, including kdist, kdist_image, ...
-include $(MA_DIR)/include/common-rules.make

kdist_config: prep-deb-files
kdist_clean: clean
    $(MAKE) $(MFLAGS) -f debian/rules clean

configure: configure-stamp
configure-stamp:
    dh_testdir
    # Add here commands to configure the package.

    touch configure-stamp

build-arch: configure-stamp build-arch-stamp
build-arch-stamp:
    dh_testdir

    # Add here command to compile/build the package.
    # $(MAKE)

    touch $$@

#k = $(shell echo $(KVERS) | grep -q ^2.6 && echo k)

# during a normal build
binary-modules:
    dh_testroot
    dh_clean -k
    dh_installdirs lib/modules/$(KVERS)/misc

    # Build the module
    $(MAKE) KVER=$(KVERS) KSRC=$(KSRC)

    # Install the module
    cp vloopbackko debian/$(PKGNAME)/lib/modules/$(KVERS)/misc

    dh_installdocs
    dh_installchangelogs
    dh_compress
    dh_fixperms
    dh_installdeb
    dh_gencontrol -- -v$(VERSION)
    dh_md5sums
    dh_builddeb --destdir=$(DEB_DESTDIR)
    dh_clean -k

```

```

build-indep: configure-stamp build-indep-stamp
build-indep-stamp:
    dh_testdir
    touch $@

build: build-arch build-indep

clean:
    dh_testdir
    rm -f build-arch-stamp build-indep-stamp configure-stamp
    dh_clean

install: DH_OPTIONS=
install: build
    dh_testdir
    dh_testroot
    dh_clean -k
    dh_installdirs

    # Create the directories to install the source into
    dh_installdirs -p$(psource)  usr/src/modules/$(sname)/debian

    # Copy only the driver source to the proper location
    cp vloopback.c  debian/$(psource)/usr/src/modules/$(sname)/.
    # Copy the needed debian/ pieces to the proper location
    cp debian/*modules.in* \
        debian/$(psource)/usr/src/modules/$(sname)/debian
    cp debian/rules debian/changelog debian/copyright \
        debian/compat debian/$(psource)/usr/src/modules/$(sname)/debian/
    cd debian/$(psource)/usr/src && tar c modules | bzip2 -9 > $(sname).tar.bz2 && rm -rf modules

    # Add here commands to install the package into debian/vloopback.
    # $(MAKE) DESTDIR=$(CURDIR)/debian/vloopback install

    dh_install

# Build architecture-independent files here.
# Pass -i to all debhelper commands in this target to reduce clutter.
binary-indep: build install
    dh_testdir -i
    dh_testroot -i
    dh_installchangelogs -i
    dh_installdocs -i
    dh_installexamples -i
    dh_installman -i
    dh_link -i
    dh_compress -i
    dh_fixperms -i
    dh_installdeb -i
    dh_gencontrol -i
    dh_md5sums -i
    dh_builddeb -i

# Build architecture-dependent files here.
binary-arch: build install
    dh_testdir -s
    dh_testroot -s
    dh_installdocs -s
    dh_installinfo -s
    dh_installchangelogs -s
    dh_compress -s
    dh_fixperms -s
    dh_installdeb -s
    dh_shlibdeps -s
    dh_gencontrol -s
    dh_md5sums -s
    dh_builddeb -s

binary: binary-indep binary-arch
.PHONY: build clean binary-indep binary-arch binary install configure binary-modules kdist kdist_configure kdist_image kdist_clean

```

6.7 パッケージのビルドおよびインストール

パッケージのビルドを行う際には、通常のパッケージ作成と変わりません。debuild などを使ってパッケージの作成を行ってください。作成されたパッケージをインストールすると、`/usr/src/modules` にドライバソースイメージがインストールされます。

```

% debuild
% sudo dpkg -i ../vloopback-source_1.1.2-1_all.deb

```

6.8 モジュールパッケージの作成

6.8.1 module-assistant を使ったコンパイル

ドライバパッケージのコンパイルには module-assistant コマンドを利用します。このコマンドは GUI を持っており、GUI でドライバのコンパイルやコンパイル環境のアップデートを行う事ができます。GUI で選択できるドライバ群は module-assistant パッケージ内で持っているリスト (/usr/share/modass/compliant.list) のみが対象になっているため、新しく作成したものは選択できません。このような場合は、モジュール名を指定することによってコンパイル可能です。

```
$ sudo m-a build vloopback
Extracting the package tarball, /usr/src/vloopback.tar.bz2, please wait...
/usr/bin/make clean
make[1]: ディレクトリ '/usr/src/modules/vloopback' に入ります

<snip>

make[3]: ディレクトリ '/usr/src/linux-headers-2.6.25-2-686' に入ります
CC [M] /usr/src/modules/vloopback/vloopback.o
Building modules, stage 2.
MODPOST 1 modules
CC /usr/src/modules/vloopback/vloopback.mod.o
LD [M] /usr/src/modules/vloopback/vloopback.ko
make[3]: ディレクトリ '/usr/src/linux-headers-2.6.25-2-686' から出ます
make[2]: ディレクトリ '/usr/src/modules/vloopback' から出ます
# Install the module
cp vloopback.ko debian/vloopback-modules-2.6.25-2-686/lib/modules/2.6.25-2-686/misc
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol -- -v1.1.2-1+2.6.25-6
dh_md5sums
dh_builddeb --destdir=/usr/src
dpkg-deb: '/usr/src/vloopback-modules-2.6.25-2-686_1.1.2-1+2.6.25-6_i386.deb' にパッケージ 'vloopback-modules-2.6.25-2-686' を構築しています。
dh_clean -k

<snip>
```

6.8.2 make-kpkg を使ったコンパイル

modules-assistant 以外にも make-kpkg の `-added-modules` オプションを使うことによってコンパイルすることができます。しかし、make-kpkg の場合にはライバソースイメージを展開してくれないので、手動で展開する必要があります。kernel-package はカーネルパッケージ専用と考えたほうがよさそうです。(そのために modules-assistant ができたので。)

6.9 まとめ

今回は dh_make からの作成でしたが、cdbs を使うともっと簡単かつ簡潔に作成することができます。過去の勉強会資料にちょっとだけ載っているのですが、参考にしてください。しかし、カーネルドライバ側の debian/rules は cdbs での対応が行われていないので、対応する必要があるでしょう。あと、最近はドライバも linux ツリーに取り込まれやすくなったのですが、Linux のリリース間隔などを考えると、新しいドライバは簡単に使えるものではありません。新しいドライバを早く使いたい人には、ドライバパッケージ群が重要なパッケージになってくると思います。そのためには、今後はこれらのパッケージをどのようにメンテナンスしていくかなどについて考えていく必要がありそうです。

6.10 おまけ

module-assistant でサポートされているパッケージがどれくらいコンパイルできるのか調べてみました。アーキテクチャは i386、ディストリビューションは sid です。パッケージは /usr/share/modass/compliant.list にかかっているものです。結果は 82 個中、コンパイル OK: 22/ コンパイル NG: 22/ パッケージなし: 38 になりました。パッケージなしなのは、module-assistant のリストがメンテナンスされていなかからだと思います。あとは、アーキテク

チャ非対応など。既にカーネルにマージされているものがけっこうあり、コンパイルエラーは各メンテナがサボっているでしょう。これらは今後 BTS していくつもりです。

月例 Nexenta Operating System 上川 純一

先月につづいて Nexenta をいじって見たのでお伝えします。

実用的なアプリケーションをビルドするまでに前は至りませんでした。そこで、今回は `dlsym` を使って自由にライブラリをロードできるようにしてみるところからまずやってみましょう。

`dlopen` でロードできる最低限の共有ライブラリを作成するところから始めてみましょう。

まず、関数一つだけを定義した C のファイルから最低限の共有ライブラリを作成してみ、それを実行するだけのプログラムと、`dlopen` 経由で利用するプログラムを作成してみました。

```
// a.c : サンプルの共有ライブラリのコード
#include <stdio.h>

void func1 (int i)
{
    printf(" hello world %i\n", i);
}
```

```
// use.c: 通常の共有ライブラリ利用
#include <stdio.h>

void func1 (int i);

main()
{
    func1(1);
}
```

```
// dlopen.c: dlopen でのバイナリ利用
#include <dlfcn.h>
static int (*myfunc1)(int i) = NULL;
int main()
{
    void* h=dlopen("./liba.so", RTLD_NOW);

    myfunc1=dlsym(h, "func1");

    myfunc1(10);

    return 0;
}
```

これらをコンパイル、リンクしてみて、動作を確認してみました。どうやら Linux とあまりかわらないようです。

```
+ gcc -shared a.c -o liba.so
+ gcc use.c ./liba.so -o use
+ ./use
hello world 1
+ gcc dlopen.c -o dlopen
+ ./dlopen
hello world 10
```

ただし、これだけだとバージョンシンボルを利用した場合に `dlsym` ではどのバージョンを利用するのが指定できないような雰囲気がただよっています。というところまで確認したところで今月も力尽きました。また来月続きをながめてみましょう。もしかするとバージョンシンボルの作成とその利用をするかもしれません。



Debian 勉強会資料

2008 年 7 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）
