

A large, stylized pink brushstroke graphic that forms a partial circle, framing the central text.

東京エリア Debian 勉強会

第 136 回 2016 年 2 月度

野島貴英

2016 年 2 月 13 日

Agenda

- 事前課題発表
- 最近あった Debian 関連のイベント報告
 - 第 135 回東京エリア Debian 勉強会
- Debian Trivia Quiz
- Debian GNU/Linux 上での省電力設定について
- libhinawa というライブラリを Debian プロジェクトに ITP/RFS した話
- 今日の宴会場所

A large, stylized pink circular graphic element, resembling a thick brushstroke or a swirl, is positioned on the right side of the image. It frames the text.

事前課題

- ① Q.hack time に何をしますか？
A.hinawa-utils の開発。あるいは ALSA の開発。
<https://github.com/takaswie/hinawa-utils>
- ② Q. 本勉強会をどこでお知りになりましたか？(任意回答)
A. ML

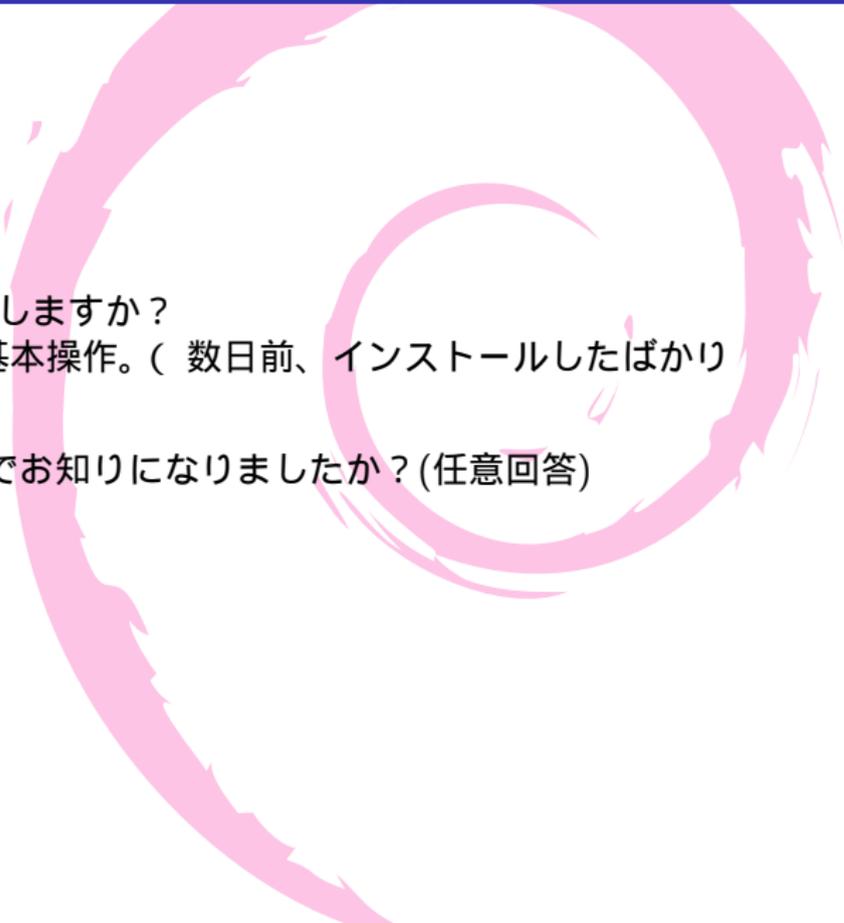
① Q.hack time に何をしますか？

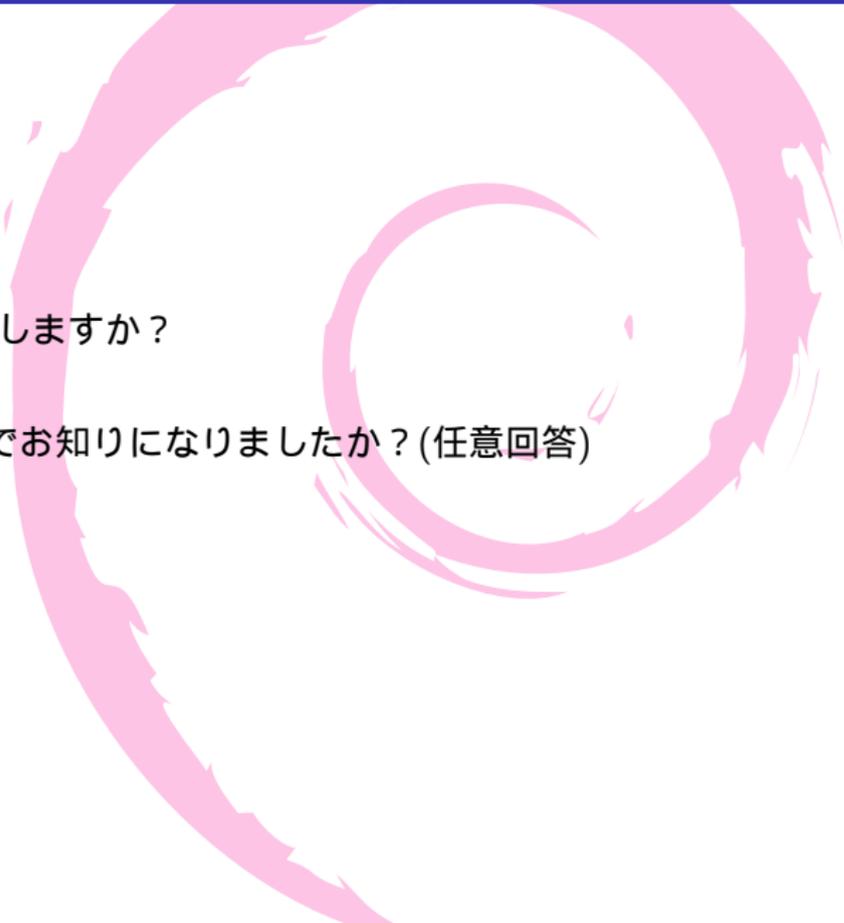
A.

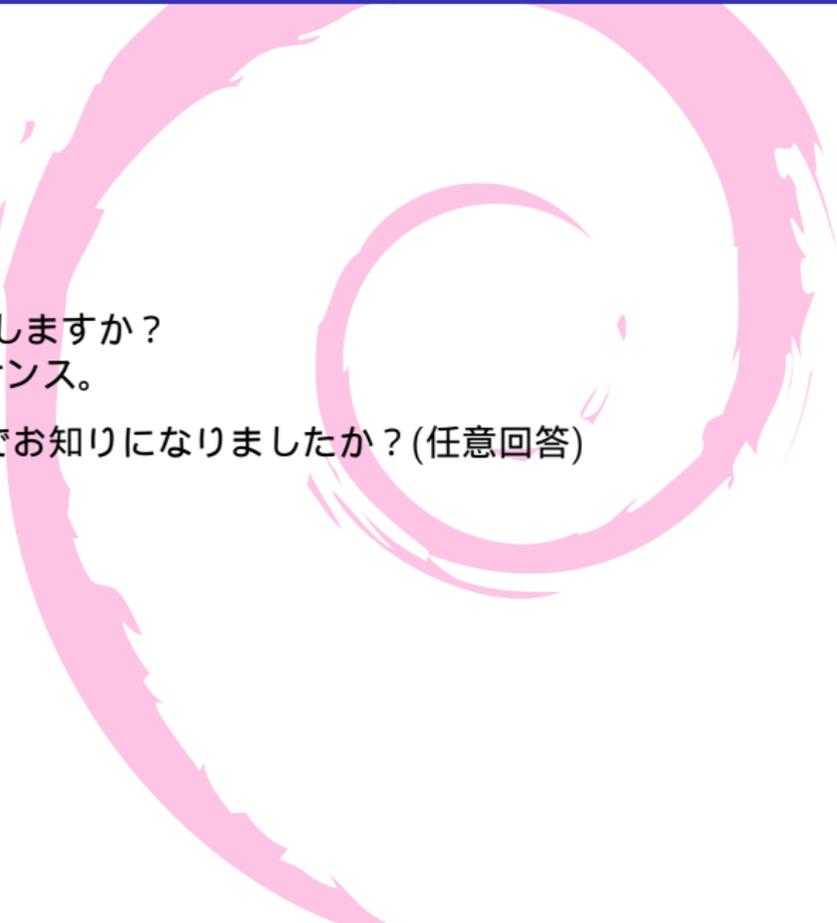
- <https://qa.debian.org/developer.php?login=mkouhei@palmtb.net> のメンテ。
- パッケージ (<https://qa.debian.org/developer.php?login=mkouhei@palmtb.net>) のメンテナンス
- keybase.io を signing-party(caff) のキーサーバーとして利用できないかの検証

② Q. 本勉強会をどこでお知りになりましたか？(任意回答)

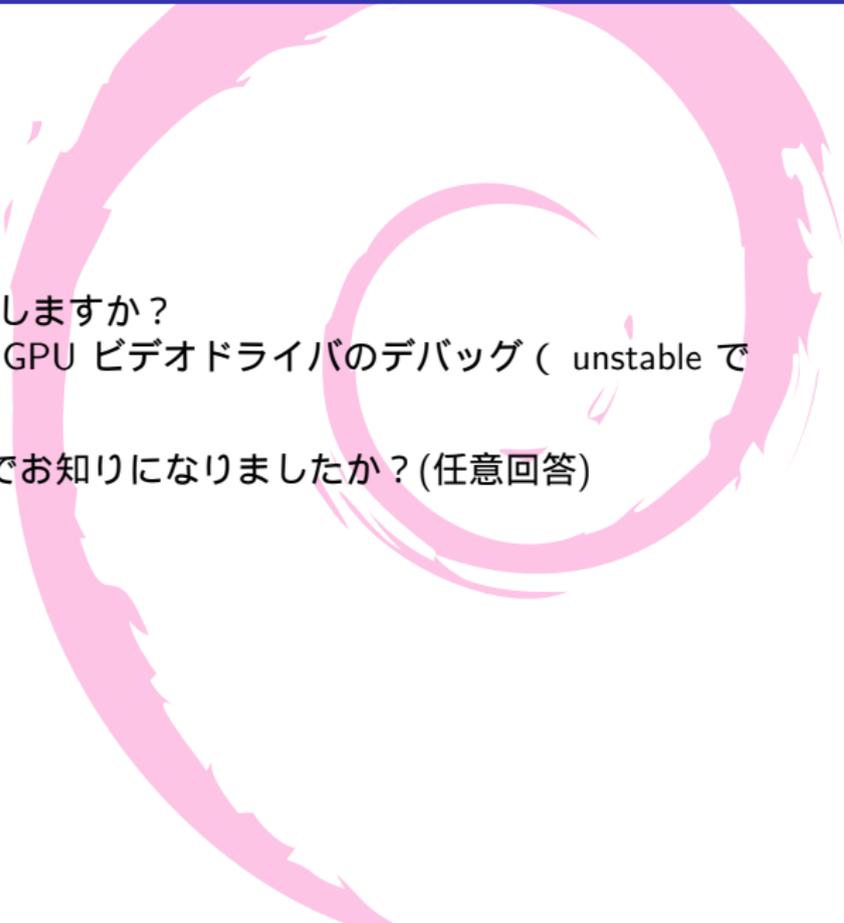
A. 常連

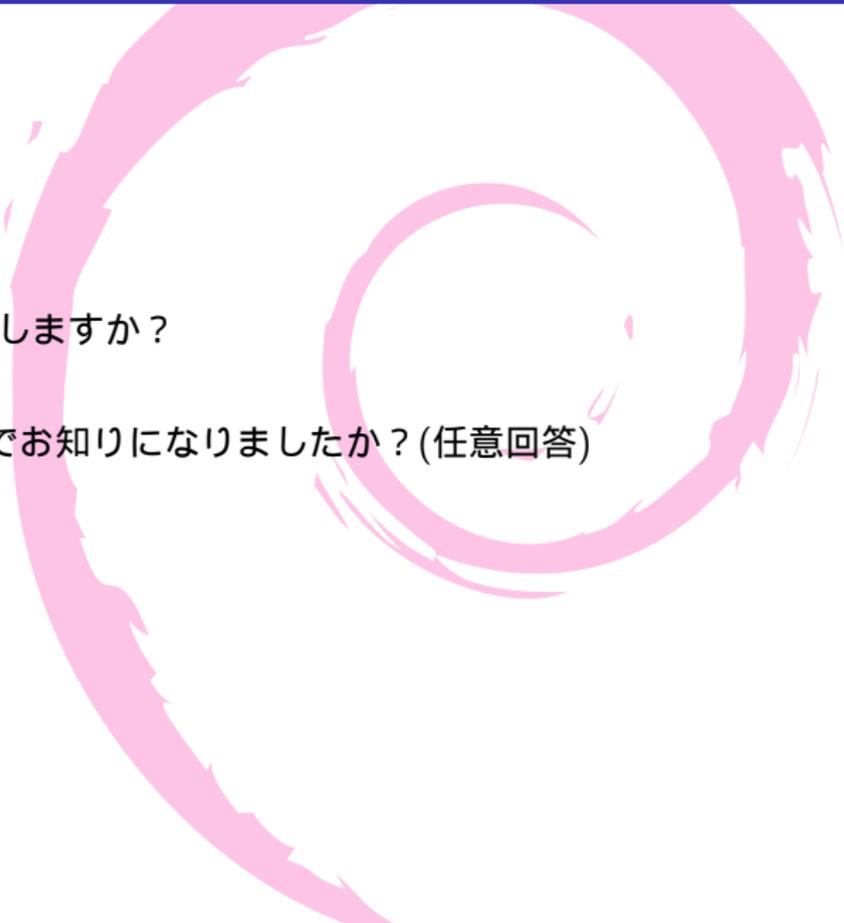
- 
- ① Q.hack time に何をしますか？
A. 「 Debian 」 の基本操作。(数日前、インストールしたばかりなので)
 - ② Q. 本勉強会をどこでお知りになりましたか？(任意回答)
A. Web

- 
- ① Q.hack time に何をしますか？
A. 未定。
 - ② Q. 本勉強会をどこでお知りになりましたか？(任意回答)
A. ML

- 
- ① Q.hack time に何をしますか？
A. package メンテナンス。
 - ② Q. 本勉強会をどこでお知りになりましたか？(任意回答)
A. ML

- ① Q.hack time に何をしますか？
A. パッケージング。

- 
- ① Q.hack time に何をしますか？
A. kfreebsd の Intel GPU ビデオドライバのデバッグ (unstable で動かなくなった)
 - ② Q. 本勉強会をどこでお知りになりましたか？(任意回答)
A. Web

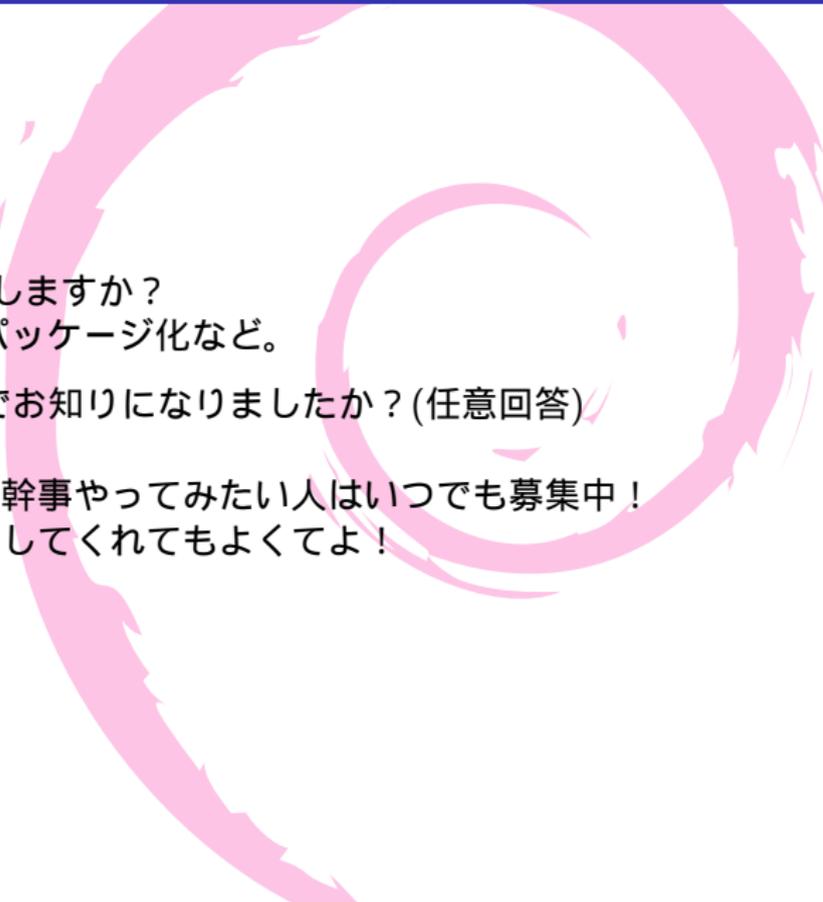
- 
- ① Q.hack time に何をしますか？
A. d-i に関する作業
 - ② Q. 本勉強会をどこでお知りになりましたか？(任意回答)
A. twitter

- ① Q.hack time に何をしますか？
A. tilegx



- ① Q.hack time に何をしますか？
A. DDTSS



- 
- ① Q. hack time に何をしますか？
A. DDTSS、xmris パッケージ化など。
 - ② Q. 本勉強会をどこでお知りになりましたか？(任意回答)
A. 幹事なので！
ところで 세미나か、幹事やってみたい人はいつでも募集中！
貴殿好みの勉強会にしてくれてもよくてよ！



イベント報告

第135回東京エリア Debian 勉強会

- 場所は dots さんをお借りしての開催でした。
- 参加者は10名でした。
- セミナ内容は、2本建てで、
 - ① 参加者皆さんによる「Debian 今年の半年分の計画を立ててみた」
 - ② 野島さんによる「Debian で Linux Ftrace まわりをいじってみた」でした。
- 残りの時間で hack time を行い、成果発表をしました。

第 135 回東京エリア Debian 勉強会 (つづき)

「Debian 今年の半年分の計画を立ててみた」は、2016 年 6 月末までの Debian プロジェクトで行う目標を書いて頂きました。みなさん、がっちり目標を立てていただきました。あとは、実行あるのみということで、よろしくおねがいします！

第 135 回東京エリア Debian 勉強会 (つづき)

「Debian で Linux Ftrace まわりをいじってみた」は、linux kernel に搭載されているデバッグ I/F である Ftrace を Debian で利用してみた事について語られました。Debian sid に搭載されている kernel が 4.3.3 であることから、USDT を利用してみたり、kprobe を利用した話をしました。Debian に搭載されている Ftrace 利用のツールである、perf-tool が古かった件についても、BUG Report¹ が行われました。

¹<https://bugs.debian.org/813769>

第 135 回東京エリア Debian 勉強会 (つづき)

Debian は、パッケージが古いなどの件も BUG Report することでも改善を図ることが出来ます。もし、他に、

- パッケージが古い、
- 導入したパッケージのソフトウェアの挙動がおかしい (バグっている)
- パッケージはこうして欲しい

などの要望がありましたら、「誰でも」reportbug ツールを使って簡単に改善の要請をすることが出来ます。是非、皆様もどしどし BUG Report してくださいませ。

第 135 回東京エリア Debian 勉強会 (つづき)

また、hack time ですが、成果を titanpad.com に記載する試みを行いました。以下は前回の hack time 中の成果の一覧となります。

- henrich さん
proposal for Debian idea
 - integrate piuparts into repository pipeline
 - replace dput → lintian + piuparts, then upload + prevents regression into repository
- kenhys
libhinawa の debian/* に手をいれていくつか PR を投げたり、issue を立てたりした
<https://github.com/takaswie/libhinawa/pull/20>
<https://github.com/takaswie/libhinawa/pull/21>
<https://github.com/takaswie/libhinawa/pull/24>

第135回東京エリア Debian 勉強会 (つづき)

- tai さん

- キーサインを行った。caff ではなく手動で
Keysigning with the GNU/Linux Terminal
<http://www.phillylinux.org/keys/terminal.html>
に沿ってやってみた。しかし最後の署名したキーを送
る所が Gmail 経由になり署名・暗号化せずに返送してし
まったので、caff がやっぱりいいのか …
- Debian Wiki の翻訳し残しの続きをもくもくと。

- wskoka さん

tilegx 用パッケージ作成 (10 くらい。トータルは 1 2
0 0 個くらいある。)

第 135 回東京エリア Debian 勉強会 (つづき)

- rosh さん

- caff でキーサインが出来た
- 前マージされた Linkstation DTS [0] をベースにて、新 Linkstation デバイス (LS-QVL) の DTS を別のユーザ様が出来たとの連絡がありました。今週受け入れられた Linkstation DTS [1] にマージを行いました。これから ARM kernel lists にアップする予定です。

- ① Kernel tree:
arch/arm/boot/dts/kirkwood-lswvl,lswxl.dts
- ② <http://lists.infradead.org/pipermail/linux-arm-kernel/2016-January/400949.html>

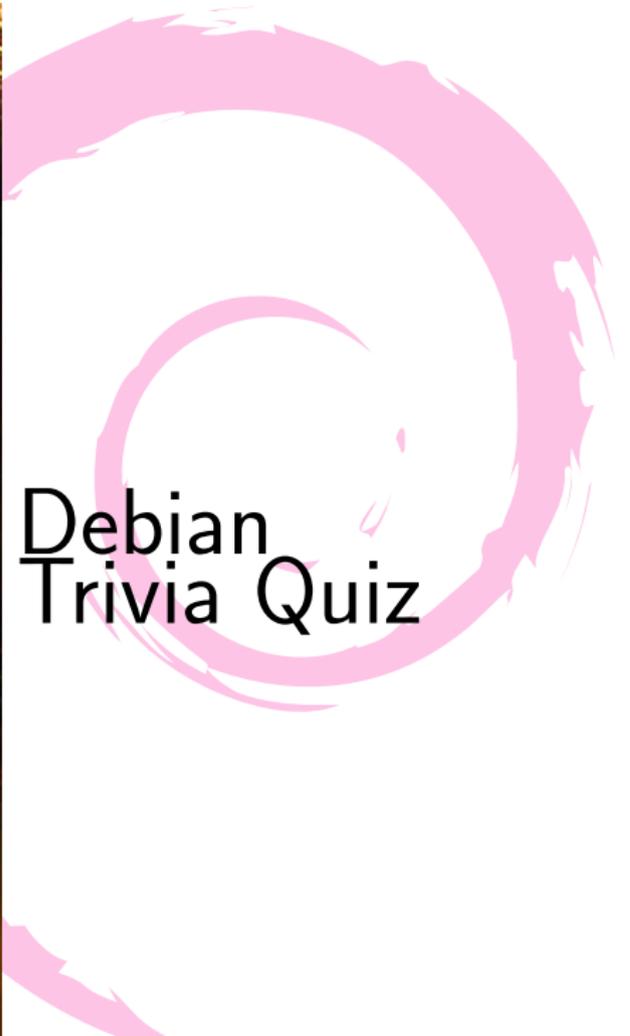
第 135 回東京エリア Debian 勉強会 (つづき)

- dictoss さん
 - raspberry pi2 の debian jessie から bluetooth テザリングできた
 - キーサインしました
 - 東京エリア Debian 勉強会の web サイトを HTML5 対応とスマートフォン対応作業 (リポジトリの場所確認、ソースコードを読んで中身と作りを把握中)²
- y.y さん
 - 参加者と key sign を行った。
 - Caff で、溜まっていた keysigning 処理のキューをフラッシュできた。

²2016 年 2 月 27 日に無事リリースされ、tokyodebian.alioth.debian.org は、祝!モバイルフレンドリーになりました!

第 135 回東京エリア Debian 勉強会 (つづき)

- takaswie さん
 - Echo Audio Corp. の Fireworks デバイスマジュール向けコマンドラインツールがだいたい書けた。
<https://github.com/takaswie/hinawa-utils>
- yy-y-ja-jp さん
 - uim-qt5 バグレポート
 - キーサイン
 - DDTSS
- 野島さん
 - 9 個の DDTSS のレビューを行いました。なお、yorick は typo の "M-x Yorick" を見落とし、request しました。



Debian Trivia Quiz

Debian Trivia Quiz

Debian の常識、もちろん知ってますよね？ 知らないなんて
恥ずかしくて、知らないとは言えないあんなことやこんな
こと、みんなで確認してみましよう。

今回の出題範囲は

`debian-devel-announce@lists.debian.org`,
`debian-news@lists.debian.org` に投稿された内容などか
らです。

問題1. 2016/2/3にて、debttagのtag付けについての変更が流れました。以下のどれ？



A tag 付け廃止



B tag 付けについてユーザ認証付きにする



C tag のレビューをさらに強固にする

問題1. 2016/2/3にて、debttagのtag付けについての変更が流れました。以下のどれ？

-  A tag 付け廃止
-  B tag 付け
証付きにて
-  C tag のし
固にする

答えは:



B

問題 1. 2016/2/3 にて、debtags の tag 付けについての変更が流れました。以下のどれ？

解説：debtags を編集できるサイトがいろいろリニューアルするというアナウンスが流れ、実際いくつも実行されたようです。まず、匿名による tag 付けの編集を廃止し、代わりに sso.debian.org によるシングルサインオンでユーザ認証しないと編集出来ないようにしたとのこと。また、tag 編集の URL が変更になっており、<https://debtags.debian.org/> となりました。他にもいろいろ変更が出ていますので、詳しくは <https://lists.debian.org/debian-develop/2016/02/msg00000.html> 参照。

問題2. 2016/1/12にて、debian sidにphpの新しいバージョンを入れた件がアナウンスされました。どのバージョン？

-  A php 7.0
-  B php 5.6
-  C phpって何？

問題2. 2016/1/12にて、debian sidにphpの新しいバージョンを入れた件がアナウンスされました。どのバージョン？

-  A php 7.0
-  B php 5.6
-  C php 7.0

答えは:



A

問題2. 2016/1/12にて、debian sidにphpの新しいバージョンを入れた件がアナウンスされました。どのバージョン？

解説：php7.0がdebian sidにてリリースされました。ちなみに、php7の目玉機能は大幅な実効速度改善です。これで次期安定版バージョンであるstretchで、php7が利用できる見込みがとて高くなってきましたね！

問題3. dbgsym パッケージですが、こちらを保管するミラー先はどこでしょう？

-  A mirrors.debian.org
-  B debug.mirrors.debian.org
-  C ftp.jp.debian.org

問題3. dbgsym パッケージですが、こちらを保管するミラー先はどこでしょう？

答えは:

•  A mirrors.debian.org

•  B debug.mirrors

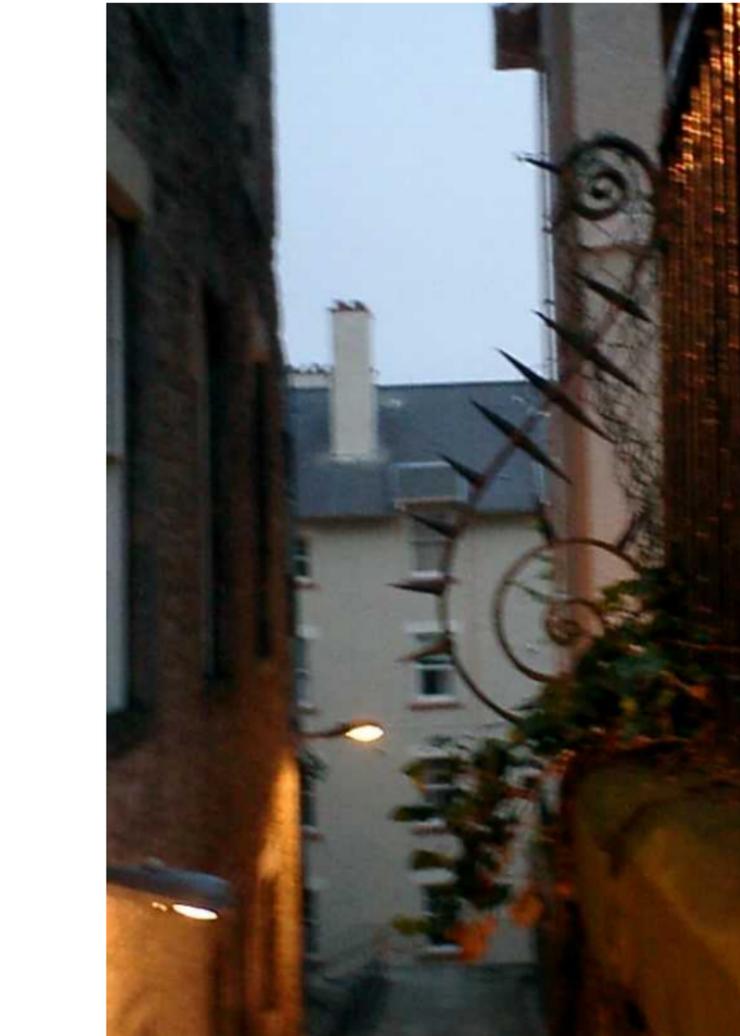
•  C ftp.jp.debian



B

問題3. dbgsym パッケージですが、こちらを保管するミラー先はどこでしょう？

解説：debhelper 9.20151219 以降にて、常にデバッグ用シンボルを収めたパッケージ（名前はパッケージ名-dbgsym）を生成するようになりました。この-dbgsym パッケージの保管先がアナウンスされ、
<http://debug.mirrors.debian.org/debian-debug/>と、
<http://snapshot.debian.org/archive/debian-debug/>になったようです。



Debian
GNU/Linux
上での省電
力設定につ
いて

はじめに

- Linux がインストールされたノート PC を利用している時、スペック通りにバッテリーが持たない
- デフォルトの設定では省電力設定ができていない。
- デフォルトの設定ではなく、パラメータを調整して、よりよいノート PC 生活を送りたい
- Debian GNU/Linux での省電力設定を解説

省電力設定するためには

- CPU と制御
- 動作しているデバイス
- 動作しているプログラム

CPU と制御

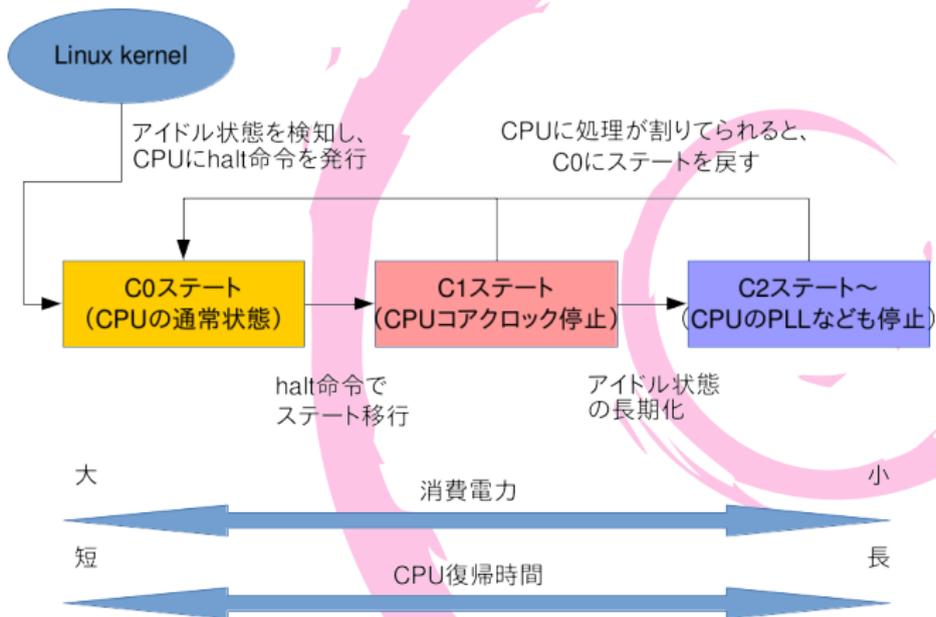


Figure: CPU 状態遷移簡易図

CPU 周波数スケーリング機能と cpufreq-info

Linux の場合は CPU 周波数スケーリング機能があり、cpufreq 機構として実装されている。

```
$ cpufreq-info
cpufrequtils 008: cpufreq-info (C) Dominik Brodowski 2004-200
Report errors and bugs to cpufreq@vger.kernel.org, please.
analyzing CPU 0:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by soft
  maximum transition latency: 0.97 ms.
  hardware limits: 800 MHz - 2.90 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 800 MHz and 2.90
                  The governor "powersave" may decide which s
                  within this range.
  current CPU frequency is 1.90 GHz.
....
```

Figure: cpufreq-info 実行結果

ガバナーと現在設定されているポリシー

ガバナー	内容
ondemand	CPU 負荷が大きい、または小さい時に CPU クロックを大きくに切り替える
conservative	CPU 負荷が大きい、または小さい時に CPU クロックを徐々に切り替える
performance	最大周波数で CPU を動作させる
powersave	最小周波数で CPU を動作させる
userspace	ユーザーが指定した周波数で CPU を動作させる

Table: 指定できるガバナー

- CPU が 800MHz から 2.90GHz までをサポートしている
- powersave governor で動作している
- 最大値と最小値のポリシー設定により、800MHz から 2.90GHz の間で変動させている

```
$ cpufreq-info
cpufrequtils 008: cpufreq-info (C) Dominik Brodowski 2004-2009
Report errors and bugs to cpufreq@vger.kernel.org, please.
analyzing CPU 0:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 0.97 ms.
  hardware limits: 800 MHz - 2.90 GHz
  available cpufreq governors: performance, powersave, ondemand
  current policy: frequency should be within 800 MHz and 2.90 GHz.
                   The governor "powersave" may decide which speed to use
                   within this range.
  current CPU frequency is 1.90 GHz.
....
```

ガバナーとCPUクロックの設定

設定には cpufreq-set コマンドを使う。
ガバナーを設定する

```
$ sudo cpufreq-set -c CPU 番号 -g ガバナー名  
または  
$ sudo sh -c "echo ガバナー名 > \  
/sys/devices/system/cpu/cpuCPU 番号/cpufreq/scaling_governor"
```

ガバナーとCPUクロックの設定

最小クロックを設定する

```
$ sudo cpufreq-set -c CPU番号 -d クロック値
```

または

```
$ sudo sh -c "echo クロック値 > \  
/sys/devices/system/cpu/cpuCPU番号/cpufreq/scaling_min_freq"
```

ガバナーとCPUクロックの設定

最大クロックを設定する

```
$ sudo cpufreq-set -c CPU番号 -u クロック値
```

または

```
$ sudo sh -c "echo クロック値 > \  
/sys/devices/system/cpu/cpuCPU番号/cpufreq/scaling_max_freq"
```

ガバナーとCPUクロックの設定

現在のクロックを設定する

```
$ sudo cpufreq-set -c CPU番号 -f クロック値
```

または

```
$ sudo sh -c "echo クロック値 > \  
/sys/devices/system/cpu/cpuCPU番号/cpufreq/scaling_cur_freq"
```

ガバナーとCPUクロックの設定

- sysfs を使った設定は再起動すると消えてしまう。
- /etc/sysfs.conf を使うか、cpufreqd パッケージで設定できるようにする。

動作しているデバイスの設定

- 使わないデバイスを有効にするだけで電力を消費する。
- 環境に応じてデバイスの電力等を制御する。
- ここではよく利用されるデバイスに対する制御方法について紹介。

ラップトップモード

ラップトップモード設定

```
$ sudo sh -c "echo 5 > /proc/sys/vm/laptop_mode"
```

NMI watchdog 無効化

```
$ sudo sh -c "echo 0 > /proc/sys/kernel/nmi_watchdog"
```

USB

USB は `/sys/bus/usb/devices/` 以下に対して設定。

```
$ sudo sh -c "echo off > \  
  /sys/bus/usb/devices/usb1/power/control"  
$ sudo sh -c "echo auto > \  
  /sys/bus/usb/devices/usb1/power/autosuspend"
```

起動時に再設定したい場合は `udev` の `rules` ファイルを用意する。

```
$ cat /etc/udev/rules.d/70-my-usb-power.rules  
ACTION=="add", SUBSYSTEM=="usb", ATTRS{idVendor}=="0x046d", \  
  ATTR{idProduct}=="0x08cb", TEST=="power/control", \  
  ATTR{power/control}="off"
```

無線 LAN

無線 LAN は iw パッケージに含まれる iw コマンドを使って設定する。

```
$ sudo iw dev wlan0 set power_save on
```

udev の rules ファイルを使って設定方法

```
$ cat /etc/udev/rules.d/70-my-wifi-power.rules  
ACTION=="add", SUBSYSTEM=="net", KERNEL=="wlan*", \  
    RUN+="/usr/bin/iw dev %k set power_save on"
```

サウンド

```
$ sudo sh -c "echo 1 > \  
/sys/module/snd_hda_intel/parameters/power_save"
```

PCI/PCI-Express

PCI/PCI-Express の省電力に設定するには power/control を auto に設定する。

```
$ sudo sh -c "echo auto > \  
/sys/bus/pci/devices/0000:00:00.0/power/control"
```

動作しているプログラムについて

- top コマンドである程度わかるが、細かい数字はわからない。
- PowerTOP を使って確認する。

省電力設定するためのツール

- 省電力設定するには専門の知識が必要。
- 細々とした設定をひとつづつやっていくのは大変。
- 専門知識がなくても、ざっくりとやってくれるツールがある。
- これらのツールを紹介

- Intel が開発しているソフトウェア。
- Linux カーネル、ハードウェア、ユーザランドで制御可能な省電力項目を有効にするツール。
- プロセスを監視して、CPU 負荷やデバイスドライバの使用状況のレポートなどが行える

インストール

```
% sudo apt-get install powertop
```

起動

```
PowerTOP 2.8 Overview Idle stats Frequency stats Device stats Tunables
The battery reports a discharge rate of 13.0 W
The estimated remaining time is 5 hours, 34 minutes
Summary: 684.6 wakeups/second, 69.3 GPU ops/seconds, 0.0 VFS ops/sec and 12.1% CPU use
  Usage      Events/s    Category      Description
  18.6 ms/s   171.0       Process       /usr/lib/chromium/chromium --show
  59.1 ms/s   113.0       Process       /usr/lib/chromium/chromium --type
  0.9 ms/s    87.4        Timer         tick_sched_timer
  9.3 ms/s    66.9        Process       /usr/lib/chromium/chromium --type
  688.4 us/s  84.3        Timer         hrtimer_wakeup
  663.2 us/s  61.3        Interrupt     [52] i915
  174.1 us/s  36.3        Interrupt     [6] tasklet(softirq)
  8.9 ms/s    26.7        Process       /usr/lib/chromium/chromium --type
  528.1 us/s  19.2        Process       xfwm4 --display :0.0 --sm-client-
  1.2 ms/s    23.1        Interrupt     [48] xhci_hcd
  1.3 ms/s    12.8        kWork         hci_rx_work
  77.3 us/s   16.7        Process       [rcu_sched]
  72.6 us/s   4.6         Process       /usr/sbin/mouseemu
  3.1 ms/s    2.2         Process       /usr/lib/xcorg/Xorg -l -seat_seat0
```

Figure: PowerTOP 起動画面

Tunables タブ

- Tunables タブには調整可能なシステムの設定が表示される。
- Bad が省電力に有効な項目にもかかわらず無効な設定。
- Good が既に有効になっている設定。

```
PowerTOP 2.8 Overview Idle stats Frequency stats Device stats Tunables
>> Bad Wireless Power Saving for interface wlan0
Bad Autosuspend for USB device xHCI Host Controller [usb2]
Bad Runtime PM for I2C Adapter i2c-8 (DPDDC-C)
Bad Autosuspend for USB device Bluetooth USB Host Controller [Apple Inc.]
Bad Runtime PM for I2C Adapter i2c-3 (i915 gmbus dpc)
Bad Runtime PM for I2C Adapter i2c-7 (DPDDC-B)
Bad Runtime PM for I2C Adapter i2c-4 (i915 gmbus dpb)
Bad Autosuspend for USB device xHCI Host Controller [usb1]
Bad Runtime PM for I2C Adapter i2c-2 (i915 gmbus panel)
Bad Runtime PM for I2C Adapter i2c-0 (i915 gmbus ssc)
Bad Autosuspend for USB device Android [Android]
Bad Autosuspend for USB device Apple Internal Keyboard / Trackpad [Apple In
Bad Runtime PM for I2C Adapter i2c-5 (i915 gmbus dpd)
Bad Runtime PM for I2C Adapter i2c-6 (DPDDC-A)
Bad Runtime PM for I2C Adapter i2c-1 (i915 gmbus vga)
Bad Runtime PM for PCI Device Intel Corporation Haswell-ULT Integrated Grap
Good NMI watchdog should be turned off
Good Bluetooth device interface status
Good Enable Audio codec power management
```

Figure: Tunables 画面

この状態ではまだシステムに最適化された設定になってい

キャリブレーション

実行するとデバイスなどから使用状況を読み取り、マシンに対して適切な設定を行う。ノートPCの場合はいきなりモニターのバックライトが消えるので注意。

```
$ sudo powertop --calibrate
```

キャリブレーション

- キャリブレーションが終わると、
`/var/cache/powertop/saved_parameters.powertop`
以下にデータが保存される。
- 次の PowerTOP 起動時からはキャリブレーション
データを元に省電力にされた環境で起動する。

キャリブレーション後

キャリブレーション後に起動すると、「The battery reports a discharge rate ...」の項目に表示される消費電力値が変わり、システム全体で省電力で稼働している事が確認できるはず。

PowerTOP の起動時有効化

- PowerTOP は起動すると保存されている設定を元に省電力状態にしてくれますが、PC を立ち上げるたびに PowerTOP 自体を立ち上げる必要がある。。
- 起動時に自動的に PowerTOP を立ち上げるようにするには、以下のように systemd のユニットファイルを用意し、有効にする。

```
$ cat /etc/systemd/system/powertop.service
```

```
[Unit]  
Description=PowerTOP
```

```
[Service]  
Type=oneshot  
ExecStart=/usr/bin/powertop  
Environment="TERM=xterm"
```

```
[Install]  
WantedBy=multi-user.target
```

TLP を使った設定

- PowerTOP のように詳細なレポートは出さないが、AC 接続時などの状況に応じたスクリプトが準備されており、インストールするだけである程度省電力設定を行ってくれる便利なツール。
- Debian ではパッケージ化されており、apt でインストールできる。

```
$ sudo apt-get install tlp
```

TLP を使った設定

- 無線 LAN の設定等に NetworkManager を使っているなら tlp-rdw パッケージもインストールしておくことで無線 LAN、Bluetooth 関連の設定も行ってくれる。
- デフォルトの設定は `/etc/default/tlp` にあり、このファイルを変更して環境に合わせた省電力設定を行う。

TLP を使った設定

- 設定はよく使われる項目しかなく、使っている環境の設定がない場合もある。このような場合は自分で設定を追加するか、先に説明したように sysfs / procfs 経由の設定を別途行う必要がある。

```
# Set to 0 to disable, 1 to enable TLP.
```

```
TLP_ENABLE=1
```

```
# Operation mode when no power supply can be detected: AC,
```

```
# Concerns some desktop and embedded hardware only.
```

```
TLP_DEFAULT_MODE=AC
```

```
# Seconds laptop mode has to wait after the disk goes idle
```

```
# Non-zero value enables, zero disables laptop mode.
```

```
DISK_IDLE_SECS_ON_AC=0
```

```
DISK_IDLE_SECS_ON_BAT=2
```

```
# Dirty page values (timeouts in secs).
```

```
MAX_LOST_WORK_SECS_ON_AC=15
```

```
MAX_LOST_WORK_SECS_ON_BAT=60
```

省電力設定後

- 省電力設定した後に PowerTOP で消費電力を確認。
- 消費電力が 13W から 11W に下がっている。
- PC 稼働時間も 5 時間半から 6 時間 50 分に伸びている

```
PowerTOP 2.8 Overview Idle stats Frequency stats Device stats Tunables
The battery reports a discharge rate of 11.1 W
The estimated remaining time is 6 hours, 50 minutes
Summary: 401.8 wakeups/second, 30.1 GPU ops/seconds, 0.0 VFS ops/sec and 2.6% CPU use
Power est. Usage Events/s Category Description
13.4 W 10.0% Device Display backlight
11.2 mW 0.9 ms/s 46.4 Process xfwm4 --display :0.0 --sm-client-
6.95 mW 4.6 ms/s 12.9 Process /usr/lib/xorg/Xorg :0 -seat seat0
5.30 mW 3.5 ms/s 7.6 Process python /usr/share/blueproximity/p
4.72 mW 1.7 ms/s 10.5 Process xfce4-terminal --geometry=231x68
4.60 mW 2.6 ms/s 32.1 Process /usr/lib/x86_64-linux-gnu/xfce4/p
2.95 mW 1.9 ms/s 1.4 Process powertop
2.85 mW 1.9 ms/s 99.8 Interrupt [9] acpi
2.37 mW 1.6 ms/s 1.3 Process powertop
2.24 mW 688.3 us/s 4.4 Process /usr/lib/x86_64-linux-gnu/xfce4/p
2.15 mW 1.4 ms/s 51.6 Process /usr/sbin/cpufreqd -f /etc/cpufre
1.25 mW 0.8 ms/s 3.6 Process sh
1.20 mW 175.1 us/s 3.9 Process /usr/lib/x86_64-linux-gnu/xfce4/p
534 uW 350.1 us/s 21.5 Interrupt [52] i915
```

Figure: 省電力設定後

まとめ

- Debian GNU/Linux での省電力設定について説明した。
- 現在の状態をとりあえず確認するには `cpufreq-info` を使い、カーネルの設定やドライバの設定は `sysfs` や `proc fs` 経由で設定する。
- プログラムやプロセスの詳細な状態の確認するには `PowerTOP` を使う。省電力設定できる項目もわかり、ユーザインターフェイスから各種設定ができるようになってきている。
- 細かい設定を行わなくても、とりあえず省電力設定を行いたい場合は `TLP` を使うのがよい。ただ全ての PC をサポートしているわけではないので、環境に合わせてプログラムを修正するなどの対応が必要となる。
- 細かい設定をしたい場合は `sysfs` や `procfs` から設定する必要がある。この場合専門の知識が必要となる。



libhinawa と
いうライブ
ラリを De-
bian プロ
ジェクトに
ITP/RFS
した話

libhinawa というライブラリを Debian プロジェクトに ITP/RFS したので、その背景や設計、機能に関して、ALSA の upstream での活動も踏まえてお話しします。

内容

- ALSA について
- libhinawa について
- 今回の ITP/RFS について

質疑は適宜行ってください。

ALSA の概要

ALSA

- Linux sound subsystem
- ユーザースペースのアプリケーションに対し、サウンドデバイスを制御する手段を提供
- 特定のサウンドデバイスに対しては、CUI や GUI を伴うツールを提供

ALSA を利用しているもの

- Linux カーネルを採用した製品全般

ALSA がサポートするサウンドデバイス

- PCI/PCI-Express バスに接続するもの
- プラットフォーム固有のペリフェラルバスに接続するもの
- USB に接続するもの
- 他、レガシーな汎用バスに接続するもの (IEEE 1394 バス)

ALSA と IEEE 1394 バス

2000-2010 年にかけて、IEEE 1394 バスに接続して使うサウンドデバイスが発売されていた。

- 主にレコーディング用途
- 一部の高級オーディオ機器

ALSA でのサポートは 2010 年に始まった。

- Linux 2.6.39 に初期のコードをマージ
- 以降の開発は進まず

ユーザースペースのドライバ実装

Linux FireWire subsystem が、キャラクタデバイス越しにデバイスを制御する手段をユーザースペースに提供

- `/dev/fw[0-9]*`

FFADO

- ユーザースペースのドライバ開発プロジェクト
- <http://ffado.org/>
- libffado2 として、C のライブラリ API を公開
- 2004 年あたりに開発開始

ALSA とは異なる API

- ALSA のアプリケーションから使えない
- プロトコル仕様 (IEC 61883-1/6) 的に、ALSA-FFADOブリッジが成立しない。

ALSA の firewire スタックの成長

2013 年 1 月時点。2 モデルをサポート。

- 2,600 行程度

2016 年 2 月現在、約 140 モデルをサポートするに至る。

- 20,000 行程度

ALSA の firewire スタックの設計

- リアルタイムデータ転送以外の機能はユーザースペースに置く
- どうしてもカーネルランドに必要な機能は ALSA の HwDep を使ってユーザースペースからアクセス可能にする

当初は FFADO の資産を再利用するつもりだった

- が、それは困難を伴うことが後に判明

FFADO の困難

FFADO のツールを開発に用立てるのはむづかしい

- libffado2 の設計上の欠陥

FFADO のコードベースの古さ

- FFADO のツールが Python2 で書かれている

開発に用立てるツールの必要

- libhinawa の開発に着手

キャラクタデバイスに対する入出力の抽象

- /dev/fw[0-9]*
- /dev/snd/hwC[0-9]*D0

GObject Introspection による、言語バインディングからの利用

- 140 デバイスそれぞれ固有の操作が必要
- デバイス操作方法を見出す試行錯誤が必要
- LL で書きたかった

libhinawa の開発

- 2014 年 9 月あたりに開発着手。C のライブラリ API を公開 (0.2.0) status/transactions to ALSA FireWire devices
<http://mailman.alsa-project.org/pipermail/alsa-devel/2014-September/081732.html>
- 2015 年 1 月に GObject Introspection 対応 (0.3.0)
ALSA/FireWire transaction
<http://mailman.alsa-project.org/pipermail/alsa-devel/2015-January/086371.html>
- その 2 週間後に alsa-tools.git に対する RFC (0.4.0)
applications of FireWire devices
<http://mailman.alsa-project.org/pipermail/alsa-devel/2015-January/086969.html>
- 2015 年 2 月に alsa-lib への依存を破棄 (0.5.0)

libhinawa の開発 (つづき)

- 2015 年 3 月に初期の debian/rpm パッケージングとそれに伴う修正 (0.6.0)
- 2016 年 1 月に Debian への ITP とそれに伴う修正 (0.7.0)
<http://mailman.alsa-project.org/pipermail/alsa-devel/2016-February/103771.html>

メインラインの作業

- Linux 3.16 - 63 パッチ
- Linux 3.19 - 31 パッチ
- Linux 4.2 - 22 パッチ
- Linux 4.4 - 68 パッチ

Linux stable kernel

- バグが報告されたら、その修正パッチを作って stable に送る。

libhinawa の ITP/RFS

作業は github を使って進める。

- <https://github.com/takaswie/libhinawa/>

1/14 あたり

- 林さんと、libhinawa の ITP の話をする。

1/18

- 林さんから最初の PR

1/24 あたり

- Ubuntu 16.04 の DebianImportFreeze(2/17) に間に合わせる方針

1/28 あたり

- Autotools 周りの修正終了
- debian/* の用意終了

libhinawa の ITP/RFS(つづき)

1/31

- Debian プロジェクトに ITP することを、Linux FireWire subsystem, Linux sound subsystem、FFADO に対して連絡。
- <http://mailman.alsa-project.org/pipermail/alsa-devel/2016-January/103693.html>

2/2

- ITP
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=813474>
- RFS
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=813489>

Debian への ITP/RFS 後

2/6

- 樋口大輔さんにスポンサーしてもらおう
- New queue 入りする

2/7

- 朝、ftp-master 入り。unstable リポジトリに入る
 - <https://packages.debian.org/source/sid/libhinawa>
- 夜、Ubuntu の Debian AutoSync bot が捕捉。universe リポジトリに入る。
 - <https://launchpad.net/ubuntu/+source/libhinawa>
- 深夜、アップストリームに報告
 - <http://mailman.alsa-project.org/pipermail/alsa-devel/2016-February/104046.html>

まとめ

- libhinawa を Debian プロジェクトに ITP しました



Hack time



成果を記入下さい!!

今回、Hack time 時の成果を記録に残してみます。

<https://debianmeeting.titanpad.com/8>

に皆さんアクセス頂き（認証不要です）各自成果を 18:30
までに記録するようにお願いします。



今後のイベント

今後のイベント

- 2/16 Debian developer 来日迎撃 GPG Beer Signing(2016/2) Debian にて、LXDE のパッケージメンテナをされている Andrew Lee さんが来日され、henrich さんにて、飲み会しながらのキーサイン会が行われます。掲示：
<http://debianjp.compass.com/event/26965/>
- 関西エリア Debian 勉強会。

今後のイベント

- 2/27 (土) OSC 2016 Tokyo/Spring にて、Debian-JP 出展予定。iwamatsu さんが講師。
<http://www.ospn.jp/osc2016-spring/>
- 3/5 (土) 第 137 回東京エリア Debian 勉強会
場所はサイボウズ株式会社さんにて！この日、henrich さんのお手配により、ドイツから公式 Debian Developer の John Paul Adrian Glaubitz さんが合流されます！海外 Developer の Key Sign のチャンス、海外のアクティブな Debian Developer の方とぶっちゃけトークなど出来る貴重なチャンス！さあ、そこの貴方も参加するのだ。

今後のイベント（つづき）

- 参考：Andrew Lee さんの活動

[https:](https://qa.debian.org/developer.php?login=ajqlee)

[//qa.debian.org/developer.php?login=ajqlee](https://qa.debian.org/developer.php?login=ajqlee)

主に LXDE デスクトップ環境のパッケージなど。

- 参考：John Paul Adrian Glaubitz さんの活動

- 詳しくは、<https://qa.debian.org/developer.php?login=glaubitz%40physik.fu-berlin.de> Debian の MATE デスクトップ環境の多数のパッケージ群など。
- m68k/sh4/sparc64 移植作業



今日の宴会
場所

今日の宴会場所

未定

