

**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Розрахунково-графічна робота**

з дисципліни  
«Архітектура програмного забезпечення»  
на тему  
«Архітектурні діаграми та бенчмарки»

Виконали:

студенти групи ІП-95  
Білокоренко Артем Андрійович  
Владіміров Артем Олексійович

Перевірів:

Мазур Р. Ф.

## Завдання:

1. Для 2-гої роботи, підтвердьте лінійний час виконання вашої функції перетворення чи обчислення вхідного виразу:

## Сирцевий код

---

```
package lab2

import (
    "fmt"
    "math"
    "math/rand"
    "testing"
    "unsafe"
)

type test struct {
    elements int
    expression string
}

var tests []test

func pickOperation() string {
    var chance = rand.Float64()
    if chance < 0.25 {
        return "+"
    } else if chance < 0.5 {
        return "-"
    } else if chance < 0.75 {
        return "*"
    } else {
        return "/"
    }
}

func pickNumber() string {
    var value = rand.Intn(9) + 1
    return fmt.Sprintf("%v", value)
}

func Clone(s string) string {
    c := make([]byte, len(s))
    copy(c, s)
    return *(*string)(unsafe.Pointer(&c))
}
```

```
}
```

```
var UpperBound = int(math.Pow(2.0, 18.0))
```

```
func generateTests() {  
    var prefixExpression string = "+ 2 2"  
    for i := 1; i <= UpperBound; i *= 2 {  
        for j := i; j < i*2 && j <= UpperBound; j++ {  
            var num = pickNumber()  
            var op = pickOperation()  
            prefixExpression = fmt.Sprintf("%s %s ", op, num) + prefixExpression  
        }  
        var newTest = test{  
            elements: len(prefixExpression),  
            expression: Clone(prefixExpression),  
        }  
        tests = append(tests, newTest)  
    }  
}
```

```
var result string
```

```
func BenchmarkPrefixEvaluation(b *testing.B) {  
    generateTests()  
    fmt.Println(UpperBound)  
    for _, item := range tests {  
        b.Run(fmt.Sprintf("%d", item.elements), func(b *testing.B) {  
            for i := 0; i < b.N; i++ {  
                result, _ = PrefixEvaluation(item.expression)  
            }  
        })  
    }  
}
```

---

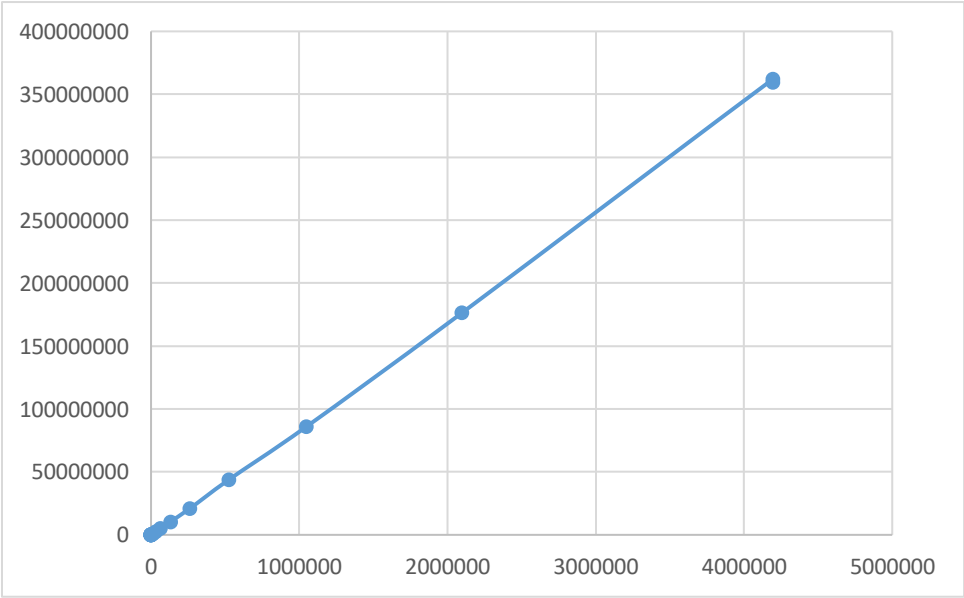
# Логн

OK: 14 passed  
1048576  
goos: windows  
goarch: amd64  
pkg: github.com/vlad1m1r0v/APZ-2  
cpu: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz

BenchmarkPrefixEvaluation/9	90956	12428 ns/op
BenchmarkPrefixEvaluation/17	93715	12894 ns/op
BenchmarkPrefixEvaluation/33	82545	14334 ns/op
BenchmarkPrefixEvaluation/65	72039	16950 ns/op
BenchmarkPrefixEvaluation/129	55436	21827 ns/op
BenchmarkPrefixEvaluation/257	37796	31625 ns/op
BenchmarkPrefixEvaluation/513	23318	51400 ns/op
BenchmarkPrefixEvaluation/1025	12944	91283 ns/op
BenchmarkPrefixEvaluation/2049	6732	172465 ns/op
BenchmarkPrefixEvaluation/4097	3633	329990 ns/op
BenchmarkPrefixEvaluation/8193	1798	652411 ns/op
BenchmarkPrefixEvaluation/16385	937	1286537 ns/op
BenchmarkPrefixEvaluation/32769	465	2563281 ns/op
BenchmarkPrefixEvaluation/65537	234	5164339 ns/op
BenchmarkPrefixEvaluation/131073	100	10519685 ns/op
BenchmarkPrefixEvaluation/262145	60	20951755 ns/op
BenchmarkPrefixEvaluation/524289	27	43781881 ns/op
BenchmarkPrefixEvaluation/1048577	14	86003379 ns/op
BenchmarkPrefixEvaluation/2097153	6	176335350 ns/op
BenchmarkPrefixEvaluation/4194305	3	362002500 ns/op
BenchmarkPrefixEvaluation/4194309	3	359522200 ns/op

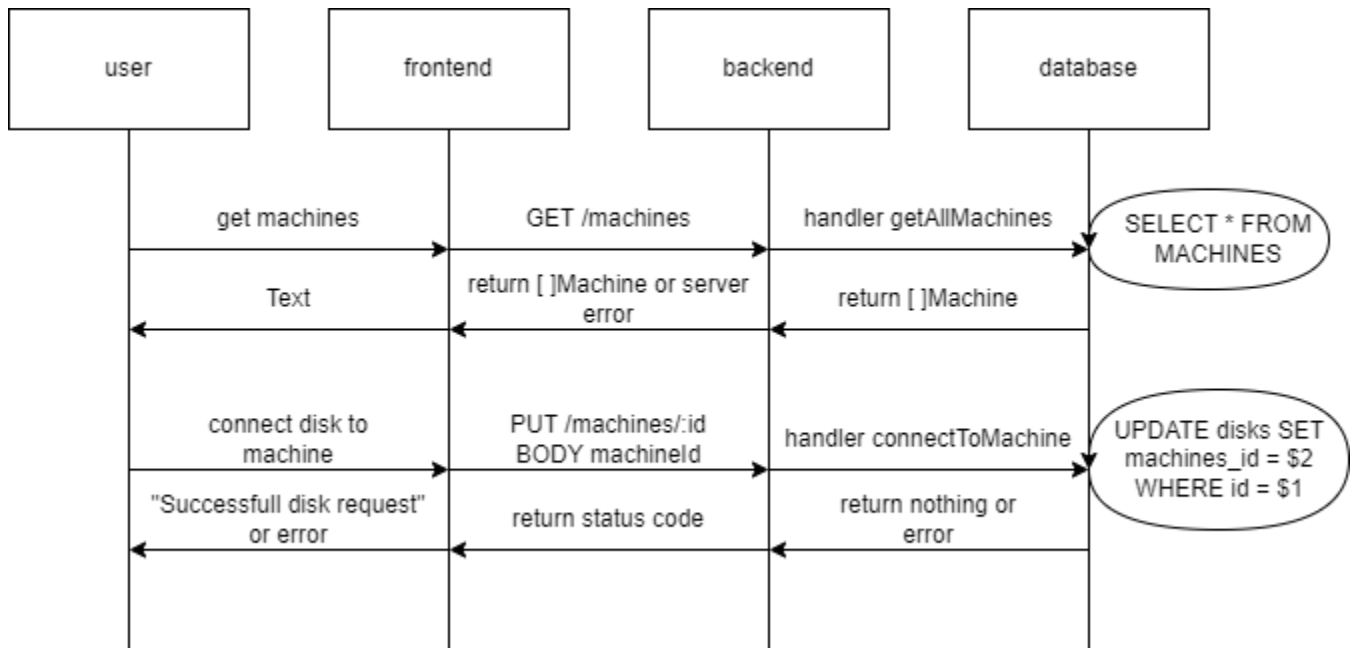
PASS  
ok github.com/vlad1m1r0v/APZ-2 348.818s

# Графік



2. Для 3-тої роботи, побудуйте діаграму взаємодії компонентів у вашій імplementації:

### Діаграма



3. Для 4-ої роботи, побудуйте діаграму взаємодії для вашої реалізації (на ній, скоріш за все, мають опинитися компоненти парсера, черги команд, ядра цикла) та підтвердьте лінійний час роботи вашого парсера команд.

### Сирцевий код

```

package main

import (
    "fmt"
    "io/ioutil"
    "math"
    "strings"
    "testing"
)

func ExtendSplitCommand(times int) {
    for i := 0; i < times; i++ {
        input, _ := ioutil.ReadFile("./example/example-test.txt")
        parts := strings.Fields(string(input))
        cmd := parts[0]
        params := parts[1]
        delimiter := parts[2]
        params = fmt.Sprintf("%s-%s", params, params)
    }
}
  
```

```
newCommand := fmt.Sprintf("%s %s %s", cmd, params, delimiter)
ioutil.WriteFile("./example/example-test.txt", []byte(newCommand), 0644)
}
}

func SetupFile() {
    ioutil.WriteFile("./example/example-test.txt", []byte("split uno-dos-tres-cuatro-cinco -"), 0644)
}

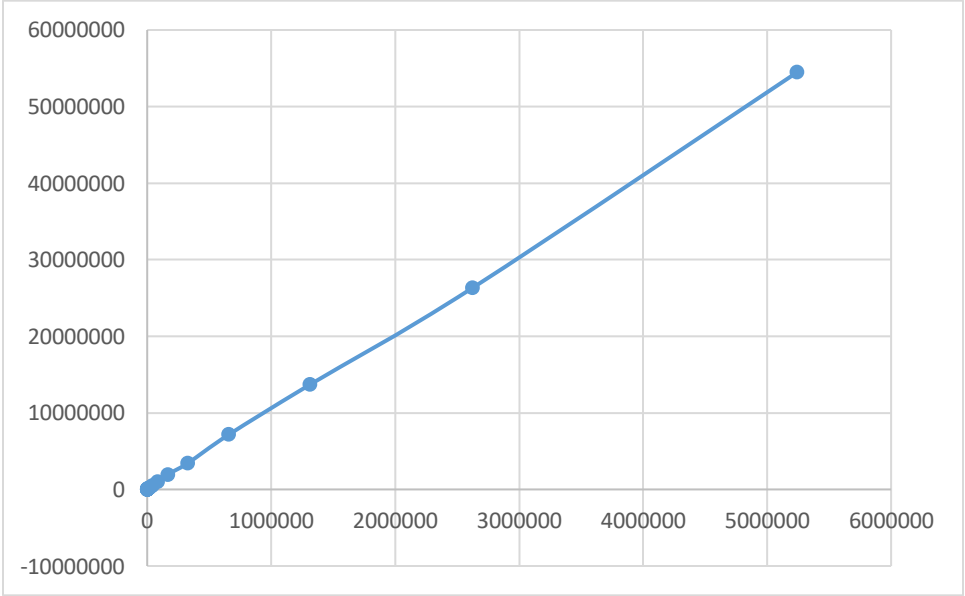
func BenchmarkParser(b *testing.B) {
    for i := 1; i <= 20; i++ {
        SetupFile()
        ExtendSplitCommand(i)
        command_ := ioutil.ReadFile("./example/example-test.txt")
        b.Run(fmt.Sprintf("%d elements", 5 * int(math.Pow(2, float64(i)))), func(b *testing.B) {
            for i := 0; i < b.N; i++ {
                parse(string(command))
            }
        })
    }
}
```

---

## Логи

```
goos: windows
goarch: amd64
pkg: github.com/vlad1m1r0v/APZ-4/cmd
cpu: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
BenchmarkParser/10_elements      3459094      361.7 ns/op
BenchmarkParser/20_elements      2704849      448.9 ns/op
BenchmarkParser/40_elements      1590595      643.5 ns/op
BenchmarkParser/80_elements      1080207      1058 ns/op
BenchmarkParser/160_elements     563618       2100 ns/op
BenchmarkParser/320_elements     342333       3416 ns/op
BenchmarkParser/640_elements     182833       6803 ns/op
BenchmarkParser/1280_elements    87210        13891 ns/op
BenchmarkParser/2560_elements    39362        26371 ns/op
BenchmarkParser/5120_elements    21478        51459 ns/op
BenchmarkParser/10240_elements   9598         122265 ns/op
BenchmarkParser/20480_elements   5266         232082 ns/op
BenchmarkParser/40960_elements   2666         442412 ns/op
BenchmarkParser/81920_elements   1212         990677 ns/op
BenchmarkParser/163840_elements   530         1939211 ns/op
BenchmarkParser/327680_elements   296         3440148 ns/op
BenchmarkParser/655360_elements   154         7127008 ns/op
BenchmarkParser/1310720_elements   85         13673016 ns/op
BenchmarkParser/2621440_elements   40         26292640 ns/op
BenchmarkParser/5242880_elements   20         54492325 ns/op
PASS
ok      github.com/vlad1m1r0v/APZ-4/cmd 35.267s
```

## Графік



## Діаграма

