

Монструозный Makefile и оптимальная альтернатива

Даниил Подольский



О докладчике



- Go-евангелист со стажем
- Работаю в YADRO в отделе разработки ОС, в группе общей платформы
- YADRO известна как разработчик и производитель всего, что можно увидеть в серверной и ЦОД: систем хранения данных, серверов, коммутаторов. С недавних пор компания также создает телеком-оборудование операторского класса. Go-разработчики заняты как в СХД, так и в телекоме.

О докладе

- Как мы сегодня собираем наши сервисы
- Какие у нас есть с этим проблемы
- Как мы их решили



О докладе

- Как мы сегодня собираем наши сервисы
- Какие у нас есть с этим проблемы
- ~~Как мы их решили~~



Что такое сборка Go-проекта



- Запуск утилит кодогенерации (`go generate`)
- Запуск `go build`
- Запуск статических анализаторов
 - не только go-кода
- Запуск тестов
 - Не только unit

Что такое сборка Go-проекта



- Установка утилит кодогенерации
- Установка утилит статического анализа
- Запуск утилит кодогенерации (`go generate`)
- Запуск `go build`
- Запуск статических анализаторов
- Запуск тестов
- Запуск утилит обновления (`renovate`)
- Запуск утилит поддержки консистентности зависимостей (`go mod tidy`)
- Запуск утилит контроля безопасности
 - `govulncheck`, `n1laway`



- Отставание по версиям Go
 - Необходимо поддерживать возможность компиляции
- Необходимо постоянно обновлять зависимости
 - Но сохранять совместимость
- Multi-repo
 - Вопреки распространенному мнению, использование multi и single repo - вопрос не только вкуса

СУР-специфика



- Как это - было? - сделано у нас
 - Makefile
 - С инклюдями
 - С git submodules
 - С универсальным сур-go-build docker image

СУР-специфика

- И что с этим не так?



СУР-специфика



- И что с этим не так?



Evgeniy Shkuropat 3 weeks ago

Общий здрасти

А это нормально что у нас в мейк файле есть [diff-coverage](#) и [coverage-diff](#) ?

8 replies



Ovanes Khandzhyan 3 weeks ago

одно создает, другое сравнивает)

```
diff-coverage: #@makehelp Creates a report on the difference in coverage locally
```

```
coverage-diff: #@makehelp Compares test reports obtained using the cyp-coverage-tool
```

таргеты не отражают суть, согласен



Svetlana Zhurova 3 weeks ago

Кавераджа никогда не бывает много...



Daniil Podolskiy 3 weeks ago

facepalm.jpg



- И что с этим не так?
 - Makefile на поверку оказывается shell-скриптом
 - Останавливаться на вопросах поддерживаемости shell-скриптов нам не позволяет человеколюбие
 - Отслеживание связей внутри Makefile оказывается весьма нетривиальной задачей
 - Версионирование сабмодулей - это тема отдельной драмы. Или трагедии?..
 - Самое главное: make это совершенно другого назначения инструмент
 - Он был создан собирать программы, написанные на C
 - Остальные применения - side effects abuse

СУР-специфика



- Disclaimer: make применять все-таки можно

```
сур-docs >  Makefile
1  #!/usr/bin/make -f
2
3  PUMLS=$(shell find ./ -name '*.puml')
4
5  puml:|
6      plantuml -tsvg $(PUMLS)
7      plantuml -tpng $(PUMLS)
```



- Disclaimer: make применять все-таки можно, **но недолго**

```
cyp-docs > Makefile
1  #!/usr/bin/make -f
2
3  CI_UTILS_DIR = .ci_utils
4  -include $(CI_UTILS_DIR)/common.mk
5
6  $(CI_UTILS_DIR)/common.mk:
7      @git submodule update --init --recursive
8
9  PUMLS=$(shell find ./ -name '*.puml')
10
11 help: help-common #@makehelp Show this help
12
13 puml: #@makehelp Render PlantUML diagrams
14     plantuml -tsvg $(PUMLS)
15     plantuml -tpng $(PUMLS)
16
17 lint: #@makehelp Lint directory structure, file names, etc.
18 @echo "=> Linting 'proposals' folder..."
19 @if [ `ls proposals/ | wc -l` \
20     -ne \
21     `find proposals/* -maxdepth 0 -type d | grep -P '^proposals\\d{4}(-\\w+)+$$' | wc -l` ]; \
22 then \
23     echo "ERROR: 'proposals' must contain directories ONLY, and they must fall under regex '\\d{4}(-\\w+)+'"; \
24     exit 1; \
25 fi
26 @for d in `find proposals/* -maxdepth 0 -type d`; do \
27     if [ ! -f $$d/README.md ]; then \
28         echo "ERROR: '$$d' proposal doesn't contain a README.md. All proposals must have a README.md as a starting point."; \
29         exit 1; \
30     fi; \
31     # Conventionally (for now), any proposal's README.md must have 'Status' in the 5th line. \
32     STATUS=`sed -n '5p' "$$d/README.md" | cut -d'|' -f3 | xargs`; \
33     ALLOWED_STATUS_LIST="ACCEPTED DECLINED POSTPONED"; \
34     STATUS_ALLOWED=1; \
35     for s in $$ALLOWED_STATUS_LIST; do \
36         test "$$STATUS" = "$$s" && STATUS_ALLOWED=0; \
37     done; \
38     if [ $$STATUS_ALLOWED -ne 0 ]; then \
39         echo "ERROR: Current status of proposal '$$d' is '$$STATUS', but it must be one of '$$ALLOWED_STATUS_LIST' before you merge the PR!"; \
40         exit 1; \
41     fi; \
42 done
43 @echo "=> OK"
44
45 # Digest date can be set in env manually in various formats
46 DIGEST_DATE ?= $(shell date +%Y-%m-%d)
47 # Internal date representation used in generation
48 export DIGEST_DATE_GEN ?= $(shell date -d "$(DIGEST_DATE)" +%Y-%m-%d)
49 export DIGEST_DATE_HUMAN ?= $(shell date -d "$(DIGEST_DATE)" +%d.%m.%Y)
50 DIGEST_YEAR ?= $(word 1, $(subst -, ,$(DIGEST_DATE_GEN)))
51 DIGEST_FILE ?= digest-$(subst -, ,$(DIGEST_DATE_GEN)).md
52 digest: #@makehelp Create a skeleton of CYP Digest document for DIGEST_DATE date (default is today)
53 @test -d "/cyp/digest/$(DIGEST_YEAR)" || mkdir -p "/cyp/digest/$(DIGEST_YEAR)"
54 @test -f "/cyp/digest/$(DIGEST_YEAR)/$(DIGEST_FILE)" && \
55     >&2 echo "ERROR: Digest already exists: /cyp/digest/$(DIGEST_YEAR)/$(DIGEST_FILE)" \
56     && exit 1 || true
57 @cat /cyp/digest/digest-template.md | envsubst > "/cyp/digest/$(DIGEST_YEAR)/$(DIGEST_FILE)"
58 @echo "=> Digest created from template: cyp/digest/$(DIGEST_YEAR)/$(DIGEST_FILE)"
```



А в чем, собственно, проблема?



- Философское отступление: о природе времени
 - Линейное время
 - Спиральное время
 - Время в мультивселенной

А в чем, собственно, проблема?



- Наши продукты живут не в линейном времени
 - Похоже, они живут в мультивселенной
 - А мы живем в линейном
- Императивный программы (shell-скрипты, например) живут в линейном времени
 - То, что мы наблюдаем в нашем Makefile - это проекция мультивселенной на прямую
 - И конформное ли это отображение - мы не знаем. скорее всего - нет.

А в чем, собственно, проблема?



- Чего мы хотим:
 - Сервис должен собираться самым старым Go, какой только есть в продуктах, его использующих
 - Для генерации кода должны применяться самые новые совместимые инструменты
 - Эти инструменты должны быть собраны самым новым совместимым Go
 - Обновления файлов контроля зависимостей должны производиться самым старым Go, какой только есть в продуктах, использующих сервис
 - Должны использоваться самые новые инструменты контроля безопасности
 - Должны использоваться самые новые совместимые средства анализа кода
 - Локальное исполнение, без докера (!)

А в чем, собственно, проблема?

- Что такое “совместимый”:
 - Тот, на котором проходят тесты



Что же делать



- Все-таки написать Makefile
 - pros:
 - олдскульно
 - cons:
 - неподдерживаемо

Что же делать



- написать все на императивном языке: shell/go/python/etc
 - pros:
 - духоподъемно
 - cons:
 - сложна-сложна-сложна
 - могут быть проблемы с бюджетом
 - коллеги могут посмотреть косо

Что же делать



- Взять <https://taskfile.dev/>
 - pros:
 - декларативно!
 - это решение с нами давно, его многие успели попробовать
 - cons:
 - Установку утилит, до правильной версии и компиляцию правильных версий правильными версиями надо прописывать эксплицитно
 - Task is a task runner / build tool that aims to be simpler and easier to use than, for example, GNU Make.
 - Может быть трудно убедить коллег, что шило надо менять на мыло
 - Есть запутанные места в районе приоритетов переменных

Что же делать



- Взять <https://github.com/aquaproj/aqua>
 - pros:
 - декларативно!
 - хорошо работающий контроль версий утилит
 - cons:
 - это Version Manager для утилит, не система сборки

Что же делать



- Взять <https://mise.jdx.dev/>
 - pros:
 - декларативно!
 - Ровно то, что нам надо
 - Постоянно обновляется
 - cons:
 - Документация может забегать вперед
 - Есть баги
 - Постоянно обновляется
 - Надо договариваться с коллегами



**Отдельное спасибо Алексу Эфросу
за предоставленные результаты
личных исследований**

Спасибо



Вопросы?