



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ОТЧЕТ

по лабораторной работе № 5

Название: Основы асинхронного программирования на Golang

Дисциплина: Языки Интернет-программирования

Студент

ИУ6И-31Б
(Группа)

(Подпись, дата)

С. Гун

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Москва, 2024

Цель работы:

Изучение основ асинхронного программирования с использованием языка Golang.

Ход работы:

1. Ознакомились с разделом "3. Map, файлы, интерфейсы, многопоточность и многое другое" курса <https://stepik.org/course/54403/info>
2. Сделали форк данного репозитория в GitHub, клонировали получившуюся копию локально, создали от мастера ветку dev и переключились на неё
3. Выполнили задания:

а. Задание 1. calculator

```
package main

import "fmt"

func main() {

    chan1, chan2 := make(chan int), make(chan int)
    stop := make(chan struct{})
    r := calculator(chan1, chan2, stop)
    go func() {
        chan1 <- 3
        chan2 <- 3
        close(stop)
    }()
    fmt.Println(<-r)
}

func calculator(firstChan <-chan int, secondChan <-chan int, stopChan
<-chan struct{}) <-chan int {
    res := make(chan int)
    go func() {

        select {
        case num := <-firstChan:
            res <- num * num
        case num := <-secondChan:
            res <- num * 3
        case _ = <-stopChan:
        }
        close(res)
    }
```

```

    }()
    return res
}

```

b. Задание 2. pipeline

```

package main

import (
    "fmt"
)

func main() {

    inputStream := make(chan string)
    outputStream := make(chan string)
    go removeDuplicates(inputStream, outputStream)

    go func() {
        inputStream <- "a"
        inputStream <- "a"
        inputStream <- "b"
        inputStream <- "b"
        inputStream <- "c"
        close(inputStream)
    }()

    for x := range outputStream {
        fmt.Print(x)
    }
    fmt.Print("\n")
}

func removeDuplicates(inputStream chan string, outputStream chan string)
{
    prev_str := ""
    cur_str := ""
    for value := range inputStream {
        prev_str = cur_str
        cur_str = value
        if cur_str != prev_str {
            outputStream <- cur_str
        }
    }
    close(outputStream)
}

```

```
}
```

с. Задание 3. work

```
package main

import (
    "fmt"
    "sync"
    "time"
)

func work() {
    time.Sleep(time.Millisecond * 50)
    fmt.Println("done")
}

func main() {
    wg := new(sync.WaitGroup)

    for i := 0; i < 10; i++ {
        wg.Add(1)
        go func(wg *sync.WaitGroup) {
            defer wg.Done()
            work()
        }(wg)
    }
    wg.Wait()
}
```

4. Зафиксировали изменения, сделали коммит и отправили полученное состояние ветки dev в удаленный репозиторий GitHub
5. Через интерфейс GitHub создали Pull Request dev --> master