


Programming Assignment: Утилита tree

You have not submitted. You must earn 1/1 points to pass.

 It looks like this is your first programming assignment. [Learn more](#)



Deadline Pass this assignment by January 6, 11:59 PM PST

Instructions My submission

Discussions

hw1_tree.zip

Во всех заданиях курса тесты являются частью задания, не пытайтесь понять задачу только по описанию, смотрите тесты

Утилита tree.

Выводит дерево каталогов и файлов (если указана опция -f).

Необходимо реализовать функцию `dirTree` внутри `main.go`. Начать можно с <https://golang.org/pkg/os/#Open> и дальше смотреть какие методы есть у результата.

Код писать в файле `main.go`

Запускать тесты через `go test -v` находясь в папке с заданием. После запуска вы должны увидеть такой результат:

```
1 $ go test -v
2 === RUN   TestTreeFull
3 --- PASS: TestTreeFull (0.00s)
4 === RUN   TestTreeDir
5 --- PASS: TestTreeDir (0.00s)
6 PASS
7 ok      coursera/homework/tree    0.127s
8
```

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

```

1 go run main.go . -f
2   └─main.go (1881b)
3   └─main_test.go (1318b)
4     └─testdata
5       └─project
6         └─file.txt (19b)
7         └─gopher.png (70372b)
8       └─static
9         └─css
10        └─body.css (28b)
11        └─html
12          └─index.html (57b)
13          └─js
14            └─site.js (10b)
15        └─zline
16          └─empty.txt (empty)
17          └─zzfile.txt (empty)
18

```

```

1 go run main.go .
2   └─testdata
3     └─project
4     └─static
5       └─css
6       └─html
7       └─js
8       └─zline
9

```

Замечания:

- Перенос строки - unix-style (\n)
- Отступы - символ графики + символ табуляции (\t)
- Для расчета символа графики в отступах подумайте про последний элемент и префикс предыдущих уровней. Там довольно простое условие. Хорошо помогает проговорить вслух то что вы видите на экране.
- Если вы пользуетесь windows - помните, что там и в linux разделители директорий различаются - используйте лучше `string(os.PathSeparator)`
- Рекурсивный алгоритм проще всего. Но можно реализовать и не-рекурсивно
- Вы можете реализовать любые нужные вам функции, вы не ограничены в единственной dirTree. Если вам нужно больше аргументов - создайте другую функцию и вызывайте её рекурсивно. dirTree в этом случае может быть только входной точкой.
- Символы графики лучше копируйте не из текста задания (который вы читаете сейчас), а из исходного кода теста (main_test.go)
- **Результаты (список папок-файлов) должны быть отсортированы по алфавиту. Т.е. у вас должен быть код который отсортирует уровень. Смотрите для этого пакет sort. Это самая частая причина непрохождения тестов. Тесты запускаются в среде linux. В задании есть докер-файл для тестов ровно в тех же условиях, он сразу выявит все проблемы.**

- У вас может быть соблазн использовать глобальные переменные, но вариант с рекурсией проще получается без них, а в не-рекурсивном варианте они вообще не нужны
- сигнатуру функции dirTree (количество параметров) менять нельзя, тесты на сервере не пройдут
- если вы столкнётесь с несовместимостью os.File и bytes.Buffer - смотрите видео "Написание тестов для программы уникализации", uniq/wint_tests в коде в уроку, а так же ссылку на хабр ниже
- На MacOS может быть проблема с системным файлом ``.DS_Store`` - его можно просто игнорировать в самой программе, или же настроить `dockerignore`:

```
1  **/*.DS_Store
2  **/.git
```

Материалы в помощь:

- <https://habrahabr.ru/post/306914/> - пакет io
- <https://golang.org/pkg/sort/>
- <https://golang.org/pkg/io/>
- <https://golang.org/pkg/io/ioutil/>

