

東京エリア デビアン 勉強会

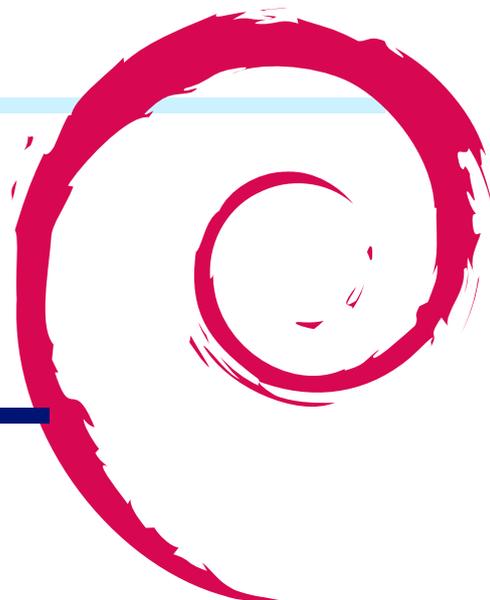


Debian勉強会幹事 上川純一

2009年12月12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 をがりがりとす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。

2009 年の計画は仮です。

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(岩松さん)
11. 3D graphics 開発 (藤沢さん)
12. Debian サーバ + VMware + 各種 OS、他の仮想化ツール (vserver etc.)、忘年会

会場候補としては下記があります：

- 大学
- 恵比寿 SGI ホール
- Google オフィス
- 公民館 (あんさんぶる荻窪等)
- 都立会議室 (無線 LAN)
- 健保の施設

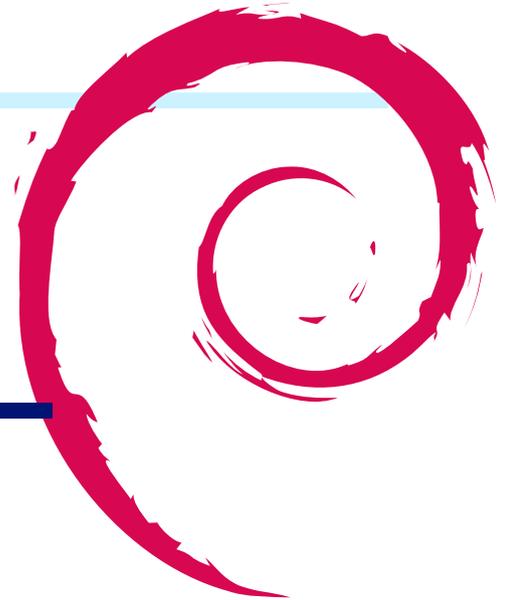
*1 <http://www.library.chiyoda.tokyo.jp/>

目次

1	Introduction	1
2	事前課題	3
2.1	上川純一	3
2.2	吉田俊輔	3
2.3	まえだこうへい	4
2.4	やまねひでき	4
2.5	本庄弘典	5
2.6	キタハラ	5
2.7	日比野 啓	5
2.8	岩松 信洋	6
2.9	吉野与志仁	6
3	最近の Debian 関連のミーティング報告	7
3.1	東京エリア Debian 勉強会 58 回目報告	7
3.2	Adam C Powell IV 迎撃	7
3.3	Dirk Eddelbuettel 迎撃	7
4	東京エリア Debian 勉強会 2009 年度各種イベント開催実績と総括	8
4.1	Debian 勉強会の野望の進捗具合	8
4.2	運営方法	8
4.3	基本的な数値	8
5	関西 Debian 勉強会	
	2009 年度各種イベント開催実績と総括	12
5.1	運営状況	12
5.2	開催実績	13
6	2009 年を振り返ってみる	15
6.1	最近のトレンドと今後の推移	15
7	qemubuilder 2009 年アップデート	17
7.1	qemubuilder の基本コンセプト	17
7.2	qemubuilder の使い方	17
7.3	qemubuilder の課題	17
7.4	qemubuilder の今後の進め方	18
8	lxc コンテナを使ってみる	19
8.1	はじめに	19
8.2	導入してみる	19
8.3	lxc の仕組みを見てみる	25
8.4	まとめ	26
8.5	参考文献	26

2 事前課題

上川 純一



今回の事前課題は以下です:

1. 2009 年を振り返って自分は何をしたか
2. 2009 年を振り返って Debian 関連で世間では何が起きたか

この課題に対して提出いただいた内容は以下です。

2.1 上川純一

2.1.1 2009 年を振り返って自分は何をしたか

2009 年、半分くらいしか Debian 勉強会には参加してません。

DDTSS をこそこそやってましたが、気づいたら今年は DDTSS のサーバがとまったりといろいろと起きたようです。

Debian パッケージをぼちぼちとメンテナンスしていました。月に一回以上は活動できていたと思います。

新しい開発の方向としては、qemubuilder をいろいろといじっていました。ARM 向けの qemu をいじっているのが中心です。これがしたかったのは Android 携帯をゲットしたからで、Android アプリもいくつか書きました。

2.1.2 2009 年を振り返って Debian 関連で世間では何が起きたか

ARM の携帯できる Linux デバイスが一気に増えた気がします。日本初の Android 携帯 HT03A が発売され、Sharp の Netwalker がリリースされました。

Intel CPU の高速化はそこまですすんでいない感じですが、4 コアの CPU がリリースされて良い感じに盛り上がっていると思います。

2.2 吉田俊輔

2.2.1 2009 年を振り返って自分は何をしたか

Debian 関係だと DDTSS でレビューを中心に 180 件程度やっていたようです。DDTSS は休んだりやったり、まったりやっています。イベント関連では 2 月に Linux conf.au に行って、秋は JLS にボランティア参加、KOF 行って関西 Debian の方とキーサインをしたりしていました。あとは例年通り? あんどきゅめんでっどでびあんのゴーストエディター兼売り子として、某所で勉強会資料をまとめて本にして頒布しています。

2.2.2 2009 年を振り返って Debian 関連で世間では何が起きたか

やはり Lenny リリースが一番でしょう。基本的に stable を使うチキンなので、メジャーバージョンアップは大きなニュースです。最近の Debian は定期的にリリースされるので各種ソフトもバージョンが割と新しく、便利に感じ

ています。おかげで使いたいソフトのために、sid 等からバックポートしたり、そのほか tar ボールからビルドする必要等も少なくなり、またその場合の手間も昔 (woody の頃) に比べてかなり減っておりありがたいです。

2.3 まえだこうへい

2.3.1 2009 年を振り返って自分は何をしたか

ヨメに Debian (Sid) を使わせ始めた、Hack Cafe, Debian JP Project の選挙管理委員、Debian 勉強会の運営や、はじめての ITP, DebConf9 参加、はじめての執筆等、ちょこちょこやってきましたが、やろうと思っていたことはなかなかできていないのが現状です。(ITP っ放しにしてしまっているし...。)

来年は、もうすこしマルチタスクで (要領良く) 動けるようにしたい、というかしないといかなあと思う年の瀬でした。

2.3.2 2009 年を振り返って Debian 関連で世間では何が起こったか

OpenBlockS 600 が出ましたね。OpenBlockS 266 ほど面白みはないですが、スペックが上がっても消費電力が低い (5W) のは良いですね。

2.4 やまねひでき

2.4.1 2009 年を振り返って自分は何をしたか

以下で幾つかの項目に分けて振り返ってみます。

パッケージの作成と維持

主にフォントのパッケージについて追加を実施しました。以下が追加されたパッケージです。

- ttf-umefont - 「梅フォント」のパッケージです
- ttf-umeplus - PCLinuxOS の標準フォント umeplus のパッケージです。
- ttf-ipafont/ttf-ipafont-jisx0208/otf-ipafont - IPA フォントです。
- ttf-monapo - mona フォントと IPA フォントの合成フォントです
- ttf-sawarabi-gothic - 「さわらびゴシック」フォントです。
- ttf-misaki - 「美咲フォント」です。
- ttf-kanjistrokeorders - 「漢字の書き順」フォントです。日本語学習者に人気です。

あとは ITP したまま作業が止まっていたり reject されて再調整ができていないのが幾つかありますので、それを片付けていければと思います。

翻訳作業

ウェブについては、Developers News の査読や翻訳作業、それから po-debconf の更新と新規翻訳作業を実施しています。残念ながら i18n.debian.net サーバが正常に稼働していないので、po-debconf の進捗状況は昨年と比べてどの程度上がっているのかは不明ですが、後退はしていないはず (現状で 70% 以上は作業している)。

あとは upstream で何かできれば、と思い、ここ数日ではあるものの GNOME の翻訳更新作業に手を出しています。2.30 リリースの頃 (2010/3 かな?) に慌てずに査読までできれば良いですね。

イベント / 広報系な活動

Debian/Ubuntu 活動関連で記事を少し書かせていただいたり、対談に参加させてもらったりしていました。

- Software Design 2009/07 特集「初心者にはやさしく、ベテランも満足!」Debian GNU/Linux 5.0 (Lenny) をお勧めする理由」
- Software Design 2009/10 「Debian GNU/Linux カンファレンス「Debconf9」取材レポート Debian 開発者

が集うスペイン 14 日間」

- ThinkIT「TOMOYO Linux 徹底解剖 第 3 回：Debian で TOMOYO Linux を使う」
- ascii.jp「行っとけ！Ubuntu 道場」
- アスキーメディアワークス「週刊アスキー別冊 さくさく Ubuntu!」

また、イベントに幾つか参加しました。今年は体調が悪かったこともあり、昨年より少なめです (KOF とか行けなかった...)

- Linux Consortium 10 Years Event !!「オープンソースデスクトップの未来」(2009/01)
- 第 17 回オープンソーステクノロジー勉強会@ GREE Labs (2009/04)
- Ubuntu 9.04 オフラインミーティング (2009/04)
- Open Source Conference Tokyo/Fall (2009/11)

上記の中で、実質的に発表したのは GREE Labs だけでしたので、来年はもう少し体調を整えて、ネタをかませれば、と思います。

Debconf9 参加

ようやく Debconf に参加することができました。が、とりあえず行っただけで終わってるので、もう少し実績を積むことと英語でのコミュニケーションを改善できればと思っています。

2.4.2 2009 年を振り返って Debian 関連で世間では何が起こったか

うーん、世間というとそんなにインパクトは与えられていないかなーと個人的には感じています。

2.5 本庄弘典

2.5.1 2009 年を振り返って自分は何をしたか

仕事が忙しく勉強会も欠席しがちでした。明日から頑張る。

2.5.2 2009 年を振り返って Debian 関連で世間では何が起こったか

SmartQ5 が出ました。

2.6 キタハラ

2.6.1 2009 年を振り返って自分は何をしたか

直接 Debian でなく申し訳ないのですが…。この秋、実家に ADSL 引きまして、Ubuntu インストールしたノート PC を置いてきました。ネットサーフィン専用機ですが、得に文句もなく普通に使っているようです。debian 系 linux ユーザを一人増やしたということで。

2.6.2 2009 年を振り返って Debian 関連で世間では何が起こったか

同件多数かも知れませんが、Lenny リリースですかね。世間へのインパクトは…、私のまわりではあまりなかったような?!

2.7 日比野 啓

2.7.1 2009 年を振り返って自分は何をしたか

- Debian 勉強会で C や Perl 以外の言語にも注目してもらおうと、OCaml や Common Lisp の布教活動をしました。
- Debian 勉強会で Common Lisp ネタの発表をしました。

2.7.2 2009 年を振り返って Debian 関連で世間では何が起こったか

- lenny リリース。会社のマシンもアップデートが必要になって大変忙しかった。
- Debian 勉強会の宴会でのヨタ話もきっかけの一つになって、日本では初の OCaml 特化イベント、OCaml Meeting Tokyo 2009 で開催。みなさんに感謝。

2.8 岩松 信洋

2.8.1 2009 年を振り返って自分は何をしたか

- Debian Developer になった。
Debian Developer になったので、スポンサーを行うようにした。いままでやってもらったことをやるように心がけるようにしている。
- Debian JP Project Leader になった。
なぜか DJPL になっている。
- Debian SH4 Port 復活
Debconf9 にいて 他の Porter と作業ができたのが大きい。この成果物を利用した製品も出たようです。
- 子供が生まれた
子供が生まれたので、生活が一変した。

2.8.2 2009 年を振り返って Debian 関連で世間では何が起こったか

- Lenny がリリースされた
- GPG 鍵 4096R への移行
- glibc から eglibc への移行

2.9 吉野与志仁

2.9.1 2009 年を振り返って自分は何をしたか

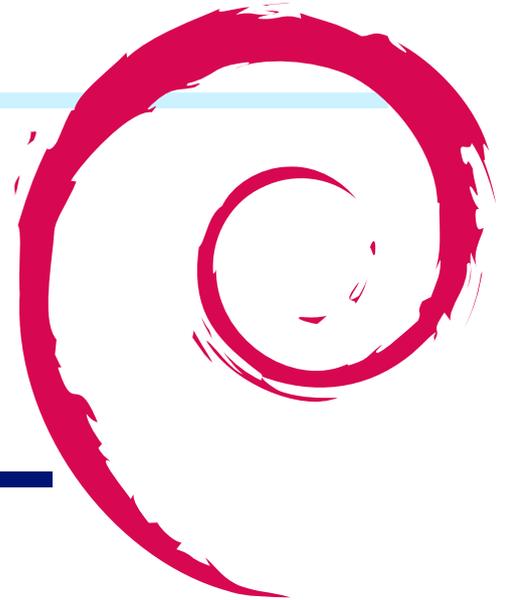
DDTSS は途中サーバが落ちていて滞っていましたが Translations: 165, Reviews: 206 くらいやっていたようです。

2.9.2 2009 年を振り返って Debian 関連で世間では何が起こったか

やっぱり Lenny release でしょうか... 来年の kFreeBSD official 化 (による盛り上がり) に期待したいですね。

3 最近の Debian 関連のミーティング報告

まえだこうへい



3.1 東京エリア Debian 勉強会 58 回目報告

東京オリンピック青少年総合センターで開催しました。参加者は、あけどさん、なかおさん、吉田さん、やまねさん、山本さん、本庄さん、日比野さん、あらきさん、高橋 (emasaka) さん、つじたさん、岩松さん、まえだの 12 名でした。

Debian での数学ことはじめとして、gnuplot, Octave, R の入門についての話をまえだが行いました。gnuplot と R でのプロットのやり方と、R と Octave のそれぞれの拡張パッケージである CRAN, Octave-Forge からの Debian パッケージ化するためのポイントポイントについて議論できたので、参加された方は gnuplot と R を自習して使えるようになると同時に、Octave-Forge と CRAN の拡張パッケージをガンガン Debian パッケージにするべく ITP していることでしょう。

もう一つのネタとして岩松さんが Debian auto-builder ネットワークについての話をされました。Debian を支える重要なシステムであるにも関わらず、このシステムを理解している DD も少ないとのことなので、とても意義のあるセッションでした。特にドキュメント化されていない loop-depends の対応の話には溜飲を下ろした人が多かったのではないかと思います。今回の話で、auto-builder に興味を持ったヒトは、ぜひ Debian/SH4 のポーティングに参加しましょう。

宴会は、石臼挽きたて蕎麦と酒肴の店 心粋 参宮橋店にて。Rolf さんが宴会から合流され、宴会場でのキーサインパーティが行われました。

3.2 Adam C Powell IV 迎撃

11 月 29 日に来日していた Adam C Powell IV を上川が迎撃しました。Debian Beowulf Project で昔からやりとりしていた仲です。秋葉原にいて大喜びしていました。

その直前に Debian Developer の荒木さんとも一緒に飲みに行っていたようです。

3.3 Dirk Eddelbuettel 迎撃

11 月 26 日に来日していた Dirk Eddelbuettel を迎撃しました。gtach さん、だいごさん、岩松さん、前田さん、Dirk Eddelbuettel 夫妻、上川 x2 にて宴会。Debian と金融について熱く語る新宿の夜でした。



4 東京エリア Debian 勉強会 2009 年度各種イベント開催 実績と総括

上川 純一



今月で 5 年目の Debian 勉強会が終了しました。Debian Developer になった人がいたり変化もみられ、私生活の面でも結婚したメンバーが多数いたり、転職したメンバーがいたり、当時と所属が変わっていないメンバーのほうがめずらしくなってきました。

4.1 Debian 勉強会の野望の進捗具合

今年は Debian 勉強会にとって大きなできごとがありました。当初の目標であった Debian Developer の育成という目標がすこしづつ実現してきたのです。東京エリア Debian 勉強会常連の岩松さん、関西 Debian 勉強会立ち上げ時期に尽力した矢吹さんが Debian Developer になりました。苦節 5 年。おめでとうございます。

2009 年までは基本的な Debian 開発者になるまでの基本的な教養の共有を目的にやってきました。これからは少し方向性を変えて開発に必要な実践的な内容にシフトしていったらいいかと考えています。2010 年はスポンサー、NMU、BSP の行い方をより実践的に行える方向を模索したいと思います。

4.2 運営方法

2009 年の勉強会には上川半分くらいしか出席しておらず、岩松・前田が中心として運営にあたりました。

事前課題は latex のソースコードに対する git format-atch の出力を ML に投稿するという手順で、宴会君と atnd を補助的に利用しました。勉強会会場の予約はとりあえず 30 人くらいが入れる場所を確保しましたが、宴会会場の予約は人数の確定する開催二日前、でした。

勉強会の会費は 500 円を維持しています。過去荻窪の「あんさんぶる荻窪」という公民館を利用していたので、費用計算もそこを基準におこなってきていました。2009 年はいろいろな会場を試しており、公営ではない施設も試しており、会場費用の高騰にともない赤字決算になった回もあります。公営の施設を利用していると会場が非常に安かったので印刷費用がまかなえていました。

幹事は昨年までは上川が集中的にしていたので運営についてのノウハウが十分伝わっていなかった苦労もあったみたいです。

4.3 基本的な数値

Debian 勉強会は毎回事前課題事後課題を設定しており、予習復習を必要だとうたっている勉強会です。実際にどれくらいの人が出席しているのか、またその人たちがどれくらい事前課題・事後課題を提出しているのか、確認して

みましょう。図1です。値は一年の移動平均です。

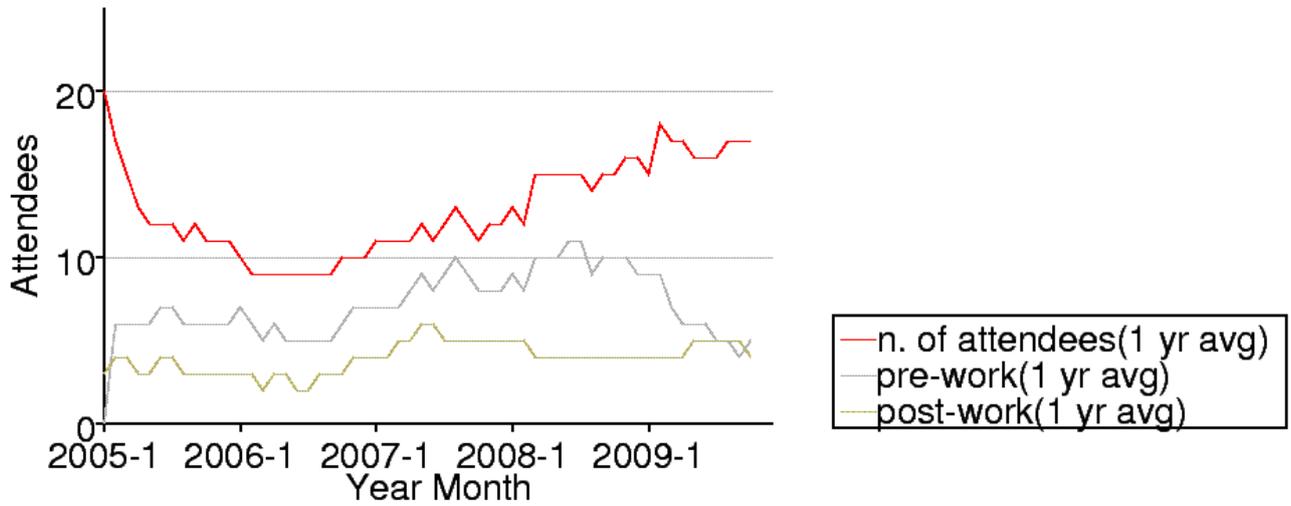


図1 東京エリア Debian 勉強会事前課題・事後課題提出実績 (12ヶ月移動平均)

毎回の参加者の人数と、その際のトピックを見てみます。今年の場合は、平日の夜に行った GPG キーサインパーティーの参加者の多さが目立っています。

表 1 東京エリア Debian 勉強会参加人数 (2005-2006 年)

	参加人数	内容
2005 年 1 月	21	秘密
2005 年 2 月	10	debhelper 1
2005 年 3 月	8	(早朝) debhelper 2、social contract
2005 年 4 月	6	debhelper 3
2005 年 5 月	8	DFSG、dpkg-cross、lintian/linda
2005 年 6 月	12	alternatives、d-i
2005 年 7 月	12	toolchain、dpatch
2005 年 8 月	7	Debconf 参加報告、ITP からアップロードまで
2005 年 9 月	14	debconf
2005 年 10 月	9	apt-listbugs、バグレポート、debconf 翻訳、debbugs
2005 年 11 月	8	DWN 翻訳フロー、statooverride
2005 年 12 月	8	忘年会
2006 年 1 月	8	policy、Debian 勉強会でやりたいこと
2006 年 2 月	7	policy、multimedia
2006 年 3 月	30	OSC: debian 勉強会、sid
2006 年 4 月	15	policy、L ^A T _E X
2006 年 5 月	6	mexico
2006 年 6 月	16	debconf、cowdancer
2006 年 7 月	40	OSC-Do: MacBook Debian
2006 年 8 月	17	13 執念
2006 年 9 月	12	翻訳、Debian-specific、oprofile
2006 年 10 月	23	network、i18n 会議、Flash、apt
2006 年 11 月	20	関西開催: bug、sid、packaging
2006 年 12 月	14	忘年会

表 2 東京エリア Debian 勉強会参加人数 (2007-2008 年)

	参加人数	内容
2007 年 1 月	15	一年を企画する
2007 年 2 月	13	dbcs、dpatch
2007 年 3 月	80	OSC 仮想化
2007 年 4 月	19	quilt、darcs、git
2007 年 5 月	23	etch、pbuilder、superh
2007 年 6 月	4	エンジンバラ開催: Debconf7 実況中継
2007 年 7 月	18	Debconf7 参加報告
2007 年 8 月	25	cdn.debian.or.jp
2007 年 9 月	14	exim
2007 年 10 月	30	OSC Tokyo/Fall(CUPS)
2007 年 11 月	19	live-helper、tomoyo linux kernel patch、server
2007 年 12 月	11	忘年会
2008 年 1 月	23	一年を企画する
2008 年 2 月 29+3 月 1 日	36	OSC
2008 年 3 月	37	データだけのパッケージ、ライセンス
2008 年 4 月	17	バイナリパッケージ
2008 年 5 月	20	複数のバイナリパッケージ
2008 年 6 月	10	debhelper
2008 年 7 月	17	Linux kernel patch / module パッケージ
2008 年 8 月	10	Debconf IRC 会議と Debian 温泉
2008 年 9 月	17	po4a、「Debian メンテナのお仕事」
2008 年 10 月	11?	OSC Tokyo/Fall
2008 年 11 月	17	「その場で勉強会資料を作成しちゃえ」Debian を使った L ^A T _E X 原稿作成合宿
2008 年 12 月	12	忘年会

表 3 東京エリア Debian 勉強会参加人数 (2009 年)

	参加人数	内容
2009 年 1 月	12	一年を企画する
2009 年 2 月	30	OSC パッケージハンズ オン
2009 年 3 月	23	Common Lisp, パッ ケージ作成
2009 年 4 月	15	Java Policy, ocaml, 開 発ワークフロー
2009 年 5 月	13	MC-MPI パッケージ化、 Erlang、Android アプ リ、DDTP
2009 年 6 月	14	DDTP・DDTSS、 bsdstats パッケージ、 Debian kFreeBSD
2009 年 7 月	?	スペインにて Debconf 9
2009 年 8 月	14	スペイン Debconf 9 参 加報告
2009 年 9 月	26	GPG キーサインパーテ ィー
2009 年 10 月	?	OSC Tokyo Fall
2009 年 11 月	12	Octave, R, gnuplot, auto-builder
2009 年 12 月	?	忘年会

5 関西 Debian 勉強会 2009 年度各種イベント開催 実績と総括

倉敷・佐々木・野方



5.1 運営状況

関西は運営に関わっている人に学生が多いので、いろいろ無理をお願いする場面も多かったような気がします。

5.1.1 勉強会全体

今年度途中 (7 月) より、運営担当が山下尊也から倉敷・佐々木・野方の三名体制に交代しました。これは山下の身辺が多忙になり身動きがとれないという理由からです。幸い、以前より分担に向け運営の見直しを進めていたこともあり、大きな混乱もなく継続することができました。

年度当初、ライブ中継に若干盛り上がりを見せましたが、その後、うまく継続できませんでした。問題としては IP アンリーチャブルな会場をメインにしていることと、中継の実作業を担っていた人が運営側にシフトし、余力を回せなくなっていることが原因と思われます。

5 月には神戸市を中心とした関西地域の新型インフルエンザ流行により、勉強会を中止する出来事がありました。社会的な要因により勉強会開催の判断を迫られる状況は初めてでしたが、こういう事は二度とあって欲しくないですね。

9 月は京都市サテパークにお邪魔して勉強会初の京都で開催しました。会場を変えると、いつもとは違う参加者も増えるので、たまに場所を変えるのもよいのではと思いました。

講師については現状、固定化している中、継続して常連参加者への講師依頼をするほかに、DMC を取り入れたり、LT 発表も可能な参加者自己紹介の常設などをおこないました。

LT 発表可能な参加者自己紹介は、話題にバリエーションが加わったなど興味深いこともあった反面、年度後半は関西の勉強会参加者も参加している Open Street Map にトピックを持っていかれてしまった感もあり、うまくバランスを取る必要がありそうです。

また、今年度は佐々木、山下の 2 名が Package Maintainer として Debian の New queue に新しくパッケージを送り込みました。来年もこの流れを維持できればと思います。

5.1.2 扱ったテーマ

勉強会の内容としては、パッケージ開発自体に加えて、Debian の体制にまつわる話 (gpg や mentors など) や、周辺ツールの利用 (bash や reportbug や gdb など) をとりあげました。来年度のテーマについては、年末年始に相談をする予定をしています。

翻訳関連では、東京での流れに乗り DDTSS のハンズオン実習をしましたが、予想外に反応がありました。もともと需要があったのか、実習したことで身近になったのか、はよくわかりませんが...

5.1.3 イベント関連

例年通り、夏のオープンソースカンファレンス Kansai@Kyoto(OSC) と、秋の関西オープンフォーラム (KOF) に出展しました。

セッションでは、OSC では大浦さんによる Debian GNU/kFreeBSD について、KOF では矢吹さんに DD になるまでの軌跡をお話してもらいました。矢吹さんは、関西 Debian 勉強会立ち上げの立役者なので、できれば勉強会にも来て欲しいところですが、最近は、なかなかご多忙で難しいとのことでした。

また、四国ではじまったオープンソース勉強会 *2 と、岡山でのオープンセミナー@岡山 *3 に、野方が参加して Debian Live やノウハウの紹介などを行いました。

5.2 開催実績

関西 Debian 勉強会の出席状況を確認してみましょう。グラフで見ると図 2 になります。表で見ると表 6 です。

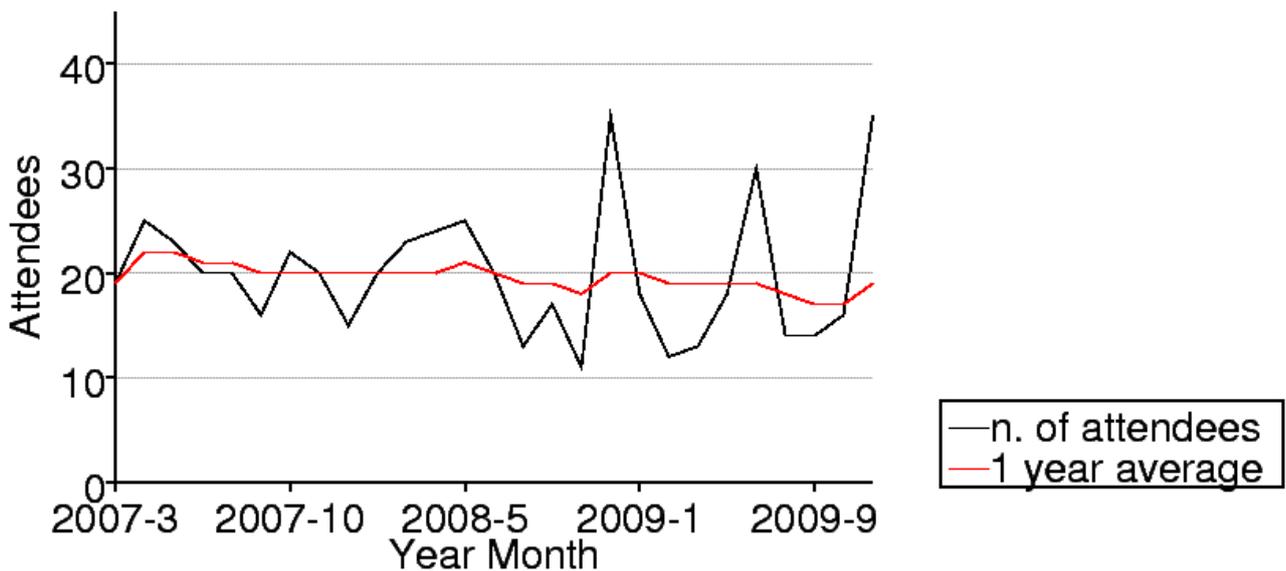


図 2 関西の参加人数推移

*2 <http://openforce.project2108.com/>

*3 <http://openseminar.okaya.ma/>

表 4 関西 Debian 勉強会参加人数 (2007 年)

	参加人数	内容
2007 年 3 月	19	開催にあたり
2007 年 4 月	25	goodbye、youtube、プロジェクトトラッカー
2007 年 6 月	23	社会契約、テーマ、debian/rules、bugreport
2007 年 7 月	20 前後	OSC-Kansai
2007 年 8 月	20	Inkscape、patch、dpatch
2007 年 9 月	16	ライブラリ、翻訳、debtorrent
2007 年 10 月	22	日本語入力、SPAM フィルタ
2007 年 11 月	20 前後	KOF
2007 年 12 月	15	忘年会、iPod touch

表 5 関西 Debian 勉強会参加人数 (2008 年)

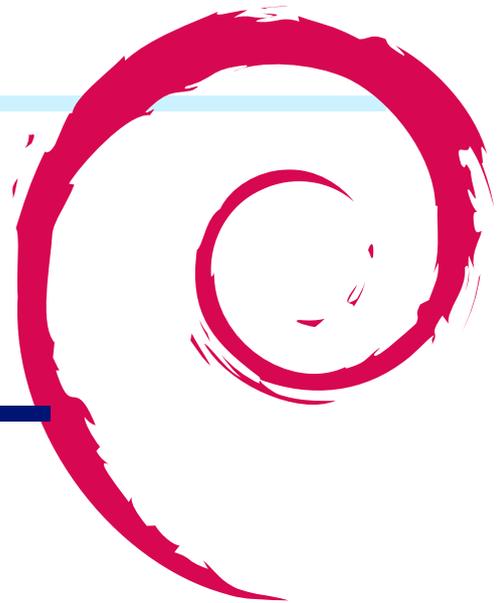
	参加人数	内容
2008 年 2 月	20	PC Cluster, GIS, TeX
2008 年 3 月	23	bug report, developer corner, GPG
2008 年 4 月	24	coLinux, Debian GNU/kFreeBSD, sid
2008 年 5 月	25	ipv6, emacs, us-tream.tv
2008 年 6 月	20	pbuilder, hotplug, ssl
2008 年 8 月	13	coLinux
2008 年 9 月	17	debian mentors, ubiquity, DFSG
2008 年 10 月	11	cdbs,cdn.debian.or.jp
2008 年 11 月	35	KOF
2008 年 12 月	?	TeX 資料作成ハンズオン

表 6 関西 Debian 勉強会参加人数 (2009 年)

	参加人数	内容
2009 年 1 月	18	DMCK, LT
2009 年 3 月	12	Git
2009 年 4 月	13	Installing sid, Man-coosi, keysign
2009 年 6 月	18	Debian Live, bash
2009 年 7 月	30?	OSC2009Kansai
2009 年 8 月	14	DDTSS, lintian
2009 年 9 月	14	reportbug, debian mentors
2009 年 10 月	16	gdb, packaging
2009 年 11 月	35	KOF2009
2009 年 12 月	??	GPS program, Open-StreetMap

6 2009 年を振り返ってみる

上川 純一



6.1 最近のトレンドと今後の推移

最近どんなことがあって、これからどういうことがあるでしょうか。みんなで予想してみましょう。

2007	2008	2009	2010	2011
<p>VT・AMD-V(仮想化技術)が普及(ML115!)、玄箱(ARM)、Open-Blocks(PPC?)、iPhone登場、HS-DPA月額5000円くらいに、google mobile、VISTAリリース、Leopardリリース、GPL3.0、メモリ2Gがコモディティーに、SparcT2がオープン、ニコニコ動画、</p>	<p>python 3.0 ruby 1.9 wine 1.0, wine64登場 RoR 2.0 登場で普及に 4コア・64bitのCPUがデスクトップに普及、Core2Quad値下げ。 ニコニコ動画1000万ユーザ突破、初音ミクブームに地デジ関連のPC製品の普及 勉強会の普及(楽天とか) 公衆無線LAN(wireless gate) 携帯電話の売上が落ちる、iPhone、Android登場、emobile 100円PC抱きあわせ(eeePC, Dell mini9) Zaurus販売終了。 Chumby発売。 サーバの仮想化ESXi・シンククライアント MacBook Air発売、無線802.11nが実機に SystemZ10発表 世界経済の崩壊(IT投資緊縮財政、職を失う人が増加) FreeBSD 7 (malloc, ZFS?) Debian次世代育成計画始動 Debian Maintainer制度始動 セキュリティー関連(OpenSSL事件、DNS事件) クラウド関連が流行? Nintendo DSi</p>	<p>政権交代、スパコン事業仕分け、円高 Windows7, Snow Leopard発売 Netwalker発売 MacBookからIEEE1394が消えた。 メモリがDDR3に移行中、メモリ高騰 マジコン販売取り締まり ラブプラス, OSSを使ったエロゲー登場(OpenCV), AR, セカイカメラ JLSでLinux来て大騒ぎ DD, 2世誕生 デジタルサイネージ Google Voice, Wave, Chrome, Chrome OS, Go, 日本語入力, 徒歩ナビ CouchDB Twitter,*なうブーム Eye-Fi, Kindle2, DS LL, PSP-GO, POKEN Cell終了のお知らせ tile window manager boom? Lennyリリース Debian結婚ブーム デスクトップ、4コア、8GB ノートパソコン、2コア、4GB Linuxが標準インストールのPC。(Dell) SSDの値段と容量がこなれる(まあまあ) HDDがなくなる? 高くなる?(ならず) SSD特化したFSが出てきた ipv6使えるようになってる(来年) DL禁止法? torrentに逆風?</p>	<p>Debian OAuth サービス開始のお知らせ SolidICE使ったDebian VDIサービス開始のお知らせ Chrome OS, Android 統合のお知らせ Willcom終了のお知らせ Netbook, クラウド, Amebaなう終了のお知らせ Debian Cloudリリースのお知らせ 10GbE, SSD普及 新iPhoneリリース 自民復活 SIMロックフリー延期のお知らせ 新Android端末(日本以外) 次世代用FS:btrfs, NILFS Lenny and Halfリリース Squeezeリリース遅延, kFreeBSD, SH4 オフィシャルアーキテクチャに ToyStory3リリース 消費税上昇に伴う繁忙期 クラウドにより、単純なホスティング業者がつかない? 一部は自社でもつようになる? USB 3.0搭載、wireless USB vs Bluetooth? 組み込みCPUはAtomに統一? Armは残ってる? ruby 2.0リリース? Perl6リリース?</p>	<p>Windowsドライバシグニチャチェックがなくなる IEEE1394終了のお知らせ LTEが徐々に普及 地デジ延期のお知らせ Debian 11日本開催 Googleに統合(オフィスソフト、グループウェア、メール、ファイルサーバ止めようぜ運動) Scalaのエンタープライズ利用 Windows11リリース C/Sを意識せずにアプリ開発できる環境(コンパイラが自動判断)</p>



7 qemubuilder 2009 年アップデート

上川純一

7.1 qemubuilder の基本コンセプト

qemubuilder は Debian パッケージをビルドするためのツールです。debootstrap のクロスブートストラップ機能を利用してベースイメージを作成した後、qemu を利用して各アーキテクチャ用の仮想マシンを実行し、その中でパッケージをビルドします。ネイティブビルドと変わらない使用感でパッケージのビルドができるので面倒なクロスビルドの設定が必要ありません。特に Debian パッケージは buildd でネイティブビルドされ、クロスビルドされない前提なので、buildd でビルドできるようなパッケージの作成・デバッグに便利です。

7.2 qemubuilder の使い方

利用したいアーキテクチャ向けのカーネルと設定ファイルを用意します。手元の設定では次のような設定になっています。

```
KERNEL_IMAGE=vmlinuz-2.6.24-1-versatile-armel
ARCH=armel
BASEPATH=/home/dancer/tmp/base-armel.qemu
INITRD=
```

イメージをまず作成します。これで、BASEPATH に指定したファイル名に qemu の RAW ディスクイメージが作成されます。

```
# qemubuilder --configfile arm.config --create
```

ディスクイメージをアップデートします。

```
# qemubuilder --configfile arm.config --update
```

パッケージをビルドするには dsc ファイルを指定します。

```
# qemubuilder --configfile arm.config --build xxx.dsc
```

7.3 qemubuilder の課題

カーネルと設定ファイルの取得が今一番面倒なところですが、Debian の標準のカーネルと initrd でできた時期もあったのですが、現在そういうようにはなっていません。

アーキテクチャの組み合わせがあまりにも多いため、動く組み合わせを同定することや、デバッグが困難です。ス

クリプトで自動化すること、またテストの自動化が必要ではないかと考えています。

また、qemu の-append コマンドでカーネルにブートパラメータを指定できることを装丁していますが、実際にはそれができないアーキテクチャ(ppc など)があり、そのままでは動きません。^{*4}

最近では kFreeBSD アーキテクチャなども登場してきていますが、このままの設計ではそこまで手が回らなさそうです。

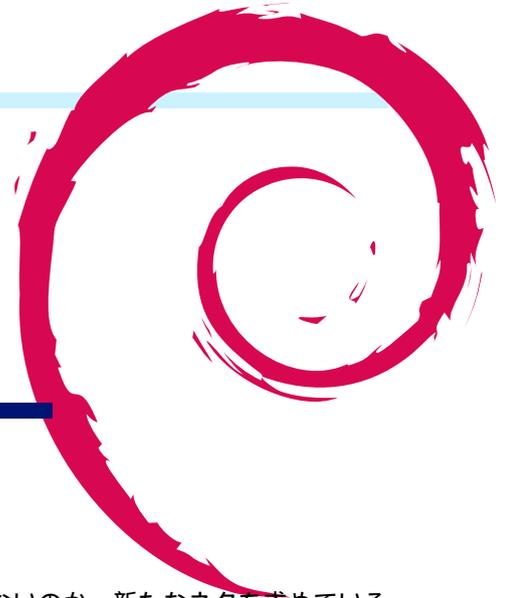
7.4 qemubuilder の今後の進め方

どうしましょうね。

^{*4} いまでもそうかはわかりません

8 lxc コンテナを使ってみる

まえだこうへい



Debian 勉強会に参加されている面々は、そろそろ Xen や KVM 以外に何かないのか、新たなネタを求めている向きが多いかと思います。そこで、最近、Linux Kernel に新たにマージされた機能を使って実現している、lxc という仮想化技術について紹介します。

8.1 はじめに

8.1.1 lxc の概要

lxc^{*5} は、正式名称を Linux Containers と言い、コンテナ自体が稼働するためのカーネルの機能と、コンテナを管理するためのユーザツールから構成されます。lxc で使用している Kernel の機能 (Control Group, 以下 Cgroup と省略) は、kernel 2.6.29 で完全にマージされ、カーネルにパッチを当ててビルドする必要がなくなってなっています。kernel 2.6.29 より前のカーネルでは、kernel 2.6.27 以降であればパッチを当てれば使うことができます。

lxc は GPL2 ライセンスで公開されており、開発およびメンテナンスは、Daniel Lezcano 氏が実質一人で行っています。

Debian では、Squeeze/Sid からパッケージ化されており、一つ前の最新版^{*6}がパッケージとなっています。

8.1.2 他のコンテナ型仮想化技術との比較

コンテナ型の仮想化技術というと有名なのは、Solaris Containers や FreeBSD jail がありますが、Linux では Linux-VServer, OpenVZ^{*7} などがあります。いずれも既に使ったことがある方が多いのではないのでしょうか。

lxc で提供されるサービスは、大きく分類してシステムコンテナと、アプリケーションコンテナの 2 つがあります。前者は、いわゆる OS まるごとの仮想化です。init から起動して、仮想 OS の空間を提供します。後者は、chroot によるアプリケーションの分離に近いです。単一アプリケーションを分離するだけなので、とても軽くシンプルなのが特徴です。

lxc は現状、一人で開発・メンテナンスされており、今後プロジェクトがどうなるのか先行き見えないところではあります。現在、メーリングリストを見ている限りでは開発は続いているようです。

8.2 導入してみる

8.2.1 ソースコードを取得する

ユーザスペースのツールのソースコードは、SourceForge にあります。ソースコードは git で管理されており、最新版は git リポジトリから取得できます。

^{*5} <http://lxc.sourceforge.net/>

^{*6} 2009 年 11 月 27 日現在、0.6.3。

^{*7} Parallels Virtuozzo Containers の OSS 版。

```
$ git clone git://lxc.git.sourceforge.net/gitroot/lxc/lxc
```

Debian では前述のとおり、Squeeze/Sid でパッケージになっている*⁸ ため、最新版である必要がなければ、ソースコードは特に必要ありません。

8.2.2 カーネルオプションを有効にする

lxc の機能をフルに活用するには以下のカーネルオプションが有効になっている必要があります。

```
* General
* Control Group support
-> namespace cgroup subsystem
-> cpuset support
-> Group CPU scheduler
-> control group freeze subsystem
-> Basis for grouping tasks (Control Groups)
-> Simple CPU accounting
-> Resource counters
-> Memory resource controllers for Control Groups
-> Namespace support
-> UTS namespace
-> IPC namespace
-> User namespace
-> Pid namespace
* Network support
-> Networking options
-> Network namespace support
```

これらが有効になっているかを確認するには、lxc のソースツリーに含まれている、src/lxc/lxc-checkconfig.in というシェルスクリプトを実行すれば、現在起動中のカーネルでどのカーネルオプションが無効になっているかをチェックできます。

また、Debian パッケージでは、/usr/bin/lxc-checkconfig としてインストールされています。Squeeze/Sid で Debian のカーネルパッケージ*⁹を使っている環境で確認すると以下の結果になります。Cgroup memory controller のみが無効になっているようです。

```
$ lxc-checkconfig
Kernel config /proc/config.gz not found, looking in other places...
Found kernel config file /boot/config-2.6.30-2-amd64
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup namespace: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: disabled
Cgroup cpuset: enabled

--- Misc ---
Veth pair device: enabled
Macvlan: enabled
File capabilities: enabled
```

8.2.3 Debian でのインストール

ここから先は、Squeeze/Sid でパッケージを使うことを前提として話を進めますが、このままでは、cgroup でのメモリ管理は無効になっていますので、カーネルオプション CONFIG_CGROUP_MEM_RES_CTLR を有効にしてリビルドしてください。それ以外で実際に Debian で lxc を使うために必要なパッケージは何かというと lxc だけです。

```
$ sudo apt-get install lxc
```

これでアプリケーションコンテナを試すことはできます。README にも載っている手順ですが、次のコマンドを

*⁸ <http://packages.debian.org/search?keywords=lxc&searchon=names&exact=1&suite=all§ion=all>

*⁹ 2009 年 11 月 27 日現在、linux-image-2.6.30-2-amd64

実行すると、即席のコンテナを起動できます。

```
$ uname -a
Linux silicon 2.6.32 #1 SMP Sun Dec 6 02:30:30 JST 2009 x86_64 GNU/Linu
$ sudo lxc-execute -n hoge -f ./lxc-macvlan.conf /bin/bash
# uname -a
Linux alpha 2.6.32 #1 SMP Sun Dec 6 02:30:30 JST 2009 x86_64 GNU/Linux
```

他のコンソールからコンテナが起動しているか確認してみます。

```
$ sudo lxc-info -n hoge
'hoge' is RUNNING
$ lxc-ps -n hoge
CONTAINER   PID TTY          TIME CMD
            4747 pts/3        00:00:00 bash
            5692 pts/3        00:00:00 lxc-ps
            5693 pts/3        00:00:00 ps
```

ちゃんと確認できましたね。今回は、これで以上です、と言いたいところですが、この環境はコンテナを起動させただけでしかなく、はっきり言って役に立ちません。bash を sudo で起動しているだけで、ホスト OS のファイルシステムにもアクセスできてしまいます。

コンテナだけを起動させて満足、はい、終了とするのであれば良いかもしれませんが、実際に lxc を活用しようと考えているなら次に挙げる他のパッケージをインストールし、さらにコンテナ用のクローズ環境を作る必要があります。

- iproute : コンテナのネットワーク設定を行うため。
- debootstrap : コンテナのイメージを作成するため。

今回は、クローズな Debian 環境を簡単に作るための方法を紹介します。^{*10}

8.2.4 ネットワークの設定

ブリッジの設定

必要なパッケージをインストールしたら、まずはブリッジの設定を行う必要があります。/etc/network/interfaces で直接ブリッジの設定をすれば良いと思いますが、シェルスクリプトを用意して、それを post-up で実行させれば良いでしょう。

```
#!/bin/sh
brctl addbr br0                                <- ブリッジデバイスの追加
brctl setfd br0 0                              <- ブリッジデバイス br0 の設定
ifconfig br0 192.168.0.1 promisc up
brctl addif br0 eth0
ifconfig eth0 0.0.0.0 up
route add -net default gw 192.168.0.254 br0 <- ホスト OS のゲートウェイ
```

interfaces は以下のように設定します。

^{*10} 厳密に言うと、コンテナからホスト OS で稼働しているプロセスや、カーネルメッセージが見えてしまうのでクローズにはまだなっていないとは言えませんが、まだ開発中でもあるのでそのうち改善されるでしょう。

```
$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.0.101
    netmask 255.255.255.0
    broadcast 192.168.0.255
    pre-up /etc/init.d/iptables start
    post-up /etc/network/if-up.d/brctl.sh
```

以上のあと、ブリッジの設定がきちんとされているか確認してみると、以下のようになります。

```
$ /usr/sbin/brctl show
bridge name      bridge id          STP enabled      interfaces
br0              8000.00wwyyyxxno  eth0
                veth0_14820
                veth0_15932
                veth0_17164
```

ちなみに“veth0_”の後ろの数字は、コンテナで起動した init プロセスのプロセス ID です。

IP フォワードと NAT の設定

ホスト OS とコンテナ、コンテナとホスト OS の外部のネットワーク、コンテナ間での通信は、上記の設定だけでなく、IP フォワードや NAT の設定をする必要があります。例えば、コンテナ起動後、ホスト OS から ssh でログインするのも、IP フォワードが必要となります。

```
$ sudo bash -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'
```

IP フォワードを設定すると、ホスト OS の外部のネットワークへの通信や、コンテナ同士の通信も行えるようになります。^{*11}

一方、ホスト OS の外部ネットワークからはこのままではアクセスできません。アクセスを許可するには、ホスト OS でポートフォワーディングや、宛先 NAT などを行う必要があります。ポートフォワーディングを行うのであれば、以下のようなルールを設定します。

```
iptables -t nat -A PREROUTING -d 192.168.0.1 -p tcp --dport 80 -i br0 -j DNAT --to 192.168.0.101
iptables -t nat -A PREROUTING -d 192.168.0.1 -p tcp --dport 5984 -i br0 -j DNAT --to 192.168.0.102
```

ここでの設定は、デフォルトポリシーが全て ACCEPT であることを前提にしていますが、実際には当然 DROP にすると思いますので、FORWARD のルールなども定義する必要があります。Netfilter のルールと IP フォワーディングの許可はスクリプトにして、/etc/network/interface で pre-up として設定しておくといいでしょう。

なお、FORWARD チェインのデフォルトポリシーが ACCEPT である場合は問題ありませんが、DROP や REJECT に設定してある場合は、コンテナ同士での通信はできません。FORWARD チェインでの ACCEPT ルールが必要になります。

8.2.5 システムコンテナの作成

では、システムコンテナのイメージを作成してみます。lxc のパッケージに含まれる、/usr/share/doc/lxc/examples/lxc-debian.gz を使って、Debian のシステムコンテナを作成します。

まず、このファイルをホームディレクトリなどで伸張します。

```
$ zcat /usr/share/doc/lxc/examples/lxc-debian.gz > ~/lxc-debian
```

^{*11} 同じブロードキャストドメインの場合。

次に、cgroup ファイルシステムをマウントします。/etc/fstab に下記一行を追記します。マウントポイントは任意の場所で構いません。

```
cgroup /var/local/cgroup cgroup defaults 0 0
```

追記したら、マウントポイントのディレクトリを作成し、マウントします。

```
$ sudo mkdir /var/local/cgroup
$ sudo mount cgroup
```

それでは、先ほどの lxc-debian スクリプトを使ってコンテナイメージを作成します。このスクリプトでは debootstrap を使ってイメージが作成されますが、ディストリビューションは lenny になっています。前述した通り本環境は Squeeze/Sid です。コンテナも Squeeze/Sid の方が良ければ、コンテナイメージを作成する前に予めスクリプトを書き換えておく必要があります。また、パッケージも、デフォルトで apache2 がインストールされたり、一方、sudo や vi, dig コマンドが無かったりするので、そのままインストールすると不便であったりするので、予めインストールするパッケージの指定を変更しておく必要があります。lxc-debian スクリプト内の debootstrap コマンドの--include オプションでパッケージの指定は変更できます。また、sshd の設定や、ネットワークの設定も予め設定しておくとう便利です。以下、変更済みファイルとの diff の結果を掲載しておきます。

```
$ diff -u a/lxc-debian b/lxc-debian
--- a/lxc-debian      2009-11-30 21:22:59.000000000 +0900
+++ b/lxc-debian      2009-10-30 17:58:20.000000000 +0900
@@ -8,8 +8,8 @@
 MNTFILE=
 TMPMNTFILE=
 UTSNAME=
 -IPV4="172.20.0.21"
 -GATEWAY="172.20.0.1"
 +IPV4="192.168.0.101"
 +GATEWAY="192.168.0.1"
 MTU="1500"

 # These paths are within the container so do not need to obey configure prefixes
@@ -99,14 +99,14 @@
 SyslogFacility AUTH
 LogLevel INFO
 LoginGraceTime 120
 -PermitRootLogin yes
 +PermitRootLogin no
 StrictModes yes
 RSAAuthentication yes
 PubkeyAuthentication yes
 IgnoreRhosts yes
 RhostsRSAAuthentication no
 HostbasedAuthentication no
 -PermitEmptyPasswords yes
 +PermitEmptyPasswords no
 ChallengeResponseAuthentication no
 EOF
 }
@@ -259,8 +259,8 @@
 # download a mini debian into a cache
 echo "Downloading debian minimal ..."
 debootstrap --verbose --variant=minbase --arch=$ARCH \
- --include ifupdown,locales,libui-dialog-perl,dialog,apache2,netbase,net-tools,iproute,openssh-server \
- lenny $CACHE/partial-$ARCH http://ftp.debian.org/debian
+ --include ifupdown,locales,libui-dialog-perl,dialog,sudo,vim-tiny,dnsutils,netbase,net-tools,iproute,openssh-server \
+ sid $CACHE/partial-$ARCH http://cdn.debian.or.jp/debian

 RESULT=$?
 if [ "$RESULT" != "0" ]; then
```

それでは、コンテナを作成します。コンテナを作成する場所は任意の場所にできます。通常、lxc-debian スクリプトを実行を実行したディレクトリの下に debian ディレクトリを自動的に作成し、その下にコンテナイメージである rootfs が作成されます。配置したいディレクトリが無ければ作成し、そのディレクトリへ移動し、sudo lxc-debian create を実行します。

```

$ sudo mkdir /var/cache/lxc
$ cd /var/cache/lxc
$ sudo bash /home/kohei/lxc-debian create
What is the name for the container ? [debian] hoge          <- コンテナの名前
What hostname do you wish for this container ? [hoge]      <- コンテナのホスト名
What IP address do you wish for this container ? [192.168.0.101] <- コンテナの IP アドレス
What is the gateway IP address ? [192.168.0.1]           <- コンテナから見たデフォルトゲートウェイのアドレス
What is the MTU size ? [1500]
Specify the location of the rootfs [./rootfs.hoge]
Specify the location for an extra fstab file [(none)]
(snip)
Choose your architecture
1) amd64
2) i386
#? 1                                                       <- Core 2 Duo のマシンなのでアーキテクチャは amd64 を選択。
Architecture amd64 selected
Checking cache download ...Found.
Copying rootfs ...Done.
(snip)
update-rc.d: using dependency based boot sequencing
Done.

You can run your container with the 'lxc-start -n hoge'

```

これで、`/var/cache/lxc/debian/rootfs.hoge` という名前でコンテナイメージのディレクトリが作成され、ここに、`debootstrap` による Debian イメージのコピーが作成されます。初めて `lxc-debian create` を実行すると、`debootstrap` でダウンロードされるキャッシュイメージが、`/var/cache/lxc/debian/rootfs-architecture` として作成されます。^{*12}

コンテナ自体のメタ情報は、`/var/lib/lxc` ディレクトリのコンテナ名のディレクトリ以下にあります。ツリー表示すると以下のようになっています。

```

$ tree hoge/
couchdb/
|-- cgroup
|-- config
|-- fstab
|-- init
|-- network
|   |-- veth0
|       |-- ifindex
|       |-- link
|       |-- mtu
|       |-- name
|       |-- up
|-- nsgroup -> /var/local/cgroup/hoge
|-- pts
|-- rootfs
|   |-- rootfs -> /var/cache/lxc/debian/rootfs.hoge
|-- state
|-- tty
'-- utsname

5 directories, 13 files

```

8.2.6 システムコンテナの起動

システムコンテナの起動の前に、やることがあります。コンテナイメージに、ログイン可能なユーザアカウントを作成することと、`/etc/hosts` を書き換えることです。前者は、コンテナ作成後にそのまま起動させたのでは、ログインすることができません。ホスト OS 側と同じユーザアカウントで問題なければ、`/var/cache/lxc/debian/rootfs.hoge/etc/ディレクトリ` の下の、`passwd`, `shadow`, `group` をホストのそれで上書きしておくといいでしょう^{*13}。後者はコンテナ自身の名前解決の設定が目的です。現状では、コンテナの `/etc/hosts` は、一律、

```

127.0.0.1    localhost
127.0.1.1    ホスト OS のホスト名

```

となっており、上記の 2 行目が、`/etc/hostname` と異なるため、ユーザアカウントを作っても、ログイン時に名前解決で時間が掛かってしまいます。ですので、予め、ホスト OS のホスト名になっている部分をコンテナのホスト名

^{*12} 今回は `amd64` を選択しているので、`/var/cache/lxc/debian/rootfs-amd64` となります。

^{*13} 今後作成するコンテナ全てに共通して言えることなので、キャッシュイメージである、`/var/cache/lxc/debian/rootfs-amd64/etc/ディレクトリ` 以下を上書きしておいた方が良くもありません。

に書き換えておく必要があるわけです。^{*14}

上記の設定を終えたら、システムコンテナを起動します。システムコンテナの起動は、`lxc-start` コマンドを使います。一つの環境で複数のコンテナを起動できるので、コンテナ名の指定が必ず必要です。コンテナ名の指定には、オプション `-n` を使います。

```
$ sudo lxc-start -n hoge
```

このまま起動させると、現在のシェルでそのままコンテナのコンソールが表示されます。ログインするにはそのままコンソールログインすれば良いでしょう。バックグラウンドで起動させるには、`-d` オプションをつけます。

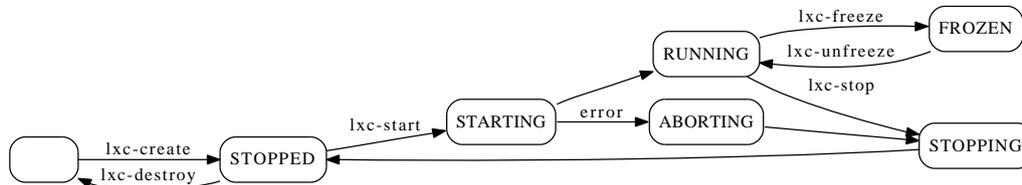
```
$ sudo lxc-start -n hoge -d
```

KVM や Xen などのようにカーネルから起動させる訳ではなく、`init` プロセスから起動させるので、起動完了までに要する時間はわずかです。

8.3 lxc の仕組みをしてみる

8.3.1 コンテナのライフサイクル

Linux コンテナのライフサイクルは他の仮想化技術と大きく異なるところはなく、次の図の様になります。



この図での状態を遷移させるためのコマンド、つまりコンテナを管理するためのコマンドは以下の通りです。

status	コマンド	実行後の状態	備考
コンテナの起動	<code>lxc-start</code> <code>lxc-execute</code>	STARTING, RUNNING	システムコンテナ アプリケーションコンテナ
コンテナの一時停止	<code>lxc-freeze</code>	FROZEN	
コンテナの再開	<code>lxc-unfreeze</code>	RUNNING	
コンテナの停止	<code>lxc-stop</code>	STOPPING, STOPPED	
コンテナの再起動	<code>lxc-restart</code>	STOPPING, STOPPED STARTING, RUNNING	
コンテナの作成	<code>lxc-create</code>	STOPPED	<code>lxc-debian</code> は内部で <code>lxc-create</code> を実行。 <code>lxc-fedora</code> , <code>lxc-sshd</code> など同様。
コンテナの破棄	<code>lxc-destroy</code>		

8.3.2 リソースを制御する

`lxc` では、コンテナのリソースを制御するために、`cgroup` というカーネルの機能を使っています。`cgroup` とは、Control group の略です。Linux Kernel は通常プロセス単位でのリソース制御を行っていましたが、`cgroup` を使うと、同じ `cgroup` に所属しているプロセス間でのリソースの共有ができます。`lxc` では、`lxc-cgroup` コマンドを用い、コンテナのリソースの設定を表示したり、値をセットすることができます。

設定を変更する。

`cgroup` の変数を表示する場合は、“`lxc-cgroup -n コンテナ名 変数名`” で、変数の値を変更するには、“`sudo lxc-cgroup -n コンテナ名変数名 設定する値`” とします。

例えば、CPU リソースの割り当てには、`cpu.shares` というパラメータを用いますが、あるコンテナに割り当てられている比率を算出するには、任意のコンテナの `cpu.shares` / 各コンテナの `cpu.share` の総和を計算する必要があります。具体的には、現在の各コンテナの `cpu` リソースの割り当て設定を確認すると、

^{*14} これは、`lxc-debian` スクリプト自体でコンテナのホスト名を書き換えるようにしておけば済む話だと思っただけだね。

```

$ lxc-ls
couchdb git hoge octave w3m web
$ for i in `lxc-ls`; do echo -ne "$i"\t"; lxc-cgroup -n $i cpu.shares; done
couchdb      2048
git          1024
hoge         1024
octave       1024
w3m         1024
web          1024

```

総和は 7168 で、各コンテナは、couchdb は約 28.6%、他のコンテナは約 14.3% ほどの比率で CPU リソースをシェアするということが分かります。

コンテナ全体のリソースは？

上記の通り、lxc-cgroup コマンドは、コンテナを明示的に指定する必要があるため、cgroup 管理下全体での使用率などを把握するには不便です。cgroup で制御されるリソースは 8.2.5 節でマウントした、cgroup ファイルシステムからアクセスすることもできます。今回はマウントポイントを /var/local/cgroup にしているためこのディレクトリの下を見てみると以下の様になっています。^{*15}

```

$ cd /var/local/cgroup
$ ls -F
couchdb/          cpuset.memory_pressure      git/
cpu.shares        cpuset.memory_pressure_enabled hoge/
cpuacct.stat      cpuset.memory_spread_page   net_cls.classid
cpuacct.usage     cpuset.memory_spread_slab   notify_on_release
cpuacct.usage_percpu cpuset.mems                  octave/
cpuset.cpu_exclusive cpuset.sched_load_balance   release_agent
cpuset.cpus       cpuset.sched_relax_domain_level tasks
cpuset.mem_exclusive devices.allow                 w3m/
cpuset.mem_hardwall devices.deny                  web/
cpuset.memory_migrate devices.list

```

/var/local/cgroup ディレクトリ以下にある、“xxx.yyyy”の形式をとっているのが、cgroup で管理するリソース項目で、このディレクトリ直下はシステム全体のリソースです。一方、hoge/などのディレクトリがありますが、これは各コンテナに割り当てられているリソース項目です。ですので、このディレクトリを使って、lxc 全体と各コンテナのリソースの管理を行うことができます。

8.4 まとめ

Linux Kernel の標準機能だけを使ったコンテナである、Linux Containers について説明しました。まだまだ機能的には不十分なところもあり、開発体制も不安なところがありますが、特別なカーネルパッチを適用せずに試せる点では非常に気軽に使えるのではないかと思います。

また、lxc は libvirt のサポート対象にもなっています。本資料を作成前に libvirt で扱えるか検証してみたところ、定義ファイルを作るところまではできたものの、virsh でコネクトするとうまくリソースにアクセスできないといった問題もありますが、やることが多いのでハックするには良いネタになるのではないのでしょうか。

8.5 参考文献

- LXC
<http://lxc.sourceforge.net/lxc.html>
- LXC: Linux コンテナツール ; IBM developer Works Japan
<http://www.ibm.com/developerworks/jp/linux/library/l-lxc-containers/>^{*16}
- Cgroup And Memory Resource Controller
<http://www.linux-foundation.jp/uploads/seminar20081119/CgroupMemcgMaster.pdf>

^{*15} lxc-cgroup コマンド自体は、各コンテナの値を /var/lib/lxc/コンテナ名/ns/cgroup/以下から取得しますが、これは、/var/local/cgroup/コンテナ名 への symlink になっています。

^{*16} 今年の 2 月の記事ですが内容が若干古く、現在存在しないコマンドもあるので要注意。



Debian 勉強会資料

2009年12月12日 初版第1刷発行
東京エリア Debian 勉強会（編集・印刷・発行）
