



第 13 回 東京エリア Debian 勉強会 事前資料

Debian 勉強会会場係 上川純一 *

2006 年 2 月 18 日

目次

1	Introduction To Debian 勉強会	2
1.1	講師紹介	2
1.2	事前課題紹介	2
2	Debian Weekly News trivia quiz	4
2.1	2006 年 4 号	4
2.2	2006 年 5 号	4
2.3	2006 年 6 号	5
2.4	2006 年 7 号	5
3	最近の Debian 関連のミーティング報告	6
3.1	東京エリア Debian 勉強会 12 回目報告	6
4	Debian policy 第 2 回	7
4.1	はじめに	7
4.2	Debian アーカイブについて	7
4.3	バイナリファイル	11
5	Debian multimedia project	14
5.1	agnula/DeMuDi	14
5.2	Debian multimedia policy	14
5.3	今 Debian でできること	14
5.4	MIDI コネクション	15
5.5	JACK で接続する	15
5.6	LADSPA:エフェクトをかける	15
5.7	Linux オーディオ処理におけるリアルタイムの必要性	15
6	Debian Multimedia Audio application 概観	17
6.1	フレームワーク系	17
6.2	ソフトウェアシンセ	20
6.3	楽譜編集	23
6.4	音声編集系	25
7	次回	29

1 Introduction To Debian 勉強会

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？

目的として下記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を情報共有する場をつくる
- まとまっていない Debian を利用する際の情報をまとめて、ある程度の塊として出してみる

また、東京には Linux の勉強会はたくさんありますので、Debian に限定した勉強会にします。Linux の基本的な利用方法などが知りたい方は、他でがんばってください。Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりと作りながらスーパーハッカーになれるような姿を妄想しています。

Debian をこれからどうするという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、というのが目的です。次回は違うこと言ってるかもしれませんが、御容赦を。

1.1 講師紹介

- 岩松さん Debian new maintainer になるべく修行中です。
- 上川純一 宴会の幹事です。

1.2 事前課題紹介

今回の事前課題は「普段の生活でフリーソフトウェアライセンスについて感じること」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。

1.2.1 吉田さん

普段の生活でフリーソフトウェアライセンスについて感じること

ライセンスについては、野良パッケージを外に公開するときによく気になりますね。Debian や upstream のソースとそれに対するパッチの両方のライセンスをちゃんと調べないとならない(当然ですが...)。

そのため、パッケージ作成より、ライセンス調査の方が時間がかかったり、ライセンス的にグレーとか判断されている物は公開を見送ったりします。

個人的に 1 から作る物については修正 BSD ライセンスが好きです。ソースも含めて好きに使ってくれというスタンスです。

GPL(Version2) は誤解を受けやすいライセンスだなということはよく思います。

たとえば GPL(Version2) ライセンスのソフトウェアの付属ドキュメント(英語)を翻訳した物で別に配布されている物のライセンスは？

本体は GPL(Version2) だけれど、ライセンス条項のないディレクトリに格納されているソースでコンフィグオプションによりリンクされるソースのライセンスは？等いろいろ悩ましいです。

1.2.2 中島さん

ソフトはライセンスなんて無くていいと思う。僕の周囲の人間もライセンスを気にして使ってる人はいない。だから必要ない。なぜなら使うのに、いちいちこうしろああしろと言われてもメンドウだからだ。ソフトがフリーなんて騒ぐことでもなんでもない。それは当たり前だ。フリーじゃなきゃ使い辛い。例えばもし空気が各地域ごとに、この地域の空気は大きく吸って小さくゆくり吐く。だとか1秒だけ吸って5秒で吐く。などと決められていたら逆に息苦しいだけだ。つまり世の中そこらじゅうにコンピュータだらけなのだからコードは空気みたいなものなので、そんなものにライセンスがどーのこうのと言われても生きていけなくなるじゃないか。だからライセンスなんて無いほうがいい。

1.2.3 大坪 知さん

ソフトに対してお金を払いたくない私にとって、debian は、全てのソフトを無料で公開しているディストリビューションとして、極めてありがたい存在です。また私は、もしかしてプログラムソフトとは、売買の対象にするものではなくて、みんなで共同して作り出して、みんなで共有して利用するものとなっていくものかもしれないとも思っています。そしてまた、私はインターネット空間は将来、誤ったプログラムソフトによってもたらされる被害を最小限にするために、ソースコードが公開されたプログラムソフトしか流通できなくなるのではとも思っています。

1.2.4 上川

フリーソフトウェアライセンスについてしばらく平穏な時期が過ぎていたが、今回、GPLv3 の策定に際してまた話題にのぼってくるようになった。そもそも GPL さえ選んでおけば良いという時代ではないが、だれもが同意できるような GPLv3 になって、分裂が押えられるというのが理想だ。

GFDL が DFSG Free でないことについて Debian と FSF が対立しているのが現状大きな問題だが、GPLv3 も DFSG Free でなくなってしまうと、Debian の存在意義があやうくなる。Debian としては、DFSG Free である GPLv3 を目指したい所である。

2 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。

2.1 2006 年 4 号

<http://www.debian.org/News/weekly/2006/04/> にある 1 月 24 日版です。

問題 1. Debian のフォーラムをつくらうという提案に関して却下した理由は

- A フォーラムにはフォーラムの主がいついてしまうからダメだ
- B フォーラムは 2ch 化してしまうから不適切
- C メーリングリストの参加者とフォーラムの参加者は質的に違う

問題 2. 1 月 1 日に Anthony Towns が発表した Debian のリリース対象のアーキテクチャは

- A alpha、amd64、hppa、i386、ia64、mips、mipsel、powerpc
- B i386、powerpc、amd64
- C i386、m68k

問題 3. 今後の標準で kaffe で利用する java コンパイラはどれか

- A jikes
- B gcj
- C ecj

2.2 2006 年 5 号

<http://www.debian.org/News/weekly/2006/05/> にある 1 月 31 日版です。

問題 4. 2006 年の Debian Day が開催されるのはいつか

- A 5 月 13 日
- B 6 月 13 日
- C 7 月 13 日

問題 5. /var/run/ 以下のサブディレクトリをつかう場合はどうするべきか

- A パッケージに含める
- B サブディレクトリは使わない
- C 起動時に毎回作成する

問題 6. launchpad を Debian の開発にしようという提案に対して出た反論は

- A ubuntu の成果なんてつかえない
- B non-free であるため、それに依存するのはよくない
- C ウェブインタフェースなんて使いたくない

2.3 2006 年 6 号

<http://www.debian.org/News/weekly/2006/06/> にある 2 月 7 日版です。

問題 7. Extremadura のハッキングセッションで何ができたか。

- A 政治的な思想の熟成
- B D-I の GUI 版
- C 日焼け

問題 8. 2006 年の Debian Project Leader 選挙に最初に立候補したのは誰か

- A Branden Robinson
- B Junichi Uekawa
- C Lars Wirzenius

問題 9. FLUG は何か

- A Finland の Linux Users Group
- B Finland の Lost Users Group
- C Finland の Legal Users Group

2.4 2006 年 7 号

<http://www.debian.org/News/weekly/2006/07/> にある 2 月 14 日版です。

問題 10. iBook G4 の無線がどうなったか

- A 仕様が公開された
- B あいかわらず動かない
- C 動くようになった

問題 11. 商標について Debian はどういう立場をとっているか

- A 変更と配布の邪魔にならないようであれば問題ない
- B 商標は自由の思想に反するので許せない
- C なにそれ、おいしい？

問題 12. <http://lists.debian.org/msgid-search/> はどうやって使うのか

- A <http://lists.debian.org/msgid-search/> /メールのメッセージ ID
- B <http://lists.debian.org/msgid-search/> /送信者の名前
- C <http://lists.debian.org/msgid-search/> /メールのサブジェクト

3 最近の Debian 関連のミーティング報告

上川純一



3.1 東京エリア Debian 勉強会 12 回目報告

1 月の第 12 回 Debian 勉強会を実施しました。今回は岩松さんが debian-policy について今後どうやって解説していきたいかということを説明しました。今回の参加人数は 8 人でした。

まず最初に Debian 勉強会の経緯について紹介、事前課題の紹介、会場の諸注意など。関根さん曰く、オーストラリアの SLUG はレストランで勉強会を開催しているということで、食事ができる場所で勉強会を開催できるのもよいな、と思いました。

Debian weekly news quiz は今月は全問正解は難しかったようです。backports.org についての話題が出た時に、RedHat には security advisory とは別に Bug Fix advisory というのがある、というのを関根さんに紹介いただきました。

岩松さんが、Debian policy について紹介しました。Debian policy と一口にいっても大量のドキュメントがあり、debian-policy 本体についても dpkg の動作の話などがたくさんはっており、大きなドキュメントです。変更の手順などについて説明がありました。今年じっくりといろいろな部分について解説していこうということなので、期待です。debian-policy の日本語訳があまりすすんでいないとのことで、そちらも少し手をつけるのがよいのかな、と思います。

久しぶりにグループワークをしました。個人課題として今年の Debian 勉強会でやりたいことというのを提案していただきました。

宴会は「土間土間」にて開催。

4 Debian policy 第 2 回

岩松



4.1 はじめに

今回から実際に Debian Policy の中身を見ていこうと思います。対象は Debian アーカイブについて、バイナリファイルのポリシーについてです。

4.2 Debian アーカイブについて

Debian には大量 (1 万パッケージ以上) のパッケージがあります。それらを管理し、フリーなオペレーティングシステムを目指しています。このフリーという言葉はどういうものなのか、Debian ではどのように扱うのかということガイドラインにしたものが Debian フリーソフトウェアガイドライン (以下、DFSG) です。

4.2.1 Debian フリーソフトウェアガイドライン とは？

Debian GNU/Linux システムのガイドラインである DFSG とはどのような内容なのか、確認してみましょう。

- 自由な再配布

Debian システムを構成するソフトウェアのライセンスは、そのソフトウェアを複数の異なる提供元から配布されているプログラムを集めたソフトウェアディストリビューションの一部として、誰かが販売したり無料配布したりすることを制限してはいけません。また、ライセンスはそのような販売に対して使用料やその他の手数料を要求してはいけません。

Debian にインストールされるソフトウェアのライセンスは自由に配布でき、無償またはお金を取って配布することが可能なライセンスでないといけない、ということです。ただし、ディストリビューションに含まれるプログラムのライセンスの内容に配布に対して料金を請求したりするものがあってはいけないということです。

この項目に合わないライセンスの一つとして aladdin フリー公衆利用許諾契約書 (Aladdin Free Public License) があります。このライセンスは配布において手数料を取るのを禁じています。

- ソースコード

プログラムにはソースコードが含まれていなければならず、かつ実行形式での配布に加えてソースコードでの配布をも許可していなければなりません。

ソースコードの配布を許可していないライセンスは Debian にはインストールされることはないということです。

これはそのままです。

- 派生ソフトウェア

ライセンスは、ソフトウェアの修正や派生ソフトウェアの作成を認めていなければなりません。そして、これらをオリジナルソフトウェアのライセンスと同じ条件の下で配布することが可能でなければなりません。

あるソフトウェアを改変し、それを配布するときも改変元と同じライセンスで配布できないと Debian にはイ

インストールされないということであり、派生を認めたライセンスでないといふことです。

この項目に合わないライセンスの一つとして Qmail のライセンスがあります。改変された場合には配布を認められてないからです。

- 原作者によるソースコードの整合性維持

ライセンスは、プログラムを構築時に変更する目的で「パッチファイル」をソースコードとともに配布することを容認している場合に限り、ソースコードを修正済の形式で配布することを制限することができます。この場合、そのライセンスは修正済のソースコードから構築されたソフトウェアの配布を明示的に許可してなければなりません。またライセンスは派生ソフトウェアにオリジナルソフトウェアと異なる名前を付けること、あるいは異なるバージョン番号を付けることを要求できます (これは妥協案です。Debian グループは全ての作者に、ファイル、ソース、バイナリについての変更を制限しないよう奨めています)。

パッチを配布するときに許可が必要とか、ソフトウェアのバージョンを替えてはいけないうこととす。

この項目に合わないライセンスの一つとして AT&T 公衆利用許諾契約書 (AT&T Public License) があります。このライセンスはパッチを公開するときには連絡しないといけません。

- すべての個人、団体の平等ライセンスは、すべての個人や団体を差別してはなりません。

Debian には入れるな！とか、そんなライセンスはだめといふことです。

- 目標分野の平等

ライセンスは、人々が特定の目標分野でプログラムを利用することを制限してはいけません。たとえば、商用利用や、遺伝学の研究でのプログラムの使用を制限してはいけません。

商業のみでしか使えないライセンスや研究目的のみで使用可能なライセンスではだめといふことです。

この項目に合わないライセンスとして Jahia コミュニティソースライセンス (Jahia Community Source License) があります、このライセンスは学術のみで使用可能なライセンスになっています。

- ライセンスの配布

プログラムに付随する権利は、プログラムが再配布されたすべての人々に対して、追加ライセンスの履行を必要とすることなく、適用されなければなりません。

- ライセンスは Debian に限定されない

プログラムに付随する権利は、プログラムが Debian システムの一部であるかどうかには左右されてはいけません。プログラムが Debian から取り出され Debian とは別に使用または配布されるとしても、その他の点でそのプログラムのライセンス条項を満たしているならば、プログラムが再配布されたすべての当事者は Debian システムにおいて付与されたのと同じ権利を与えられなければなりません。

Fedora なら AT&T ライセンスだけど、Debian にインストールされるなら GPL にしていいよ といったライセンスでは不適合といふことです。また、Debian 専用のライセンスではいけないうこととす。

- ライセンスは他のソフトウェアを侵害しない

ライセンスは、そのソフトウェアとともに配布される他のソフトウェアに制約を加えてはなりません。たとえば、同じ媒体で配布される他のソフトウェアがすべてフリーソフトウェアでなければならないと要求してはいけません。

- フリーなライセンスの例

”GPL”、”BSD”、および ”Artistic” ライセンスは私たちが「フリー」と判断しているライセンスの例です。

他のライセンスに関しては <http://www.debian.org/legal/licenses/> に書かれています。

以上の 9 つの項目全てに該当するパッケージが Debian のシステムとしてインストールされています。

4.2.2 セクション

先にも書いたように Debian には大量のパッケージがあります。その中には DFSG に沿ったパッケージ以外のものや、輸出に制限があるパッケージも存在します。それらを区別するために Debian ではセクションを用いて分類しています。ここではこのセクションについて示されています。

- main セクション

main セクションに入るパッケージは以下のことを満たしたパッケージである必要があります。

- DFSG に準拠したパッケージであること。
- コンパイル時に main に含まれないパッケージに依存してはいけない。
- バグだらけなパッケージであってはいけない。
- Debian Policy マニュアルに全て適合していないといけない。

main セクションに入っているパッケージは DFSG に準拠したパッケージであり、それらのパッケージは Debian システムの一部です。

- contrib セクション contrib セクションに入るパッケージは以下のことを満たしたパッケージである必要があります。

- DFSG に準拠したパッケージであること。
- バグだらけなパッケージであってはいけない。
- Debian Policy マニュアルに全て適合していないといけない。

このセクションに入るパッケージの例としてコンパイルや実行するときに Debian に存在しないものを必要とするパッケージ。フリーではないプログラム用のラッパーパッケージや、フリーではないプログラム向けのフリーな付属物などが当てはまります。

contrib セクションに入っているパッケージは Debian システムとして認められていません。

実際のパッケージでは

- atokx
理由：atok for linux をインストールするプログラム
- quake2
理由：ゲームをするために Quake 2 の CD が必要（データがフリーではない。）

があります。

- non-free セクション

DFSG に準拠していないパッケージや、特許や法律上、問題のあるパッケージが non-free セクションに入ります。

must meet all policy requirements presented in this manual that it is possible for them to meet.

実際のパッケージとして

- fglrx-driver
ATI のデバイスドライバ
- lha
lzh アーカイバー

があります。

non-free セクションに入っているパッケージは Debian システムとして認められていません。

- non-US セクション

sarge から non-US セクションが廃止され、同じアーカイブに収録されることになりました。現在、non-US のパッケージや apt-line は存在しません。

4.2.3 著作権の問題について

Debian にあるパッケージは著作権を示すファイルもポリシーとして決められています。

Debian では全てのパッケージがインストールされたときに、著作権やライセンスが `/usr/share/doc/ipackage-name/copyright` として配布されないといけません。しかし、そのようなことができないパッケージは non-free に分類されるべきであると示されています。また、バイナリのみ配布は禁止されており、Debian の FTP にもミラーにも置いてはいけないことが示されています。

国際著作権法も挙げられており、著作権が明記されていないプログラムにも著作権が存在し、このようなプログラムに手を加えることによって著作権侵害で訴えられることもありえるのでこれらのプログラムは注意すべきであるとも書かれています。

4.2.4 サブセクション

パッケージをさらに種類別に分割したものです。main セクションと contrib セクション、non-free セクションにはさらにサブセクションが設けられています。このサブセクションは control ファイルの Section レコードに指定する必要があります。

main セクションの場合、サブセクションが x11 であれば Section: x11

contrib セクションの場合は Section: contrib/x11

non-free セクションの場合は Section: non-free/x11

と指定する必要があります。

現在指定できるサブセクションは以下の通りです。admin, base, comm, contrib, devel, doc, editors, electronics, embedded, games, gnome, graphics, hamradio, interpreters, kde, libs, libdevel, mail, math, misc, net, news, non-free, oldlibs, otherosfs, perl, python, science, shells, sound, tex, text, utils, web, x11.

これらのサブセクションの分類方法は規定されていません。(base サブセクション以外) (ほんとかな?)

4.2.5 プライオリティ

それぞれのパッケージにプライオリティが設定されるべきであるとポリシーに示されています。このプライオリティは Debian システムでのパッケージの重用度を定めています。この情報は Debian のパッケージ管理ツールが優先順位の高いパッケージを優先順位の低いパッケージから選択する際に使用します。

required Debian のシステムに必要なパッケージにあたえられる重要度です。例えば base-passwd とか。アンインストールしようすると警告メッセージが表示されます。

important どんな Unix ライクなシステムにおいて存在することを期待されているプログラムはこの重要度を指定すべきです。しかし、X-Window-system や Emacs の大規模なプログラムは含まれません例えば、manpages パッケージがあります。^{*1}

standard スタンダードなアプリケーションがこの重要度を指定すべきです。perl や python , Emacs など。

optional X-window-system などがこの重要度が指定すべきです。とりあえず、入れとくか程度のもののようです。optional なパッケージは互いに conflict しないように設定しないといけないようです。

extra プライオリティとして required , important ,standard ,optional のいずれかに指定されている他のパッケージと衝突するパッケージはこの重要度が指定されます。しかし、Conflicts で指定している extra のパッケージもあれば、指定していないパッケージもあります。また、パッケージは自分のプライオリティより低いプライ

^{*1} こういうのは人によって異なると思うのですが。

オリティがあたえられたパッケージに依存してはいけません。(ビルド時の依存は除きます)。

4.3 バイナリファイル

Debian では、dpkg と呼ばれるパッケージ管理システムをベースにしています。よって、Debian で配布される全てのパッケージは.deb 形式で提供しなければなりません。

ここではこの deb 形式での配布方法等について示されています。

4.3.1 パッケージ名

全てのパッケージ名は Debian アーカイブ内において重複しない名前でないといけません。パッケージ名はアルファベット小文字、数字 (0-9) と +,1, ピリオド (.) のみで構成されてないといけません。詳細は Debian ポリシーのセクション 5.6.7 で定義されています。

4.3.2 パッケージのバージョンについて

全てのパッケージはコントロールファイルの Version フィールドに書かれている必要があります。バージョンについては Debian Policy の 5.6.12 節で説明されているので、次回あたりで詳細を説明します。

日付によるバージョン番号のつけ方についても決められています。これはスナップショットでリリースされているアプリケーションにバージョン番号をつけるときに使用します。YYYYMMDD の形式でバージョン番号を付けるようにすべきであると書かれています。

4.3.3 パッケージのメンテナについて

ここでは、パッケージメンテナについて示されています。

全てのパッケージには必ず一人以上のメンテナを持たないといけなく、連絡可能なメールアドレスを持たなければなりません。グループでメンテナンスすることも可能ですが、この場合でも共通の一つのメールアドレスを持つ必要があります。

メンテナは control ファイルの Maintainer フィールドに正しい名前と連絡可能なメールアドレスを指定します。メンテナが複数のパッケージをメンテナンスしているときは、パッケージ毎に異なった名前やメールアドレスを使うことはやめて、同じものを使うことが推奨されています。

メンテナがパッケージをメンテナンスすることを辞めた場合、他のメンテナが見つかるまで Debian QA グループがメンテナンスを引き継ぎます。このようなパッケージは orphaned (みなしご) パッケージと呼ばれます。

4.3.4 パッケージの説明について

ここではパッケージの説明文について示されています。

全てのパッケージには control ファイルの description フィールドに説明文が記入されている必要があります。簡単なパッケージの説明を記入するラインは半角 80 文字以内である必要があります。詳細な説明文を記入するエリアがあり、ここは上の簡単なパッケージの説明文とわけて書く必要があります。内容はパッケージが何をするか、Debian システムにどのような機能を追加するのかを書くべきであると示されています。

ソフトウェアの公式サイトに書いてある説明文をそのまま書くと、わかりにくいと BTS が来るときもあるのでよく考えて書きましょう。

4.3.5 依存関係について

ここにはパッケージの依存関係について示されています。

全てのパッケージはパッケージそれぞれが正常に動作するために必要なパッケージパッケージを依存情報として指定されている必要があります。

動作に必要なライブラリなどを依存情報として指定しておかないと、プログラムをインストールしても正常に動

作しないからです。例外もあって、Essential が指定されているパッケージは依存情報に指定する必要はありません。(2.8 で説明します。)

また、あるパッケージが、それをインストールする際に別のパッケージがインストールされ、且つ設定されている必要があるときがあります。この場合、そのパッケージには Pre-Depends フィールドに指定する必要があります。

例えば、coreutils パッケージがインストールされる場合に、libacl1 パッケージや libc6 パッケージがインストールされている必要があるので、coreutils の control ファイルの Pre-Depends フィールドには libacl1 や libc6 が指定されています。

この Pre-Depends は勝手に設定していいものではなく、debian-devel@list.debian.org でその設定が正しいものなのか、必要なものなのか議論して決めるべきですと書かれています。

4.3.6 バーチャル (仮想) パッケージについて

同じような機能を持つパッケージを仮想パッケージとして定義する方法について示されています。例えば、httpd の機能を持ったパッケージは apache や boa, lighttpd などがあります。これらのパッケージで提供される機能は同じようなものであり、これらをまとめて、仮想パッケージとして定義することで想定できるパッケージをずらず書かなくても良くなります。

例えば、これらの機能を必要とするパッケージ、例えば viewcvs などは httpd の機能が必要なのですが、コントロールファイルの Depends フィールドに httpd と書くだけでよくなります。

仮想パッケージは物理的には存在せず、論理的に存在します。例えば、apt-get install httpd と実行すると httpd を仮想パッケージとして指定している (コントロールファイルの Provides フィールドで指定) パッケージがずらずらと表示されます。

仮想パッケージ名は勝手に決めてはいけません。debian-devel@list.debian.org で議論する必要があると思います。現在指定可能な仮想パッケージ名は /usr/share/doc/debian-policy/virtual-package-names-list.txt.gz に書かれています。

4.3.7 ベースシステムについて

Debian のベースシステムについて示されています。Debian のベースシステムは Debian GNU/Linux システムとして最小のパッケージで構成されています。これらのパッケージのほとんどはプライオリティが required か important で、Essential が指定されています。また、Section フィールドに base が指定されています。

勝手に Section フィールドに base を指定してはいけません。debian-devel@list.debian.org で議論して同意を得る必要があります。

4.3.8 エッセンシャルパッケージについて

エッセンシャルパッケージのポリシーについて示されています。エッセンシャルパッケージとは Debian GNU/Linux システムとして必要不可欠なパッケージのことを指します。Essential が指定されているパッケージは control フィールドに Essential: yes が指定されており、Debian のシステムとして必ず必要なパッケージであることを示します。

勝手にパッケージ Essential を指定してはいけません。debian-devel@list.debian.org で議論して同意を得る必要があります。

4.3.9 メンテナスクリプトについて

ここでいうメンテナスクリプトとは、パッケージのインストールの際に実行されるスクリプトのことを指します。debconf を使ったり、オリジナルのスクリプトを使ったユーザーへのデータ入力方法や制限が示されています。

インストールする際に毎回設定をユーザーに対して質問するのではなく、設定ファイルをうまく用いて行うよう努力するべきであると書かれています。また、設定ファイルは /etc の適切な場所に置く必要があり、このことについてのドキュメントも書く必要があると書かれています。質問を行うためのプログラムを呼び出すスクリプトも postinst

か config にすべきであり、インストールに失敗したときにも適切な処理が行われるように設計されている必要があると書かれています。

5 Debian multimedia project

上川



Debian には Multimedia Project というサブプロジェクトがあります。そこでは、Multimedia 関連のツールについての議論や調整が行われています。そこで議論されている内容は大きく、ビデオとオーディオに分割できます。その内の、上川が担当している、オーディオ関連について現状何がなされていて、何ができるようになっているのかを説明します。

5.1 agnula/DeMuDi

Debian Multimedia Distribution というプロジェクトがあります。これは Debian に、リアルタイムカーネルを追加し、いくつかのパッケージをカスタマイズして作成したものです。Debian 用語でいう、「CDD:Custom Debian Distribution」の一つで、インストール直後からオーディオアプリケーションが使える便利なディストリビューションです。

Debian 本体とパッケージ自体はあまりかわりませんが、一部のパッケージが追加されております。

5.2 Debian multimedia policy

multimedia 関連のツールを利用する際には、複数のツールを相互に作用させる必要があり、相互作用のための規格は Debian 外部で活発に議論されていまして。たとえば linux-audio-dev メーリングリスト周辺では音楽関連のセッション管理や相互通信のための規格が議論されています。

Debian に関しては過去それぞれのアプリケーションが独立しており、まだポリシーを策定できるような状況ではありませんでした。ただ、最近はいろいろとツールも出そろって来た感があるため、そろそろ相互運用を考えたポリシーの策定が必要になって来ています。

ただ、多くのソフトウェアが準拠できないような高いハードルを設定しても、各パッケージが従わないだけなので、現状準拠することに議論が出なさそうな部分から標準ポリシーとして策定していこうと画策しています。

一番策定したいのは、プラグイン機構、相互接続、デフォルトのオーディオと MIDI デバイスの指定の部分について Debian で統一した操作感を提供する部分です。

5.3 今 Debian でできること

Debian を Digital Audio Workstation (DAW) のプラットフォームとして音楽活動をしようとするとなんができるのか、なんができないのか、追求してみようと思います。

現在存在しているアプリケーションはおおまかに、下記に分類できます。

- フレームワーク系、相互運用のために必要な基本的なドライバなど。ALSA や jack、ladspa など。
- MIDI(音譜) 編集系
- マルチトラック/音声編集系
音声データを受け取り、録音し、編集し、音声データを出力するもの
- ソフトウェアシンセサイザー
MIDI を入力として受け取り、音声を出力するもの
- エフェクト

LADSPA プラグイン。

5.4 MIDI コネクション

音符情報を扱うための統一企画として、MIDI があります。MIDI は、音程と音量を指定して発声を指示するための通信プロトコルで、昔から使われているものです。現在の電子楽器などではほとんど利用できます。

また、ALSA の MIDI シーケンサ機能を利用すると、仮想的にアプリケーションとアプリケーションを接続して、情報の通信プロトコルとして MIDI を利用することができます。

GUI で操作する場合は、qjackctl などで制御します。

5.5 JACK で接続する

発声と録音の経路について、UNIX らしく、各アプリケーションがそれぞれの役割をもつ、という場面を考えてみると、MIDI のような通信プロトコルが必要になります。

jack はそのプロトコルを提供します。

オーディオアプリケーションは jackd というデーモンを介して相互に音声データを送受信します。また、リアルタイム (録音しながらそれを再生しながら処理) に処理をするために工夫がこらされています。

5.6 LADSPA:エフェクトをかける

音声を扱う各アプリケーションはそれぞれ音声に対してエフェクトをかけることができます。一般的にはディレイや、ディストーション、コンプレッサーなどがあります。それらを、各アプリケーション毎に再実装するのは無駄なため、共有化しようということで生まれたのが LADSPA という規格です。

LADSPA という規格にのったプラグインがそれぞれのエフェクト提供し、各アプリケーションがそれを利用してエフェクトをかけます。

現状 LADSPA エフェクトを利用したり提供できるパッケージを確認するには、ladspa-plugin を Provides: していたり、Recommends:しているパッケージを確認すればよいです。^{*2}

swh-plugin などはいいエフェクトだと定評があります。

```
> apt-cache showpkg ladspa-plugin
Package: ladspa-plugin
Versions:

Reverse Depends:
  sweep,ladspa-plugin
  ecasound,ladspa-plugin
  xmms-ladspa,ladspa-plugin
  terminatorx,ladspa-plugin
  sweep,ladspa-plugin
  snd,ladspa-plugin
  rosegarden4,ladspa-plugin
  glame,ladspa-plugin
  galan,ladspa-plugin
  ecasound,ladspa-plugin
  audacity,ladspa-plugin
Dependencies:
Provides:
Reverse Provides:
tap-plugins 0.7.0-2
swl-plugins 0.4.14-1
ladspa-sdk 1.1-4
cmt 1.15-3
caps 0.3.0-1
blop 0.2.8-3
```

5.7 Linux オーディオ処理におけるリアルタイムの必要性

オーディオデータを入力したものを加工して再生すると、その間に処理遅延が生じます。処理遅延のバッファとして 1024 フレーム利用するとしてみましょう。通常の 44.100kHz のオーディオデータで 1024 フレームということは、23

^{*2} 今気づいたのですが、現状その依存関係をもっていないパッケージがあるため、現実より少ない数のパッケージしか出て来ません。

ミリ秒程度です。これは、23 ミリ秒分のデータを取得して、jackd に接続している全プロセスがそのバッファに関連した必要な処理をして、23 ミリ秒以内に出力用バッファに配置するということを意味します。23 ミリ秒以内に配置できなかった場合にはその回の音声は途切れ、ユーザはブチという音を効くことになります。この状態を ALSA 用語では「xrun」と呼びます。^{*3}

実際問題として、Linux をそのまま利用していると、23 ミリ秒以内に絶対に全プロセスが処理を完了するということは難しいです。

最近の Linux カーネルは下記の追加で改善してきてはいますが、まだまだ進歩が求められています。特に、ライブやレコーディングでは、数時間にわたって jack の処理が時間内に終了しないということが一度でもあってはならないという条件になるため、それなりにチューニングの手腕が求められます。

- カーネルが持っている長時間の処理を分割
- preempt 対応により、カーネル空間の処理でもタイムスライスにより CPU を明け渡すようになった
- リアルタイム対応により、SCHED FIFO などのスケジューリングに対応、オーディオ関連のスレッドの優先度をあげることができるようになった

^{*3} 実際は period 数を増やすことでダブルバッファリングみたいなことをしているため、この限りではないようだが、簡単に原理だけは伝わただろうか。

6 Debian Multimedia Audio application 概観

上川



現在 Debian に存在しているアプリケーションにどのようなものがあるかみてみましょう。ここにあるリストは全てを網羅しているわけではなく、また、途中であきらかに力尽きてます。今後また続きをやります。

6.1 フレームワーク系

Debian では音楽関連のフレームワーク系も独自に管理しています。この関連について議論する場所は debian-multimedia@lists.debian.org メーリングリストです。

6.1.1 ALSA

Linux のオーディオの次期標準といわれつつけて早何年目か。カーネルモジュール (最近は標準) とユーザランドのライブラリ (libasound2) といくつかのツールがあります。

```
$ aplay -l
**** ハードウェアデバイス PLAYBACK のリスト ****
カード 0: IXP [ATI IXP], デバイス 0: ATI IXP AC97 [ATI IXP AC97]
  サブデバイス: 1/1
  サブデバイス #0: subdevice #0
カード 0: IXP [ATI IXP], デバイス 1: ATI IXP IEC958 [ATI IXP IEC958 (AC97)]
  サブデバイス: 1/1
  サブデバイス #0: subdevice #0
カード 2: Device [KC USB Audio Device], デバイス 0: USB Audio [USB Audio]
  サブデバイス: 1/1
  サブデバイス #0: subdevice #0
```

6.1.2 jack-audio-connection-kit

各音楽関連のアプリケーションが利用する音声経路ルーティングプロトコルです。jackd というデーモンが、利用しているユーザの権限で起動し、それを経由して通信します。

コマンドラインで起動する場合は

```
jackd -d alsa -d デバイス名 -r サンプルレート
```

のように指定します。

```
jackd -d alsa -d ixp -r 48000
```

ポートの接続はコマンドラインからでも操作できます。

jackd には複数の ALSA サウンドカードを同時に使えないという制限があります。そういう場合は現状としては、ecasound などの jack と ALSA 対応のアプリケーションをかましてしのいでいます。ALSA 側の機能で対応することでもできます。

```
$ jack_lsp
alsa_pcm:capture_1
alsa_pcm:capture_2
alsa_pcm:playback_1
alsa_pcm:playback_2
$ ecasound -i alsaplugin,2,0,0 -o jack_generic,usbaudio &
$ jack_lsp
alsa_pcm:capture_1
alsa_pcm:capture_2
alsa_pcm:playback_1
alsa_pcm:playback_2
alsa_pcm:playback_3
alsa_pcm:playback_4
alsa_pcm:playback_5
alsa_pcm:playback_6
ecasound:usbaudio_1
ecasound:usbaudio_2
$ jack_connect ecasound:usbaudio_1 alsa_pcm:playback_1
$ jack_connect ecasound:usbaudio_2 alsa_pcm:playback_2
```

類似したシステムとして esound^{*4}や arts^{*5}、NAS^{*6}がありますが、それらは音をできるだけ切らせないことを主眼としているため、余裕をもったバッファを利用しており、レイテンシの問題でほとんどのオーディオアプリケーションでは利用できません。

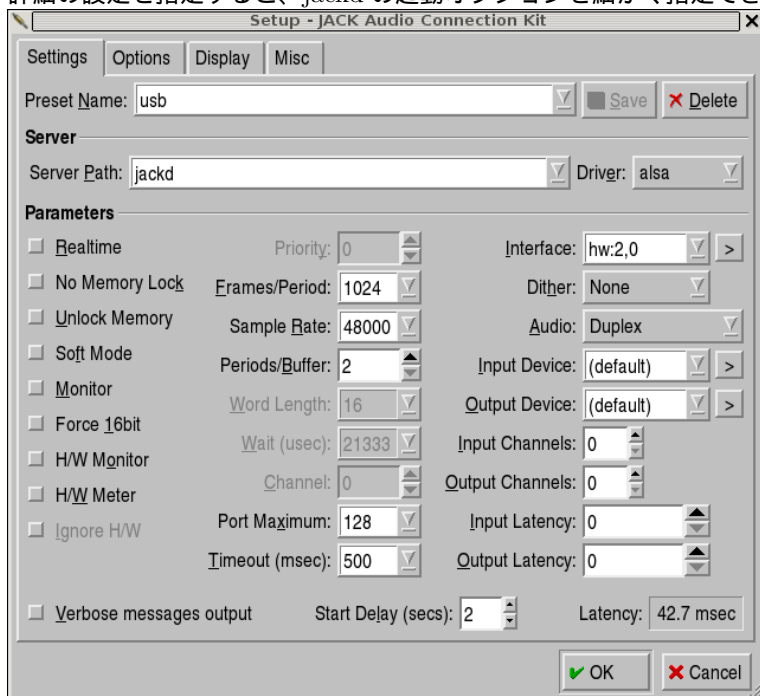
6.1.3 qjackctl

ポートの接続や、jackd の起動/停止は、qjackctl で GUI 経由で操作できます。

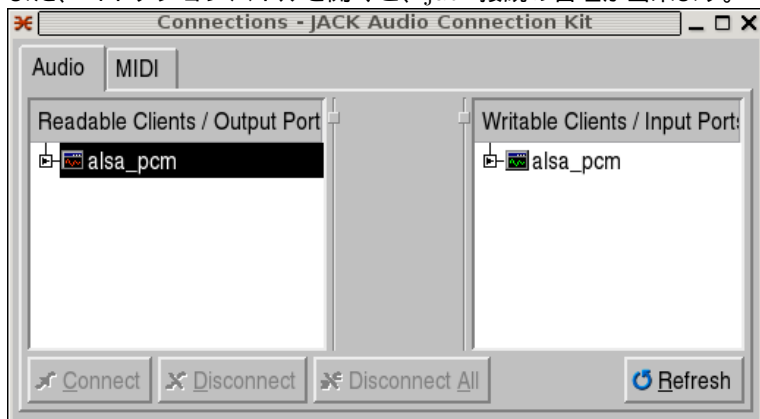
まず、起動したら、パネルが起動します。



詳細の設定を指定すると、jackd の起動オプションを細かく指定できます。



また、コネクションパネルを開くと、jack 接続の管理が出来ます。



^{*4} GNOME

^{*5} KDE

^{*6} ネットワーク経由を主目的としたオーディオプロトコル

6.1.4 ladspa

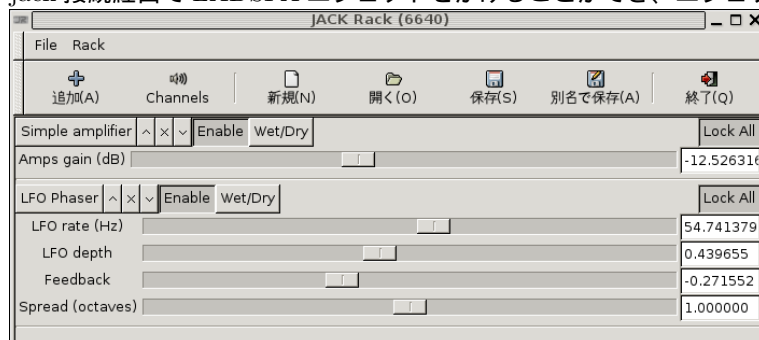
オーディオのエフェクトを処理するための、プラグインインタフェースです。また、ladspa-dev パッケージが存在しており、そのパッケージに含まれている /usr/include/ladspa.h を利用することが推奨されています。LADSPA 自体がポリシーを定義していますが、Debian の ladspa パッケージは、追加で /usr/share/doc/ladspa-sdk/README. Debian にて定義されている下記のポリシーにしたがっています。

/usr/lib/ladspa/ にパッケージが提供する LADSPA プラグインを提供すること。LADSPA_PATH 環境変数が定義されていない場合には、/usr/local/lib/ladspa:/usr/lib/ladspa をデフォルトの検索パスとして利用すること。

6.1.5 jack-rack

apt-get install jack-rack でインストールできます。

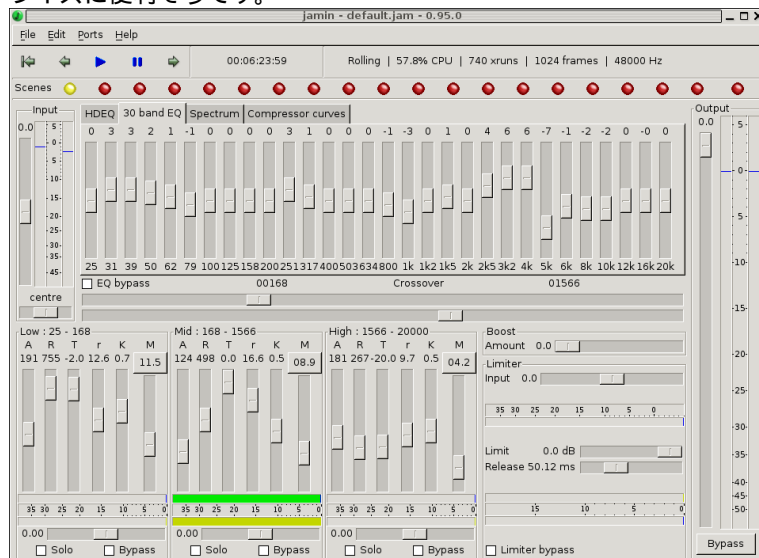
jack 接続経由で LADSPA エフェクトをかけることができ、エフェクトのパラメータを GUI で制御できます。



6.1.6 jamin

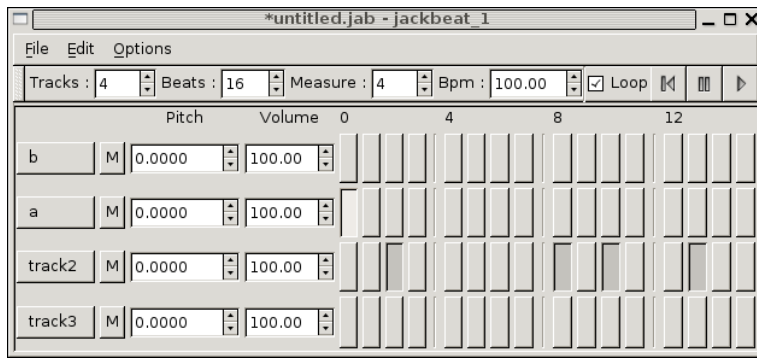
apt-get install jamin でインストールできます。

jack 経由で細かいイコライザーやコンプレッサーの設定が出来るツールです。レコーディングの最終段階のフィナライズに便利そうです。



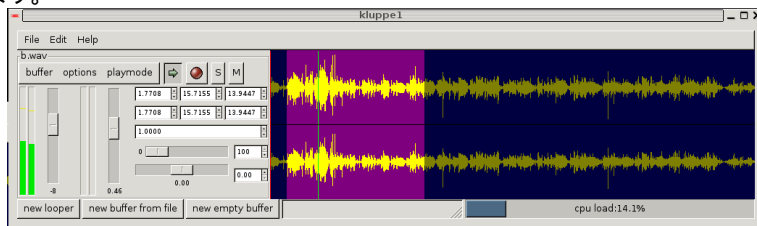
6.1.7 jackbeat

apt-get install jackbeat でインストールできます。wav ファイルをリズムループ用のサンプルとして利用して、単純なドラムシーケンサーとして利用できるようです。



6.1.8 kluppe

apt-get install kluppe でインストールできます。jack 経由で接続し、wav ファイルをループさせることができます。



6.1.9 ladcca

LADCCA というフレームワークが存在しているようです。気づいたら [lash http://www.nongnu.org/lash/](http://www.nongnu.org/lash/) というプロジェクトにかわってしまっているようです。セッション管理のためのフレームワークです。jack の導入にともない、アプリケーションが接続できるのはよいのですが、毎回アプリケーションをユーザが接続する必要があります。その処理を簡便化するためのプロトコルのようです。

6.2 ソフトウェアシンセ

MIDI の接続は ALSA の MIDI 接続が事実上の標準プロトコルとして利用されています。qjackctl の接続画面に MIDI 接続タブがあるので、それを利用して接続してあげればよいです。また、aconnect というコマンドラインインタフェースがあり、それを利用することも可能です。

```
$ aconnect -i -l
クライアント 0: 'System' [タイプ=カーネル]
  0 'Timer'
  1 'Announce'
    接続先: 15:0, 128:0
クライアント 14: 'Midi Through' [タイプ=カーネル]
  0 'Midi Through Port-0'
クライアント 130: 'Virtual Keyboard' [タイプ=ユーザ]
  0 'Virtual Keyboard'
    接続先: 129:0
$aconnect -o -l
クライアント 14: 'Midi Through' [タイプ=カーネル]
  0 'Midi Through Port-0'
クライアント 129: 'FLUID Synth (7910)' [タイプ=ユーザ]
  0 'Synth input port (7910:0)'
    接続元: 130:0
```

6.2.1 TSE3

シーケンサエンジンのようです。KDE 関連のアプリはこれを利用しているような気がしています。どうなのよ。

6.2.2 timidity

ちょっとインタフェースに古めかしい感がありますが、事実上の標準の MIDI シーケンサエンジンです。MIDI データから WAV を生成するためのインタフェースとして利用されています。

6.2.3 freepats

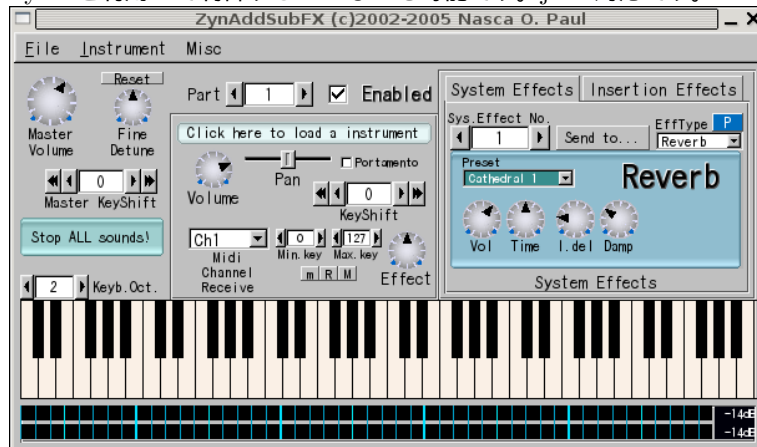
Debian にて、フリーのサウンドフォント集です。形式が pat 形式です。timidity から使える設定になっているようです。

sf2 形式じゃないとほとんどのアプリから利用できないので意味が無い。

6.2.4 zynaddsubfx

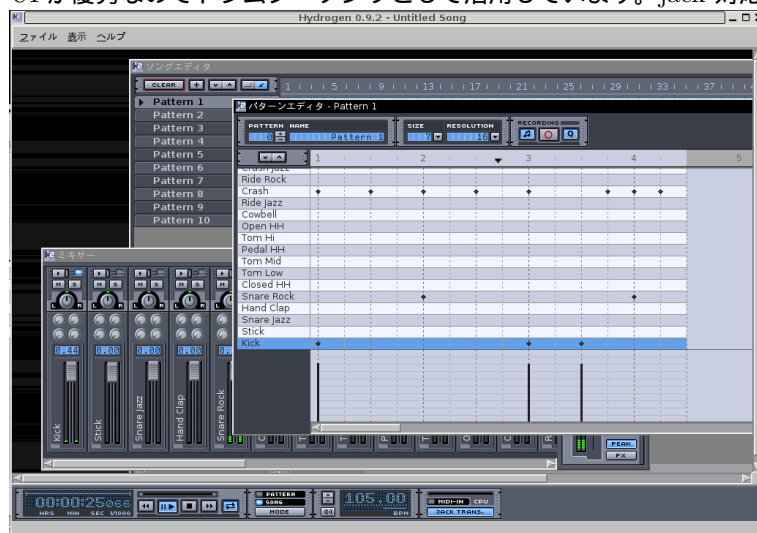
apt-get install zynaddsubfx でインストールできます。

オルガン系の音やパッド系の音が結構使えます。仮想キーボードの UI がお手軽です。MIDI 制御可能なため、vkeybd を利用して制御することなども可能です。jack 対応です。



6.2.5 hydrogen

UI が優秀なのでドラムシーケンサとして活用しています。jack 対応です。MIDI 入力にも対応しているようです。



6.2.6 pd

UI がかなり前時代的ですが、シンセを GUI で編集するという系では元祖みたいな存在です。使い方がわかりません。誰か教えてください。

6.2.7 beast

GTK シンセ。これも結構頑張っています。使い方がわかりません。誰か教えてください。

6.2.8 csound

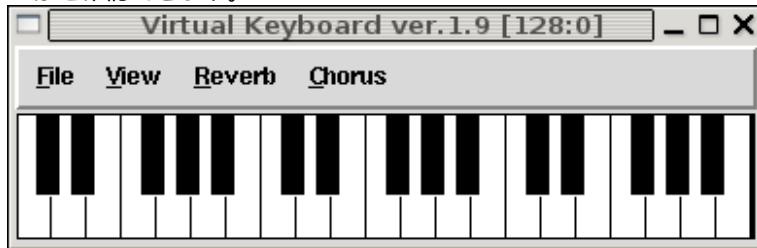
学術系の人々の中で長い間つかわれてきたものらしく、過去の遺産が大量にあります。ちょっと学術的すぎて個人的には使っていません。誰か使い方教えてください。

コマンドラインで操作できる、というよりむしろプログラム言語です。

方程式で波形を設計したいあなたに。

6.2.9 vkeybd

キーボードで操作できる MIDI キーボードです。alsa の MIDI デバイスとして動作します。とりあえず試すのには便利です。こいつを起動して、qjackctl で接続してあげれば他のアプリケーションが MIDI 入力にどういう反応をするのかを確認できます。



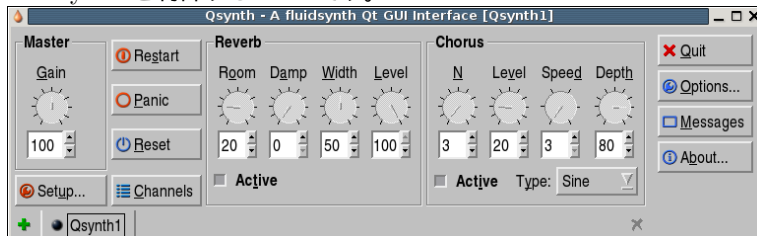
6.2.10 fluidsynth

ソフトウェアシンセのようです。sf2 形式のファイルをサポートしているようです。<http://www.hammersound.net/> などに多数のサウンドフォントが存在していて、そのうちの適当なファイルを読み込んで利用することが出来ます。jack へ音声を出力することが可能です。

Debian 内で、sf2 のファイルが見付かりません。ウェブを探すとたくさんあるようです。

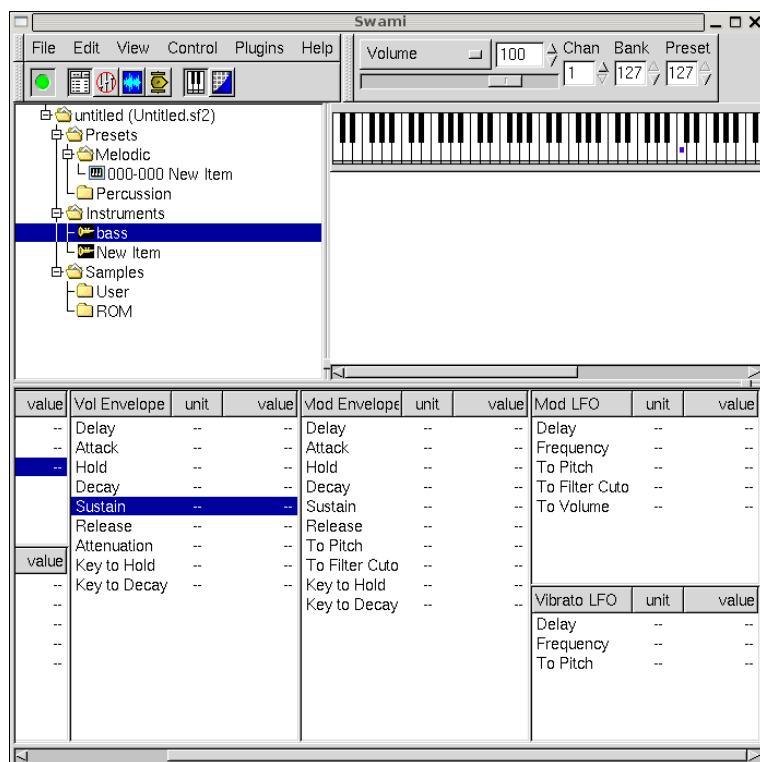
6.2.11 qsynth

fluidsynth を制御する GUI です。



6.2.12 swami

サウンドフォント (sf2 ファイル) を編集するツールです。fluidsynth を内部では利用しています。



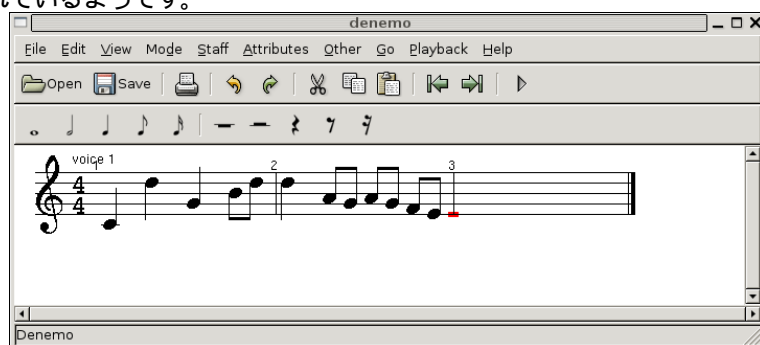
6.3 楽譜編集

6.3.1 lilypond

$\text{T}_{\text{E}}\text{X}$ で楽譜を作成しよう、というパッケージ。まともなクラシックの楽譜を作成するような作業をする際にはこれでやってました。 $\text{T}_{\text{E}}\text{X}$ をがんがん使いたいあなたに。

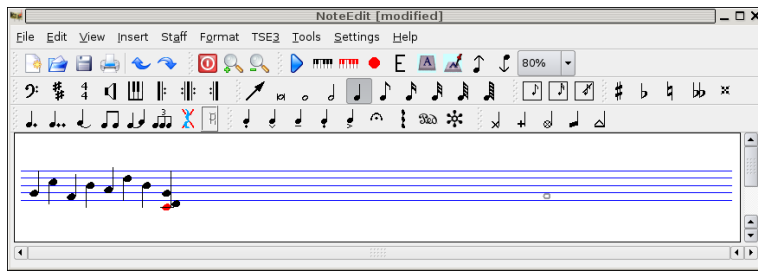
6.3.2 denemo

今までは上川はこれで一小節程度の楽譜ならちょこちょっと作成して用を足して来ました。キーバインドも数字で音符の長さが決まっていたり、キーの上下で操作できます。久しぶりに見てみるとインターフェースが大幅に改善されているようです。



6.3.3 notedit

MIDI のインポートもできるらしい。とりあえず楽譜を表示することはできるっぽい。

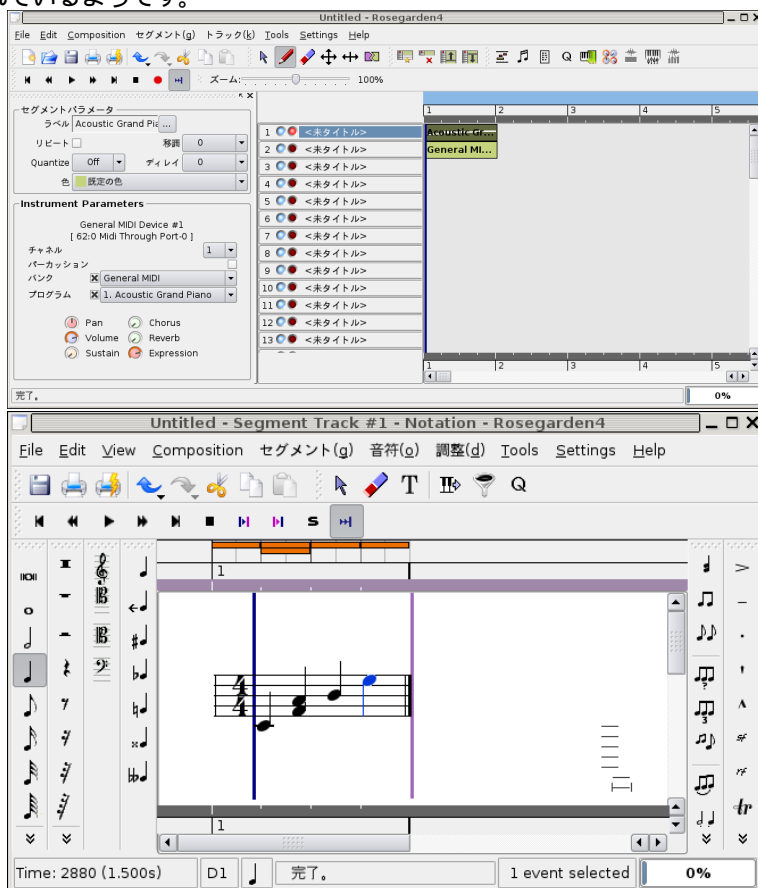


6.3.4 rosegarden4

apt-get install rosegarden4 でインストール。

一応楽譜が編集できます。ステップ録音などもできます。デバッグメッセージが大量に出て来るのとなんだか反応が鈍い感じはします。

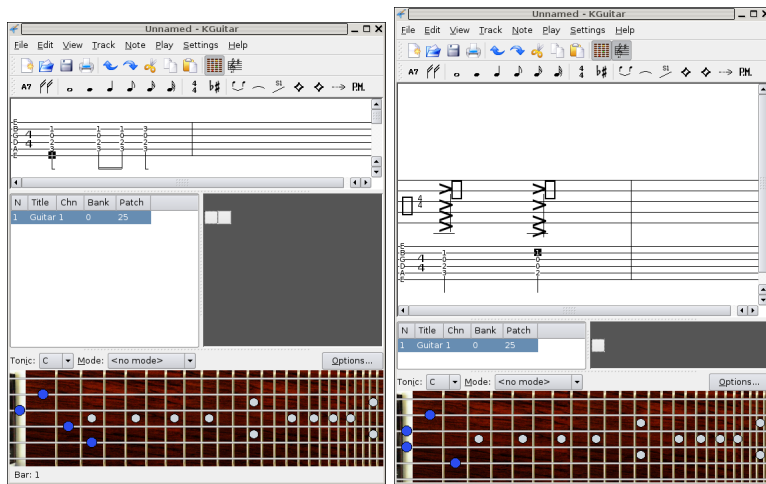
rosegarden という古くから何度も書き直されつづけているプログラムの最新版です。いまのところまともに開発されているようです。



6.3.5 kguitar

apt-get install kguitar でインストール。

ギターのタブ譜を編集できるソフトウェアのようです。使い方が分からないので、困りました。楽譜が出るはずのようですが、出てません。ギターの絵が素敵です。MIDI 入出力ができることになっているようです。



6.4 音声編集系

6.4.1 ecasound

apt-get install ecasound でインストール。

コマンドラインベースで音声加工をするためのツールです。

よく使うコマンドは

音量をノーマライズする。(可能な最大の音量まであげる)

```
$ ecanormalize in.wav
```

in.wav にコンプレッサーエフェクトをかけて、out.wav を生成する。

```
$ ecasound -i in.wav -o out.wav -eca
```

とりあえず録音する

```
$ ecasound -i alsahw,0,0,0 -o /tmp/a.wav
*****
*          ecasound v2.4.3 (C) 1997-2005 Kai Vehmanen and others
*****
- [ Session created ] -----
- [ Chainsetup created (cmdline) ] -----
- [ Connecting chainsetup ] -----
(eca-chainsetup) 'rt' buffering mode selected.
(eca-chainsetup) Audio object "alsahw", mode "read".
(audio-io) Format: s16_le, channels 2, srate 44100, interleaved.
(eca-chainsetup) Audio object "/tmp/a.wav", mode "read/write".
(audio-io) Format: s16_le, channels 2, srate 44100, interleaved.
- [ Chainsetup connected ] -----
(eca-control-objects) Connected chainsetup: "command-line-setup".
- [ Controller/Starting batch processing ] -----
- [ Engine init - Driver start ] -----
```

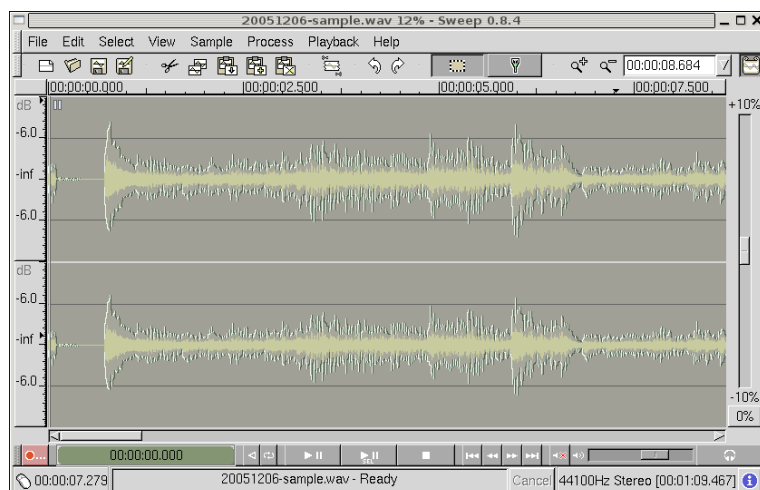
ここで ctrl-C で停止

```
- [ Controller/Processing stopped (cond) ] -----
- [ Engine exiting ] -----
(eca-control-objects) Disconnecting chainsetup: "command-line-setup".
- [ Chainsetup disconnected ] -----
- [ Controller/Batch processing finished ] -----
```

6.4.2 sweep

apt-get install sweep でインストール。

メモリ上に wav ファイルを展開するので、大きい wav ファイルは編集できないです。プリビューが優秀。ダブルクリックしたらそこから再生したりしてくれる。こまかい波形の切りだしなどに上川愛用。しかし、jack に対応していないなどの欠点があります。



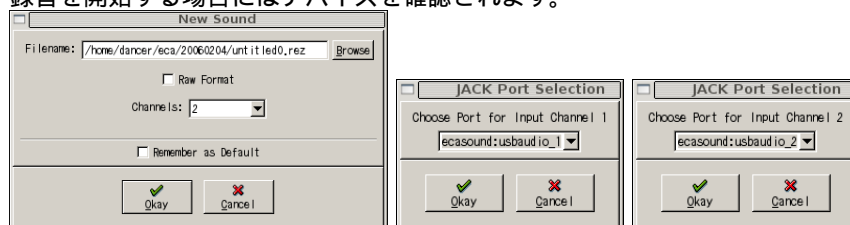
6.4.3 rezound

apt-get install rezound でインストール。

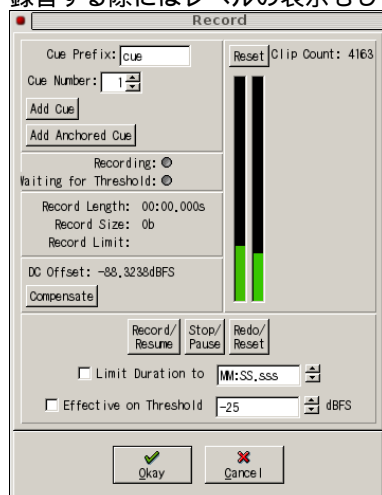
サウンドの編集用のツールです。jack 出力をサポートしています。

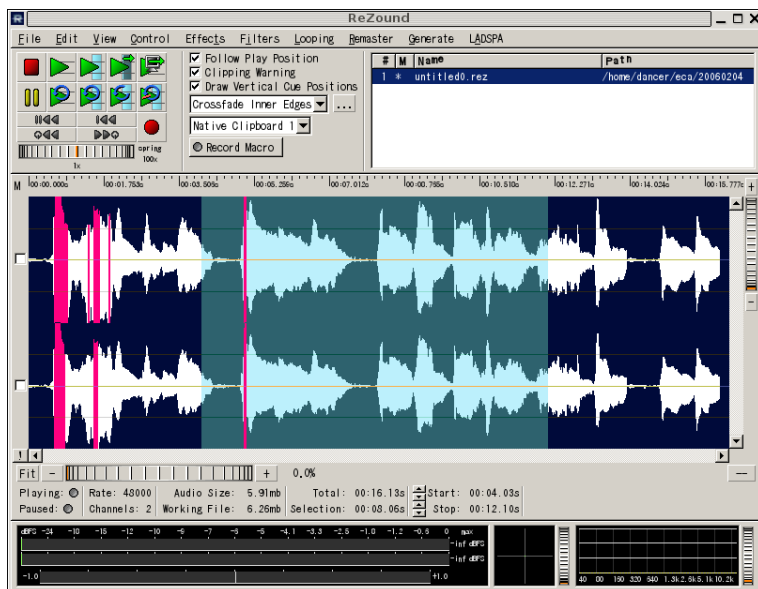
```
$ rezound --audio-method jack
```

録音を開始する場合にはデバイスを確認されます。

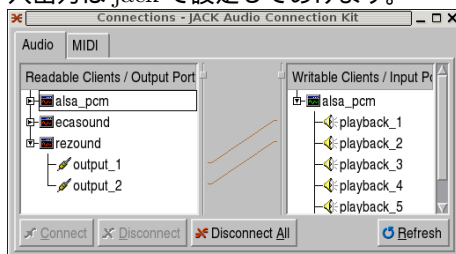


録音するにはレベルの表示もしてくれます。





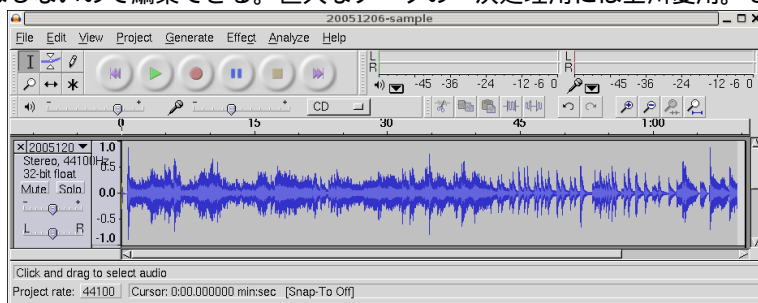
入出力は jack で設定してあげます。



6.4.4 audacity

apt-get install audacity でインストール。

audacity で起動。マルチトラックのオーディオ編集に最適。巨大な波形データもメモリ上に全てをロードしようとはしないので編集できる。巨大なデータの一次処理用には上川愛用。しかし、jack をサポートしていません。



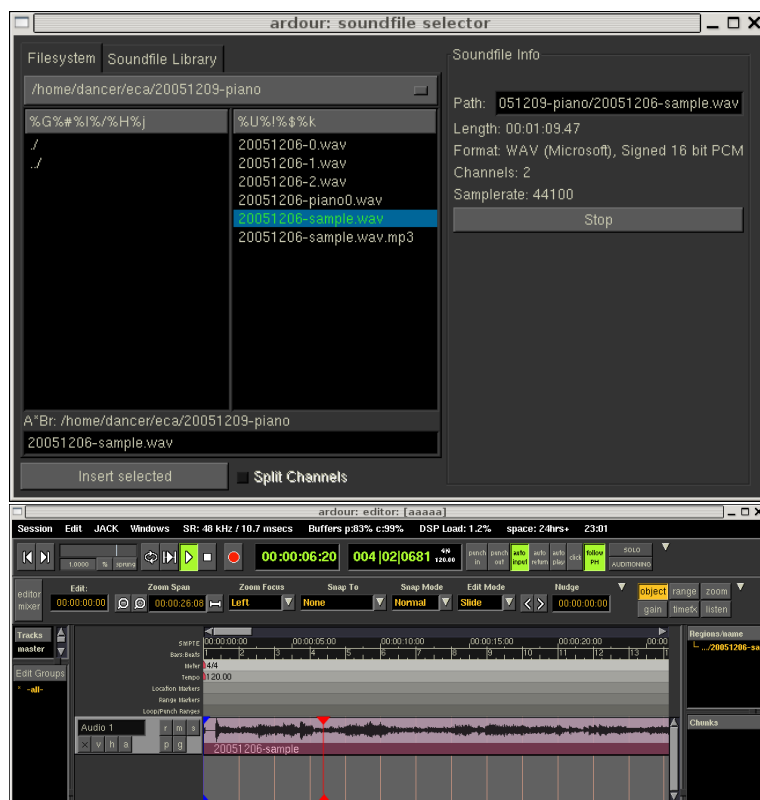
以前は日本語インターフェースを利用しようとするとは悲惨でしたが、直っているようです。まだ一部問題が残っているようですが、全く使えないほどではないです。

6.4.5 ardour

apt-get install ardour-gtk でインストール。

マルチトラックの音声ファイルは編集できるけど、音符が編集できそうな雰囲気は無いです。

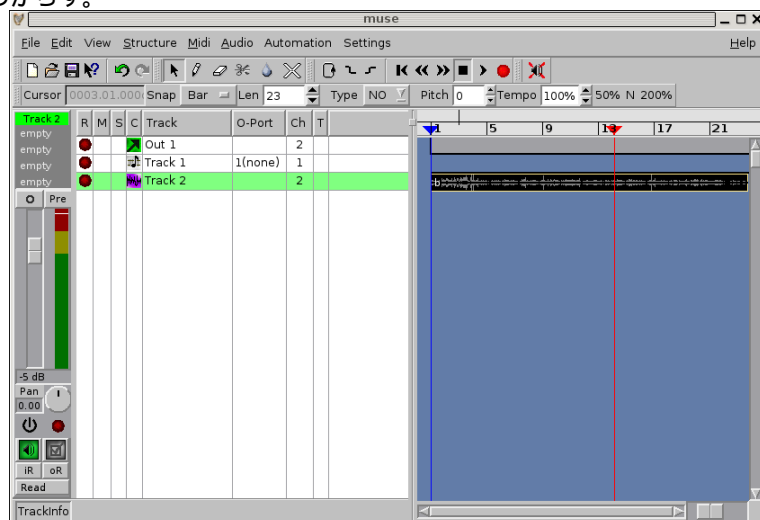
Paul Davis という人が中心に開発していて、彼はこのために jack と LADSPA を実装し、Hammerfall のサウンドカードのドライバを ALSA 用に実装したというよくわからないけど凄い代物です。



6.4.6 muse

apt-get install muse でインストール。

MIDIトラックと音声トラックが同時に扱えるようです。jack と ALSA MIDI に対応しているようです。使い方がわからず。



6.4.7 snd

音声業界での emacs と呼ばれています。使い方がわかりません。誰か教えて下さい。

7 次回



3 月はオープンソースカンファレンスにお邪魔する予定です。内容は本日決定予定です。
参加者募集はまた後程。



Debian 勉強会資料

2006 年 2 月 18 日 初版第 1 刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
