A large, light pink circular brushstroke graphic is centered in the background, partially overlapping the title box.

Golang で書かれたツールを Debian パッケージにする

まえだこうへい

2014 年 4 月 19 日

自己紹介

Debian: blockdiag シリーズ、django REST framework、
yrmcds などのメンテナ

言語: 主に Python。
最近 Golang も使おうと勉強中

家庭: 猫一匹 ()、娘二人、妻一人

勉強会: ほぼ Debian 勉強会のみ。でも今日は約半年ぶり

仕事: Python で LDAP ごによごによとか、
django REST framework で API 開発とか

今日の話の流れと前提

- Golang のツールをパッケージ化しようと思った動機
- Debian での Golang を使う方法
- Debian パッケージ化に必要な Golang の知識
- パッケージ化するためのお作法
- Golang のツールを Debian パッケージにする

前提条件： Sid を準備すること

今日の話の流れと前提

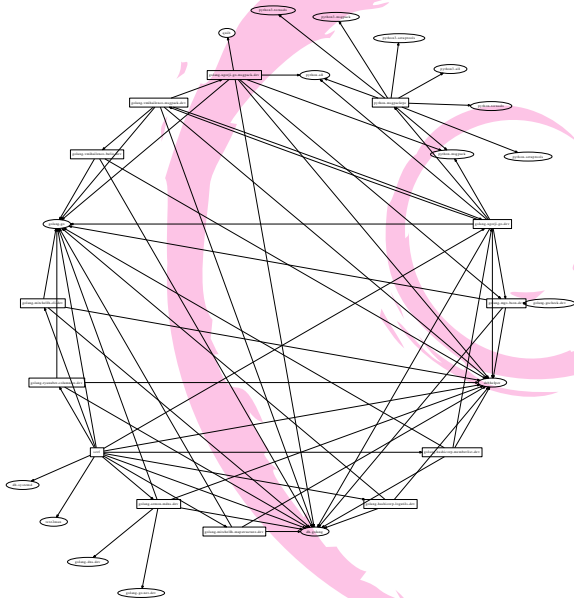
- Golang のツールをパッケージ化しようと思った動機
- Debian での Golang を使う方法
- Debian パッケージ化に必要な Golang の知識
- パッケージ化するためのお作法
- Golang のツールを Debian パッケージにする

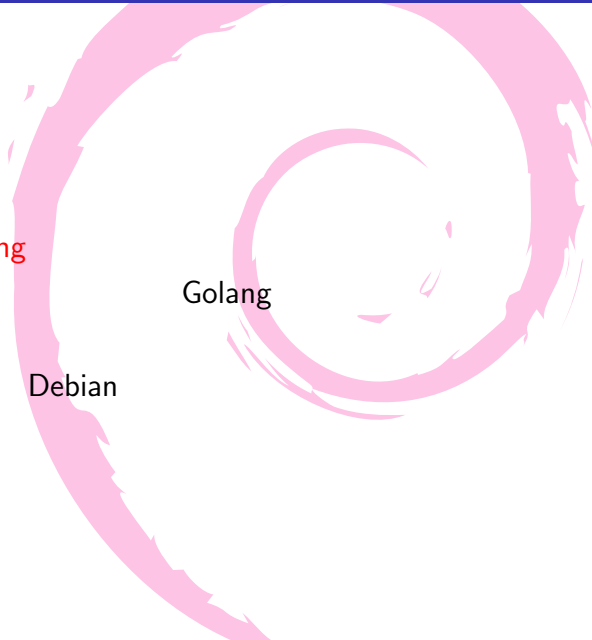
前提条件： Sid を準備すること

Serf を使おうと思ったので

0. やりたいと思っていることを Serf でできそう？
1. main にないので Debian パッケージを作成すっか
2. 依存関係のパッケージも軒並みパッケージ化必要！
3. 全部作ったけど多い！ pkg-go チームに加入！
4. pkg-go のポリシー見たら、パッケージ名など修正必要
5. 修正終わったので、ML 投稿と ITP やるかー イマココ

Debian パッケージ化が必要なものども (四角)



- 
- A large, stylized pink circular brushstroke graphic that frames the text on the right side of the slide. It consists of several overlapping, hand-drawn style loops in a light pink color.
- Golang のツールをパッケージ化しようと思った動機
 - Debian での Golang を使う方法
 - Debian パッケージ化に必要な Golang の知識
 - パッケージ化するためのお作法
 - Golang のツールを Debian パッケージにする

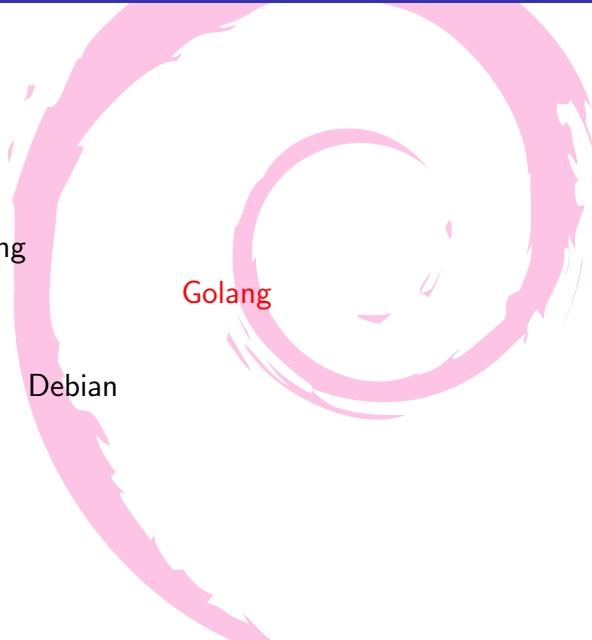
Debian での Golang の環境構築

```
$ sudo apt-get install golang
```


インストールされるパッケージ

- golang-doc
- golang-go
- golang-go-linux-amd64¹
- golang-go.tools
- golang-src
- libjs-jquery
- javascript-common

¹amd64 の場合

- 
- A large, stylized pink circular brushstroke graphic that frames the text on the right side of the slide. It consists of several overlapping, hand-drawn style loops in various shades of pink.
- Golang のツールをパッケージ化しようと思った動機
 - Debian での Golang を使う方法
 - Debian パッケージ化に必要な Golang の知識
 - パッケージ化するためのお作法
 - Golang のツールを Debian パッケージにする

Golang のコンパイル方法

hello world(hello.go) を用意

```
package main

import "fmt"

func main() {
    fmt.Println("Hello world.")
}
```

Golang のコンパイル方法

go build でコンパイル

```
$ go build hello.go
```

Golang のコンパイル方法

実行

```
$ ./hello  
Hello world.
```

Golang のコンパイル方法

静的リンクされていることを確認

```
$ file hello
hello: ELF 64-bit LSB executable, x86-64, \
version 1 (SYSV), statically linked, not stripped
```

標準ライブラリ以外に依存する場合

- 環境変数 GOPATH を設定
- 標準ライブラリ以外の Golang の Debian パッケージは、`/usr/share/gocode` にインストールされる

```
$ export GOPATH=/usr/share/gocode
```

標準ライブラリ以外に依存する場合

golang-dns-dev(Go用のDNSライブラリ)を試してみる

```
$ sudo apt-get install golang-dns-dev
$ dpkg -L golang-dns-dev
(snip)
/usr/share/gocode
/usr/share/gocode/src
/usr/share/gocode/src/github.com
/usr/share/gocode/src/github.com/miekg
/usr/share/gocode/src/github.com/miekg/dns
/usr/share/gocode/src/github.com/miekg/dns/zscan_rr.go
/usr/share/gocode/src/github.com/miekg/dns/zscan.go
```


標準ライブラリ以外に依存する場合

Golang のツールで依存パッケージをインストール場合
ユーザディレクトリにインストールする例

```
$ mkdir $HOME/gocode  
$ export GOPATH=$HOME/gocode  
$ go get github.com/miekg/dns
```

標準ライブラリ以外に依存する場合

import のみ抜粋²

```
package main

import (
    "github.com/miekg/dns"
    "fmt"
    "flag"
)

(snip)
```

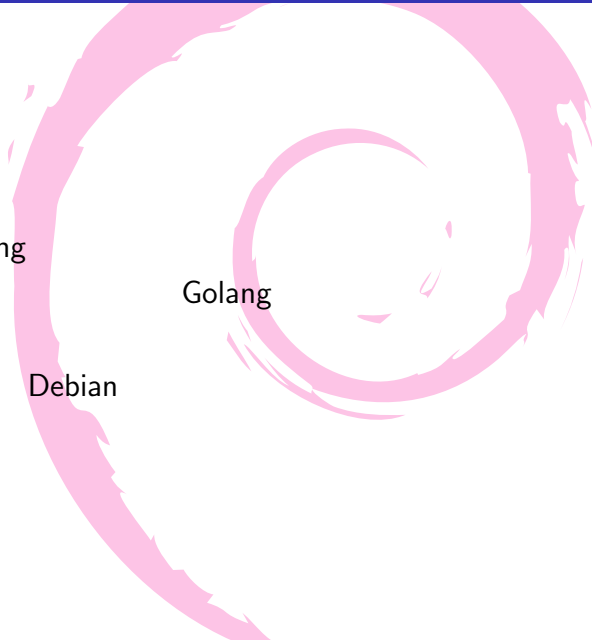
²参照: <https://github.com/mkouhei/godig/blob/devel/dig.go>

標準ライブラリ以外に依存する場合

実行

```
$ go run dig.go
example.org.      2022      IN        SOA       sns.dns.icann.org. noc.dns.icann.org.\
2013103528 7200 3600 1209600 3600

$ go run dig.go -d debian.or.jp.
debian.or.jp.    600      IN        SOA       ns.debian.or.jp. root.debian.or.jp.\
2013091001 600 86400 2419200 1440
```

- 
- A large, stylized pink circular brushstroke graphic is positioned on the right side of the slide, partially overlapping the text area. It consists of several concentric, hand-drawn style rings in various shades of pink, creating a sense of motion and focus.
- Golang のツールをパッケージ化しようと思った動機
 - Debian での Golang を使う方法
 - Debian パッケージ化に必要な Golang の知識
 - **パッケージ化するためのお作法**
 - Golang のツールを Debian パッケージにする

Golang の Debian パッケージ化のお作法

バイナリのみの場合

- 基本的には Depends は必要ない
- デーモン化すると必要になるケースもある
- パッケージ名は Upstream の名称と同じでもよい

Golang の Debian パッケージ化のお作法

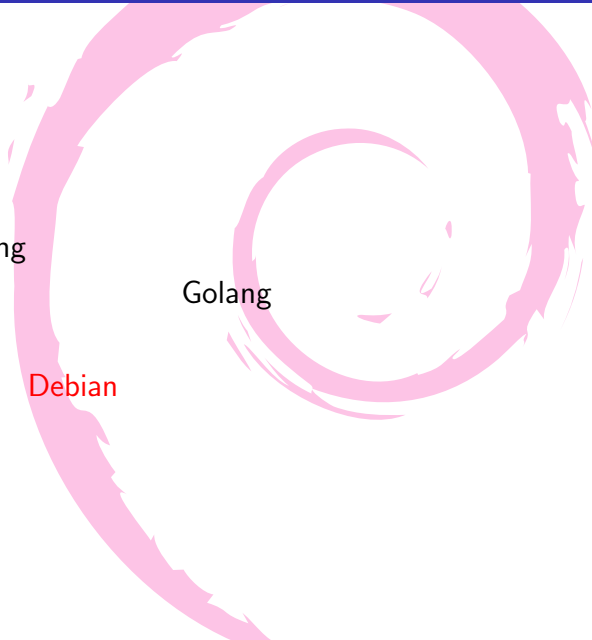
ライブラリのみの場合

- ソースコードの配布だけなのでコンパイルは不要
- バイナリと一緒に配布する場合は、バイナリと同様
- ソースパッケージ、バイナリパッケージとも "golang-" という prefix が必要
- バイナリパッケージには "-dev" という suffix も必要

Golang の Debian パッケージ化のお作法

共通

- ビルドやテスト実行時に、`go get` を使ってはアカン
- 伝統的なバージョンニングがされていなければ”0.0 git20140419-1”というバージョンをつける
- ソースパッケージは `git-buildpackage` で管理する

- 
- A large, stylized pink circular brushstroke graphic that frames the text on the right side of the slide. It consists of several overlapping, hand-drawn style loops in a light pink color.
- Golang のツールをパッケージ化しようと思った動機
 - Debian での Golang を使う方法
 - Debian パッケージ化に必要な Golang の知識
 - パッケージ化するためのお作法
 - Golang のツールを Debian パッケージにする

ソースコードの入手

- tarball があればそれをなれば、リポジトリから入手
- バージョンを確認する³
- ソースパッケージ用の Git リポジトリを作り、`git import-orig` コマンドでインポートする

³なければ前述の 0.0 git20140419 のようなバージョンをつける

ソースパッケージの作成

1. `dh_make` コマンドで `debian` ディレクトリを生成
2. 不要なテンプレートを削除
3. `control`, `copyright`, `rules`, `changelog` などを修正
4. `git-buildpackage` コマンドでパッケージをビルド⁴

⁴main に入っていないパッケージに依存していない場合は、`pbuilder` の方が個人的には楽

- 共通
 - Build-Depends に golang-go
- ライブラリ
 - Build-Depends に dh-golang
 - Depends に golang-go を追記
 - Architecture: all
- バイナリ
 - Architecture: any

- 共通
 - dh コマンドに`--buildsystem=golang` オプションを指定
- ライブラリ
 - 環境変数 `DH_GOPKG` に `GOPATH` で指定したディレクトリしたに展開される名前空間を指定
 - dh コマンドに`--with=golang` オプションを指定
 - `override_dh_auto_install` ターゲットで、
`dh_auto_install -O-buildsystem=golang` を設定
- バイナリ
 - dh コマンドに`--with=golang` オプションが不要

まとめ

- Debian パッケージを作る上で必要な Golang パッケージ、コンパイルの方法
- Golang の Debian パッケージ化
- pkg-go team による Golang パッケージのポリシー