

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 134 回 2015 年 12 月度


野島貴英

2015 年 12 月 19 日

- 事前課題発表
- 最近あった Debian 関連のイベント報告
 - 第 133 回東京エリア Debian 勉強会
- Debian Trivia Quiz
- Debian おすすめのパッケージを参加者に聞いてみた
- Debian モバイル wifi ルータ化
- 今日の宴会場所




事前課題

- 
- ① Q.hack time に何をしますか？
A. バグレポートを読んで、直せそうなところを整理します。

- ① Q.hack time に何をしますか？
A. tilegx の debian 化

- ① Q.hack time に何をしますか？
A. 自分の担当するパッケージをメインテナンスする (wide-dhcpv6 と adjtimex)

- 
- ① Q.hack time に何をしますか？
A. もくもくとパッケージングリハビリと bugs.d.o の落穂ひろいを
します。

- ① Q.hack time に何をしますか？
A. 予定は未定

- ① Q.hack time に何をしますか？
A. kfreebsd 関連作業

- 
- ① Q.hack time に何をしますか？
A. windows タブレットに debian インストール

- 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. 前回参加できなかったので、今回は wayland で簡単なサンプルを作りたい。
参考にしたいのは
<http://d.hatena.ne.jp/devm33/20140414/1397473785>

- ① Q.hack time に何をしますか？
A. DDTSS とか



イベント報
告

第133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9名でした。
- セミナ内容は杉本さんによる「Debian GNU/kFreeBSD セットアップガイド 2015 年版」でした。
- 残りの時間で hack time を行い、成果発表をしました。

第 133 回東京エリア Debian 勉強会 (つづき)

「Debian GNU/kFreeBSD セットアップガイド 2015 年版」は、kFreeBSD のセットアップについて最近のやり方についての発表でした。昨今、検索エンジンで検索できる kFreeBSD のセットアップ方法について内容が古いものが多いということです。本発表の資料で今時のやり方にアップデートが行われる（少なくとも検索上位になる）と良いですね。

また、発表後の hack time 中に、

kFreeBSD を一般の IaaS 型クラウド
サービスへ直接インストールに成
功した

強者が現れるなど、面白い展開をしていました。



Debian おの
すすめ の
パッケージ ズ
を参加者
に聞いて
みた

はじめに

前回の東京エリア Debian 勉強会で、「参加者に Debian のおすすめパッケージを聞いてみたらおもしろいんじゃない？」という企画案がでました。今回、東京エリア Debian 勉強会の参加登録時のアンケートを利用して聞いてみました。

screenfetch パッケージ

推薦者 henrich さん

コメント cowsay みたいなちょっとユーモラスなパッケージです。

adjtimex パッケージ

推薦者 rosh さん

コメント 自分がメンテする adjtimex をお勧めいたします。adjtimex パッケージに adjtimexconfig コマンドを実行すれば、70 秒をかけて、kernel に時間に関する tick と frequency のお勧め値が計算し、`/etc/default/adjtimex` に保存されます。再起動の際に、その値が kernel に設定されます。NTP client と違って、バックグラウンド service がありません。起動時に kernel パラメータを設定することで、システム時間がある程度早く・遅くにならないように、精度を向上できます。

推薦者 taiさん

コメント パッケージというよりパッケージビルダー fpm
がお気に入りです。aufs や unionfs とくっつけ
て、`chroot ... sh -c './configure && make' &&`
`fpm ... /unionfs/cow-dir` で手抜き deb や rpm を
量産していました。なぜ fpm は Debian package
化されていないのだろうか？

geeqie パッケージ

推薦者 dictoss さん

コメント 画像ビューア。快適に動きます。

geeqie パッケージ (つづき)

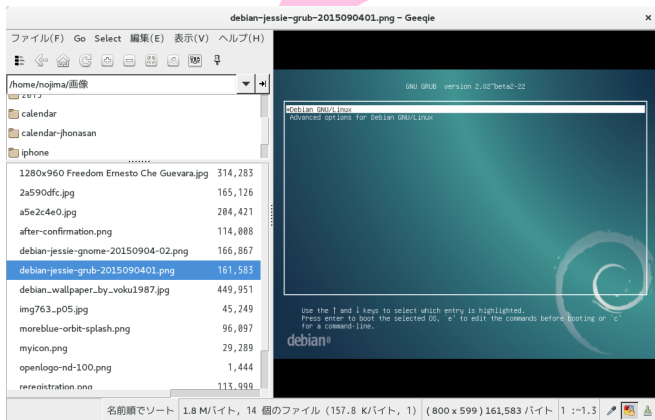


Figure: geeqie の様子

openvpn パッケージ

推薦者 dictoss さん

コメント 自宅へ接続するときに便利です。

python-dateutil パッケージ

推薦者 dictoss さん

コメント ISO 8601 形式の日時文字列から日時オブジェクトへ変換するのに便利なライブラリです。

apache2 パッケージ

推薦者 nametake

コメント 構成に独特のこだわりを感じる

推薦者 yy-y-ja-ja

コメント 自動でお勧めしましょう

wavemon パッケージ

推薦者 **かじ**

コメント wavemon の表示がかっこいい。



wavemon パッケージ (つづき)


```
wavemon
-----
wlan0 (IEEE 802.11abgn, WPA/WPA2), ESSID: "dots.WIFI"
--Level--
link quality: 77% (54/70)
*****
signal level: -56 dBm (2.51 nW)
*****
--Statistics--
RX: 95,738 (23.42 MiB), invalid: 0 ruid, 0 crypt, 0 frag, 79 misc
TX: 25,573 (5.76 MiB), mac retries: 0, missed beacons: 0
--Info--
mode: Managed, access point: F8:9C:E9:4F:77:EB
freq: 5.22 GHz, channel: 44, bitrate: 65 Mbit/s
power mcs: on, tx-power: 15 dBm (31.82 mW)
rate: short limit 7, rts/cts: off, frag: off
encryption: off (no key set)
--Network--
mac: 68:87:28:0F:5A:30, ip: 10.0.1.252/16
1 | info | 2 | hlist | 3 | scan | 4 | | 5 | | 6 | | 7 | prefs | 8 | state | 9 | about | 10 | quit
```

Figure: 起動後の画面

```
wavemon
-----
Level histogram
-----
-----10-----
-----15-----
-----20-----
-----25-----
-----30-----
-----35-----
-----40-----
-----45-----
-----50-----
-----55-----
-----60-----
-----65-----
-----70-----
-----75-----
-----80-----
-----85-----
-----90-----
-----95-----
-----100-----
[?] sig lvl (-57...-54 dBm) [?] no lvl (unknown) [?] S-N ratio (unknown)
1 | info | 2 | hlist | 3 | scan | 4 | | 5 | | 6 | | 7 | prefs | 8 | state | 9 | about | 10 | quit
```

Figure: hlist の画面

wavemon パッケージ (つづき)



The screenshot shows a terminal window titled "wavemon" with a menu bar containing "ファイル(F)", "編集(E)", "表示(V)", "検索(S)", "端末(T)", and "ヘルプ(H)". The main text area displays "--help" at the top left and "don't panic." in the center. At the bottom, a navigation bar includes buttons for "Info", "hist", "can", "quit", "prefs", "help", "about", and "quit".

Figure: help の画面



The screenshot shows a terminal window titled "wavemon" with a menu bar containing "ファイル(F)", "編集(E)", "表示(V)", "検索(S)", "端末(T)", and "ヘルプ(H)". The main text area displays "About" at the top left, followed by the following text:
wavemon - status monitor for wireless network devices
version 0.7.0 (built Sat Apr 25 23:26:11 UTC 2015)
original by jan morganstern <jan@ja-music.de>
distributed under the GNU general public license v3
wavemon uses the wireless extensions by
jean tourrilhes <jt@npl.hp.com>
please send suggestions and bug reports to
gerrit@eng.abdn.ac.uk
<http://www.eng.abdn.ac.uk/orgsps/wavemon/>

Figure: about の画面 (動画でお見せできないのが残念)

推薦者 野島

コメント これ一本だけで、DHCP サーバ、PXE サーバ、簡易 DNS サーバを、少ない設定量であっという間に作れる。設定項目も多すぎず少なすぎずの絶妙な設計であり、使い捨ての環境を作るのにとても便利。参考：第 109 回東京エリア Debian 勉強会「Debian で dnsmasq を使う」
<http://tokyodebian.aliath.debian.org/2014-02.html>

kvm パッケージ

推薦者 野島

コメント コマンドラインにちょこちょこっと Live イメージの iso とともに指定して起動するだけで、仮想環境で様々な OS を試せる。例：Android-x86 の Live デモ起動：
`kvm -m 2G -soundhw es1370 -net nic -net user -cdrom ./android-x86-5.1-rc1.iso`

kvm パッケージ (つづき)

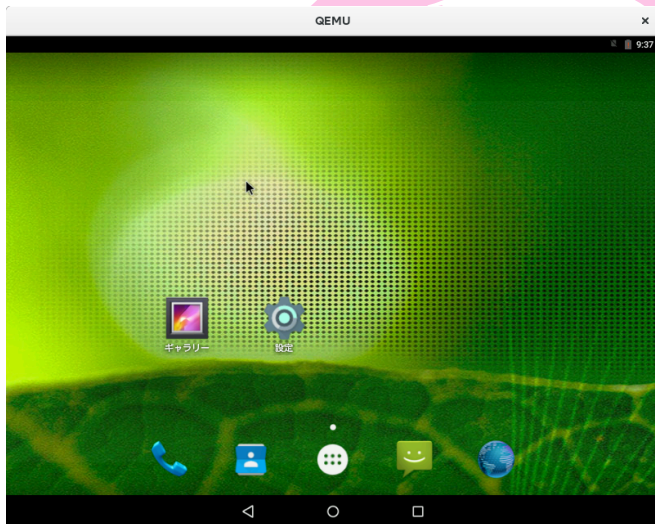


Figure: kvm で Android-x86 Live CD を起動

おわりに

もっといろいろ素敵なパッケージがありましたら、随時
おしえてくださいませ。



Debian モ
バイル wifi
ルータ化

はじめに

最近のモバイルルータは、7GBytes/月、300MBytes/日などの、一定の通信量を超えるとたちまち通信制限がかかってしまい、とても実用にならないぐらいに通信帯域を絞られてしまいます。

たまたま、手元に通信制限が非常にゆるい（というか気にならない）FOMAのモデムがありましたので、こちらとDebianを使ってモバイルルータが作れないかを試してみました。また、Linuxで無線APを作る時の仕組みについてもちょっと調べてみました。

用意するもの

用意するものは次のとおり。

- Debian の動くモバイル PC
- DoCoMo 社 L-05A (モデム。データ定額制の契約であること。)
- BUFFALO WLI-UC-GNM2 (備考：Ralink 製 Ralink RT3070 搭載。900 円/1 個ぐらいの小型 USB 無線 LAN アダプタ)

bridge を作る

Step 1-1 apt install bridge-utils

Step 1-2 vi /etc/network/interface して以下を追記

/etc/network/interface の追記部分：

```
auto br0
iface br0 inet static
    address 192.168.0.1
    netmask 255.255.255.0
    bridge_ports none
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

bridge を作る

Step 1-3 ifup br0

Step 1-4 vi /etc/sysctl.d/bridge-filter-workaround.conf

/etc/sysctl.d/bridge-filter-workaround.conf 中身 :

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

bridge を作る (つづき)

Step 1-5 vi /etc/sysctl.d/forward-yes.conf

/etc/sysctl.d/forward-yes.conf 中身 :

```
net.ipv4.ip_forward=1
```

bridge を作る (つづき)

Step 1-6 `sysctl -p /etc/sysctl.d/bridge-filter-workaround.conf`

Step 1-7 `sysctl -p /etc/sysctl.d/forward-yes.conf`

ここまでの設定を図示

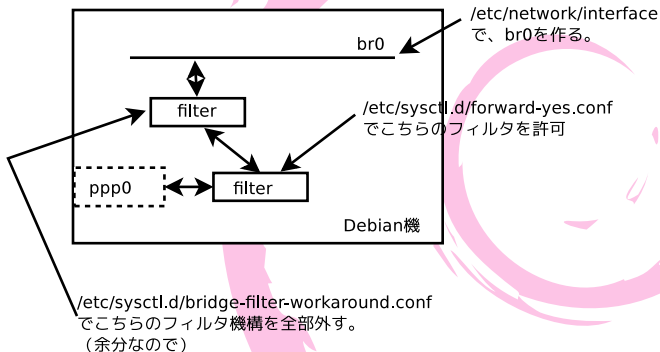


Figure: bridge の設定の状況

L-05A 側設定

Step 2-1 apt install ppp

Step 2-2 vi /etc/ppp/peers/l-05a

L-05A 側設定 (つづき)

/etc/ppp/peers/l-05a の中身 :

```
hide-password
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/l-05a"
debug
/dev/ttyACM0
115200
defaultroute
noipdefault
user ""
remotename l-05a
ipparam l-05a
persist
usepeerdns
idle 300
```

L-05A 側設定 (つづき)

Step 2-3 vi /etc/chatscripts/l-05a

/etc/chatscripts/l-05a の中身 :

```
ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE  
ABORT 'NO DIALTONE' ABORT 'NO DIAL TONE'  
ABORT 'NO ANSWER' ABORT DELAYED  
' ' ATZ  
OK-AT-OK "ATDT*99***5#"  
CONNECT \d\c
```


L-05A 側設定 (つづき)

Step 2-4 `chown root:dip /etc/ppp/peers/l-05a
/etc/chatscripts/l-05a`

Step 2-5 `chmod 640 /etc/ppp/peers/l-05a
/etc/chatscripts/l-05a`

Step 2-6 `vi /etc/ppp/ip-up.d/bridge-up`

`/etc/ppp/ip-up.d/bridge-up` の中身 :

```
#!/bin/sh
iptables -t nat -A POSTROUTING -o $PPP_IFACE \
-j MASQUERADE
iptables -A FORWARD -i br0 -o $PPP_IFACE -j ACCEPT
iptables -A FORWARD -o br0 -i $PPP_IFACE -j ACCEPT
```

L-05A 側設定 (つづき)

Step 2-7 vi /etc/ppp/ip-down.d/bridge-down

/etc/ppp/ip-down.d/bridge-down の中身 :

```
#!/bin/sh
PATH=/bin:/usr/bin:/sbin:/usr/sbin
iptables -t nat -D POSTROUTING -o $PPP_IFACE \
-j MASQUERADE
iptables -D FORWARD -i br0 -o $PPP_IFACE -j ACCEPT
iptables -D FORWARD -o br0 -i $PPP_IFACE -j ACCEPT
```

L-05A 側設定 (つづき)

Step 2-8 `chown 755 /etc/ppp/ip-up.d/bridge-up
/etc/ppp/ip-up.d/bridge-down`

Step 2-9 `pon l-05a`

これで、L-05a はグローバルに接続されるようになります。

ここまでの設定を図示

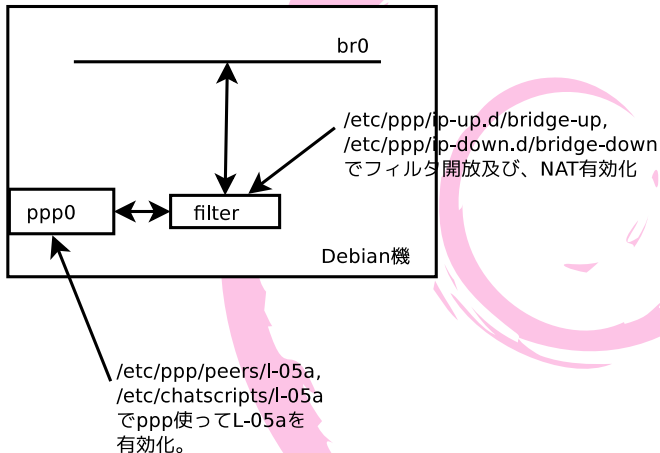


Figure: ppp の設定の状況

補足：L-05A 側トラブルシュート

つながらない時は次のとおりです。

- `tail -f /var/log/debug /var/log/messages` に詳細なログが出ます。こちらを見ると解決のためのヒントが見つかります。
- `pon l-05a` をした後、`ttyACM0` が見つからないというエラーが出る場合があります。この場合は、次の手続きを取ると治ります。

```
modprobe -r uas;modprobe uas;eject /dev/sr0
```

hostapd の設定

いよいよ無線 AP を立てます。

Step 3-1 apt install hostapd firmware-ralink

Step 3-2 ここで、WLI-UC-GNM2 を PC に差し込む。

Step 3-3 lsmod して以下のモジュールがロードされたことを確認。

lsmod の結果抜粋

rt2800usb	28672	0	
rt2x00usb	24576	1	rt2800usb
rt2800lib	90112	1	rt2800usb
rt2x00lib	53248	3	rt2x00usb,rt2800lib, rt2800usb
mac80211	630784	4	rt2x00lib,rt2x00usb, rt2800lib
cfg80211	532480	4	mac80211,rt2x00lib
rftkill	24576	5	cfg80211

hostapd の設定

Step 3-4 `ip addr show` して、`wlxXXXXXXXXXXXXX` という名前の I/F を探す。

Step 3-5 `vi /etc/hostapd/hostapd.conf`

hostapd の設定 (つづき)

/etc/hostapd/hostapd.conf の中身 :

```
interface=wlxXXXXXXXXXXXXX  
bridge=br0  
driver=nl80211  
hw_mode=g  
ieee80211n=1  
ssid=debianspot  
wpa_passphrase=abcdef  
macaddr_acl=0  
wpa=2  
channel=1  
wpa_key_mgmt=WPA-PSK  
wpa_pairwise=CCMP  
logger_syslog=-1  
logger_syslog_level=2  
ctrl_interface=/var/run/hostapd
```


hostapd の設定

Step 3-6 `chmod 600 /etc/hostapd/hostapd.conf`

Step 3-7 `systemctl start hostapd.service`

これで無線 AP が稼働します。iphone/Android 端末で見ると、SSID: debianspot という SSID が見えるはずですが、ただ、まだ、dhcp サービスを有効にしていなかったため、パスワードを入れても接続できません。

hostapd の設定までを図示

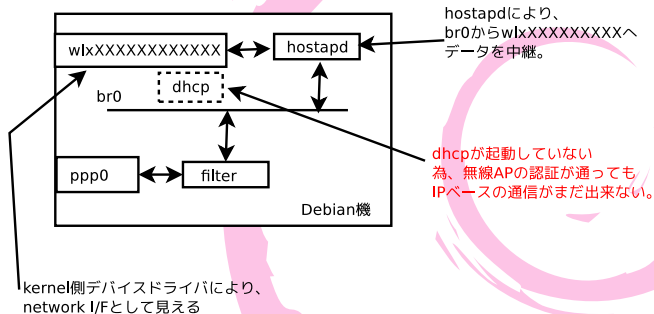


Figure: hostapd 稼働の状況

DHCP の設定

今回簡易的に dhcp サーバを立てるため、dnsmasq を利用します。

Step 4-1 apt install dnsmasq

Step 4-2 vi /etc/dnsmasq.d/dhcp.conf

dhcp.conf の中身 :

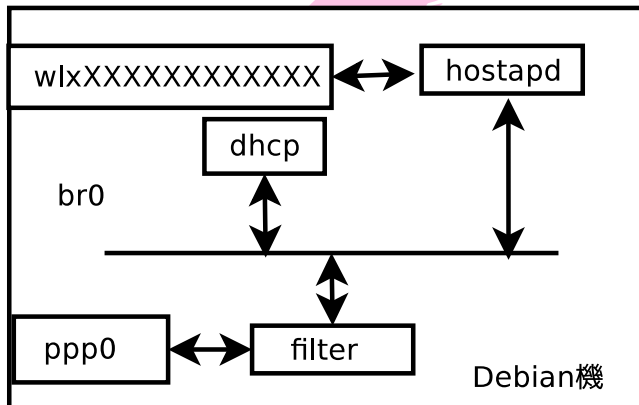
```
interface=br0  
bind-interfaces  
dhcp-range=192.168.2.129,192.168.2.254,  
255.255.255.0,1h
```

DHCP の設定 (つづき)

Step 4-3 `systemctl start dnsmasq.service`

以上で、dhcp サービスが br0 経由で開始され、無事、無線 AP として稼働します。iphone/Android からも SSID: debianspot に接続し、パスワード'abcdef' を入れると、WPA2-PSK にて接続されます。

dhcpの設定までを図示



完全な形でモバイルルータ（無線AP）として稼働。

Figure: 無線 AP 稼働の状況

hostapd はどのように WLI-UC-GNM2 を操作するの？

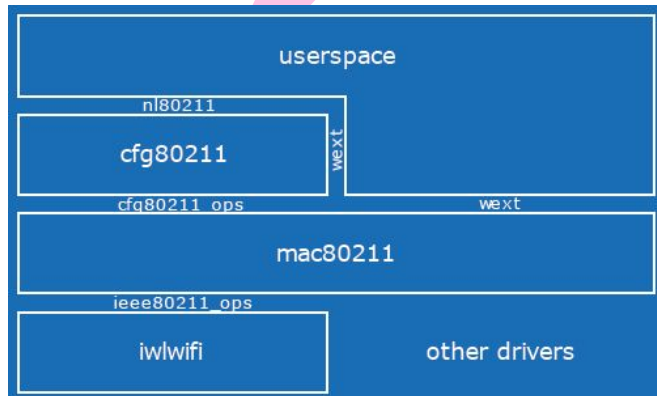


Figure: nl80211 の図示

hostapd はどのように WLI-UC-GNM2 を操作する のか？(つづき)

cfg8011 と hostapd は通信をして WLI-UC-GNM2 を操作するのですが、こちらで使われるプロトコルは、linux の NETLINK が利用されます。



今後のイベント

今後のイベント

- 関西エリア Debian 勉強会。
- 1/23(土) 14:00-19:00 第 135 回東京エリア Debian 勉強会
場所は dots さんにて。



今日の宴会
場所

今日の宴会場所

未定

