



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 _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 6₂ **по курсу «Компьютерные сети»**

«Разработка SMTP-клиента и приложения почтовой рассылки»

Студент группы ИУ9-31Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Задача 1: SMTP-клиент на Golang. Необходимо реализовать программу отправки проверочного SMTP сообщения, которое необходимо производить на ящик `danila@bmstu.posevin.ru` с одного из ящиков, доступ к которым приведен ниже. При этом работоспособность приложения необходимо продемонстрировать очно. В этом приложении должны быть реализованы следующие функции:

- ввод значения поля `To` из командной строки;
- ввод значения поля `Subject` из командной строки;
- ввод сообщения в поле `Message body` из командной строки.

Задача 2: Приложение почтовых рассылок.

1. В базе данных MySQL создать таблицу рассылки, в которой должны быть следующие поля: имя пользователя, адрес электронной почты пользователя, сообщение пользователю.

2. Реализовать рассылку по таблице рассылки, при этом в тексте письма должно быть реализовано персональное обращение к пользователю, текст письма оформлен в HTML формате, при этом как минимум приветствие должно быть выделено жирным шрифтом, текст письма курсивом и фон письма отличаться от белого, рекомендуется прочитать статьи приведенные ниже о верстке электронных писем для рассылок.

3. Необходимо реализовать два варианта рассылок: массовый спам, путем использования горутин и последовательную рассылку со случайным значением интервала задержки писем между отправками. При тестировании использовать второй способ, чтобы почтовый ящик не был забанен почтовым сервером.

2 Результаты

Исходный код программы представлен в листинге 1, 2

Листинг 1 — SMTP-client.go

```

1 package main
2 import (
3     "crypto/tls"
4     "database/sql"
5     "fmt"
6     "log"
7     "math/rand"
8     "net/smtp"
9     "time"
10
11
12     _ "github.com/go-sql-driver/mysql"
13 )
14 type EmailConfiguration struct {
15     Username string
16     Password string
17     Host      string
18     Port      string
19 }
20 type EmailData struct {
21     To      string
22     Subject string
23     Body    string
24 }
25 var emailConfig = EmailConfiguration{
26     Username: "dts21@dactyl.su",
27     Password: "12345678990DactylSUDTS",
28     Host:     "mail.nic.ru",
29     Port:     "465",
30 }
31 func main() {
32     createTableQuery := `
33     CREATE TABLE IF NOT EXISTS lab7Gorbunov (
34         id INT AUTO_INCREMENT PRIMARY KEY,
35         username TEXT NOT NULL,
36         email TEXT NOT NULL,
37         message TEXT NOT NULL
38     );`
39     db, err := sql.Open("mysql", "iu9networkslabs:Je2dTYr6@tcp(students.
yys.su)/iu9networkslabs")
40     if err != nil { log.Fatal(err) }
41     defer db.Close()
42     _, err = db.Exec(createTableQuery)
43     if err != nil { log.Fatal(err) }
44     for i := 0; i < 10; i++{
45         rows, err := db.Query("SELECT username, email, message FROM
lab7Gorbunov")
46         if err != nil { log.Fatal(err) }
47         defer rows.Close()
48         for rows.Next() {
49             var username, email, message string
50             err := rows.Scan(&username, &email, &message)
51             if err != nil { log.Fatal(err) }
52             emailData := generateEmailData(username, email, message)
53             sendEmail(emailData)
54             time.Sleep(time.Duration(randomDelay()) * time.Second)
55         }
56     }
57 }

```

Листинг 2 — SMTP-client.go(продолжение)

```
1 func sendEmail(emailData EmailData) {
2     tlsConfig := &tls.Config{
3         InsecureSkipVerify: true ,
4         ServerName:         emailConfig.Host ,
5     }
6
7     conn, err := tls.Dial("tcp", emailConfig.Host+": "+emailConfig.Port ,
8         tlsConfig)
9     if err != nil {
10         log.Fatal(err)
11     }
12     defer conn.Close()
13
14     auth := smtp.PlainAuth("", emailConfig.Username, emailConfig.
15         Password, emailConfig.Host)
16
17     client, err := smtp.NewClient(conn, emailConfig.Host)
18     if err != nil {
19         log.Fatal(err)
20     }
21     defer client.Close()
22
23     if err = client.Auth(auth); err != nil {
24         log.Fatal(err)
25     }
26
27     if err = client.Mail(emailConfig.Username); err != nil {
28         log.Fatal(err)
29     }
30
31     if err = client.Rcpt(emailData.To); err != nil {
32         log.Fatal(err)
33     }
34
35     dataWriter, err := client.Data()
36     if err != nil {
37         log.Fatal(err)
38     }
39
40     _, err = dataWriter.Write([] byte(
41         "To: " + emailData.To + "\r\n" +
42         "Subject: " + emailData.Subject + "\r\n" +
43         "Content-Type: text/html; charset=UTF-8\r\n\r\n" +
44         emailData.Body))
45     if err != nil {
46         log.Fatal(err)
47     }
48
49     err = dataWriter.Close()
50     if err != nil {
51         log.Fatal(err)
52     }
53 }
54
55 func randomDelay() int {
56     return rand.Intn(10) + 1
57 }
```

Результат запуска представлен на рисунке 1, 2

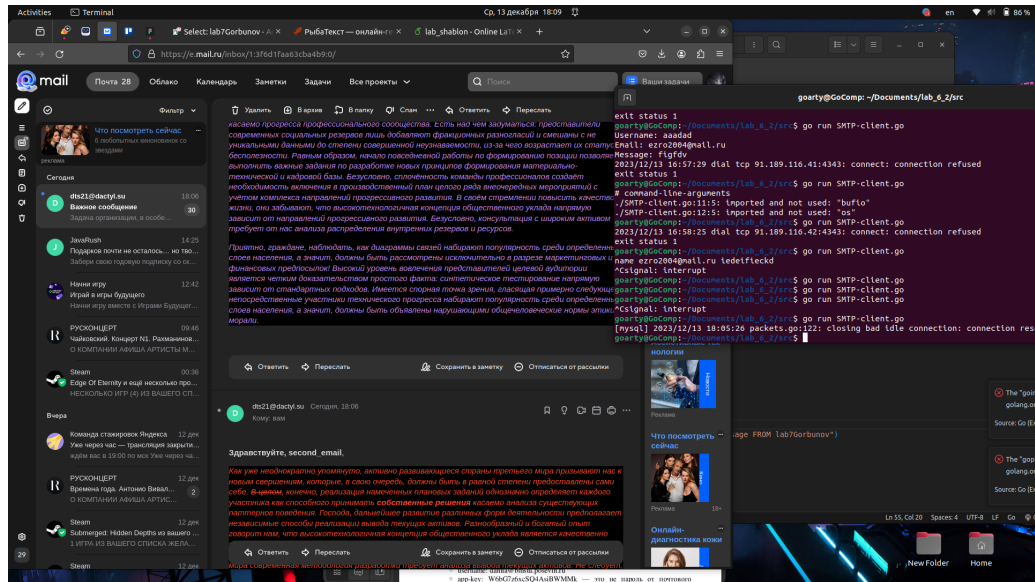


Рис. 1 — Сообщения на почте

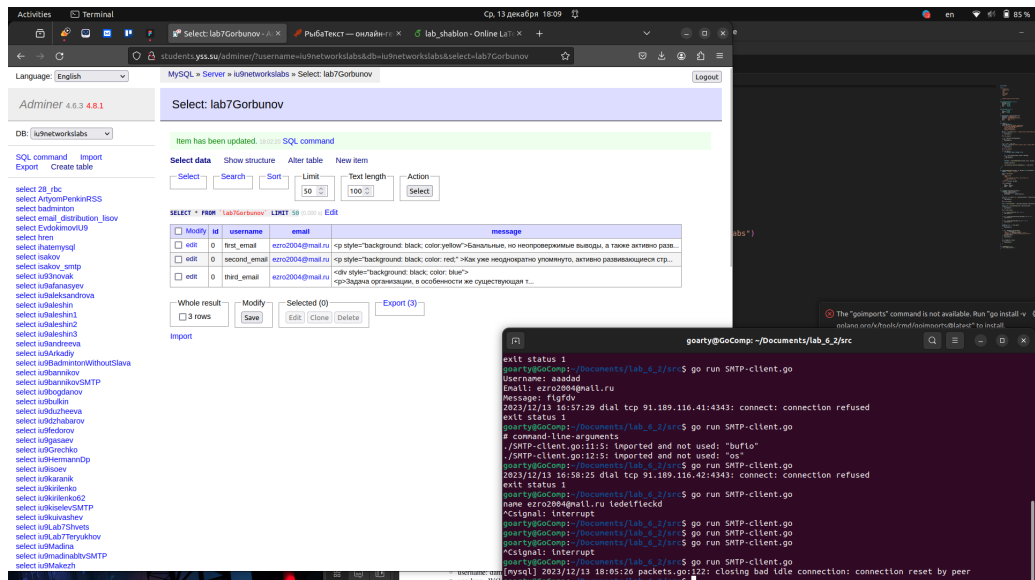


Рис. 2 — Таблица email-ов для рассылки