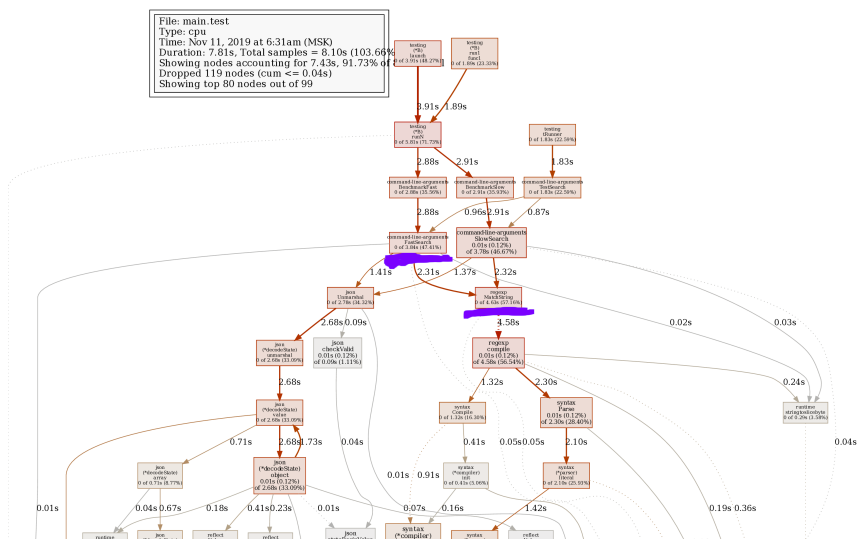


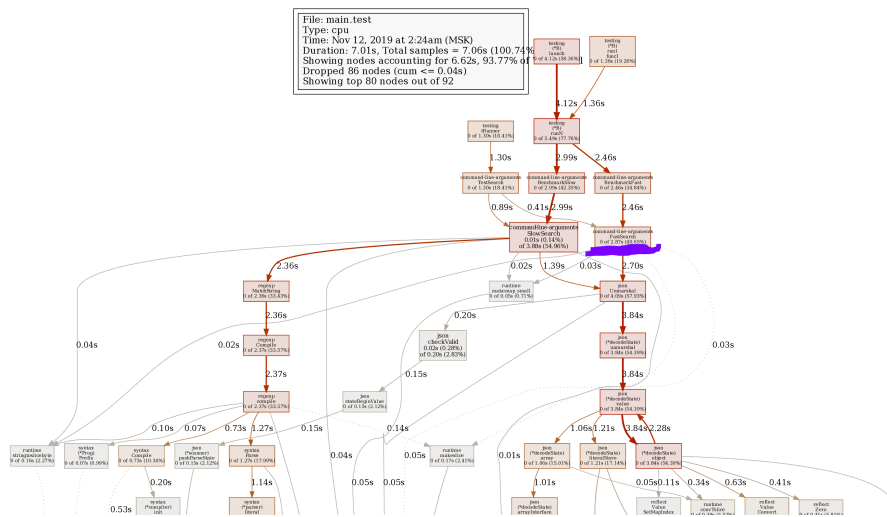
ДЗ №5. Отчет.

18 ноября 2019 г.

1. В самом начале имеем:



Заменим `regexp.MatchString()` на `strings.Contains()`, получим:



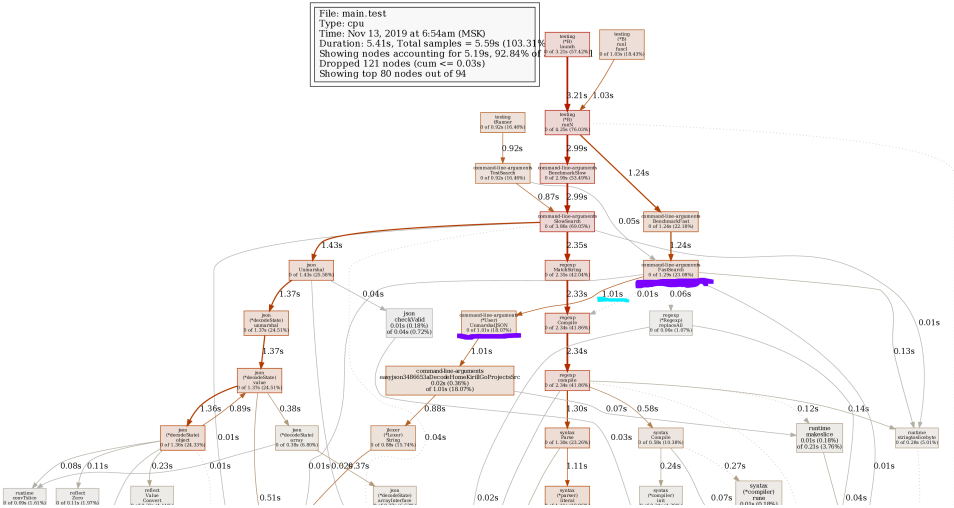
Получим следующие результаты.
Было:

```
optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      44834427 ns/op      18856268 B/op      195817 allocs/op
BenchmarkFast-4      27      43223744 ns/op      18867371 B/op      195821 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization      2.649s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      42818948 ns/op      18845349 B/op      195817 allocs/op
BenchmarkFast-4      27      42483538 ns/op      18871565 B/op      195821 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization      2.566s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      4337583 ns/op      18847256 B/op      195817 allocs/op
BenchmarkFast-4      27      43581958 ns/op      18883938 B/op      195823 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization      2.649s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

Стало:

```
optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      24      43183574 ns/op      18855581 B/op      195816 allocs/op
BenchmarkFast-4      81      15337257 ns/op      5422887 B/op      59763 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization      2.543s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      26      42028131 ns/op      18873155 B/op      195819 allocs/op
BenchmarkFast-4      88      15218797 ns/op      5422812 B/op      59763 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization      2.498s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      38      42776737 ns/op      18868665 B/op      195821 allocs/op
BenchmarkFast-4      81      15074818 ns/op      5422817 B/op      59763 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization      2.683s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

2. Создадим структуру `User{...}`, воспользуемся библиотекой `easyjson` и сгенерируем код:



Результаты тестов.
Было:

```
optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      24      43183574 ns/op      18855581 B/op      195816 allocs/op
BenchmarkFast-4     81     15337257 ns/op       5422887 B/op       59763 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.543s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      26     42828131 ns/op      18873155 B/op      195819 allocs/op
BenchmarkFast-4     88     15228797 ns/op       5422818 B/op       59763 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.498s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      38     42776737 ns/op      18868665 B/op      195821 allocs/op
BenchmarkFast-4     81     15074818 ns/op       5422917 B/op       59763 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.683s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

Стало:

```
optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28     42862462 ns/op      18871874 B/op      195828 allocs/op
BenchmarkFast-4    244     4834879 ns/op       4366997 B/op      11846 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.819s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27     42239588 ns/op      18853514 B/op      195816 allocs/op
BenchmarkFast-4    244     4762024 ns/op       4361333 B/op      11845 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.948s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28     42361942 ns/op      18857928 B/op      195819 allocs/op
BenchmarkFast-4    244     4983589 ns/op       4364855 B/op      11846 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.817s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

3. Теперь посмотрим на выделение памяти.

```
optimization: go — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-573G:~/git/golang-2019-2/5/99_hw/optimization$ go tool pprof -alloc_space main.test mem.out
File: main.test
Type: alloc_space
Time: Nov 13, 2019 at 7:48am (MSK)
Entering interactive mode (type "help" for commands, "o" for options)
(pprof) top
Showing nodes accounting for 174.49MB, 92.24% of 189.17MB total
Dropped 154 nodes (cum <= 0.95MB)
Showing top 10 nodes out of 69
      flat flat% sum% cum cum%
  63.61MB 33.62% 33.62%  63.61MB 33.62% bytes.makeSlice
  42.57MB 22.50% 56.13% 114.42MB 66.48% command-line-arguments.FastSearch
  26.33MB 14.98% 71.10%  26.33MB 14.98% regexp.syntax.(*compiler).inst
  12.93MB  6.84% 77.94%  12.93MB  6.84% github.com/mailru/easyjson/lexer.(*Lexer).String
  18.26MB  5.42% 83.30%  18.26MB  5.42% regexp.syntax.(*parser).newRegexp
  5.49MB  2.90% 86.20%  21.87MB 12.39% command-line-arguments.SlowSearch
  5.12MB  2.71% 88.97%  26.99MB 14.80% regexp.compile
  2.93MB  1.55% 90.52%  24.96MB 13.34% regexp.syntax.Parse
  1.45MB  0.77% 91.30%  26.41MB 14.49% command-line-arguments.easyjson3486653aDecodeHomeKirillGoProjectsSrc
  1.68MB  0.89% 92.24%  28.09MB 15.18% encoding/json.unquote
(pprof) list FastSearch
Total: 189.17MB
ROUTINE ===== command-line-arguments.FastSearch in /home/kirill/git/golang-2019-2/5/99_hw/optimization/fast.go
  42.57MB 114.42MB (flat, cum) 66.48% of Total
    -    -    13: return
    -    -    14: }
    -    -    15:
    -    -    16: // вам надо написать более быструю оптимальную эту функцию
    -    -    17: func FastSearch(out io.Writer) {
    -  3.73kB 18:   file, err := os.Open(filePath)
    -    -    19:   if err != nil {
    -    -    20:     panic(err)
    -    -    21:   }
    -    -    22:
    -  55.57MB 23:   fileContents, err := ioutil.ReadAll(file)
    -    -    24:   if err != nil {
    -    -    25:     panic(err)
    -    -    26:   }
    -    -    27:
    -  19.62kB 28:   r := regexp.MustCompile("o")
    -    -    29:   seenBrowsers := []string{}
    -    -    30:   uniqueBrowsers := 0
    -    -    31:   foundUsers := ""
    -    -    32:
    -  15.31MB 33:   lines := strings.Split(string(fileContents), "\n")
```

Уберем компиляцию регулярного выражения и вместо функции `r.ReplaceAllString()` будем использовать `strings.Replace()`. Тогда количество аллокаций должно уменьшиться.

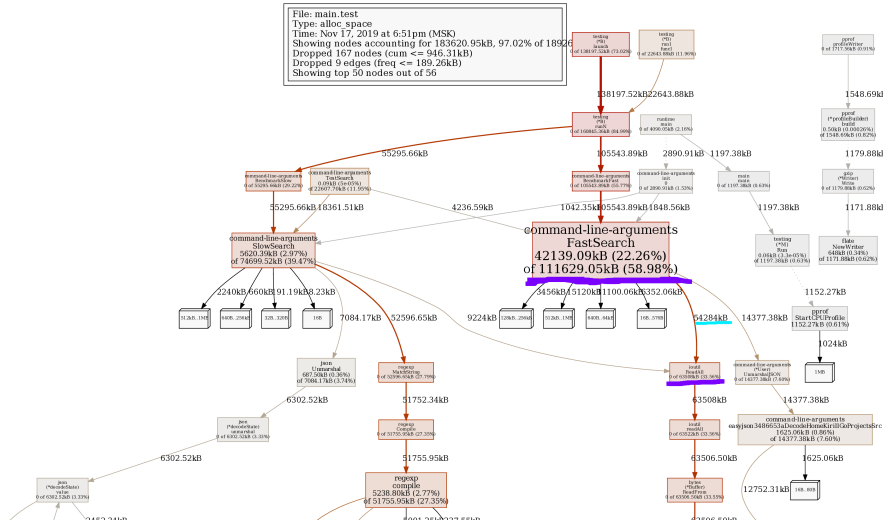
Результаты тестов.
Было:

```
optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      42862462 ns/op      18871874 B/op      195828 allocs/op
BenchmarkFast-4     244      4834879 ns/op       4366997 B/op       11846 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.019s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      42239588 ns/op      18853214 B/op      195816 allocs/op
BenchmarkFast-4     244      4762624 ns/op       4363333 B/op       11845 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.943s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      42381942 ns/op      18857928 B/op      195819 allocs/op
BenchmarkFast-4     244      4983589 ns/op       4364855 B/op       11846 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.017s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

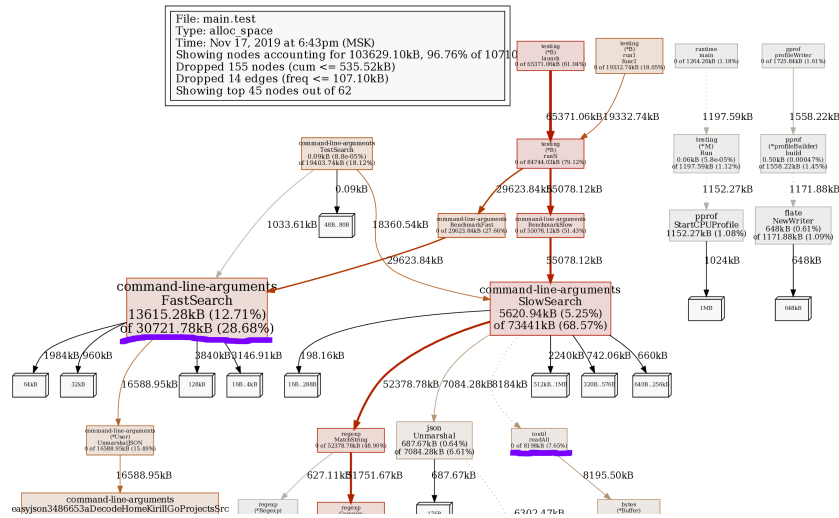
Стало:

```
optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      43611386 ns/op      18868368 B/op      195817 allocs/op
BenchmarkFast-4     248      4773221 ns/op       4322831 B/op       11627 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.043s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      42225156 ns/op      18847924 B/op      195815 allocs/op
BenchmarkFast-4     249      4734871 ns/op       4323181 B/op       11627 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.951s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      42288697 ns/op      18851288 B/op      195819 allocs/op
BenchmarkFast-4     236      4938384 ns/op       4322893 B/op       11627 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.084s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

4. Теперь посмотрим на память в виде графа. Было:



После того как изменили способ чтения файла (вместо пакета io/ioutil стали использовать пакет bufio) получилось следующее:



Результаты тестов.
Было:

```
optimization : bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      42862462 ns/op      18871874 B/op      195820 allocs/op
BenchmarkFast-4     244      4834879 ns/op       4366997 B/op       11846 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.019s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      42239588 ns/op      18853214 B/op      195816 allocs/op
BenchmarkFast-4     244      4762624 ns/op       4363333 B/op       11845 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.943s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      42381942 ns/op      18857928 B/op      195819 allocs/op
BenchmarkFast-4     244      4983589 ns/op       4364855 B/op       11846 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  3.017s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

Стало:

```
optimization : bash — Konsole <3>
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      41611924 ns/op      18851842 B/op      195814 allocs/op
BenchmarkFast-4     345      3557563 ns/op       1845217 B/op       10609 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.888s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      42813425 ns/op      18848564 B/op      195817 allocs/op
BenchmarkFast-4     348      3431348 ns/op       1845394 B/op       10609 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.884s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      44453788 ns/op      18856318 B/op      195818 allocs/op
BenchmarkFast-4     342      3451771 ns/op       1845298 B/op       10609 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization  2.882s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

5. Посмотрим на время работы в исходном коде:

```

optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kill@kill-Aspire-V5-573G:~/git/kill/kill-2019-2/5/99_hw/optimization$ go tool pprof main.test cpu.out
file main.test
Type: cpu
Time: Nov 19, 2019 at 5:56am (PMO)
Duration: 5.11s, Total samples = 5.28s (103.31%)
Entering interactive mode (type "help" for commands, "q" for options)
(pprof) list FastSearch
Total: 5.28s
ROUTINE:
0      1.25s (flat, cum) 23.67% of Total
      1: 25: foundUsers := ""
      2: 26:
      3: 27:   users := []User{}
      4: 28:
      5: 29:   for {
      6: 30:     10ms 30:     line, err := reader.ReadLine()
      7: 31:
      8: 32:     if err != io.EOF {
      9: 33:       break
     10: 34:     }
     11: 35:     data := []byte(line)
     12: 36:     user := User{}
     13: 37:     // fmt.Printf("%v %v\n", err, line)
     14: 38:
     15: 39:   }
     16: 1.19s 39:   err = user.UnmarshalJSON(data)
     17: 40:
     18: 41:   if err != nil {
     19: 42:     panic(err)
     20: 43:   }
     21: 10ms 43:   users = append(users, user)
     22: 44:
     23: 45:   }
     24: 46:   for i, user := range users {
     25: 47:
     26: 48:     isAndroid := false
     27: 49:     isMSIE := false
     28: 50:
     29: 51:     browsers := user.Browsers
     30: 52:
     31: 53:     for _, browser := range browsers {
     32: 54:       if strings.Contains(browser, "Android"){
     33: 55:         isAndroid = true
     34: 56:         notSeenBefore = true
     35: 57:         for _, item := range seenBrowsers {
     36: 58:           if item == browser {
     37: 59:             notSeenBefore = false

```

Вспомним, что все значения указываемые в файле `users.txt` нам не нужны. Следовательно, в структуре `User` можно оставить только `Name`, `Browser` и `email`. Получим:

```

File Edit View Bookmarks Settings Help
base kirill@kirill-Aspire-V5-573G:~/git/golang-2019-2/5/99_hw/optimization$ go tool pprof main.test cpu.out
(pprof) main.test
Type: cpu
Time: Nov 19, 2019 at 5:54am (HMK)
Duration: 5.11s, Total samples = 5.28s (103.37%)
Entering interactive mode (type "help" for commands, "o" for options)
(pprof) web
(pprof) list FastSearch
Total: 5.28s
outLINE ===== Command-line arguments.FastSearch in /home/kirill/git/golang-2019-2/5/99_hw/optimization/fast.go
-
- 30ms 1.20s (flat, cum) 27.7% of Total
-   : 25: foundUsers := ""
-   : 26:
-   : 27: users := []User{}
-   : 28:
-   : 29: for {
- 10ms 30:     line, _, err := reader.ReadLine()
-   : 31:
-   : 32:     if err == io.EOF {
-   : 33:         break
-   : 34:     }
-   : 35:     data := []byte(line)
-   : 36:     user := User{}
-   : 37:     // fmt.Printf("%v %v\n", err, line)
-   : 38:
-
- 1.04s 39:     err = user.UnmarshalJSON(data)
-   : 40:     if err != nil {
-   : 41:         panic(err)
-   : 42:     }
-   : 43:     users = append(users, user)
-   : 44: }
-   : 45:
10ms 46: for i, user := range users {
-   : 47:
-   : 48:     isAndroid := false
-   : 49:     isMIE := false
-   : 50:
-   : 51:     browsers := user.Browsers
-   : 52:
20ms 53:     for _, browser := range browsers {
- 10ms 54:         if strings.Contains(browser, "Android") {
-   : 55:             isAndroid = true
-   : 56:             notSeenBefore := true
-   : 57:             for _, item := range seenBrowsers {
-   : 58:                 if item == browser {

```

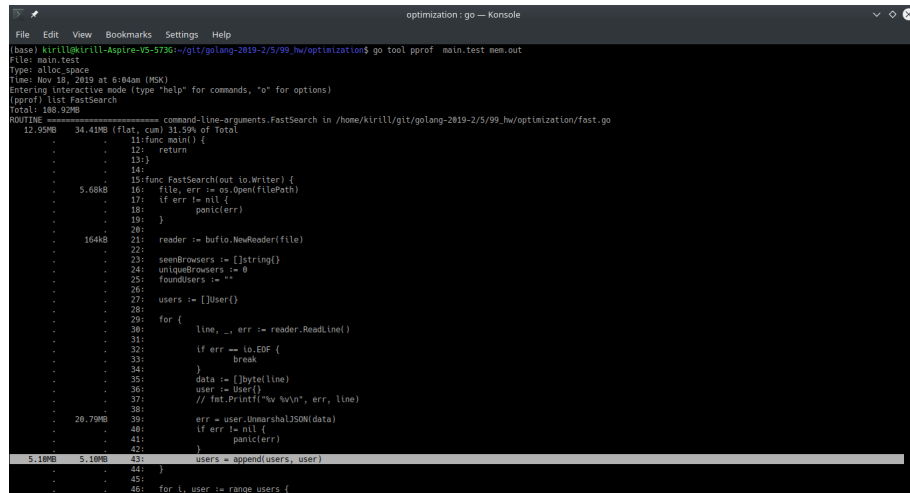
Результаты тестов.
Было:

```
optimization: bash — Konsole <3>
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      41611924 ns/op      18851842 B/op      195814 allocs/op
BenchmarkFast-4     345      3557563 ns/op      1045217 B/op      10609 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.880s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      42813425 ns/op      18848564 B/op      195817 allocs/op
BenchmarkFast-4     348      3431348 ns/op      1045394 B/op      10609 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.884s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      44453788 ns/op      18856316 B/op      195818 allocs/op
BenchmarkFast-4     342      3451771 ns/op      1045298 B/op      10609 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.882s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

Стало:

```
optimization: bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      41179439 ns/op      18842483 B/op      195814 allocs/op
BenchmarkFast-4     388      3183891 ns/op      878929 B/op      7689 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.799s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      41846251 ns/op      18836658 B/op      195816 allocs/op
BenchmarkFast-4     368      3144629 ns/op      878849 B/op      7689 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.799s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      41893343 ns/op      18859189 B/op      195818 allocs/op
BenchmarkFast-4     381      3115291 ns/op      878852 B/op      7689 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.838s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

6. Посмотрим на выделение памяти в исходном коде:



```
optimization: go - Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go tool pprof main.test mem.out
Type: alloc space
Time: Nov 18, 2019 at 6:04am (MSK)
Entering interactive mode (type "help" for commands, "o" for options)
(pprof) list FastSearch
total: 106.92MB
ROUTINE:
===== command-line-arguments.FastSearch in /home/kirill/git/golang-2019-2/5/99_hw/optimization/fast-go
12.95MB 34.41MB (flat, cum) 31.59% of Total
- 11:func main() {
- 12: return
- 13: }
- 14: }
- 15:func FastSearch(out io.Writer) {
5.08MB 16: file, err := os.Open(filePath)
- 17: if err != nil {
- 18: panic(err)
- 19: }
- 20: }
104KB 21: reader := bufio.NewReader(file)
- 22: }
- 23: seenBrowsers := []string{}
- 24: uniqueBrowsers := 0
- 25: foundUsers := ""
- 26: users := []User{}
- 27: }
- 28: }
- 29: for {
- 30: line, _, err := reader.ReadLine()
- 31: if err == io.EOF {
- 32: break
- 33: }
- 34: }
- 35: data := []byte(line)
- 36: user := User{}
- 37: // fmt.Printf("%v %v\n", err, line)
- 38: }
20.79MB 39: err = user.UnmarshalJSON(data)
- 40: if err != nil {
- 41: panic(err)
- 42: }
5.10MB 5.10MB 43: users = append(users, user)
- 44: }
- 45: }
- 46: for i, user := range users {
```

Видно, что тратится память на захват памяти в слайс структур каждый раз как добавляется новая структура. Если немного изменить логику программы и сразу выводить всех пользователей, а не запоминать их, то этой проблемы можно избежать.

Результаты тестов.
Было:

```
optimization : bash — Konsole
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      41179439 ns/op      18842483 B/op      195814 allocs/op
BenchmarkFast-4     388      3183891 ns/op       878929 B/op       7689 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.799s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      41846251 ns/op      18836658 B/op      195816 allocs/op
BenchmarkFast-4     368      3144629 ns/op       878649 B/op       7689 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.799s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      28      41993343 ns/op      18859189 B/op      195818 allocs/op
BenchmarkFast-4     381      3115291 ns/op       878852 B/op       7689 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.838s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```

Стало:

```
optimization : bash — Konsole <2>
File Edit View Bookmarks Settings Help
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      38      41188561 ns/op      18849486 B/op      195816 allocs/op
BenchmarkFast-4     384      3868848 ns/op       555238 B/op       7595 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.872s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      27      41828674 ns/op      18848258 B/op      195828 allocs/op
BenchmarkFast-4     391      3861893 ns/op       555236 B/op       7595 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.782s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$ go test -bench . -benchmem
goos: linux
goarch: amd64
BenchmarkSlow-4      21      54816116 ns/op      18887358 B/op      195824 allocs/op
BenchmarkFast-4     378      3118282 ns/op       555231 B/op       7595 allocs/op
PASS
ok      ./home/kirill/git/golang-2019-2/5/99_hw/optimization 2.824s
(base) kirill@kirill-Aspire-V5-5736:~/git/golang-2019-2/5/99_hw/optimization$
```