



## 第 5 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†

2005 年 6 月 11 日

---

\* 機密レベル public: 一般開示可能

† Debian Project Official Developer

## 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	5
2.1	2005 年 21 号 . . . . .	5
2.2	2005 年 22 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 5 回目報告 . . . . .	6
4	alternatives -選択せよ-	7
4.1	alternatives とは? . . . . .	7
4.2	update-alternatives の使い方 . . . . .	7
4.3	用語の説明 . . . . .	8
4.4	ユーザとして使う場合 . . . . .	9
4.5	パッケージメンテナとして使う場合 . . . . .	9
4.6	マニアックなコマンド . . . . .	10
4.7	一般的なオプション . . . . .	10
4.8	マニアックなオプション . . . . .	10
4.9	update-alternatives ってどうなってるの? . . . . .	11
4.10	動作 . . . . .	11
4.11	update-alternatives の改善案 . . . . .	12
4.12	dsys について . . . . .	13
5	debian-installer	14
6	個人提案課題	15
7	Keysigning Party	16
8	次回	17

## 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ふ方も，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

- kmuto さん debian-installer ハッカーです．
- えとーさん dsys の開発者です．
- 上川純一 宴会の幹事です．Debian Developer です．元超並列計算機やっていて，今は音楽関係とか，気づいたら canna とか．あと，pbuilder や，libpkg-guide などを通して，Debian の品質向上を目指しています．

### 1.2 事前課題紹介

今回の事前課題は「debian-installer をつかってみて/今後の debian-installer に期待するもの」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 みつかさん

少し前に使ったきりなので記憶が曖昧なのですが…。まず驚いたのが国際化が進んだなということです。各国語のメニューがずらっと並んだなかに日本語も含まれているというのはなんだかとても誇らしいような気持ちになりました。日本語を選択して進んでいくと普通にすんなりとインストールできてしまい感動しました。以前のインストーラーと比べて、ステップ数も少なくなったと思います。これも、インストールが簡単になったなという印象に寄与していると思います。

今後への期待としては、日本語を選択してインストールした場合

- jfbterm がデフォルトで起動したり
- ja\_JP.eucJP 辺りがデフォルトで設定されてたり

すると嬉しいかなあとかと思いました。

### 1.2.2 kaepapa

debian のインストーラは potato と woody そして sarge で使ってきたが徐々にインストールが容易なものになってきたと思います。個人的には woody の時のインストーラで十分満足できる状態でした。sarge になり cfdisk でなくなりちょっと残念です (経緯は知っている)。partitioner に関してはちょっと間違えると HDD 全域をパーティショニングしてしまうリスクを感じました。この点は十分な情報の提供が必要だと思います。RC3 当時の partitioner の不具合 (パーティションテーブル不全?) もなくなり (20050419) リリースされたらもう一度ためしてみたいと考えています (VMWare ゲスト)。またパッケージの選択等については以前よりインストーラでは最小構成のみインストールし後から aptitude で必要なものを導入するという癖がついているためあまり記憶に残っていませんが以前とかわらず最小構成でインストールが終れることに好感が持てます。このポリシーはずっと継続して欲しいと願っています。

### 1.2.3 やまねさん

使ってみた感想は昔に web に載せたことがあるので略。その頃からするとぐっとよくなってます :-)

- 改善して欲しいーという点。
  - やはり「X の自動認識」の弱さが気になる。初心者には鬼門。
  - パーティションの分割し辛さと RAID 機能の弱さ
  - パッケージダウンロードをしている間に他の設定をさせるなどは難しいのか? ずーっと待ってるのはちょっと。
  - 特定機種での問題 (Dell のとか) は結局解決したんだろうか。
  - 「定期的なアップデート」。たとえ stable といえど、新しい機種に対応していないインストーラーを勧められるのは「?」というところ。ここはインストーラだけでも半年に 1 回ぐらい更新を行うようなポリシーに切り替えて欲しい。
  - iptables でパケットをデフォルト遮断は難しい? WindowsXP SP2 など基本は『保護』の方向だし、この機能は欲しい
  - tasksel の分け方とか、本当に「最小インストール」するのが簡単になって欲しいところとか
- 思いつき
  - イベントのときに「d-i fest」でひたすらインストールできた・できない機種の情報収集するとか。
    - \* BTS 報告だと英語がネックで厳しいし
    - \* こんなイベントなら参加の敷居も低いし
    - \* レアマシン持ってくる人もいて、結構なデータ取れるかもしれないし
    - \* エスパー能力なくてもその場で解決できること多そうだし

ということで如何でしょうか :-)

#### 1.2.4 上川

これから書きます．

## 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 2005 年 21 号

### 2.2 2005 年 22 号

#### 問題 1.

- A
- B
- C

### 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 5 回目報告

前回開催した第 5 回目の勉強会の報告をします。

DWN クイズをしました。

松山さんが DFSG について語りました。

岩松さんが dpkg-cross の使い方について説明しました。

上川が lintian, linda の使い方について説明しました。Debian のパッケージの作成のフローを一度デモできるとよいねえ、という話しになりました。

グループディスカッションでは、今後のアーキテクチャ対応をどうしましょうか、という話しをしました。現在のアーキテクチャは CPU 毎にわけられており、CPU 以上の部分についてはパッケージでカバーしているのですが、task-i386 とかがあれば、各ハードウェアに必要なパッケージをごっそりいれてよいねえ、という案が出たりしました。全パッケージをスクリプトで再実装してしまい、各アーキテクチャの差分を減らす、という迷案も出ていました。

## 4 alternatives -選択せよ-

えとー



### 4.1 alternatives とは？

パッケージ管理という依存関係の管理についてのみがクローズアップされるが、パッケージ管理としては、それだけでは不足する部分がある、それを補うものの一つが alternatives です。

日本語訳すると「選択肢」、私の定義ですが、「複数のパッケージを特定機能ごとに一つにまとめて扱い機能ベースのパッケージの管理を提供するもの。」「パッケージを機能の視点で管理する。」「プログラムに OS の統一的な API を提供する。」の 3 点です。

機能としては、「類似の機能を持つプログラムの別名を提供する。」で、symlink を使って実現しています。alternatives とは、その symlink を管理するための機構と言えるでしょう。

なぜわざわざ別名が必要？

1. プログラムからの使い勝手の向上万単位のパッケージの存在する Debian では同じような機能を持つ別々のパッケージが複数存在することは珍しくありません。伝統的に Unix 系 OS ではパイプやリダイレクトなどやスクリプト、プログラムなどから他のプログラムの機能を利用することによりコード量の削減と質の向上を図っています。しかし、プログラムから呼び出す際に同じような機能をもつものを全てを把握し条件分岐などで呼んで行くのはコストから言って現実的ではありません。その解決策としてはプログラムへの統一的な API を提供することができるようすることです。awk スクリプトやコンパイラなどを思い浮べるといいと思います。
2. ユーザへの使い勝手の向上ユーザからしても同じ機能なのに別の別の名前であると不便な場合もあります。MUA や IRC クライアント からの Web Browser を呼び出す際を想像してもらえば想像し易いと思います。
3. menu との連携 Debian 独自なものとして menu がありますが、これと alternatives を連携させることによりメニューをいじらずに、違うパッケージを提供することができます。

### 4.2 update-alternatives の使い方

see man(重要)

alternatives の制御には /usr/sbin/update-alternatives というコマンドを使います。



書式一覧

```
update-alternatives --install リンク 一般名 パス 優先度 [--slave スレーブリンク スレーブ一般名 スレーブパス]
update-alternatives --remove 一般名 パス
update-alternatives --remove-all ディレクトリ
update-alternatives --all
update-alternatives --auto 一般名
update-alternatives --display 一般名
update-alternatives --list 一般名
update-alternatives --config 一般名
update-alternatives --set 一般名 パス
```

## 4.3 用語の説明

### 4.3.1 マスター

- マスター alternatives の対象とするもの
- リンク 一般的なリンクです  
例 /usr/bin/awk, /usr/bin/editor, /usr/bin/pager, /usr/bin/x-www-browser
- 一般名 一般的な名前です、  
例 awk, editor, pager, x-www-browser
- パス それぞれの alternatives のリンク先になります  
例 /usr/bin/awk: /usr/bin/nawk, /usr/bin/mawk, /usr/bin/gawk /usr/bin/editor : /bin/ed,  
/bin/nano, /usr/bin/vim /usr/bin/pager : /bin/more, /usr/bin/less, /usr/bin/w3m, /usr/bin/lv  
/usr/bin/x-www-browser : /usr/bin/mozilla, /usr/bin/kazehakase, /usr/bin/mozilla-firefox
- 優先度 auto モードで設定される際の選択基準になる整数値です

### 4.3.2 スレーブ

- スレーブ alternatives のマスターに付随するファイル、man など 複数指定可能
- スレーブリンク 一般的なリンクに付随するリンク  
例 /usr/share/man/man1/awk.1.gz, /usr/share/man/man1/nawk.1.gz, /usr/share/man/man1/editor.1.gz,  
/usr/share/man/man1/pager.1.gz, /usr/share/man/man1/x-www-browser.1.gz
- スレーブ一般名 一般的な名前に付随する一般名  
例 awk.1.gz, editor.1.gz, pager.1.gz, x-www-browser.1.gz
- スレーブパス それぞれの alternatives に付随するリンク先  
例  
/usr/share/man/man1/awk.1.gz, /usr/share/man/man1/nawk.1.gz : /usr/share/man/man1/mawk.1.gz,  
/usr/share/man/man1/gawk.1.gz  
/usr/share/man/man1/editor.1.gz : /usr/share/man/man1/ed.1.gz, /usr/share/man/man1/nano.1.gz,  
/usr/share/man/man1/vim.1.gz  
/usr/share/man/man1/pager.1.gz : /usr/share/man/man1/more.1.gz, /usr/share/man/man1/less.1.gz,  
/usr/share/man/man1/w3m.1.gz, /usr/share/man/man1/lv.1.gz  
/usr/share/man/man1/x-www-browser.1.gz : /usr/share/man/man1/mozilla.1.gz, /usr/share/man/man1/kazehakase.1.gz,  
/usr/share/man/man1/mozilla-firefox.1.gz

### 4.3.3 automatic と manual

alternatives は優先度によって自動的に選択する automatic モードと優先度を無視しユーザが選択する manual モードがあります。デフォルトは automatic モードで `--config` や `--set`、`--all` を使い変更した場合に manual モードに以降します。automatic モードに戻したければ `--auto` を使います。

## 4.4 ユーザとして使う場合

ユーザとして主に使用するコマンド

### 4.4.1 状態表示 (一般ユーザで可)

```
update-alternatives --display 一般名
update-alternatives --list 一般名
```

### 4.4.2 設定変更 (root のみ)

```
update-alternatives --auto 一般名
update-alternatives --config 一般名
update-alternatives --set 一般名 パス
```

の 4 つです。

### 4.4.3 使用例と説明

```
# update-alternatives --display editor          <-- editor という一般名の alternatives のステータスを表示
editor - status is auto.                        <-- alternatives の モード
link currently points to /usr/bin/vim           <-- 現在のリンク先
/bin/ed - priority 100                          <-- マスターリンク - 優先度 (/bin/ed)
slave editor.1.gz: /usr/share/man/man1/ed.1.gz  <-- スレーブ一般名 : スレーブリンク先 (/bin/ed)
/bin/nano - priority 40                        <-- マスターリンク - 優先度 (/bin/nano)
slave editor.1.gz: /usr/share/man/man1/nano.1.gz <-- スレーブ一般名 : スレーブリンク先 (/bin/nano)
/usr/bin/vim - priority 120                    <-- マスターリンク - 優先度 (/usr/bin/vim)
slave editor.1.gz: /usr/share/man/man1/vim.1.gz <-- スレーブ一般名 : スレーブリンク先 (/usr/bin/vim)
Current 'best' version is /usr/bin/vim.         <-- 優先度の一番高いもの

# update-alternatives --list editor             <-- editor という一般名の alternatives のマスターリンク先を表示
/bin/ed                                         <-- マスターリンク先
/bin/nano                                       <-- マスターリンク先
/usr/bin/vim                                    <-- マスターリンク先

# update-alternatives --auto editor              <-- 一般名 editor の alternatives を優先度によって変更
# update-alternatives --config editor            <-- 一般名 editor の alternatives を選択肢から手動で変更

There are 3 alternatives which provide 'editor'. <-- editor という一般名の選択肢が 3 つある

  Selection    Alternative
-----
      1        /bin/ed          <-- alternatives
      2        /bin/nano       <-- alternatives
**   3        /usr/bin/vim     <-- alternatives 現在選択されている

Press enter to keep the default[*], or type selection number: <-- ここで番号を選択する
Using '/usr/bin/vim' to provide 'editor'.                    <-- editor を /usr/bin/vim で提供する

# update-alternatives --set editor /usr/bin/vim             <-- 一般名 editor の alternatives を指定したリンクに変更する
Using '/usr/bin/vim' to provide 'editor'.                    <-- editor を /usr/bin/vim で提供する
```

## 4.5 パッケージメンテナとして使う場合

パッケージメンテナな人が主に使うコマンドパッケージインストール時に使用 (主に `postinst`)

```
update-alternatives --install リンク 一般名 パス 優先度 [--slave スレーブリンク スレーブ一般名 スレーブパス]
```

パッケージ削除時に使用 (主に `prerm`)

```
update-alternatives --remove 一般名 パス
```

#### 4.5.1 使用例

```
update-alternatives --install コマンド  
update-alternatives --install リンク 一般名 パス 優先度 [--slave スレーブリンク スレーブ一般名 スレーブパス]
```

この書式ですが、slave は省略可能で、使用する場合には複数のスレーブを指定することもできます。

vim の postinst

リンク元のファイルがインストールされてからリンクを貼るので postinst に書く

```
case "$1" in  
  abort-upgrade)  
    for i in vi view ex editor ; do  
      update-alternatives \  
        --install /usr/bin/$i $i /usr/bin/vim 120 \  
        --slave /usr/share/man/man1/$i.1.gz $i.1.gz /usr/share/man/man1/vim.1.gz  
    done  
    ;;  
  configure)  
    for i in vi view ex editor ; do  
      update-alternatives \  
        --install /usr/bin/$i $i /usr/bin/vim 120 \  
        --slave /usr/share/man/man1/$i.1.gz $i.1.gz /usr/share/man/man1/vim.1.gz  
    done  
    if [ -L /usr/doc/vim ] ; then  
      rm /usr/doc/vim  
    fi  
    ;;  
  esac
```

vim の prerm

リンク元のファイルがなくなる前に削除するので prerm に書く

```
case "$1" in  
  remove)  
    for i in vi view ex editor ; do  
      update-alternatives --remove $i /usr/bin/vim  
    done  
    ;;  
  esac
```

#### 4.6 マニアックなコマンド

update-alternatives --all

全ての選択肢に --config を使い設定を行なう

update-alternatives --remove-all

ディレクトリにある全ての選択肢を削除する（危険です）

#### 4.7 一般的なオプション

```
--verbose 冗長メッセージ  
--quiet メッセージを抑制  
--test テスト（未実装）  
--help ヘルプ
```

#### 4.8 マニアックなオプション

--altdir

リンクを置くディレクトリ

デフォルトは /etc/alternatives/

--admindir

設定ファイルを置くディレクトリ

デフォルトは /var/lib/dpkg/alternatives/

#### 4.8.1 マニアックなオプションの使用例

/usr/local/以下にソースからインストールしたものや alternatives を提供していないパッケージについて alternatives で管理したい場合などに使う 特権ユーザでなくても使えるのも特徴

使用例: /home/foo/eclipse 以下にある /home/foo/eclipse/eclipse を alternatives で管理したい。

```
$ mkdir /home/foo/altdir/
$ mkdir /home/foo/admindir/
$ mkdir /home/foo/bin/
$ /usr/sbin/update-alternatives --altdir /home/foo/altdir/ --admindir /home/foo/admindir/ \
--install /home/foo/bin/eclipse eclipse /home/foo/eclipse/eclipse 100
```

で、.bashrc などに /home/foo/bin/ を追加 IM とか、MUA の管理にも向いているのではないだろうか。

### 4.9 update-alternatives ってどうなってるの?

#### 構造

/usr/sbin/update-alternatives

alternatives の制御コマンド

perl で実装されており dpkg パッケージに含まれています。

/var/lib/dpkg/alternatives

alternatives 設定ファイルディレクトリ

設定ファイル例

```
$ cat /var/lib/dpkg/alternatives/editor
alternatives 設定ファイルディレクトリ
/usr/bin/editor          <-- マスターリンク先
editor.1.gz             <-- スレーブ一般名
/usr/share/man/man1/editor.1.gz <-- スレーブリンク先

/bin/ed                 <-- マスターリンク元 (/bin/ed)
-100                   <-- 優先度 (/bin/ed)
/usr/share/man/man1/ed.1.gz <-- スレーブリンク元 (/bin/ed)
/bin/nano               <-- マスターリンク元 (/bin/nano)
40                     <-- 優先度 (/bin/nano)
/usr/share/man/man1/nano.1.gz <-- スレーブリンク元 (/bin/nano)
/usr/bin/vim            <-- マスターリンク元 (/usr/bin/vim)
120                    <-- 優先度 (/usr/bin/vim)
/usr/share/man/man1/vim.1.gz <-- スレーブリンク元 (/usr/bin/vim)
```

/etc/alternatives/

alternatives のリンクのあるディレクトリです。

#### リンク例

```
$ ls -l /etc/alternatives/editor
lrwxrwxrwx 1 root root 12 2005-06-01 02:43 /etc/alternatives/editor -> /usr/bin/vi
```

```
/usr/share/man/man8/update-alternatives.8.gz
/usr/share/man/de/man8/update-alternatives.8.gz
/usr/share/man/es/man8/update-alternatives.8.gz
/usr/share/man/fr/man8/update-alternatives.8.gz
/usr/share/man/ja/man8/update-alternatives.8.gz
/usr/share/man/pt_BR/man8/update-alternatives.8.gz
```

alternatives マニュアルです。

### 4.10 動作

update-alternatives -auto

#### 1. 一般名を取得

2. `/var/lib/dpkg/alternatives/一般名` のステータス情報が `manual` の場合は `auto` に変更
3. `/etc/alternatives/一般名` のリンク先を `/var/lib/dpkg/alternatives/一般名` の優先度の情報を比較し最大のもののへのリンクへと変更

`update-alternatives --config` 及び `update-alternatives --set` コマンド

1. 一般名とリンク先のパスを取得
2. `/var/lib/dpkg/alternatives/一般名` のステータス情報が `auto` の場合は `manual` に変更
3. `/etc/alternatives/一般名` のリンク先を指定されたリンク先のパスへと変更

`update-alternatives --install` コマンド

1. 存在しない場合は指定されたパスのリンクを `/etc/alternatives/一般名` へと貼る
2. 存在しない場合は `/var/lib/dpkg/alternatives/一般名` ファイルを指定された情報に基づいて生成
3. `/etc/alternatives/一般名` のリンク先を `/var/lib/dpkg/alternatives/一般名` の優先度の情報を比較し最大のもののへのリンクへと変更

`update-alternatives --remove` コマンド

1. `/var/lib/dpkg/alternatives/一般名` から指定された パスの情報を削除
2. `/var/lib/dpkg/alternatives/一般名` で `alternatives` が提供されない場合は `/var/lib/dpkg/alternatives/一般名` 及び `/etc/alternatives/一般名` リンク先を削除
3. `/var/lib/dpkg/alternatives/一般名` で まだ `alternatives` が提供されている場合は `/etc/alternatives/一般名` のリンク先を `/var/lib/dpkg/alternatives/一般名` の優先度の情報を比較し最大のもののへのリンクへと変更

#### 4.11 update-alternatives の改善案

`update-alternatives` 関連の改善案を自分なりに考えてみた。

1. パッケージのファイル情報への反映  
現在の `alternatives` はパッケージをインストールされるまではそのパッケージがどんな `alternatives` を提案するのかという情報が解らない。これだと、インストールしてから、普段使用していた `alternatives` を使おうとしたら期待してたのと別のものが起動してしまい戸惑うことになるし、無駄なハマりの原因となるし、もし、そのような情報を集積できれば、機能からパッケージをより簡単に検索することができるようになる。
2. より柔軟に  
現在の `alternatives` では、`man` を `slave` にした場合などに英語のみしか表示できなくなってしまう、多言語化が進みつつある昨今これだけでは不足ということになってしまっているの、より柔軟な設定を行なえるのが望ましい。ファイルだけではなくコマンドを登録できるようにするなど。
3. `menu` 編集インターフェースとの連携  
独自に作成した `alternatives` などを `menu` に簡単に登録できるようにするなどの、応用的な使い方を

## 簡便にするインターフェースの拡充

### 4.12 dsys について

#### 4.12.1 目的

dpkg に含まれる パッケージ依存関係管理を行なう `/usr/bin/dpkg` 以外の `/usr/sbin/update-alternatives`, `/usr/sbin/dpkg-divert`, `/usr/sbin/dpkg-statoverride` の 3 つのコマンドに対応した GUI フロントエンドを提供することを目指している。

#### 4.12.2 歴史

一番初期は `update-alternatives` のみに対応したもので、`ruby 1.6` と `ruby-gtk` で書いていたが、やっぱ `gtk2` だろう！ ということで、`ruby-gtk2` に手を出してのんびりしてるうちに `galternatives` など競合で出てきてしまった。思い出したら実装したりしているがコーディング能力の低さのために遅々として改善されていない。

#### 4.12.3 機能

`/usr/sbin/update-alternatives`

一覧の表示、個々のステータスの表示、変更、追加、削除

`/usr/sbin/dpkg-divert`

一覧の表示

`/usr/sbin/dpkg-statoverride`

一覧の表示、変更、追加、削除

#### 4.12.4 TODO

まずはツールチップを適切に実装すること綺麗なコードの書き方や、デザインがわからないのでどうにかするせめて `man` くらい、`alternatives` と `statoverride` のインターフェースを意識的に替えていたがそろそろ統一 `treeview` で右クリックできるようにするショートカットの実装

#### 4.12.5 将来

設定ファイルを作るかもアイコン欲しいよね

協力者の方募集中、

## 5 debian-installer

kmuto



## 6 個人提案課題

---



名前 \_\_\_\_\_

下記の空欄を埋めてください:

Debian installer の ( )  
に注目し, 今後のネタは ( )  
します .

企画案の図 :



## 7 Keysigning Party

上川純一



### 事前に必要なもの

- 自分の鍵の fingerprint を書いた紙 (gpg --fingerprint XXXX の出力.)
- 写真つきの公的機関の発行する身分証明書, fingerprint に書いてある名前が自分のものであると証明するもの

### キーサインで確認する内容

- 相手が主張している名前の人物であることを信頼できる身分証明書で証明しているか<sup>\*1</sup>.
- 相手が fingerprint を自分のものだと言っているか
- 相手の fingerprint に書いてあるメールアドレスにメールをおくって, その暗号鍵にて復号化することができるか

### 手順としては

- 相手の証明書を見て, 相手だと確認
- fingerprint の書いてある紙をうけとり, これが自分の fingerprint だということを説明してもらう
- (後日) gpg 署名をしたあと, 鍵のメールアドレスに対して暗号化して送付, 相手が復号化してキーサーバにアップロードする (gpg --sign-key XXXXX, gpg --export --armor XXXX )

---

<sup>\*1</sup> いままで見た事のない種類の身分証明書を見せられてもその身分証明書の妥当性は判断しにくいので, 学生証明書やなんとか技術者の証明書の利用範囲は制限される. 運転免許証明書やパスポートが妥当と上川は判断している

## 8 次回



---

次回は 7 月 2 日土曜日の夜を予定しています．内容は本日決定予定です．  
参加者募集はまた後程．