Test instruction

Первым делом нужно создать еще один файл. К примеру, у нас есть основной файл с кодом, который мы хотим протестировать: program.go. Нам нужно создать еще один файл, к примеру: program\_test.go.

Дальше в основном файле program.go и в файле program\_test.go нам нужно изменить пакет с:



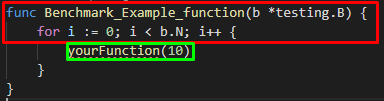
На это(можете написать любое имя пакета):



Далее в файле program\_test.go нужно импортировать “testing”:



Далее нужно добавить функцию вида:

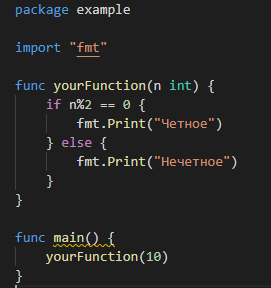
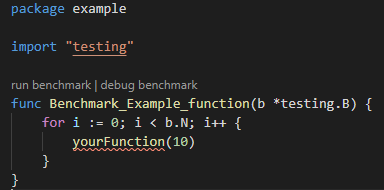


Где красным цветом написана функция для тестирования производительности. Зеленым цветом написана функция из основного файла, которую нужно протестировать. Функция бэнчмарка обязательно должна начинаться со слова Benchmark.

Цикл for i := 0; i < b.N; i++ {} будет запускать функцию yourFunction() столько раз, пока результат не будет стремится к среднему значению.

В общем виде два файла будут выглядеть так:

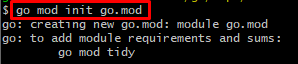
program.go program\_test.go

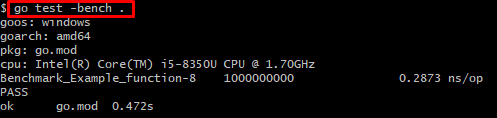
Далее в консоли переходим в директорию, в которой находится наш проект:



Далее делаем инициализацию:

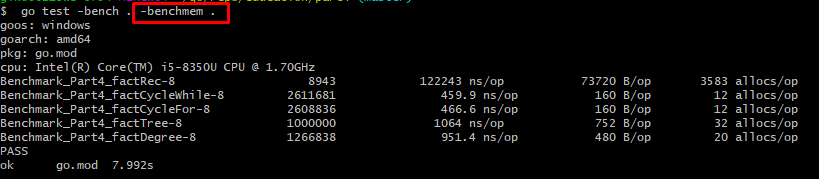


Выполняем следующую команду, и видим результат ее работы:



Каждый бэнчмарк выдает три параметра, первый(Benchmark\_Example\_function-8) – имя функции, которое мы задали в тестовом файле, второй(1000000000) – количество выполнений функции, в результате которых получается среднее время выполнения программы и третий(0.2873ns/op) – это число наносекунд, затраченное на вызов функции.

Также можно измерять количество выделяемой памяти с помощью следующей команды (рассмотрим на примере более сложной программы, для наглядности):



Четвертый столбец(B/op) – сколько выделено байт за операцию, пятый столбец(allocs/op) – сколько раз за операцию выделялась память.