

.Debian

銀河系唯一のDebian専門誌

2015年2月21日

特集：kFreeBSDにおけるJail構築



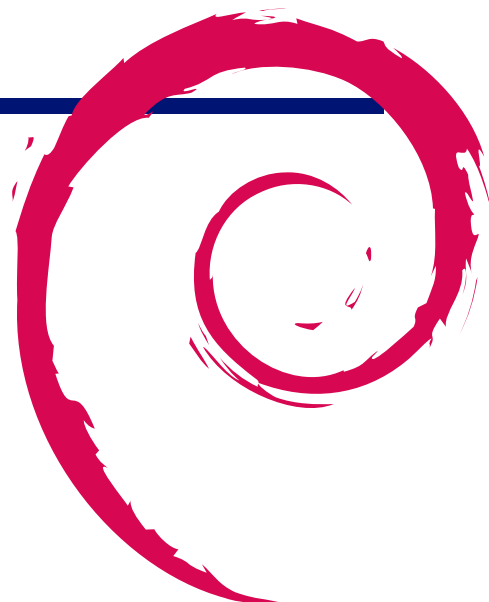
下 ビア ア シ 勉強 会

目次

1	事前課題	2	4	Debian GNU/kFreeBSD における Jail 構築を試してみた	6
1.1	野島	2	4.1	はじめに	6
1.2	alohaug	2	4.2	Debian Ports と Debian GNU/kFreeBSD	6
1.3	それがし某	2	4.3	Debian におけるコンテナ環境の構築方法	6
1.4	ryo_s	2	4.4	Debian GNU/kFreeBSD をホストとした Jail 環境の構築	7
1.5	dictoss	2	4.5	おわりに	11
1.6	hiromiso	2	4.6	参考資料	11
1.7	NOKUBI Takatsugu	2	5	会場での無線 LAN のつなぎ方	12
1.8	yy_y_ja.jp	3	5.1	はじめに	12
1.9	Roger Yan	3	5.2	wpaupplicant 及び /etc/network/interfaces を利用の場合	12
2	Debian Trivia Quiz	4	5.3	その他の無線 LAN 用パッケージを利用の場合	12
3	最近の Debian 関連のミーティング報告	5			
3.1	第 122 回東京エリア Debian 勉強会	5			
3.2	Developer Summit 2015 出展	5			

1 事前課題

野島 貴英



今回の事前課題は以下です:

1. 本日、何の作業をやるかを宣言ください。
2. (オプション) どこで今回の勉強会の開催を知りましたか?
3. (オプション) 何について聞きたい/参加者と話をしたいですか?

この課題に対して提出いただいた内容は以下です。

1.1 野島

1. Q.hack time に何をしますか?
A. RC バグ 対応 (<https://bugs.debian.org/release-critical/>) など。
2. (オプション)Q. 何について聞きたい/参加者と話をしたいですか?
A. Free! Free! Free!について。テクノロジーとしては、未来派な事。

1.2 alohaug

1. Q.hack time に何をしますか?
A. GnuK チラシの英訳。
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?
A. その他。
3. (オプション)Q. 何について聞きたい/参加者と話をしたいですか?
A. 私が、東京 Debian 勉強会に来ているわけ。(あるいは、GnuK と私)

1.3 それがし某

1. Q.hack time に何をしますか?
A. キーサインとか?(以前のキーが無くなってしまったため)
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?
A. Debian JP のメーリングリスト

1.4 ryo_s

1. Q.hack time に何をしますか?
A. 書籍読み
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?
A. 友達や知り合いから直接

1.5 dictoss

1. Q.hack time に何をしますか?
A. kfreebsd をハックする内容を決めて状況を調べる
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?
A. Debian JP のメーリングリスト

1.6 hiromiso

1. Q.hack time に何をしますか?
A. jessie のインストール
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?
A. 友達や知り合いから直接

1.7 NOKUBI Takatsugu

1. Q.hack time に何をしますか?
A. OpenCV を扱った処理
2. (オプション)Q. どこで今回の勉強会の開催を知りましたか?

A. その他のメーリングリスト

たいですか？

A. DDTSS のレビューのお願い

1.8 yy-y-ja-jp

1. Q.hack time に何をしますか？

A. DDTSS (<http://ddtp.debian.net/ddtss/index.cgi/ja>)

2. (オプション)Q. どこで今回の勉強会の開催を知りましたか？

A. その他のメーリングリスト

3. (オプション)Q. 何について聞きたい／参加者と話をし

1.9 Roger Yan

1. Q.hack time に何をしますか？

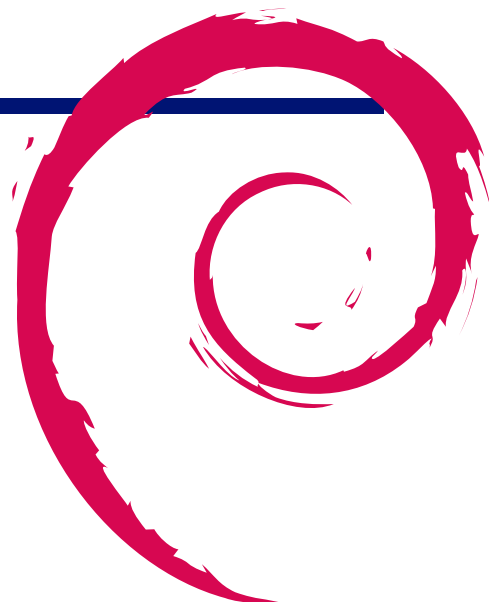
A. bug 起票について勉強するつもり

2. (オプション)Q. どこで今回の勉強会の開催を知りましたか？

A. その他

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. 2015/2/6 に、とある国に security mirror が新設されたことがアナウンスされました。どこの国？

- A 日本
- B アメリカ
- C ジャマイカ

問題 2. 2015/1/26 に、Debian Installer のリリースのアナウンスがされました。こちらのバージョンはいくつ？

- A Jessie RC 1
- B Jessie RC 2
- C Jessie RC 3

問題 3. 過去に諸々の理由により testing から Remove されてしまった Jessie 向けのパッケージについて、再度復活できる最後のチャンスの締切り日はいつでしょう？

- A 2/5 中
- B 2/14 中
- C 2 月末中

問題 4. 2015/2/13 にて GSoC 2015 の Debian に関するプロジェクト募集と GSoC のメンター募集が行われました。2015/2/15 現在寄せられていないプロジェクトはどれ？

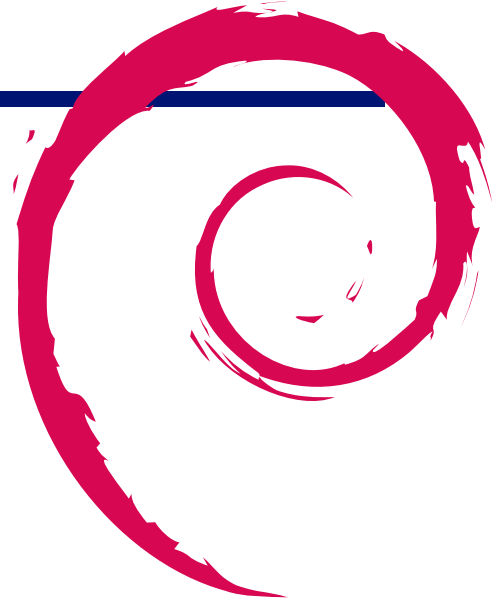
- A Apport of Debian
- B Coinstallable PHP Versions
- C Debian Gnu/Hurd enhancements

問題 5. 2015/2/13 に Reproducible Builds の報告がありました。現在 main パッケージの何 % が完了しているでしょう？

- A 62%
- B 83%
- C 100%

3 最近の Debian 関連のミーティング報告

野島 貴英



3.1 第 122 回東京エリア Debian 勉強会

- 場所はスクウェア・エニックスさんのセミナールームをお借りしての開催でした。
- 参加者は 8 名でした。
- セミナ内容は @henrich さんにより「Emacs 関連パッケージの Debian パッケージ作成について」でした。
- 残りの時間で hack time を行い、成果発表をしました。
- 宴会の代わりに、「祥龍房 新宿イーストサイドスクエア店」で夕食会をやりました。

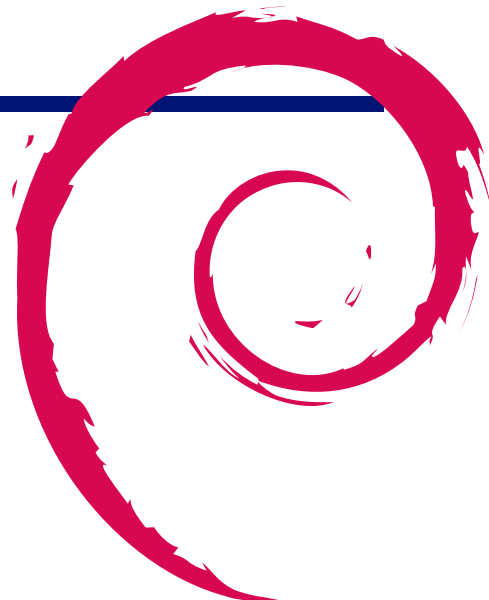
セミナですが、Debian Developer の@henrich さんにより、Emacs 関連パッケージの Debian パッケージ作成について発表がありました。emacs の elisp プログラムをパッケージ化する際、dh_make を使ってテンプレートを作ってパッケージ化するのですが、もろもろ dh_make の機能で足りない・不都合な部分を BTS して直してもらおうという事も併用されているのが印象的でした。勉強会資料にもありますが、「Policy を読んで tips を確認する手間を省くには dh-make パッケージをハックしてしまえ」というのは、Debian 開発の上で非常に正しい姿と思います。

3.2 Developer Summit 2015 出展

- 2015/2/18-2/19 にて目黒雅叙園。http://event.shoeisha.jp/devsumi/20150219/
- @henrich さんにて参加諸々調整が行われ、実現。
- コミュニティLT(2/18)の内容は @henrich さんにより「“Cohice!” adopt diversity of Linux」でした。
- 2015/2/18-2/19 の間、各 Linux ディストリビューション合同 (Debian,ubuntu,openSUSE,vine linux) 及び GnuK で出展しました。

今回、@henrich さんの働きかけにより各 Linux ディストリビューション合同で出展が実現しました。OSC など、Debian のみで出展する場合と比べ話題/ノベリティ出展物としては 4 倍の物量があるので見た目も豪華でした。また、合同で出展したメリットとして、ブースに来ていただいた方へディストリビューションについても的確に応答ができる、Debian 以外のディストリビューションと比較と要望について、参加者を巻き込んでその場で議論できる、ブースに来ていただいた方が居ない間に各技術のトレンド、各ディストリビューションのお互いの事情・動向・技術について濃密にディスカッション出来るなど、非常に新鮮で、楽しいひとときを過ごせました。

各分野で頑張っている方々とのコラボレーションは、非常に刺激的です。非常に貴重な体験かと思えます。東京エリア Debian 勉強会では、将来、他分野さんと合同で開催するようなイベントの機会が、必ずあると思しますので、その時は是非参加してみてください。



4 Debian GNU/kFreeBSD における Jail 構築を試してみた

杉本 典充

4.1 はじめに

Debian GNU/kFreeBSD は、FreeBSD カーネルで動作する Debian です。近頃は docker を実装例としたコンテナ環境に復権 (?!) の兆しがあるようにも思えます。そこで、Debian GNU/kFreeBSD をホストとして、FreeBSD Jail 環境の構築を試してみます。

4.2 Debian Ports と Debian GNU/kFreeBSD

Debian Ports ^{*1}とは、さまざまな CPU やカーネルで動作するように移植を行うプロジェクトのことです。

FreeBSD カーネルで起動する Debian を作るプロジェクトがあり、その Debian のことを「Debian GNU/kFreeBSD」と呼んでいます (k は kernel のこと)。現在では Intel CPU のアーキテクチャのみあります (kfreebsd-amd64, kfreebsd-i386)。Debian 6 (Squeeze) ではテクノロジープレビューとして初めて stable リリースされました。Debian 7 (Wheezy) でも継続して stable リリースされたのですが、Debian 8 (Jessie) では Drop されることが決まっています。^{*2}

4.3 Debian におけるコンテナ環境の構築方法

Debian において、コンテナ環境を構築するには debootstrap を使います。通常はホスト環境と同じアーキテクチャを引数に指定して実行します。^{*3}

```
# apt-get install debootstrap
# mkdir -p /srv/jail
# cd /srv/jail
# debootstrap --arch=[arch] [release] [dir] http://ftp.jp.debian.org/debian
```

ホスト環境のアーキテクチャと異なるアーキテクチャのコンテナ環境を用意すると普通は動作しません。ところが、Debian ではコンテナ環境と qemu を組み合わせることでエミュレーション動作させつつ、見た目はコンテナ環境でバイナリを実行させる方法があります。これを CrossDebootstrap と言います。^{*4}

^{*1} <https://www.debian.org/ports/>

^{*2} <https://lists.debian.org/debian-devel-announce/2014/09/msg00002.html> において、stable を維持しつつ sid の開発を進めるにはそれなりに人手が必要であるが、kFreeBSD を作業する人手は不足していることが指摘されています。

^{*3} debootstrap については昔の記事にまとめたものがあります。「debootstrap を有効活用してみよう」<http://tokyodebian.aliioth.debian.org/pdf/debianmeetingresume2013-natsu.pdf>

^{*4} <https://lists.debian.org/debian-mips/2011/02/msg00031.html> 2011 年頃に kfreebsd-mips 対応をしていた動きがあるようで、これが動くようになれば kfreebsd で CrossDebootstrap を使える日がくるかもしれません。

```
# apt-get install qemu binfmt-support qemu-user-static
# debootstrap --foreign --arch=[arch] [release] [dir] http://ftp.jp.debian.org/debian
# cp /usr/bin/qemu-[arch]-static /[dir]/usr/bin/
# chroot [dir]
# /debootstrap/debootstrap --second-stage
```

4.4 Debian GNU/kFreeBSD をホストとした Jail 環境の構築

Debian GNU/kFreeBSD 上で Jail 環境を構築してみました。環境は以下で行いました。

- ホスト環境
 - Debian GNU/kFreeBSD unstable amd64
 - FreeBSD-10.1 kernel (10.1 svn274115-2)
- ネットワーク環境
 - ホストとコンテナが同一ネットワーク (192.168.0.0/24)
- コンテナ環境
 - FreeBSD Jail にて起動する

jail コマンドは apt でインストールできます。

```
# apt-get install freebsd-utils
# which jail
/usr/sbin/jail
```

ただし現状では、jls^{*5}、jexec^{*6} コマンドはパッケージ化されていません。そのため、今回は Jail 環境で sshd を動かし、ssh ログインできるようにすることをゴールとします。

Jail 環境を作成して起動するには以下の作業を行います。

- Jail 環境の動作に必要なバイナリや設定ファイル等のディレクトリツリーを配置する
- ネットワークデバイスに IP Alias 設定を行う
- Jail 環境内の動作に必要な /dev やファイルシステムの mount を行う
- jail コマンドを実行して Jail 環境を起動する

4.4.1 kfreebsd-amd64、kfreebsd-i386 の Jail 環境

ホスト環境と同じアーキテクチャの Jail 環境を作成します。以下の手順は kfreebsd-amd64 を指定していますが、kfreebsd-i386 も同様の手順で構築することができます。

```
# mkdir -p /srv/jail
# cd /srv/jail
# debootstrap --arch=kfreebsd-amd64 wheezy jail_kf64_1 http://ftp.jp.debian.org/debian
```

Jail 環境内の設定ファイルを変更します。

```
# cd /srv/jail
# vi jail_kf64_1/etc/resolv.conf
nameserver 192.168.0.1

# vi jail_kf64_1/etc/hostname
jail_kf64_1
```

Jail 環境内で sshd が動くように設定します。

^{*5} jls は起動している jail を一覧表示するコマンドです。Jail 環境として動作しているコンテナは、kernel によって jid という番号を付与され識別されます。

^{*6} jexec は Jail 環境内でコマンドを実行するコマンドです。実行例は 「# jexec 1 /bin/sh」


```
# chroot jail_kf64_1
# apt-get install openssh-server
# passwd
# adduser user
# passwd user
```

Jail 環境を起動するには debootstrap したディレクトリを jail コマンドの引数に指定します。Debian の web サイトに jail を起動する前処理を含めたスクリプトの例があります*7。複数の Jail 環境を構築するため、スクリプトと起動パラメータを記述した設定ファイルに分割するよう修正しました。

設定ファイルは以下です。

```
$ cat jail_kf64_1
#!/bin/sh
JID=1
JID_HOST=jail_kf64_1
JID_IPADDR=192.168.0.61
JID_NETMASK=255.255.255.0
JID_SH_PARAM='-- -c'
JID_CMD='/etc/init.d/rc S && /etc/init.d/rc 2'
```

起動スクリプトは以下です。

```
$ cat jail-bootstrap.sh
#!/bin/sh

. ./j1

echo '----'
echo 'start prison.'
echo '[target] JID: $JID, name: $JID_HOST'
echo '----'
echo

NIC_DEV=em0

# add ip alias
/sbin/ifconfig $NIC_DEV alias $JID_IPADDR netmask $JID_NETMASK

# Linux-like /proc and /sys filesystems
mount -t linprocfs linprocfs /srv/jail/$JID_HOST/proc
mount -t linsysfs linsysfs /srv/jail/$JID_HOST/sys

# Ramdisk required for /run
mount -t tmpfs tmpfs /srv/jail/$JID_HOST/run

# A read-only /dev filesystem with restricted set of devices
mount -t devfs devfs /srv/jail/$JID_HOST/dev
# :XXX: ruleset 4 must be initialised as explained earlier in this Wiki page
devfs -m /srv/jail/$JID_HOST/dev rule -s 4 applyset
# Ensure the jail has some essential devices
for DEVICE in null random urandom zero
do if [ ! -e /srv/jail/$JID_HOST/dev/$DEVICE ]
then
echo 'error: device '/dev/$DEVICE' MUST be available in the jail''
exit 1
fi
done

# Ensure the jail has only a limited set of devices
#for DEVICE in mem kmem
#do if [ -e /srv/jail/$JID_HOST/dev/$DEVICE ]
# then
# echo 'error: device '/dev/$DEVICE' MUST NOT be available in the jail''
# exit 1
# fi
#done

# Compatibility symlink from /dev/shm to /run/shm
ln -s /run/shm /srv/jail/$JID_HOST/dev/

# Optionally enable networking
HOSTNAME=$JID_HOST
# :XXX: this IP address *must* be assigned to one of the host's interfaces before the guest can use it
IP=$JID_IPADDR

mkdir -p /var/run/jail
jail -J /var/run/jail/$JID.jid -c jid=$JID \
name=jail$JID \
path=/srv/jail/$JID_HOST \
host.hostname=$JID_HOST \
ip4.addr=$IP \
command=/bin/sh $JID_SH_PARAM '$JID_CMD'
```

*7 https://wiki.debian.org/Debian_GNU/kFreeBSD/Jails

スクリプトを実行して Jail 環境を起動します。

```
$ sudo ./jail-bootstrap.sh jail_kf64_jail
----
start prison.
[target] JID: 1, name: jail_kf64_1
----

devfs rule: ioctl DEVFSIO_SAPPLY: No such process
[info] Using makefile-style concurrent boot in runlevel S.
hostname: the specified hostname is invalid
kern.module_path: /lib/modules/10.1-0-amd64
sysctl: kern.module_path=/lib/modules/10.1-0-amd64: Operation not permitted
[....] Loading devfs rules...devfs ruleset: ioctl DEVFSIO_SUSE: Operation not permitted
(warning).
mount: tmpfs: Operation not permitted
mount: tmpfs: Operation not permitted
[ ok ] Activating swap...done.
mount: /: unknown special file or file system
mount: /run: unknown special file or file system
mount: /proc: unknown special file or file system
mount: /sys: unknown special file or file system
[ ok ] Activating lvm and md swap...done.
[....] Checking file systems...fsck from util-linux 2.25.2
done.
[ ok ] Cleaning up temporary files... /tmp.
[ ok ] Mounting local filesystems...done.
/etc/init.d/mountall.sh: 59: kill: No such process

[ ok ] Activating swapfile swap...done.
mount: tmpfs: Operation not permitted
mount: tmpfs: Operation not permitted
[ ok ] Cleaning up temporary files....
[....] Starting device state change daemon : devddevd: Can't open devctl device /dev/devctl: Device or resource busy
failed!
[....] Configuring network interfaces...ifconfig: up: permission denied
ifconfig: socket(family 28,SOCK_DGRAM: Protocol not supported
done.
[ ok ] Cleaning up temporary files....
[FAIL] startpar: service(s) returned failure: hostname.sh kldutils ... failed!
[info] Using makefile-style concurrent boot in runlevel 2.
[....] Starting enhanced syslogd: rsyslogdln: failed to create symbolic link '/dev/xconsole': Operation not permitted
. ok
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
```

Jail 環境へ ssh ログインします。

```
$ ssh user@192.168.0.61
user@192.168.0.61's password:

user@jail_kf64_1:~$
```

Jail 環境内のプロセスをすべて停止したい場合は Jail 環境内で以下コマンドを実行します。

```
root@jail_kf64_1:~# sh /etc/init.d/rc 0
```

4.4.2 FreeBSD-10.1-RELEASE の Jail 環境

FreeBSD のディレクトリツリーを取得して展開します。

```
# mkdir -p /srv/jail/jail_fb101_1
# cd /srv/jail/jail_fb101_1
# wget http://ftp.jaist.ac.jp/pub/FreeBSD/releases/amd64/amd64/10.1-RELEASE/base.txz
# tar xvf base.txz
# ls
COPYRIGHT  bin dev lib media proc root sys usr
base.txz  boot etc libexec mnt rescue sbin tmp var
```

Jail 環境内の設定ファイルを変更します。

```
# vi etc/resolv.conf
nameserver 192.168.0.1
```

Jail 環境内で sshd が動くように設定します。

```
# vi etc/rc.conf
sshd_enable='YES'
sendmail_enable='NONE'

# cd ..
# chroot jail_fb101
# passwd
# adduser user
# passwd user
```

FreeBSD-10.1-RELEASE の Jail 環境を起動するスクリプトの設定ファイルは以下です。

```
$ cat jail_fb101_1
JID=3
JID_HOST=jail_fb101_1
JID_IPADDR=192.168.0.63
JID_NETMASK=255.255.255.0
JID_SH_PARAM='''
JID_CMD='/etc/rc''
```

Jail 環境を起動します。

```
$ sudo ./jail-bootstrap.sh jail_fb101_1
----
start prison.
[target] JID: 3, name: jail_fb101_1
----

mount: /srv/jail/jail_fb101_1/usr/src/sys: No such file or directory
mount: /srv/jail/jail_fb101_1/run: No such file or directory
devfs rule: ioctl DEVFSIO_SAPPLY: No such process
/etc/rc: WARNING: $hostname is not set -- see rc.conf(5).
Creating and/or trimming log files.
ln: /dev/log: Operation not permitted
Starting syslogd.
ELF ldconfig path: /lib /usr/lib /usr/lib/compat /usr/local/lib
32-bit compatibility ldconfig path: /usr/lib32
Clearing /tmp (X related).
Updating motd:.
Performing sanity check on sshd configuration.
Starting sshd.
Starting cron.

Mon Feb 16 22:43:36 UTC 2015
```

Jail 環境へ ssh ログインします。

```
$ ssh user@192.168.0.63
Password for user@jail_fb101_1:
user@jail_fb101_1:~ %
```

Jail 環境内のプロセスをすべて停止したい場合は Jail 環境内で以下コマンドを実行します。^{*8}

```
root@jail_fb101_1:/ # sh /etc/rc.shutdown
```

4.4.3 linux-i386 の Jail 環境

【注意】 debian としては動きませんでしたので、試したところまで述べます。

FreeBSD kernel には Linux バイナリ互換機能というのがあります。32bit の linux バイナリを動かすことができます。FreeBSD において Linux バイナリ互換機能を有効にするには、linux.base Ports をインストールし、linux.ko を kldload します。Debian GNU/kFreeBSD では kernel に static link していますので linux.ko のロードは不要でした。

Linux バイナリ互換機能を動かす場合は、ホスト環境の sysctl 設定で linux kernel のバージョンを指定する必要があります。今回は linux-2.6.32 を採用している squeeze にしてみました。^{*9} ^{*10}

^{*8} <https://www.freebsd.org/doc/en/books/handbook/jails-build.html> にそう記述がありプロセスはあらかじめ停止しているように見えますが、kernel ではまだ jid が生きているようでゴミが残ってしまうようです。

^{*9} バージョンが異なった状態で chroot してコンテナ内のバイナリを実行すると「FATAL: kernel too old」というエラーが発生します。(新しくても too old のようです)

^{*10} FreeBSD ports において「linux.base-c6」が実験的に提供されており、機能は CentOS-6 の linux kernel 相当です。CentOS-6 の linux kernel は 2.6.32 を採用しており、squeeze と同じです。

```
# vi /etc/sysctl.conf
compat.linux.osrelease=2.6.32

# sysctl -p /etc/sysctl.conf
compat.linux.osrelease: 2.6.16 -> 2.6.32
```

```
# mkdir -p /srv/jail
# cd /srv/jail
# debootstrap --foreign --arch=i386 squeeze jail_linux32_1 http://ftp.jp.debian.org/debian
```

chroot でコンテナ環境に入ってみます。コンテナ環境の中でシェルは動きますので linux バイナリ互換機能は動いているようです。

```
# chroot jail_linux32_1
I have no name!@test-kf2:/#
```

しかし、debootstrap の second-stage でエラーになりました。

```
I have no name!@test-kf2:/# /debootstrap/debootstrap --second-stage
/debootstrap/debootstrap: 1303: /debootstrap/debootstrap: cannot create //test-dev-null: Operation not supported
E: Cannot install into target '/' mounted with noexec or nodev
```

すんなり動いてはくれない状況です。mount 絡みのエラーですので、mount できるようにコンテナの中を設定するか、debootstrap のスクリプトを修正するなど対処すると動くのかもしれませんが。

今回試せたのはここまででした。

4.5 おわりに

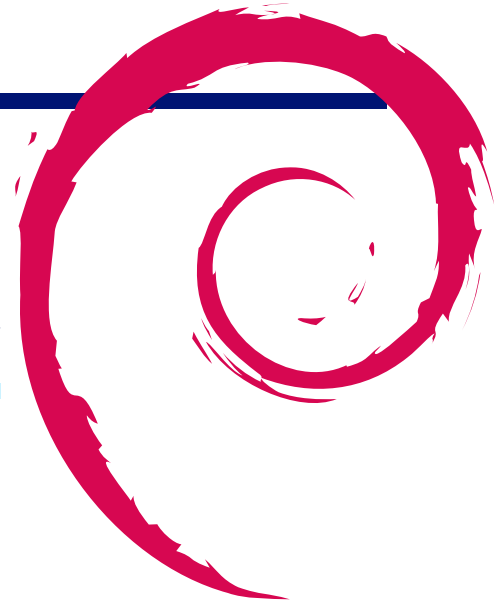
Debian GNU/kFreeBSD で FreeBSD Jail 環境の構築を試してみました。Debian GNU/kFreeBSD on FreeBSD の構成でも動作することは確認していますので試してみるのもおもしろいです。^{*11}

本家 FreeBSD の Jail のように便利に利用するツールのパッケージが足りていない状況ですので、Debian GNU/kFreeBSD の開発を進めるとどんどん使いやすくなると思います。

4.6 参考資料

- Debian/kFreeBSD で jail を使う <http://hachulog.blogspot.jp/2011/11/debiankfreebsdjail.html>
- Debian GNU kFreeBSD Jails https://wiki.debian.org/Debian_GNU/kFreeBSD/Jails
- あんどきゆめんてっど でびあん 2013 年夏号 「debootstrap を有効活用してみよう」 杉本典充 <http://tokyodebian.alioth.debian.org/pdf/debianmeetingresume2013-natsu.pdf>

^{*11} FreeBSD Ports に `/usr/ports/sysutils/debootstrap` があります。



5 会場での無線 LAN のつなぎ方

野島 貴英,Roger

5.1 はじめに

今回試験として、会場側でフィルタ無しのグローバル回線を用意しました。ただ、会場側のセキュリティポリシーにより、wpa-psk AES hidden SSID という方式での提供となります。

以下に Debian マシンでの接続方法を記載します。

また、自分の環境では違うやり方でつながったという方は、野島まで教えて下さい。こちらでもノウハウとして溜めていく予定です。

5.2 wpa_supplicant 及び/etc/network/interfaces を利用の場合

もっとも良いマニュアルは、`/usr/share/doc/wpa_supplicant/README.Debian.gz` となります。困った場合はこちらも含めてご参照下さい。

以下に/etc/network/interfaces の定義について会場の例を記載します。

```
$ sudo vi /etc/network/interfaces
----以下のエントリがなければ追記ここから-----
iface wlan0_debian inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant_debian.conf
----以下のエントリがなければ追記ここまで-----
$ sudo vi /etc/wpa_supplicant/wpa_supplicant_debian.conf
----以下のエントリを追記ここから-----
network={
    ssid=<<会場の SSID>>
    psk=<<会場のパスワード>>
    scan_ssid=1
}
----以下のエントリを追記ここまで-----
$ sudo chmod 600 /etc/wpa_supplicant/wpa_supplicant_debian.conf
$ sudo ifup wlan0=wlan0_debian
```

また、ハマってしまった時のデバッグ方法は、`/usr/share/doc/wpa_supplicant/README.Debian.gz` 中の”4. Troubleshooting” の章が便利です。

5.3 その他の無線 LAN 用パッケージを利用の場合

すみません、自分が情報を持たないため、現場で教えて下さい。



Debian 勉強会資料

2015年2月21日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
