

# TeX Live の CI テスティング

Norbert Preining



北海道大学、札幌

2018 年 11 月 10 日

# TeX Live 2018



# 概要

- ▶ Subversion と git の踊り
- ▶ T<sub>E</sub>X Live sources の CI
- ▶ T<sub>E</sub>X Live インストーラーの CI
- ▶ まとめ

# T<sub>E</sub>X Live 2018

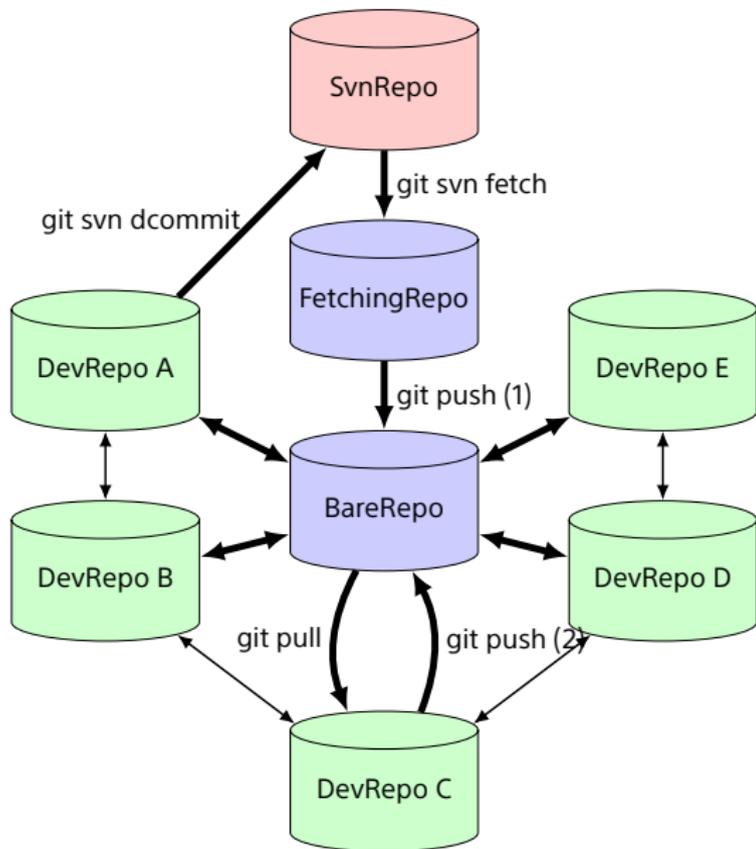


## Subversion と git の踊り

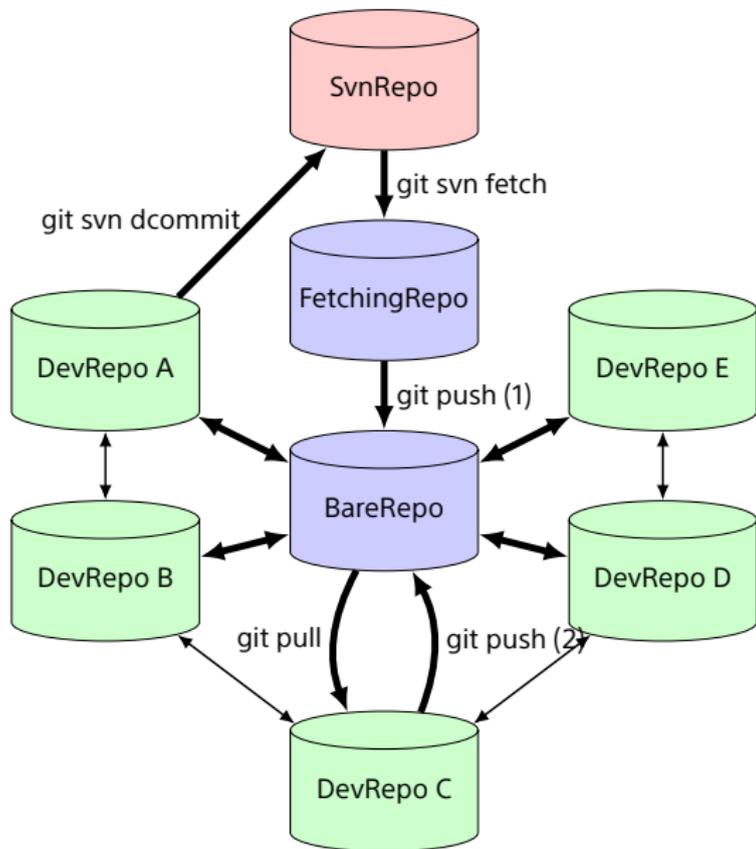
TEX Live 2018





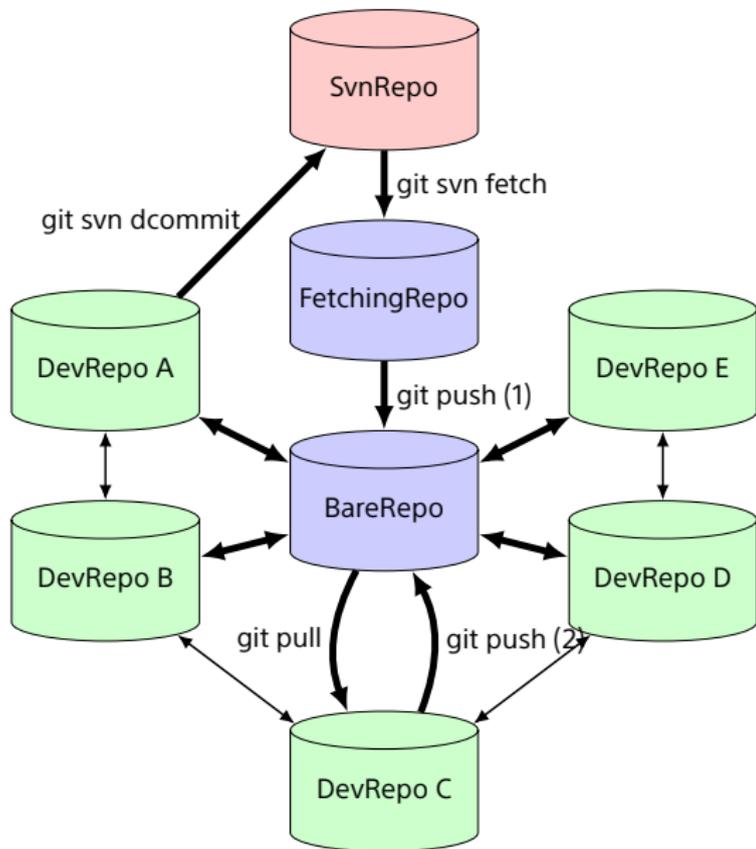


FetchingRepo  
全て  
texlive.info



FetchingRepo  
全て  
texlive.info

BareRepo  
全体の TL  
git.texlive.  
info  
(約 30Gb)



FetchingRepo  
全て  
texlive.info

BareRepo  
全体の TL  
git.texlive.  
info  
(約 30Gb)

Github:  
ソース  
インストーラー

## git リポジトリ

- ▶ 全体の T<sub>E</sub>X Live  
[git.texlive.info/texlive](https://git.texlive.info/texlive) (約 30Gb)  
アノニマスと gitolite(ssh) のアクセス

T<sub>E</sub>X Live 2018



## git リポジトリ

- ▶ 全体の T<sub>E</sub>X Live  
[git.texlive.info/texlive](https://git.texlive.info/texlive) (約 30Gb)  
アノニマスと gitolite(ssh) のアクセス
- ▶ T<sub>E</sub>X Live ソース  
[github.com/TeX-Live/texlive-source](https://github.com/TeX-Live/texlive-source)
- ▶ T<sub>E</sub>X Live インストーラー  
[github.com/TeX-Live/texlive-installer](https://github.com/TeX-Live/texlive-installer)

# T<sub>E</sub>X Live 2018



# git リポジトリ

- ▶ 全体の T<sub>E</sub>X Live  
[git.texlive.info/texlive](https://git.texlive.info/texlive) (約 30Gb)  
アノニマスと gitolite(ssh) のアクセス
- ▶ T<sub>E</sub>X Live ソース  
[github.com/TeX-Live/texlive-source](https://github.com/TeX-Live/texlive-source)
- ▶ T<sub>E</sub>X Live インストーラー  
[github.com/TeX-Live/texlive-installer](https://github.com/TeX-Live/texlive-installer)

LuaT<sub>E</sub>X もあります...

# T<sub>E</sub>X Live 2018



## ミラーの仕組み

cron ジョブで sync が行う

---

全体の T <sub>E</sub> X Live	毎 15 分	5,20,35,50
ソース	毎 30 分	0,30
インストーラー	毎 15 分	7,22,37,52
LuaT <sub>E</sub> X	毎 30 分	18,48

---

# T<sub>E</sub>X Live 2018



## sync script

---

```
git checkout trunk  
git svn fetch --all  
git push
```

---

TEX Live 2018



## sync script

---

```
git checkout trunk  
git svn fetch --all  
git push
```

---

実際にもう少し複雑 ;-)

注意 : svn tag と svn branch は git のブランチになる !

TEX Live 2018



## sync script

---

```
git checkout trunk  
git svn fetch --all  
git push
```

---

実際にもう少し複雑 ;-)

注意 : svn tag と svn branch は git のブランチになる !

その前の設定は <https://www.preining.info/blog/2018/06/git-and-subversion-collaboration/> に参考ください

TEX Live 2018



## 開発者の使い方

- ▶ 普通に git pull
- ▶ 新しいフィーチャーをフィーチャーブランチに開発
- ▶ フィーチャーブランチを git push は OK
- ▶ 直接 trunk に git push は NG
- ▶ trunk にコミットの場合は git svn dcommit で Svn にコミットする

TEX Live 2018



## ブランチ保護

直接 trunk に git push できないように！

TEX Live 2018



## ブランチ保護

直接 trunk に git push できないように！

*Gitolite*

---

```
repo texlive
RW+      = texlive-deployment
RW+ dev/ merged/ = preining
R        = preining
```

---

特別の ssh deploy key で sync command がプッシュ許可がある  
自分 dev/ と merged/ といえブランチ名だけにプッシュしていい

TEX Live 2018



# ブランチ保護

直接 trunk に git push できないように！

*Gitolite*

---

```
repo texlive
RW+      = texlive-deployment
RW+ dev/ merged/ = preining
R        = preining
```

---

特別の ssh deploy key で sync command がプッシュ許可がある  
自分 dev/ と merged/ といえブランチ名だけにプッシュしていい

*Github* Repository – Settings – Branches – Branch protection

TEX Live 2018



## CI テスティング

TEX Live 2018



# CI テスティングとは

- ▶ 新しいコミットが登録した場合は、アクションが行う
- ▶ アクションの例
  - ソースをコンパイル
  - スクリプトを起動
  - $\text{\LaTeX}$  ドキュメントをコンパイル
- ▶ いろんな設定：
  - 1つ・全てのブランチ
  - 全てのコミット最後のだけ
- ▶ 自動に行う

$\text{\TeX}$  Live 2018





# TeX-Live / texlive-source

build **passing**

[Current](#) [Branches](#) [Build History](#) [Pull Requests](#)

[More options](#)

✓ **trunk** **I3build (9nov18)**

git-sha-id: [svn://tucora/texlive/trunk/Build/source@49115-c570f23f-e606-0410-a88d-b1316a301751](#)

↪ [Commit 2bfc34](#)

↪ [Compare 43a9e15.2bfc34](#)

↪ [Branch trunk](#)

[Norbert Preining](#)

↪ **#520 passed**

🕒 [Ran for 24 min 49 sec](#)

🕒 [Total time 1 hr 31 min 8 sec](#)

📅 [15 minutes ago](#)

↪ [Restart](#)

## Build Jobs

✓ # 520.1	</> C	📦 <a href="#">tldr=x86_64-linux package=texlive-bin-x86_64-linux.tar.gz</a>	🕒 21 min 33 sec
✓ # 520.2	</> C	📦 <a href="#">tldr=i386-linux package=texlive-bin-i386-linux.tar.gz</a>	🕒 24 min 46 sec
✓ # 520.3	</> C	📦 <a href="#">tldr=x86_64-linux-musl package=texlive-bin-x86_64-musl.tar.gz</a>	🕒 21 min 44 sec
✓ # 520.4	</> Xcode: xcode7.3 C	📦 <a href="#">tldr=x86_64-darwin package=texlive-bin-x86_64-darwin.tar.gz</a>	🕒 23 min 5 sec

## CI テスティングの概要

- ▶ ソース : Linux, OS/X のうえにビルドテスト
- ▶ インストーラー : Linux, OS/X, Windows のうえにインストールテスト

TEX Live 2018



## CI テスティングの概要

- ▶ ソース : Linux, OS/X のうえにビルドテスト
- ▶ インストーラー : Linux, OS/X, Windows のうえにインストールテスト

使用されたサービス :

- ▶ TravisCI  
<https://travis-ci.org/TeX-Live/Linux, OS/X>
- ▶ AppVeyor: Windows

TEX Live 2018



# TeX Live ソース

TravisCI

<https://travis-ci.org/TeX-Live/texlive-source>

Linux (amd64, i386) – Debian/Jessie

Linux/MUSL (amd64) – Alpine Linux 3.1

OS/X (amd64)

# TeX Live 2018



# TeX Live インストーラー

TravisCI

<https://travis-ci.org/TeX-Live/installer>

Linux, OS/X

AppVeyor

[https:](https://ci.appveyor.com/project/norbusan/installer)

[//ci.appveyor.com/project/norbusan/installer](https://ci.appveyor.com/project/norbusan/installer)

Windows

# TeX Live 2018



## TravisCI のソース・テスト・スクリプト

.travis.yml

---

```
docker run -e TL_MAKE_FLAGS="-j 2" \  
  -v ${TRAVIS_BUILD_DIR}:/texlive -w /texlive \  
  -it --rm i386/debian:jessie sh -c \  
  "apt-get update; \  
  apt-get install -y --no-install-recommends \  
  bash gcc g++ make perl libfontconfig-dev \  
  libx11-dev libxmu-dev libxaw7-dev \  
  build-essential ; \  
  ./Build -C"
```

---

TEX Live 2018



# TravisCI のインストーラーのテスト・スクリプト

Profile:

---

```
selected_scheme scheme-infraonly  
TEXDIR /usr/local/texlive  
option_doc 0  
option_src 0
```

---

---

```
sudo ./install-tl -profile=texlive.profile
```

---

テスト: `tlmgr version` を起動

TEX Live 2018



# デプロイ

git tag の時、ビルドの後特別のアクションが行う：

- ▶ T<sub>E</sub>X Live ソースのコンパイルで生成されたバイナリを.tar.gz に入れる
- ▶ Github にアップする

T<sub>E</sub>X Live 2018



Latest release

📁 texlive-2018.2

🔗 f68b099

# texlive-2018.2: use Nick in Changelog

 **norbusan** released this on Sep 19 · 294 commits to trunk since this release

## ▼ Assets 5

 [texlive-bin-i386-linux.tar.gz](#)

 [texlive-bin-x86\\_64-linux.tar.gz](#)

 [texlive-bin-x86\\_64-musl.tar.gz](#)

 [Source code](#) (zip)

 [Source code](#) (tar.gz)

git-svn-id: svn://tug.org/texlive/branches/branch2018/Build/source@48691 c570f23f-eb1316a301751

# T<sub>E</sub>X Live 2018



# デプロイ

git tag の時、ビルドの後特別のアクションが行う：

- ▶ T<sub>E</sub>X Live ソースのコンパイルで生成されたバイナリを.tar.gz に入れる
- ▶ Github にアップする

新しいリリースの準備の場合は、このバイナリは Subversion に  
いれている  
(Linux, Linux/MUSL の場合は)

# T<sub>E</sub>X Live 2018



## まとめ

- ▶ バグを早く見つけられるように
- ▶ フィーチャーをバグなしまでマージしなくてテストिंगできる
- ▶ 公式の T<sub>E</sub>X Live バイナリを作ってくれる
- ▶ Windows でのインストーラーテストिंग  
(Linux のうえ毎日 [tug.org](http://tug.org) のテストिंगが行う)

# T<sub>E</sub>X Live 2018



## まとめ

- ▶ バグを早く見つけられるように
- ▶ フィーチャーをバグなしまでマージしなくてテストिंगできる
- ▶ 公式の T<sub>E</sub>X Live バイナリを作ってくれる
- ▶ Windows でのインストーラーテストिंग  
(Linux のうえ毎日 [tug.org](http://tug.org) のテストिंगが行う)
- ▶ 僕は Subversion をあまり触らないようになった！

# T<sub>E</sub>X Live 2018



## まとめ

- ▶ バグを早く見つけられるように
- ▶ フィーチャーをバグなしまでマージしなくてテストिंगできる
- ▶ 公式の T<sub>E</sub>X Live バイナリを作ってくれる
- ▶ Windows でのインストーラーテストिंग  
(Linux のうえ毎日 tug.org のテストिंगが行う)
- ▶ 僕は Subversion をあまり触らないようになった！

ありがとうございました！

# T<sub>E</sub>X Live 2018

