Taskfile

Альтернатива Makefile, которая читается легче

Michael Savin | @jtprogru_channel

Bear on Server | 2023

Makfile

Makefile – общепринятый стандарт.

Утилита `make` есть почти на каждой системе.

Используется чтобы меньше писать букв – например:

```
docker compose -f docker-compose.local.yml up -d
# vs
make rundev
```

Makefile – Example

```
SHELL := /bin/bash
.SILENT:
.DEFAULT_GOAL := help

# Global vars
export SYS_GO=$(shell which go)
export SYS_GOFMT=$(shell which gofmt)
export SYS_DOCKER=$(shell which docker)

export BINARY_DIR=dist
export BINARY_NAME=app

include .env
export $(shell sed 's/=.*//' .env)

.PHONY: run.cmd
## Run as go run cmd/app/main.go
run.cmd: cmd/app/main.go
 $(SYS_GO) run cmd/app/main.go

.PHONY: run.bin
## Run as binary
run.bin: build.bin
 source .env && ./$(BINARY_DIR)/$(BINARY_NAME)

.PHONY: run.dc
## Run in Docker
run.dc:
 $(SYS_DOCKER) compose up -d --build
```

Makefile – Usage

Теперь можно спокойно писать в консоли:

```
# Запуск контейнеров с зависимостями
make run.dc
# Запуск через go run
make run.cmd
# Сборка бинарного исполняемого файла и последующий запуск
make run.bin
```

Makefile – Help

Первый запуск проекта обычно – ознакомление.

Вопрос: Как понять, какие команды тебе доступны в Makefile?

Ответ: "Открой в редакторе и почитай! Тычо?"

Makefile – Help

```
.PHONY: help
## Show this help message
help:
 @echo "$$(tput bold)Available rules:$$(tput sgr0)"
 @echo
 @sed -n -e "/^## / { \
  h; \
  s/.*//; \
  :doc" \
  -e "H; \
  n; \
  s/^## //; \
  t doc" \
  -e "s/:.*//; \
  G; \
  s/\\n## /---/; \
  s/\\n/ /g; \
  p; \
 }" ${MAKEFILE_LIST} \
 | LC_ALL='C' sort --ignore-case \
 | awk -F '---' \
  -v ncol=$$(tput cols) \
  -v indent=19 \
  -v col_on="$$(tput setaf 6)" \
  -v col_off="$$(tput sgr0)" \
 '{ \
  printf "%s%*s%s ", col_on, -indent, $$1, col_off; \
  n = split($$2, words, " "); \
  line_length = ncol - indent; \
  for (i = 1; i <= n; i++) { \
   line_length -= length(words[i]) + 1; \
   if (line_length <= 0) { \
    line_length = ncol - indent - length(words[i]) - 1; \
    printf "\n%*s ", -indent, " "; \
   } \
   printf "%s ", words[i]; \
  } \
  printf "\n"; \
 }' \
 | more $(shell test $(shell uname) == Darwin && echo '--no-init --raw-control-chars')
```

Taskfile

Документация доступна тут на официальном сайте утилиты taskfile.dev.

Установка максимально простая:

```
# Для пользователей Homebrew
brew install go-task/tap/go-task

# Ну или с помощью скриптов
sh -c "$(curl --location https://taskfile.dev/install.sh)" -- -d

# Ну в крайнем случае go install
go install github.com/go-task/task/v3/cmd/task@latest
```

Taskfile

```yaml
# yaml-language-server: $schema=https://taskfile.dev/schema.json
---
version: "3"

set:
  - pipefail

silent: false

tasks:
  prec:
    desc: Preconditions for project
    preconditions:
      - test -f $(which go)
      - test -f go.mod
      - test -f go.sum
      - test -d dist || mkdir dist

  runcmd:
    desc: Run like go run main.go
    deps:
      - prec
    cmds:
      - go run ./cmd/gonewszer/main.go
```

Taskfile – Help

```yaml
tasks:
  default:
    silent: true
    cmds:
      - task --list --color
```

ДА! Это все! И даже если это не добавить, тебе task подскажет сам!

Taskfile vs Makefile

Отличия в использовании:

- (**!**) Придется писать task, а не make;
- Удобно читать YAML;
- Ревью YAML гораздо приятнее и удобнее, чем Makefile;

Taskfile – Demo

На примере Terraform-проекта `production` можно посмотреть использование Taskfile.

СПАСИБО

- https://taskfile.dev/usage/ – документация по Taskfile API;
- Gist Taskfile.yml – мой пример Taskfile из реального проекта;
- Gist Makefile - мой пример Makefile из реального проекта;
- @jtprogru_channel – велкам в мою телегу;