МИНИСТРЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОУ НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМ. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ ЭКОНОМИКИ И УПРАВЛЕНИЯ

КАФЕДРА " Прикладная математика и информатика"

Дисциплина "Структуры данных"

Отчет

по лабораторной работе № 2

Выполнил: Иванов Михаил Егорович

студент группы: 21 САИ

Проверил: Моисеев Антон Евгеньевич

Нижний Новгород

2023

**Цель работы:** реализоватьпереполнение стека рекурсией и достаточно большим массивом.

**Практическая часть.**

Переполнение стека происходит, когда объем данных, размещаемых на стеке, превышает его размер. Стек - это область памяти, в которой хранятся локальные переменные и параметры функций, а также адреса возврата из функций. Рекурсия - это вызов функции самой себя, при котором на стеке создается новый фрейм для каждого вызова.

Для реализации переполнения стека рекурсией была создана функция Fib, которая вычисляет n-ое число Фибоначчи по рекуррентной формуле: F(n) = F(n-1) + F(n-2). Функция Fib вызывается с отрицательным параметром -1, что приводит к бесконечной рекурсии, так как условие остановки (n == 1) никогда не выполнится. Таким образом, при каждом вызове функции Fib на стеке размещается новая переменная n и адрес возврата, что приводит к быстрому исчерпанию памяти.

Для реализации переполнения стека достаточно большим массивом была создана функция overflow, которая вызывает саму себя рекурсивно и создает массивы с увеличивающимся количеством элементов. Функция overflow вызывает другую функцию createArray, которая создает массив из n элементов и заполняет его случайными числами. Таким образом, при каждом вызове функции overflow на стеке размещается новый массив, который занимает много места в памяти.

**Код:  
  
Рекурсия:**

package main  
  
func Fib(n int64) int64 {  
 if n == 1 {  
 return 1  
 }  
 return Fib(n-1) + Fib(n-2)  
  
}  
  
func main() {  
 // Вызываем функцию overflow с начальным значением n равным 1  
 Fib(-1)  
}

### Массив:

package main  
  
import "fmt"  
  
// Создаем функцию createArray, которая будет создавать массив из n элементов и возвращать его  
func createArray(n int) []int {  
 // Создаем пустой массив с заданной длиной  
 array := make([]int, n)  
 // Заполняем массив случайными числами от 0 до 9  
 for i := 0; i < n; i++ {  
 array[i] = i % 10  
 }  
 // Возвращаем массив  
 return array  
}  
  
// Создаем функцию overflow, которая будет вызывать сама себя рекурсивно и создавать большие массивы, пока не произойдет переполнение стека  
func overflow(n int) {  
 // Создаем массив из n элементов и выводим его длину на экран  
 array := createArray(n)  
 fmt.Println("Создан массив из", len(array), "элементов")  
 // Вызываем функцию overflow снова с увеличенным на единицу значением n  
 overflow(n + 1)  
}  
  
func main() {  
 // Вызываем функцию overflow с начальным значением n равным 1  
 overflow(1)  
}

### Вывод: