



Урок Спринт 2.9

Пакет worker pool

В предыдущем примере мы ограничили количество горутин в коде функции `main`. Ранее, в процессе обучения, мы научились создавать пакеты, попробуем применить это и здесь — создадим пакет, который позволит нам более удобно создавать пул обработчиков и использовать в разных проектах. Создадим отдельный пакет `work`:

```
package work

// Интерфейс надо реализовать объектам, которые будут обрабатываться параллельно
type Worker interface {
    Task()
}

// Пул для выполнения
type Pool struct{
    // из этого канала будем брать задачи для обработки
    tasks chan Worker
    // для синхронизации работы
    wg sync.WaitGroup
}
```

Мы создали интерфейс `Worker` для того, чтобы любая структура, содержащая функцию `Task()` могла использовать наш `worker pool`. Создадим экземпляр пула:

```
// при создании пула передадим максимальное количество горутин
func New(maxGoroutines int) *Pool {
    p := Pool{
        tasks: make(chan Worker), // канал, откуда брать задачи
    }
    // для ожидания завершения
    p.wg.Add(maxGoroutines)
    for i := 0; i < maxGoroutines; i++ {
        // создадим горутину по указанному количеству maxGoroutines
        go func() {
            // забираем задачи из канала
            for w := range p.tasks {
                // и выполняем
                w.Task()
            }
        }()
    }
}
```

```
    }  
    // после закрытия канала нужно оповестить наш пул  
    p.wg.Done()  
  }()  
}  
  
return &p  
}
```

В коде выше мы создаем экземпляр пула, в котором запускаем горутин. Каждая горутина ожидает задачи из канала `tasks` и выполняет функцию `Task()`. Реализация пула готова, но мы не создали код для передачи задач в наш пул:

```
// Передаем объект, который реализует интерфейс Worker  
func (p *Pool) Run(w Worker){  
    // добавляем задачи в канал, из которого забирает работу пул  
    p.tasks <- w  
}
```

Также не забудем про корректное завершение:

```
func (p *Pool) Shutdown(){  
    // закроем канал с задачами  
    close(p.tasks)  
    // дождемся завершения работы уже запущенных задач  
    p.wg.Wait()  
}
```

Если мы хотим использовать созданный пакет, то необходимо реализовать интерфейс `Worker` и передать объект в функцию `Run`. Попробуем это сделать в задачах к уроку.

Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»