

.Deb

銀河系唯一のDebian専門誌

2019年3月16日

デーモン deb パッケージ解説

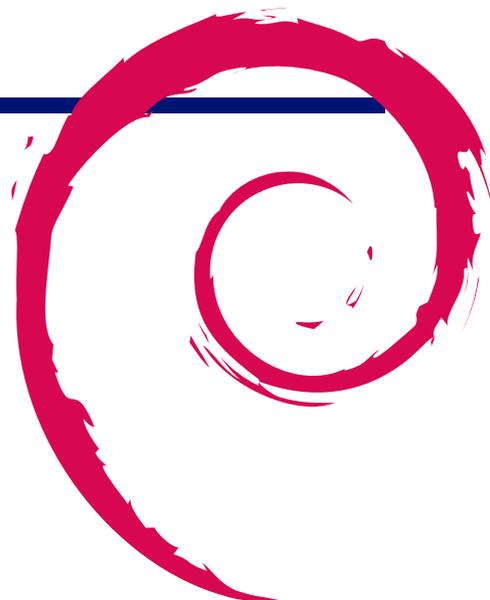


Debian 勉強会

目次	
1	最近の Debian 関連のミーティング報告
1.1	Debian Bug Squash Party Tokyo 2019-01 (第 170 回東京エリア Debian 勉強会) . . .
1.2	OSC 2019 Tokyo/spring (第 171 回東京エリア Debian 勉強会)
2	事前課題
2.1	koedoyoshida
2.2	punit
2.3	yy-y-ja.jp
2.4	su_do
2.5	dictoss
3	systemd と sysvinit のデーモンで動作する Debian パッケージの作り方を調べてみる
3.1	はじめに
3.2	Debian における init の仕組み
3.3	Debian における init が呼び出す起動・停止の設定ファイル
3.4	init 設定を行う debian パッケージ作成方法
3.5	まとめ
3.6	参考文献
4	メモ

1 最近の Debian 関連のミーティング報告

杉本 典充



1.1 Debian Bug Squash Party Tokyo 2019-01 (第 170 回東京エリア Debian 勉強会)

2019 年 1 月 19 日 (土) に第 170 回東京エリア Debian 勉強会として「Debian Bug Squash Party Tokyo 2019-01」を開催しました。会場は南麻布にあるサイオスさんをお借りして行いました。参加者は 4 名でした。

Debian 10 Buster の Transition Freeze^{*1}の直後ということでもあり、Debian Buster のリリースまでに修正が必要なバグの改修に取り組みました。

特に日本語入力環境や無線 LAN デモン (wpasupplicant) など利用環境によっては影響が大きい問題が残っており、解決に向けて議論しました。

1.2 OSC 2019 Tokyo/spring (第 171 回東京エリア Debian 勉強会)

2019 年 2 月 23 日 (土) に Debian JP Project / 東京エリア Debian 勉強会は、OSC 2019 Tokyo/Spring^{*2}に参加しました。OSC の会場は明星大学様でした。

セミナーでは第 171 回東京エリア Debian 勉強会として「Debian Updates」という表題で杉本が発表しました。参加者は 12 名でした。

ブース展示を行い、約 58 名のイベント参加者と交流しました。

^{*1} <https://lists.debian.org/debian-devel-announce/2019/01/msg00008.html>

^{*2} <https://www.ospn.jp/osc2019-spring/>

2 事前課題

dictoss

今回の事前課題は以下です。

1. あなたは Debian パッケージを作成したことがありますか（非公式で個人的で deb パッケージングを作った、も含む）
2. Hack Time は何をしますか

2.1 koedoyoshida

1. はい
2. (回答なし)

2.2 punit

1. はい
2. I'd like to work on improving libhugetlbfs packaging

2.3 yy-y-ja-jp

1. はい

2. Bug Squashing

2.4 su_do

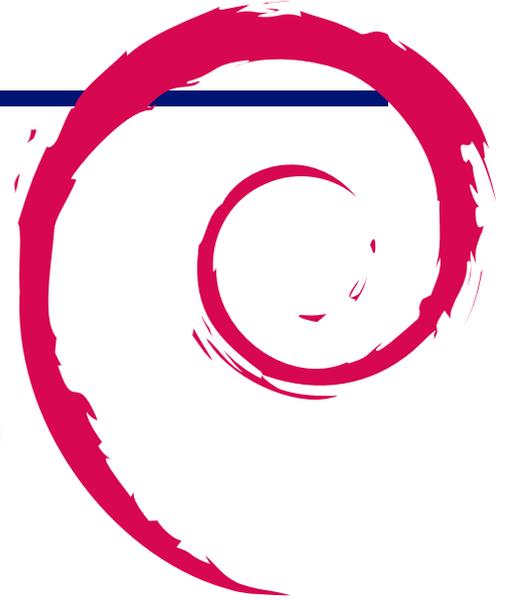
1. いいえ
2. (回答なし)

2.5 dictoss

1. はい
2. Debian Policy を読む

3 systemd と sysvinit のデーモンで動作する Debian パッケージの作り方を調べてみる

杉本 典充



3.1 はじめに

デーモンプログラムを debian パッケージ化するにはどのようにパッケージを作成すればよいのか理解していないため、調べてみました。

3.2 Debian における init の仕組み

一般的に UNIX/Linux 系の OS においてカーネルが最初に呼び出す「init」というユーザランドプログラム^{*3}があり、この init が常駐プログラム (= デーモン) の起動や停止を管理します。

Debian における init システムの解説は、Debian リファレンスの「第 3 章 システムの初期化」に記述があります^{*4}。

Debian で採用している init システムは、昔から存在する「sysvinit」、新世代の「systemd」があり^{*5}、Debian 8 からは systemd を標準の init システムとして採用しています^{*6}。

Debian では systemd と sysvinit を切り替えることができます^{*7}。そのためデーモンプログラムの debian パッケージを作るときは、systemd と sysvinit のどちらの環境でもデーモンを起動できるようにパッケージを作る必要があります。

3.3 Debian における init が呼び出す起動・停止の設定ファイル

3.3.1 systemd: /lib/systemd/system/[package].service

Debian の systemd が管理するデーモンの起動・終了は、「systemctl」コマンドで行います。

systemctl コマンドで管理するデーモンの設定ファイルは、「/lib/systemd/system/[package].service」に置いています。この「[package].service」ファイルにプログラムのファイルパスや環境設定ファイル、実行するコマンドを記述します^{*8}。

例として、nginx-full パッケージを見てみましょう。

nginx を起動する場合は以下のコマンドを実行します。

^{*3} 一般的にプロセス番号「1」のことが多いです。

^{*4} <https://www.debian.org/doc/manuals/debian-reference/ch03.ja.html>

^{*5} 他にも OpenRC という init もあります。

^{*6} systemd が標準なのは Debian GNU/Linux の場合です。systemd は linux カーネル専用のプログラムのため、kFreeBSD や Hurd では動きません。

^{*7} <https://wiki.debian.org/Init>

^{*8} service ファイルの記述方法は、<https://www.freedesktop.org/software/systemd/man/systemd.service.html> を参照。

```
# systemctl start nginx
```

nginx を停止する場合は以下のコマンドを実行します。

```
# systemctl stop nginx
```

上記の systemctl コマンドを実行したとき、以下の”nginx.service”に従って nginx の起動・終了を systemd が行っています。

```
$ cat /lib/systemd/system/nginx.service | grep -v "\#"

[Unit]
Description=A high performance web server and a reverse proxy server
Documentation=man:nginx(8)
After=network.target nss-lookup.target

[Service]
Type=forking
PIDFile=/run/nginx.pid
ExecStartPre=/usr/sbin/nginx -t -q -g 'daemon on; master_process on;'
ExecStart=/usr/sbin/nginx -g 'daemon on; master_process on;'
ExecReload=/usr/sbin/nginx -g 'daemon on; master_process on;' -s reload
ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid
TimeoutStopSec=5
KillMode=mixed

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

3.3.2 sysvinit: /etc/init.d/[package]

Debian の sysvinit が管理するデーモンの起動・終了は、”service” コマンドで行います*9。

service コマンドで管理するデーモンの設定ファイルは、”/etc/init.d/[package]”に置いています。この”[package]”ファイルはシェルスクリプトであり、プログラムのファイルパスや環境設定ファイル、実行するコマンドを記述します。

service コマンドから呼び出すスクリプトを作成するにあたり、以下のヘルパースクリプトがあります。

- /lib/init/vars.sh
- /lib/lsb/init-functions
- /sbin/start-stop-daemon

例として、nginx-full パッケージを見てみましょう。

nginx を起動する場合は以下のコマンドを実行します (systemctl の場合とコマンドの順番が違います)。

```
# service nginx start
```

nginx を停止する場合は以下のコマンドを実行します。

```
# service nginx stop
```

上記の service コマンドを実行したとき、nginx の起動・終了を ”/etc/init.d/nginx” が行っています。
長いため一部省略しています。

*9 Debian GNU/Linux 8 以降では systemd がデフォルトのため <https://wiki.debian.org/Init> の手順を行うと sysvinit に変更できません。

```

$ cat /etc/init.d/nginx

#!/bin/sh

### BEGIN INIT INFO
# Provides:          nginx
# Required-Start:    $local_fs $remote_fs $network $syslog $named
# Required-Stop:     $local_fs $remote_fs $network $syslog $named
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: starts the nginx web server
# Description:       starts nginx using start-stop-daemon
### END INIT INFO

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/sbin/nginx
NAME=nginx
DESC=nginx

# Include nginx defaults if available
if [ -r /etc/default/nginx ]; then
    . /etc/default/nginx
fi

STOP_SCHEDULE=' ${STOP_SCHEDULE:-QUIT/5/TERM/5/KILL/5} '

test -x $DAEMON || exit 0

. /lib/init/vars.sh
. /lib/lsb/init-functions

# Try to extract nginx pidfile
PID=$(cat /etc/nginx/nginx.conf | grep -Ev '\s*#' | awk 'BEGIN { RS=""; }' { if ($1 == 'pid') print $2 }' | head -n1)
if [ -z "$PID" ]; then
    PID=/run/nginx.pid
fi

if [ -n "$ULIMIT" ]; then
    # Set ulimit if it is set in /etc/default/nginx
    ulimit $ULIMIT
fi

start_nginx() {
    # Start the daemon/service
    #
    # Returns:
    # 0 if daemon has been started
    # 1 if daemon was already running
    # 2 if daemon could not be started
    start-stop-daemon --start --quiet --pidfile $PID --exec $DAEMON --test > /dev/null \
        || return 1
    start-stop-daemon --start --quiet --pidfile $PID --exec $DAEMON -- \
        $DAEMON_OPTS 2>/dev/null \
        || return 2
}

(snip)

stop_nginx() {
    # Stops the daemon/service
    #
    # Return
    # 0 if daemon has been stopped
    # 1 if daemon was already stopped
    # 2 if daemon could not be stopped
    # other if a failure occurred
    start-stop-daemon --stop --quiet --retry=$STOP_SCHEDULE --pidfile $PID --name $NAME
    RETVAL=' $? '
    sleep 1
    return "$RETVAL"
}

(snip)

case "$1" in
    start)
        log_daemon_msg "Starting $DESC" "$NAME"
        start_nginx
        case "$?" in
            0|1) log_end_msg 0 ;;
            2)  log_end_msg 1 ;;
            *)
                ;;
        esac
    ;;
    stop)
        log_daemon_msg "Stopping $DESC" "$NAME"
        stop_nginx
        case "$?" in
            0|1) log_end_msg 0 ;;
            2)  log_end_msg 1 ;;
            *)
                ;;
        esac
    ;;
    *)
        echo "Usage: $NAME {start|stop|restart|reload|force-reload|status|configtest|rotate|upgrade}" >&2
        exit 3
    ;;
esac

```

3.4 init 設定を行う debian パッケージ作成方法

3.4.1 前提

Debian GNU/Linux 9 (stretch) の環境を前提とします。

debian パッケージを作成するツールは、ほとんどのパッケージで debhelper が使われています*¹⁰。今回は、debhelper を使ってパッケージ作成を行うことを前提として説明します。

debian パッケージを作成するにあたり、基本的な事項を知るドキュメントは以下になりますので読んでみてください。

- <https://wiki.debian.org/ja/Packaging>
- packaging-tutorial パッケージの PDF ファイル
</usr/share/doc/packaging-tutorial/packaging-tutorial.ja.pdf>*¹¹
- Debian 新メンテナガイド
<https://www.debian.org/doc/manuals/maint-guide/index.ja.html>
- Debian Policy Manual
<https://www.debian.org/doc/debian-policy/>

debhelper で debian パッケージの作成を行う場合は以下のパッケージをインストールしておくといでしょう*¹²。

```
# apt-get update
# apt-get install build-essential dpkg-dev devscripts debhelper dh-make quilt
```

3.4.2 systemd に対応する設定

以下の web ページに解説ドキュメントがありますので、これを解説していきます。

- <https://wiki.debian.org/Teams/pkg-systemd/Packaging>

systemd の設定を行うには通常のパッケージ作成作業のほか、以下を追加で作業する必要があります。

- debian/rules の修正
- debian/control の修正
- ”debian/package.service” の作成

debian/rules の修正

debian/rules ファイルは、dh_make コマンドを実行したときにひな形ファイルが作成されます。

systemd で管理するデーモンの場合は、以下のように ”-with systemd” の定義を追加します。この ”-with” オプションは debhelper の addon の処理を実行するように指定するものです*¹³。

```
dh $@ --with systemd
```

この ”-with systemd” の addon 指定がある場合は debian パッケージのビルド処理で ”dh_systemd.enable”、”dh_systemd.start” を実行するようになります。

*¹⁰ <https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.ja.pdf> の 2019 年 3 月 4 日の資料において debhelper の採用率は 98% との数字があります。

*¹¹ web 版はこちら。 <https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.ja.pdf>

*¹² 昔は systemd に対応する場合には dh-systemd パッケージが必要でしたが、stretch では debhelper に統合されているためインストール不要です。

*¹³ man dh(1) を参照。

”dh_systemd_enable” は、”# systemctl {enable/disable} package” を実行するよう指定するもので、次回の OS 起動時に自動で package のデーモンを起動するか指定できます。

”dh_systemd_start” は、”# systemctl {start/stop/restart} package” を実行するよう指定するコマンドで、パッケージのインストール処理でデーモンを起動するか、再起動かなどを指定できます。

debian/control の修正

”debian/control” ファイルは、dh_make コマンドを実行したときにひな形ファイルが作成されます。

”debian/control” ファイルを書く上で、debhelper の互換性レベルを記述する”debian/compat” ファイルがあります。この”debian/compat”ファイルの中身が”10”以上であれば”debian/control” ファイルを次のように書きます。

```
Build-Depends: debhelper (>= 9.20160709)
```

もし、”debian/compat” の中身が”10”未満であれば systemd 向けの処理は別途 dh-systemd パッケージが必要のため^{*14}、次のように書きます。

```
Build-Depends: debhelper (>= 9.20150101), dh-systemd (>= 1.5)
```

debian/package.service の作成

”package.service” ファイルは、dh_make コマンドを実行したときにひな形ファイルが作成されません。

そのため、他のパッケージの”package.service” ファイルを参考にしつつ、service ファイルのマニュアルをみて作成してください^{*15}

”package.service” ファイルを作成しておく、debian パッケージのビルド処理中の”dh_installinit” の処理で”package.service” ファイルを”/lib/systemd/system” 配下へインストールするよう処理します。

3.4.3 sysvinit に対応する設定

sysvinit の設定を行うには、通常のパッケージ作成作業のほか、以下を追加で作業する必要があります。

- ”debian/package.init” の作成

debian/package.init の作成

”packages.init” ファイルは、dh_make コマンドを実行したときにひな形ファイルが作成されません^{*16}。

他のパッケージの init ファイルを参考にするか、古いバージョンの debhelper に含まれる”init.d.ex” ファイルを取得して作成してください^{*17}。

”package.init” ファイルを作成しておく、debian パッケージのビルド処理中の”dh_installinit”の処理で”package.init” ファイルを”/etc/init.d” 配下へインストールするよう処理します。

3.5 まとめ

Debian における init の説明とデーモンの起動・終了を行う debian パッケージを debhelper でどう作るか説明しました。debian/[package].service ファイル、debian/[package].init ファイルの存在を覚えておくとよいと思います。

^{*14} Debian 8 jessie 向けのパッケージを作成する場合は該当します。

^{*15} <https://www.freedesktop.org/software/systemd/man/systemd.service.html>

^{*16} 過去には dh_make コマンドで”debian/init.d.ex” というひな形ファイルが作成されたのですが、Debian 8 における systemd への移行の過程で作成されなくなりました。

^{*17} <https://sources.debian.org/data/main/d/dh-make/1.20140617/lib/debian/init.d.ex>

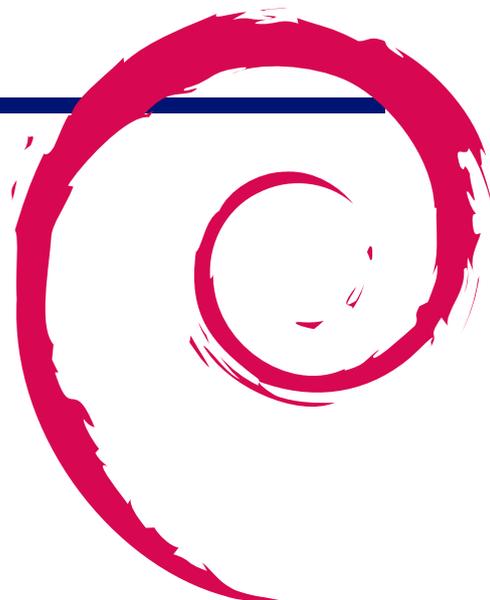
debhelper を使った debian パッケージのビルドでは dh_installinit の処理で systemd/sysvinit の設定ファイルをパッケージに配置しています。

また、1つのソースパッケージから複数のバイナリパッケージを作る場合にファイル名の命名規則が異なるなど違いも出てきます。それらについては他のパッケージを参考にして覚えていくとよいと思います。

3.6 参考文献

- 「Teams pkg-systemd Packaging」 <https://wiki.debian.org/Teams/pkg-systemd/Packaging>
- mkouhei (2014) 「How to create a Debian package of support to sysvinit, upstart, systemd」 https://d.palmtb.net/2014/01/30/how_to_create_a_debian_package_of_support_to_sysvinit_upstart_systemd.html
- @henrich (2016) 「dh-systemd は debhelper 9.20160709 で統合された」 <https://qiita.com/henrich/items/e1651e3284c6b3d0d39e>

4 メモ





Debian 勉強会資料

2019年3月16日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
