

.Debian

銀河系唯一のDebian専門誌

2019年5月18日

/usr Merge 特集



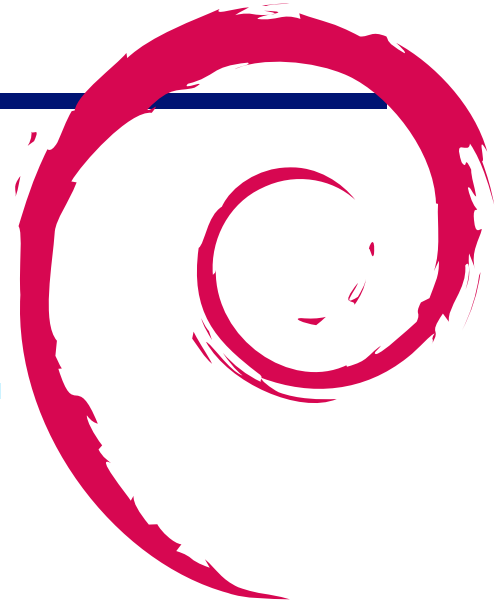
会 強 勉 研 究 会 の ア ー キ ブ ト

目次

1	最近の Debian 関連のミーテ ィング報告	2	2.3	NOKUBI Takatsugu (knok)	3
1.1	第 173 回東京エリア Debian 勉強会	2	2.4	lion (2015fuj)	3
2	事前課題	3	2.5	Kouhei Maeda (mkouhei)	3
2.1	yy-y_ja.jp	3	2.6	dictoss	3
2.2	khibino	3	3	/usr Merge について	4
			3.1	概要	4
			3.2	経緯	5
			3.3	現状	5
			3.4	今後	7
			4	メモ	9

1 最近の Debian 関連のミーティング報告

杉本 典充



1.1 第 173 回東京エリア Debian 勉強会

2019 年 4 月 20 日 (土) に第 173 回東京エリア Debian 勉強会を開催しました。会場は東銀座にある朝日ネットさんをお借りして行いました。参加者は 5 名でした。

セミナー発表は、野首さんの「grml-debootstrap を用いた USB 起動メモリの作成」でした。USB メモリに Debian をインストールして起動する使い方の便利さや USB メモリの作り方や作る時に考慮するとよい点を説明しました。

その後 Hack Time を行い、各自が用意した課題に取り組みました。

2 事前課題

dictoss

今回の事前課題は以下です。

1. Hack Time は何をしますか (How will you work on Hack Time ?)

2.1 yy-y-ja-ja

1. (回答なし)

2.2 khibino

1. (回答なし)

2.3 NOKUBI Takatsugu (knok)

1. メインを buster 環境にしたい

2.4 lion (2015fuj)

1. (回答なし)

2.5 Kouhei Maeda (mkouhei)

1. (回答なし)

2.6 dictoss

1. Debian の開発者向けドキュメントを読む



3 /usr Merge について

yy-y-ja-jp

Debian では Buster から “/usr Merge” がデフォルトになりそうです。概要と状況を整理します。

3.1 概要

“/usr Merge” とは、/bin/, /sbin/, /lib/ などの中にあるもの（バイナリ等）を /usr/ 内に移す、つまり /usr/ 内にバイナリ等をすべて統合することです。例えば /bin/ls を /usr/bin/ls に移します。また、今まで通り /bin/ls でも使えるように互換性のため /bin は /usr/bin へのシンボリックリンクにします。/sbin など同様です。

なお、このような構成にしたい理由は systemd の wiki^{*1}などにまとめられているようです^{*2}。

Debian を新規インストールするときには、Debian インストーラが内部で実行する debootstrap がこの構成を行います。既存の Debian 環境には usrmerge パッケージをインストールすればこの構成にしてくれます^{*3}。なお、以前の構成へ元に戻すことは困難です。

Stretch インストーラで作った Debian 環境の / 直下は次のようなディレクトリ構成でした。

```
$ ls -l /
合計 76
drwxr-xr-x 2 root root 4096 5月 14 00:42 bin
drwxr-xr-x 3 root root 4096 5月 14 00:43 boot
drwxr-xr-x 17 root root 2980 5月 14 00:56 dev
drwxr-xr-x 76 root root 4096 5月 14 00:56 etc
drwxr-xr-x 3 root root 4096 5月 14 00:43 home
lrwxrwxrwx 1 root root 29 5月 14 00:38 initrd.img -> boot/initrd.img-4.9.0-9-amd64
lrwxrwxrwx 1 root root 29 5月 14 00:38 initrd.img.old -> boot/initrd.img-4.9.0-9-amd64
drwxr-xr-x 15 root root 4096 5月 14 00:42 lib
drwxr-xr-x 2 root root 4096 5月 14 00:36 lib64
drwx----- 2 root root 16384 5月 14 00:35 lost+found
drwxr-xr-x 3 root root 4096 5月 14 00:35 media
drwxr-xr-x 2 root root 4096 5月 14 00:35 mnt
drwxr-xr-x 2 root root 4096 5月 14 00:35 opt
dr-xr-xr-x 80 root root 0 5月 14 2019 proc
drwx----- 2 root root 4096 5月 14 00:35 root
drwxr-xr-x 14 root root 460 5月 14 00:57 run
drwxr-xr-x 2 root root 4096 5月 14 00:56 sbin
drwxr-xr-x 2 root root 4096 5月 14 00:35 srv
dr-xr-xr-x 13 root root 0 5月 14 00:57 sys
drwxrwxrwt 8 root root 4096 5月 14 00:56 tmp
drwxr-xr-x 10 root root 4096 5月 14 00:35 usr
drwxr-xr-x 11 root root 4096 5月 14 00:35 var
lrwxrwxrwx 1 root root 26 5月 14 00:38 vmlinuz -> boot/vmlinuz-4.9.0-9-amd64
lrwxrwxrwx 1 root root 26 5月 14 00:38 vmlinuz.old -> boot/vmlinuz-4.9.0-9-amd64
$
```

一方、Buster インストーラで作った Debian 環境では次のようになります。なお、現時点ではインストーラに “/usr Merge” しないようにする設定はなさそうです^{*4}。

*1 <https://www.freedesktop.org/wiki/Software/systemd/TheCaseForTheUsrMerge/>

*2 Debian wiki <https://wiki.debian.org/UsrMerge>

*3 Buster の次のバージョン Bullseye ではインストール推奨となるかもしれません。 <https://bugs.debian.org/841666>

*4 <https://bugs.debian.org/923091>

```

$ ls -l /
合計 60
lrwxrwxrwx 1 root root    7 5月 14 00:36 bin -> usr/bin
drwxr-xr-x 3 root root 4096 5月 14 00:43 boot
drwxr-xr-x 17 root root 3160 5月 14 00:56 dev
drwxr-xr-x 67 root root 4096 5月 14 00:56 etc
drwxr-xr-x 3 root root 4096 5月 14 00:43 home
lrwxrwxrwx 1 root root    30 5月 14 00:38 initrd.img -> boot/initrd.img-4.19.0-4-amd64
lrwxrwxrwx 1 root root    30 5月 14 00:38 initrd.img.old -> boot/initrd.img-4.19.0-4-amd64
lrwxrwxrwx 1 root root    7 5月 14 00:36 lib -> usr/lib
lrwxrwxrwx 1 root root    9 5月 14 00:36 lib32 -> usr/lib32
lrwxrwxrwx 1 root root    9 5月 14 00:36 lib64 -> usr/lib64
lrwxrwxrwx 1 root root   10 5月 14 00:36 libx32 -> usr/libx32
drwx----- 2 root root 16384 5月 14 00:36 lost+found
drwxr-xr-x 3 root root 4096 5月 14 00:36 media
drwxr-xr-x 2 root root 4096 5月 14 00:36 mnt
drwxr-xr-x 2 root root 4096 5月 14 00:36 opt
dr-xr-xr-x 79 root root    0 5月 14 2019 proc
drwx----- 2 root root 4096 5月 14 00:36 root
drwxr-xr-x 15 root root 460 5月 14 00:58 run
lrwxrwxrwx 1 root root    8 5月 14 00:36/sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 5月 14 00:36 srv
dr-xr-xr-x 13 root root    0 5月 14 00:56 sys
drwxrwxrwt 8 root root 4096 5月 14 00:56 tmp
drwxr-xr-x 13 root root 4096 5月 14 00:36 usr
drwxr-xr-x 11 root root 4096 5月 14 00:36 var
lrwxrwxrwx 1 root root    27 5月 14 00:38 vmlinuz -> boot/vmlinuz-4.19.0-4-amd64
lrwxrwxrwx 1 root root    27 5月 14 00:38 vmlinuz.old -> boot/vmlinuz-4.19.0-4-amd64
$

```

3.2 経緯

“/usr Merge” は以前 Fedora で特に行われていたようです^{*5}。なお Fedora は必須化したようです。

Debian では当初 Stretch でデフォルトにする予定でしたが、問題の解決が間に合いそうになく延期されました^{*6}。

その後 Buster に向けて debootstrap 1.0.102 で再度デフォルトにすることにしました^{*7}が、問題があるためデフォルトでなくすべきではという議論があり、Debian 技術委員会 (tech-ctte) にも持ち込まれました^{*8}。

Debian 技術委員会は、debootstrap のメンテナーが決めたことは覆さない、と決定しました^{*9}。同時に、Debian における “/usr Merge” の現状と今後の望ましい姿を示しました。

3.3 現状

Debian 技術委員会の決定により、debootstrap メンテナー次第で “/usr Merge” がデフォルトになるかどうかは決まります。現時点で testing (Buster) 以降のバージョン向けに debootstrap を実行すると次のようになっており、メンテナーが変更しなければこのまま Buster リリースとなります。

- 通常的环境下では “/usr Merge” がデフォルト有効 (無効にするには `--no-merged-usr` オプションを付ける)
- パッケージビルド用の環境 (`--variant=buildd` オプション) ではデフォルト無効

なお、現状では “/usr Merge” された環境でビルドされたバイナリパッケージを “/usr Merge” されていない環境に持っていくと動作しないものがあり、それらのパッケージに随時バグ報告してくれている方がいるようです^{*10}。よくある理由としては、ビルド中にバイナリの場所を `/bin/` の中よりも先に `/usr/bin/` を探しに行き、“/usr Merge” された環境では `/usr/bin/` のパスをコンパイル結果に埋め込むからです。例えば `quilt` パッケージの少し古い修正^{*11}以前のバージョン 0.65-2 を “/usr Merge” された環境でビルドしてみます。`dh_auto_configure` 経由で実行された `./configure` が `/bin/bash` ではなく `/usr/bin/bash`、`/bin/cp` ではなく `/usr/bin/cp`、などを検出してビルドしています。この場合はビルドされたパッケージ内のシェルスクリプトの先頭が `#!/usr/bin/bash` となったため `lintian` がたまたま別件のポリシー違反として検出してくれています。

^{*5} <https://fedoraproject.org/wiki/Features/UsrMove>

^{*6} <https://lists.debian.org/debian-devel-announce/2017/01/msg00004.html>

^{*7} <https://bugs.debian.org/839046>

^{*8} <https://bugs.debian.org/914897>

^{*9} <https://lists.debian.org/debian-devel-announce/2019/03/msg00001.html>

^{*10} <https://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=usrmerge;users=md@linux.it>

^{*11} <https://bugs.debian.org/913226>

```

$ dget http://snapshot.debian.org/archive/debian/20180809T030926Z/pool/main/q/quilt/quilt_0.65-2.dsc
dget: retrieving http://snapshot.debian.org/archive/debian/20180809T030926Z/pool/main/q/quilt/quilt_0.65-2.dsc
(snip)
quilt_0.65-2.dsc:
  Good signature found
  validating quilt_0.65.orig.tar.gz
  validating quilt_0.65-2.debian.tar.xz
All files validated successfully.
dpkg-source: info: extracting quilt in quilt-0.65
(snip)
dpkg-source: info: applying fix-mail-threading
$ cd quilt-0.65/
$ debuild -us -uc
  dpkg-buildpackage -us -uc -ui
dpkg-buildpackage: info: source package quilt
dpkg-buildpackage: info: source version 0.65-2
(snip)
dh_auto_configure -- --with-docdir=/usr/share/doc/quilt --with-sendmail=/usr/sbin/sendmail --with-awk=/usr/bin/awk
  ./configure --build=x86_64-linux-gnu --prefix=/usr --includedir=\${prefix}/include --mandir=\${prefix}/share/man --infodir=\${prefix}/share/info
configure: WARNING: unrecognized options: --disable-silent-rules, --disable-maintainer-mode, --disable-dependency-tracking, --with-docdir
checking for a BSD-compatible install... /usr/bin/install -c
checking whether #! works in shell scripts... yes
checking for bash... /usr/bin/bash
checking whether /usr/bin/bash quoting works... yes
checking for gcp... no
checking for cp... /usr/bin/cp
checking for gdate... no
checking for date... /usr/bin/date
(snip)
dpkg-deb: building package 'quilt' in '../quilt_0.65-2_all.deb'.
dpkg-deb: building package 'quilt-el' in '../quilt-el_0.65-2_all.deb'.
  dpkg-genbuildinfo
  dpkg-genchanges >../quilt_0.65-2_amd64.changes
dpkg-genchanges: info: not including original source code in upload
  dpkg-source --after-build .
dpkg-source: warning: Testsuite field contains value autopkgtest, but no tests control file debian/tests/control
dpkg-buildpackage: info: binary and diff upload (original source NOT included)
Now running lintian quilt_0.65-2_amd64.changes ...
W: quilt source: orig-tarball-missing-upstream-signature quilt_0.65.orig.tar.gz
E: quilt: missing-depends-on-sensible-utils usr/share/quilt/edit
E: quilt: missing-depends-on-sensible-utils usr/share/quilt/header
E: quilt: missing-depends-on-sensible-utils usr/share/quilt/mail
E: quilt: wrong-path-for-interpretter usr/bin/quilt (#!/usr/bin/bash != /bin/bash)
E: quilt: wrong-path-for-interpretter usr/share/quilt/add (#!/usr/bin/bash != /bin/bash)
(snip)
Finished running lintian.
$

```

問題を回避するため、少なくとも Debian 公式のパッケージビルドには当面 “/usr Merge” されていない環境を使うこととなりました。また、Debian 公式レポジトリにアップロードされたパッケージは、“/usr Merge” されていない環境でビルドした結果と “/usr Merge” された環境でビルドした結果を比較してこのような差分が発生していないか Reproducible Builds プロジェクトの CI^{*12}環境で検証も行われており^{*13}、これも元にバグ報告・修正が行われています。

手元で動く pbuilder, cowbuilder や sbuild などクリーンルームでのパッケージビルドツールも同様に “/usr Merge” されていない環境が作られてその中でビルドしてくれます。新たに cowbuilder でビルド環境を作ると次のように構成されます。

```

# cowbuilder --create
I: Invoking pbuilder
I: forking: pbuilder create --buildplace /var/cache/pbuilder/base.cow --mirror http://ftp.jp.debian.org/debian --distribution sid --no-targz
W: /root/.pbuilder does not exist
I: Running in no-targz mode
(略)
Processing triggers for libc-bin (2.28-10) ...
I: Copying back the cached apt archive contents
I: unmounting dev/ptmx filesystem
I: unmounting dev/pts filesystem
I: unmounting dev/shm filesystem
I: unmounting proc filesystem
I: unmounting sys filesystem
#

```

*12 Continuous Integration、継続的インテグレーション

*13 https://tests.reproducible-builds.org/debian/issues/unstable/paths_vary_due_to_usrmerge_issue.html

```
# ls -l /var/cache/pbuilder/base.cow/
合計 80
drwxr-xr-x 2 root root 4096 5月 14 23:50 bin
drwxr-xr-x 2 root root 4096 5月 14 05:25 boot
drwxr-xr-x 2 root root 4096 5月 14 23:50 build
drwxr-xr-x 4 root root 4096 5月 14 23:50 dev
drwxr-xr-x 30 root root 4096 5月 14 23:50 etc
drwxr-xr-x 2 root root 4096 5月 14 05:25 home
drwxr-xr-x 7 root root 4096 5月 14 23:50 lib
drwxr-xr-x 2 root root 4096 5月 14 23:50 lib64
drwxr-xr-x 2 root root 4096 5月 14 23:50 media
drwxr-xr-x 2 root root 4096 5月 14 23:50 mnt
drwxr-xr-x 2 root root 4096 5月 14 23:50 opt
drwxr-xr-x 2 root root 4096 5月 14 05:25 proc
drwx----- 2 root root 4096 5月 14 23:50 root
drwxr-xr-x 4 root root 4096 5月 14 23:50 run
drwxr-xr-x 2 root root 4096 5月 14 23:50 sbin
drwxr-xr-x 2 root root 4096 5月 14 23:50 srv
drwxr-xr-x 2 root root 4096 5月 14 05:25 sys
drwxrwxrwt 2 root root 4096 5月 14 23:50 tmp
drwxr-xr-x 10 root root 4096 5月 14 23:50 usr
drwxr-xr-x 11 root root 4096 5月 14 23:50 var
#
```

パッケージを作るときは安全のためこれらのツールでビルドするようにしましょう。

なお、多くの場合 PATH 環境変数に従ってバイナリの場所を探しているので /bin/ よりも /usr/bin/ が優先されています。なのでもともと “/usr Merge” されていない環境であってもクリーンルームでビルドしていないと次のようなことも起こります。/usr/local/bin/bash が存在する環境で dpkg-buildpackage コマンドを実行すると、PATH 環境変数では /bin (や /usr/bin) よりも /usr/local/bin が優先されるので /bin/bash ではなく /usr/local/bin/bash が使われてしまいます。なおこのように /usr/local/bin/ の中のものを使うのは明らかにポリシー違反なので一部ツールでは対策されており、dpkg-buildpackage コマンドの代わりに devscripts パッケージにある debuild コマンドを使えば回避できます。これは PATH 環境変数に /usr/local/bin を除いたものを設定した上でビルドしてくれます。ですがまずは先ほどのクリーンルームなビルドツールを使うのが良いでしょう。

また、“/usr Merge” された環境 (に限らないですが) の既知の注意点として、dpkg --search (dpkg -S) が指定したファイルのパッケージを見つけてくれないことがあります*14。

```
$ which ls
/usr/bin/ls
$ dpkg -S /usr/bin/ls
dpkg-query: パターン /usr/bin/ls に一致するパスが見つかりません
$ dpkg -S /bin/ls
coreutils: /bin/ls
$
```

なお、おそらくこの挙動が変更されることはなさそうです。debhelper をはじめ様々なツールも現在の挙動に依存しています。ディレクトリの対応付けなどを認識しながらファイルを検索するオプションは別に用意されるかもしれませんが。

3.4 今後

Debian 技術委員会は今後 Debian の望ましい姿として、Buster の次のバージョン (Bullseye) では “/usr Merge” されていない環境とされている環境の両方が使えるべきで、またどちらの環境でビルドされたパッケージも両方の環境で動作すべきと示しました。

先ほどの quilt は、その次のバージョン 0.65-3 で “/usr Merge” された環境でビルドされたパッケージが “/usr Merge” されていない環境に持っていったも動作するよう次のように修正されました。debian/rules ファイルの dh_auto_configure コマンド (結果的には ./configure) 呼び出しの引数最後に --with-bash=/bin/bash を追加し、PATH 環境変数に関係なく決め打ちで /bin/bash を見に行くようにしました。なお ./configure は --with-bash= を BASH= と書くこともできます。今のところこのように差分が見つかった各パッケージを修正中のようにです。新しくパッケージングしたいアプリにも少なくとも差分があればこのような対処が必要になります。

さて、現時点でこれをどうシームレスに実現するのかは不明です。少なくとも Debian 公式レポジトリにアップ

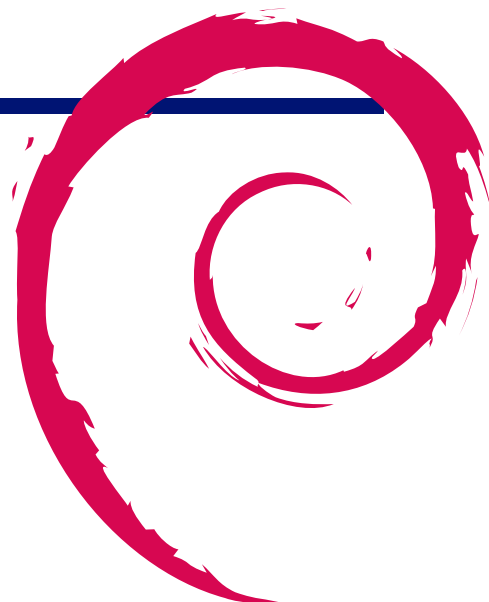
*14 <https://bugs.debian.org/858331>

ロードされた後に検出する CI 環境は既にあります。新しいアプリをパッケージングするときに事前に対応できるようにしないと真に実現したことになるでしょう。そのためには、debhelper 等のパッケージング時のビルド補助ツールが頑張り、lintian 等のパッケージング後の検査ツールも強化していく感じでしょうか。PATH 環境変数で /bin よりも /usr/bin を優先させることもできるかもしれませんが悪影響が読めません。現時点でこの問題の自衛方法としては Reproducible Builds の取り組みで使われているツールを手元で動かすくらいでしょう。diffoscope で 2 つのファイルをバイナリの中身まで再帰的に差分表示してくれます^{*15}。ただ 2 つの環境作って両方でビルドするのも、そしてこの “/usr Merge” で発生している差分を見極めるのも正直面倒なので^{*16}、まずはクリーンルームでビルドして上げておいて、後で CI 環境などに指摘されたら直すくらいが現実的ではないかと思います。

^{*15} x11-apps で発生している差分の表示例 https://tests.reproducible-builds.org/debian/dbd/buster/amd64/x11-apps_7.7+7.diffoscope.html。表示に使った diffoscope コマンド引数はこの HTML ページのタイトル (<title>) に書かれています。

^{*16} 完全な Reproducible Builds は本当は実現すべきですが大変なので、今回発生する差分だけでも対処するのが現実的です。

4 メモ





Debian 勉強会資料

2019年5月18日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
