МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра вычислительной техники



**Лабораторная работа №2**

по дисциплине: «WEB - ПРОГРАММИРОВАНИЕ»

на тему: **«Работа с файлами теневых посылок (cookies) и текстовыми файлами»**

Вариант №6

Выполнил: Проверил:

Студент гр. АВТ-219: Гунько А. В.

Завёрткин М. А.

Новосибирск

2025

СОДЕРЖАНИЕ

Введение 3

Ход работы 3

Вывод 6

# **Задание**

**Вариант 6.** Набор флажков для выбора допустимых режимов работы (чтение, запись, изменение, удаление) и кнопка отправки данных.

**Сценарии на GO**

Задача была сделана в виде одного веб-сервера на go и одной страницы. Отправка формы одновременно вызывает запись в cookies и запись в лог, который можно просмотрить отдельной кнопкой.

package main

import (

"fmt"

"io"

"log"

"net/http"

"os"

"time"

)

var file \*os.File

const post\_log = "post.log"

const form\_html = `

<form method="POST" action="">

<p><input type="checkbox" name="read">Read</p>

<p><input type="checkbox" name="write">Write</p>

<p><input type="checkbox" name="edit">Edit</p>

<p><input type="checkbox" name="delete">Delete</p>

<p><input type="SUBMIT"></p></form>

<a href="/log">View logs</a>

`

func handler(w http.ResponseWriter, r \*http.Request) {

cookie, err := r.Cookie("cookie")

cookie\_text := ""

if err == nil {

cookie\_text = cookie.Value

}

if r.Method == http.MethodPost {

form\_values := fmt.Sprintf("Read:%s, Write:%s, Edit:%s, Delete:%s",

r.FormValue("read"),

r.FormValue("write"),

r.FormValue("edit"),

r.FormValue("delete"))

cookie := &http.Cookie{

Name: "cookie",

Value: form\_values,

}

fmt.Fprintf(file, "%s:\t%s\n", time.Now().Format(time.UnixDate), form\_values)

http.SetCookie(w, cookie)

http.Redirect(w, r, "/index", http.StatusSeeOther)

}

fmt.Fprintf(w, "<h1>Just a form, bro</h1><p>%s</p><div>%s</div>", cookie\_text, form\_html)

}

func handler\_log(w http.ResponseWriter, r \*http.Request) {

file, err := os.Open(post\_log)

if err != nil {

log.Fatal(err)

}

defer file.Close()

\_, \_ = io.Copy(w, file)

}

func main() {

var err error

file, err = os.OpenFile(post\_log, os.O\_APPEND|os.O\_CREATE|os.O\_RDWR, 0644)

if err != nil {

log.Fatal(err)

return

}

defer file.Close()

http.HandleFunc("/index", handler)

http.HandleFunc("/log", handler\_log)

fmt.Println("Start listening at http://localhost:8080/index")

log.Fatal(http.ListenAndServe(":8080", nil))

}