

# .Debian

銀河系唯一のDebian専門誌

2016年2月13日

特集 : Debian の消費電力管理



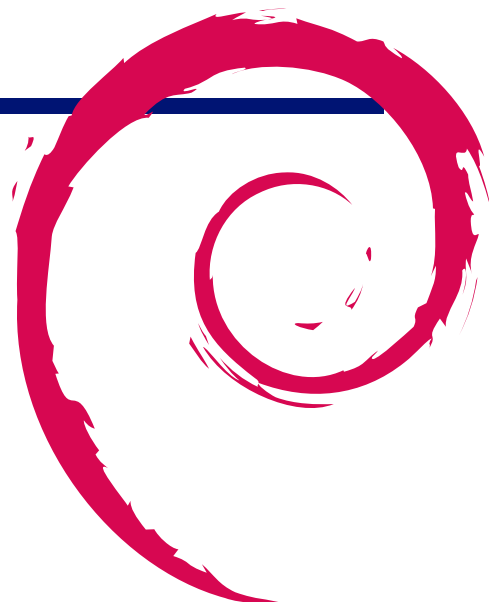
# 会 強 勉 の ア ビ ト

---

## 目次

1	事前課題	2	1.11 野島	3
1.1	takaswie	2	2 Debian Trivia Quiz	4
1.2	mkouhei	2	3 最近の Debian 関連のミーテ ィング報告	5
1.3	Poeto	2	3.1 第 135 回東京エリア Debian 勉強会	5
1.4	kenhys	2	4 Debian GNU/Linux 上での省 電力設定について	8
1.5	iwamatsu	2	4.1 はじめに	8
1.6	Charies	2	4.2 省電力設定するためには	8
1.7	dictoss	2	4.3 省電力設定するためのツール	11
1.8	rosh	3	4.4 まとめ	14
1.9	wskoka	3		
1.10	yy-y-ja.jp	3		

---



## 1 事前課題

野島 貴英

今回の事前課題は以下です:

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

この課題に対して提出いただいた内容は以下です。

### 1.1 takaswie

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

### 1.2 mkouhei

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 ) のキーサーパ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

### 1.3 Poeto

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

### 1.4 kenhys

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

### 1.5 iwamatsu

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

### 1.6 Charies

1. Q.hack time に何をしますか?  
A. パッケージング。

### 1.7 dictoss

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

## 1.8 rosh

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

## 1.9 wskoka

1. Q.hack time に何をしますか？  
A. tilegx

## 1.10 yy-y-ja-jp

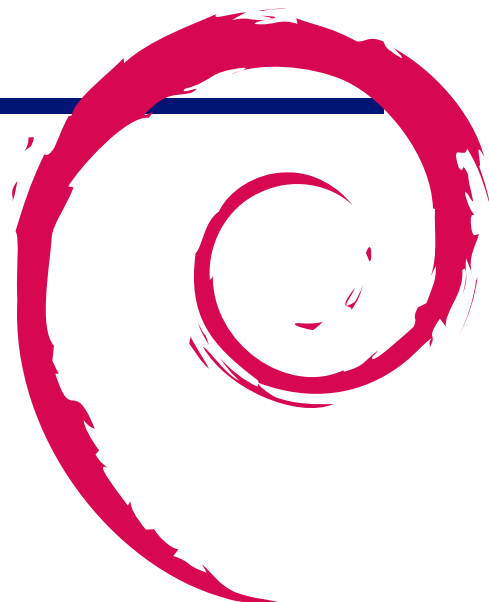
1. Q.hack time に何をしますか？  
A. DDTSS

## 1.11 野島

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集中！  
貴殿好みの勉強会にしてくれてもよくてよ！

## 2 Debian Trivia Quiz

野島 貴英



Debian の昨今の話題についての Quiz です。

今回の出題範囲は `debian-devel-announce@lists.debian.org` や `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、`debtags` 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 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？

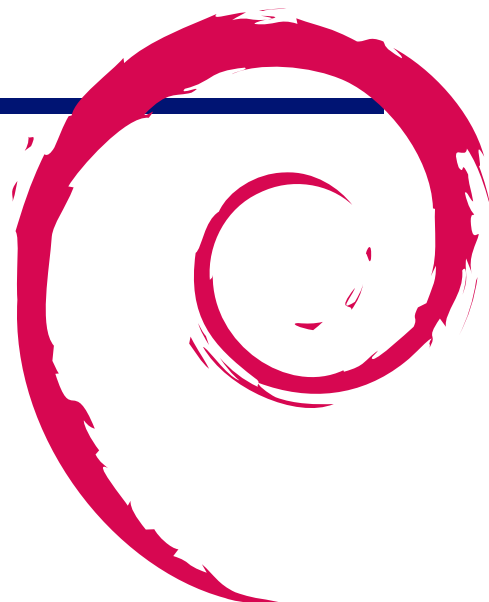
- A `php 7.0`
- B `php 5.6`
- C `php` って何？

問題 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`

## 3 最近の Debian 関連のミーティング報告

野島 貴英



### 3.1 第 135 回東京エリア Debian 勉強会

今回場所は dots さんをお借りしての開催でした。参加者は 10 名でした。  
セミナー内容は、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。

「 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けました。あとは、実行あるのみですね。

「 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 Report<sup>\*1</sup>が行われました。

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。表は前回の hack time 中の成果の一覧となります。(敬称略)

---

<sup>\*1</sup> <https://bugs.debian.org/813769>

参加者	成果
henrich	<p>proposal for Debian idea</p> <ul style="list-style-type: none"> <li>• integrate piuparts into repository pipeline</li> <li>• replace dput → lintian + piuparts, then upload + prevents regression into repository</li> </ul>
kenhys	<p>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。参考：</p> <ul style="list-style-type: none"> <li>• <a href="https://github.com/takaswie/libhinawa/pull/20">https://github.com/takaswie/libhinawa/pull/20</a></li> <li>• <a href="https://github.com/takaswie/libhinawa/pull/21">https://github.com/takaswie/libhinawa/pull/21</a></li> <li>• <a href="https://github.com/takaswie/libhinawa/pull/24">https://github.com/takaswie/libhinawa/pull/24</a></li> </ul>
tai	<ul style="list-style-type: none"> <li>• キーサインを行った。caff ではなく手動で。Keysigning with the GNU/Linux Terminal(<a href="http://www.phillylinux.org/keys/terminal.html">http://www.phillylinux.org/keys/terminal.html</a>) 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…</li> <li>• Debian Wiki の翻訳し残しの続きをもくもくと。</li> </ul>
wskoka	<p>tilegx 用パッケージ作成 (10 ころうい。トータルは 1 2 0 0 個ころういある。)</p>
rosh	<ul style="list-style-type: none"> <li>• caff でキーサインが出来た</li> <li>• 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。 <ol style="list-style-type: none"> <li>1. Kernel tree: arch/arm/boot/dts/kirkwood-lswvl,lswxl.dts</li> <li>2. <a href="http://lists.infradead.org/pipermail/linux-arm-kernel/2016-January/400949.html">http://lists.infradead.org/pipermail/linux-arm-kernel/2016-January/400949.html</a></li> </ol> </li> </ul>
dictoss	<ul style="list-style-type: none"> <li>• raspberry pi2 の debian jessie から bluetooth テザリングできた</li> <li>• キーサインしました</li> <li>• 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)*<sup>2</sup></li> </ul>
y・y	<ul style="list-style-type: none"> <li>• 参加者と key sign を行った。</li> <li>• Caff で、溜まっていた keysigning 処理のキューをフラッシュできた。</li> </ul>

表 1 hack time 成果

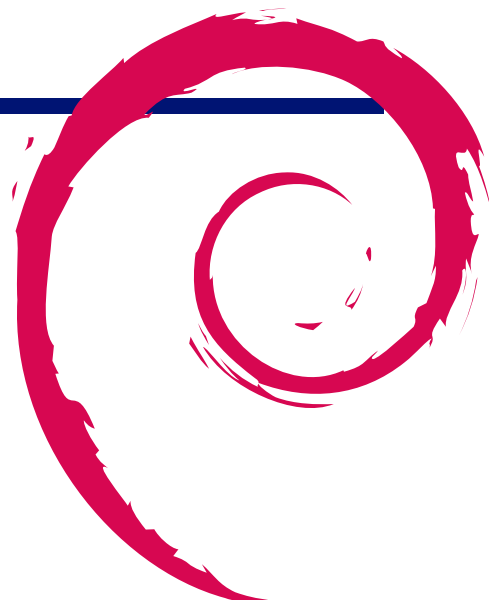
参加者	成果
takaswie	Echo Audio Corp. の Fireworks デバイスモジュール向けコマンドラインツールがだいたい書けた。 <a href="https://github.com/takaswie/hinawa-utils">https://github.com/takaswie/hinawa-utils</a>
yy-y-ja-jp	<ul style="list-style-type: none"> <li>• uim-qt5 バグレポート</li> <li>• キーサイン</li> <li>• DDTSS</li> </ul>
野島	9 個の DDTSS のレビューを行いました。なお、yorick は typo の”M-x Yorick” を見落とし、requeue しました。

表 2 hack time 成果 (つづき)



## 4 Debian GNU/Linux 上での省電力設定について

岩松



### 4.1 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場合が多々あります。数年前と比べると Linux 上の電源管理対応も進んでいますが、まだ Windows などの OS にはまだまだ及ばないところがあります。とは言ってもデフォルトの状態を使うより、少しパラメータを操作してみたり、管理用のツールをインストールするだけでバッテリーの持ちはよいものになります。少しでも Debian でのノート PC ライフを過ごすために今回は Debian GNU/Linux を題材にした 省電力設定方法について説明します。

### 4.2 省電力設定するためには

省電力設定するためにはどのような点で電力を消費しているのかを簡単に理解しておくことが重要です。Linux の場合、大きく分けて以下の点が重要となってきます。

- 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

これらについて、Debian での対応方法について簡単に説明します。

#### 4.2.1 CPU と制御

CPU ですが、Intel 製 CPU の CPU 状態遷移は図 4.2.1 のようになっており、CPU 稼働状態とスリープ状態の切り替えが頻繁に行われています。CPU にはステート (C ステート) という状態があり、この状態によって消費電力が異なり、CPU の復帰時間も変わってきます。最新の CPU ではもっと細かいステートが提供されるようになっています。これらを制御しているのが ACPI となります。

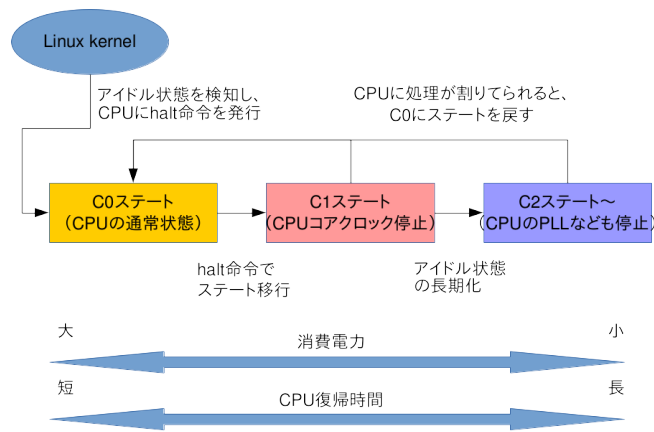


図1 CPU 状態遷移簡易図

実際の CPU は C ステートだけではなく、CPU の電圧やクロック数なども関連してきます。Linux の場合は CPU 周波数スケール機能を使うことで、OS が自動的に制御できるようになっています。これは Linux カーネルの cpufreq によって実装されています。現在の cpufreq に設定されている内容を確認するには cpufrequtils パッケージで提供されている cpufreq-info (2) コマンドを使います。

```

$ 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
  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.
.....

```

図2 cpufreq-info 実行結果

いくつか項目がありますが、重要となるのは「available cpufreq governors」と「current policy」です。available cpufreq governors は CPU の調整可能な速度調整名 (ガバナー) であり、以下のものが用意されています。

ガバナー	内容
ondemand	CPU 負荷が大きい、または小さい時に CPU クロックを大きめに切り替える
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

表3 指定できるガバナー

これらは実際には設定できる値カーネルや環境によって異なる点に注意が必要です。「current policy」は現在設定されているガバナーが表示されます。

この以上から、上記の結果では

- 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

ということがわかります。

これらを設定するには `sysfs` 経由で操作するか、`cpufrequtils` に含まれる `cpufreq-set` コマンドを使います。

- 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"
```

- 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"
```

- 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"
```

- 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"
```

上記のようにして CPU クロックを制御することにより、環境によって無駄なく CPU を利用できるようになります。設定した値は再起動すると消えるので、`/etc/sysfs.conf` に設定しておくか、`cpufreq` を設定するデーモン `cpufreqd` を使うとよいでしょう。

#### 4.2.2 動作しているデバイスの設定

使っている個々のデバイスの設定も重要となります。例えば無線 LAN や Bluetooth を使わないのに有効にしているとそれだけで電力を消費してしまいます。よって使用する環境に応じてこれらを制御する必要が出てきます。

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説明します。

- ラップトップモード

Linux の場合はカーネルのモードとして、ラップトップモードが設定できるようになっています。これは以下のようにして設定します。

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

あと NMI の watchdog (`nmi_watchdog`) も無効化しておきます。これはこれはカーネルハングアップを定期的にチェックする機構をコントロールするフラグです。

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

- USB

USB は `/sys/bus/usb/devices/` 以下に対して設定を行います。例えば、`/sys/bus/usb/devices/usb1` に対して電源供給を切りたい場合は `/sys/bus/usb/devices/usb1/power/control` を `off` に設定します。自動的にサスペンドさせたい場合には `/sys/bus/usb/devices/usb1/power/autosuspend` に対して `1` を設定します。

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

設定を起動時に適用したい場合は、再起動時に初期化されてしまう事とデバイスの USB 位置が変わる事があ

りますので、udev の rules ファイルを使って設定するのがよいでしょう。

```
$ 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"
```

注意しなければいけない点としては USB をなんでも設定してしまうとキーボードが動作しなくなる可能性もあるため、ベンダー ID、デバイス ID などを確認した上で設定しましょう。

- 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します。無線 LAN が wlan0 の場合は以下のように設定することによって制御できます。

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

これも udev の rules ファイルを使って設定すると良いです。

```
$ 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

サウンドの場合も sysfs 経由で設定します。ドライバによって設定出来ない場合がありますが、INTEL のサウンドコントローラの場合は、power\_save があるので、これを 1 に設定することによってパワーセーブモードに設定できます。

```
$ 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 に設定します。この場合も sysfs 経由で設定します。PCI も USB と同様に設定先がどのようなデバイスなのか確認してから設定するようにしましょう。

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

#### 4.2.3 動作しているプログラムについて

動作しているプログラムは top コマンドなのでざっくりとした CPU 占有率を確認できますが、実際にどれぐらいの頻度で使われているのかわかりません。アイドル状態であるにもかかわらず、CPU 割り込みが多いプログラム・プロセスが省電力の効果が得にくいものとなりますので、このようなプログラム・プロセスを調べる必要があります。これらを調べるには下記で説明する PowerTop を使うとよいでしょう。

### 4.3 省電力設定するためのツール

先では長々と書きましたが、知識がないユーザが上記を一つづつやっていくのは非常に大変です。Linux では専門の知識がなくとも使っているマシンを省電力状態に設定できるツールがいくつか準備されています。以下ではそれらの使い方について紹介します。

#### 4.3.1 PowerTOP

PowerTOP は Intel が開発しているソフトウェアで、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です。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からプロセスの操作を行う事ができます。

1. インストール

PowerTOP は Debian でも提供されており、apt でインストールできます。

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

## 2. 起動

起動すると図 2 のような画面が表示されます。「The battery reports a discharge rate ...」に現在の消費電力が表示され、現在動作しているプロセスと使用状況がわかります。筆者の環境で、何も設定しない場合は 13W のようです。

```
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/xdm/xdm --seat_seat0
```

図 3 PowerTOP 起動画面

Tunables タブを選択すると調整可能なシステムの設定が表示されます。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となっています。

```
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 (DPDCC-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 (DPDCC-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 (DPDCC-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
```

図 4 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ないため、一度終了し、キャリブレーションを行います。

## 3. キャリブレーション

設定する PC の状態を取得するためにキャリブレーションを行います。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います。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しましょう。

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

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ます。次回の 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します。

## 4. 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ことが確認できるでしょう。

## 5. 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ります。

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、有効にしておきます。

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

[Unit]
Description=PowerTOP

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

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

```
$ sudo systemctl enable powertop
```

#### 4.3.2 TLP を使った設定

PowerTOP の他に TLP というツールもあります。これは PowerTOP のように詳細なレポートは出してくれませんが、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です。もちろん、Debian ではパッケージ化されており、apt でインストールできます。

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

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ます。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います (図 4.3.2)。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ります。このような場合は T 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ります。

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

# Operation mode when no power supply can be detected: AC, BAT
# Concerns some desktop and embedded hardware only.
TLP_DEFAULT_MODE=AC

# Seconds laptop mode has to wait after the disk goes idle before doing a sync.
# 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
...
```

図 5 /etc/default/tlp 例

TLP は systemd やその他 init 用の起動ファイルが用意されているので PC 起動時に設定が反映されるのも良い点です。

#### 4.3.3 省電力設定後

図 4.3.3 が省電力設定した後に PowerTOP で消費電力を確認した内容です。消費電力が 13W から 11W に下がっていることがわかります。また PC 稼働時間も 5 時間半から 6 時間 50 分に伸びていることがわかります。

```

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

```

図 6 省電力設定後

#### 4.4 まとめ

Debian での省電力設定について説明しました。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` 経由で設定します。プログラムやプロセスの詳細な状態の確認するには `PowerTOP` を使います。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いました。また再立ち上げすると省電力設定を再設定する必要がありますので、`systemd` 用の `service` ファイルを別途用意するなどの対策が必要です。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` を使うのがよいでしょう。ただ全ての PC をサポートしているわけではありませんので、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ります。



**Debian 勉強会資料**

2016年2月13日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）

---