



Когда нельзя,
но очень хочется? GO!

Лазаренков Егор
Исполнительный директор, SberInfra

О спикере

Егор Лазаренков

Разработал свою первую ERP систему на Delphi 7 и MS SQL 2000
18 лет назад и с того времени не перестаю изучать новые технологии, языки программирования и писать коммерческий код.

- Лидер Golang community
- Исполнительный директор в SberInfra
Управление развития облачных решений


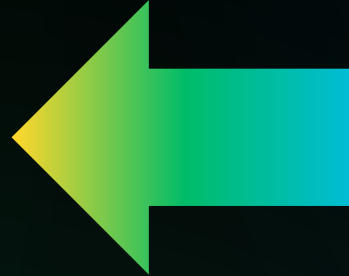

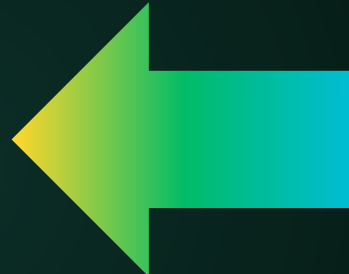


**golang
community**
#SBERTeam

Чем может быть полезно?

- Упрощает разработку Unit тестов, аналог mockito в Java
- Добавление нового функционала в стороннюю библиотеку, которая не поддерживает его
- Удаление из исполняемых файлов опасного и неиспользуемого кода
- Понять, как взламываются программы и разрабатывать более защищенные приложения с учетом этого

О чем речь?

	Языки с указателями	Языки без указателей
Примеры языков	C, C++, C#, Go, Rust, Swift	Python, Java, Kotlin
Правки оперативной памяти в обход ООП и прочих правил используемого языка		<p>Получаем прямой доступ к памяти:</p>  <ul style="list-style-type: none">• Используем низкоуровневые интерфейсы, например JNI• Либо используем специальные библиотеки, например sun.misc.unsafe
Правки исполняемого файла ABI без исходников		<p>Компиляция JIT:</p> <ol style="list-style-type: none">1. Декомпилируем в исходники - Dex2jar, Java Decompiler2. Правим и собираем снова – ApkTool <p>Компиляция AOT:</p>  <ul style="list-style-type: none">• GNU Compiler for Java• GraalVM• Cython

Работаем с БД Postgres

```
1 package examples
2
3 import (
4     "context"
5     "testing"
6
7     "github.com/jackc/pgx/v5"
8     "github.com/stretchr/testify/require"
9 )
10
11 var ctx = context.Background()
12
13 const connStr = "postgresql://postgres:*****@tvlds-nimbs0107.delta.sbrf.ru"
14
15 run test | debug test
16 func TestConnect(t *testing.T) {
17     conn, err := pgx.Connect(ctx, connStr)
18     require.NoError(t, err)
19
20     defer conn.Close(ctx)
21
22     rows, err := conn.Query(ctx, "SELECT 1")
23     require.NoError(t, err)
24     rows.Close()
25 }
```

PASS

Process 62707 has exited with status 0

В поисках трейсера

```
// Connect establishes a connection with a PostgreSQL server with a connection string. See
// pgconn.Connect for details.
func Connect(ctx context.Context, connString string) (*Conn, error) {
    connConfig, err := ParseConfig(connString)
    if err != nil {
        return nil, err
    }
    return connect(ctx, connConfig)
}
```

```
func ParseConfig(connString string) (*ConnConfig, error) {
    return ParseConfigWithOptions(connString, ParseConfigOptions{})
}
```

```
// ConnConfig contains all the options used to establish a connection. It must be created by ParseConfig and
// then it can be modified. A manually initialized ConnConfig will cause ConnectConfig to panic.
```

```
type ConnConfig struct {
    pgconn.Config
```

```
→ Tracer QueryTracer
```

```
    // Original connection string that was parsed into config.
    connString string
```

```
    // StatementCacheCapacity is maximum size of the statement cache used when executing a query with "cache_statement"
    // query exec mode.
    StatementCacheCapacity int
```


И где же ты?

```
func ParseConfigWithOptions(connString string, options ParseConfigOptions) (*ConnConfig, error) {
    config, err := pgconn.ParseConfigWithOptions(connString, options.ParseConfigOptions)
    if err != nil {
        return nil, err
    }

    statementCacheCapacity := 512
    if s, ok := config.RuntimeParams["statement_cache_capacity"]; ok {
        delete(config.RuntimeParams, "statement_cache_capacity")
        n, err := strconv.ParseInt(s, 10, 32)
        if err != nil {
            return nil, fmt.Errorf("cannot parse statement_cache_capacity: %w", err)
        }
        statementCacheCapacity = int(n)
    }

    descriptionCacheCapacity := 512
    if s, ok := config.RuntimeParams["description_cache_capacity"]; ok {
        delete(config.RuntimeParams, "description_cache_capacity")
        n, err := strconv.ParseInt(s, 10, 32)
        if err != nil {
            return nil, fmt.Errorf("cannot parse description_cache_capacity: %w", err)
        }
        descriptionCacheCapacity = int(n)
    }
}
```

```
defaultQueryExecMode := QueryExecModeCacheStatement
if s, ok := config.RuntimeParams["default_query_exec_mode"]; ok {
    delete(config.RuntimeParams, "default_query_exec_mode")
    switch s {
    case "cache_statement":
        defaultQueryExecMode = QueryExecModeCacheStatement
    case "cache_describe":
        defaultQueryExecMode = QueryExecModeCacheDescribe
    case "describe":
        defaultQueryExecMode = QueryExecModeDescribe
    case "exec":
        defaultQueryExecMode = QueryExecModeExec
    case "simple":
        defaultQueryExecMode = QueryExecModeSimple
    default:
        return nil, fmt.Errorf("invalid default_query_exec_mode: %s", s)
    }
}

type ConnConfig struct {
    pgconn.Config

    Tracer QueryTracer

    // Original connection string that was parsed in ParseConfigWithOptions
    connString string

    // StatementCacheCapacity is maximum size of the statement cache.
    // query exec mode.
    StatementCacheCapacity int

    // DescriptionCacheCapacity is the maximum size of the description cache.
    // "cache_describe" query exec mode.
    DescriptionCacheCapacity int
}
```

```
connConfig := &ConnConfig{
    Config:          *config,
    createdByParseConfig: true,
    StatementCacheCapacity: statementCacheCapacity,
    DescriptionCacheCapacity: descriptionCacheCapacity,
    DefaultQueryExecMode: defaultQueryExecMode,
    connString:        connString,
}
```



```

// ConnectWithOptions behaves exactly like Connect with the addition of options. At the present options is only used to
// provide a GetSSLPassword function.
func ConnectWithOptions(ctx context.Context, connString string, options ParseConfigOptions) (*Conn, error) {
    connConfig, err := ParseConfigWithOptions(connString, options)
    if err != nil {
        return nil, err
    }
    return connect(ctx, connConfig)
}

// ConnectConfig establishes a connection with a PostgreSQL server with a configuration struct.
// connConfig must have been created by ParseConfig.
func ConnectConfig(ctx context.Context, connConfig *ConnConfig) (*Conn, error) {
    // In general this improves safety. In particular avoid the config.Config.OnNotification mutation from affecting other
    // connections with the same config. See https://github.com/jackc/pgx/issues/618.
    connConfig = connConfig.Copy()

    return connect(ctx, connConfig)
}

// New creates a new Pool. See [ParseConfig] for information on connString format.
func New(ctx context.Context, connString string) (*Pool, error) {
    config, err := ParseConfig(connString)
    if err != nil {
        return nil, err
    }

    return NewWithConfig(ctx, config)
}

func ParseConfig(connString string) (*ConnConfig, error) {
    return ParseConfigWithOptions(connString, ParseConfigOptions{})
}

```

Кто умеет
работать
с трейсером?


```

173 // NewWithConfig creates a new Pool. config must have been created by [ParseConfig].
174 func NewWithConfig(ctx context.Context, config *Config) (*Pool, error) {
175     // Default values are set in ParseConfig. Enforce initial creation by ParseConfig rather than setting defaults from
176     // zero values.
177     if !config.createdByParseConfig {
178         panic("config must be created by ParseConfig")
179     }
180
181
182
183
184
185
186
187
188 var err error
189 p.p, err = puddle.NewPool(
190     &puddle.Config[*connResource]{
191         Constructor: func(ctx context.Context) (*connResource, error) {
192             atomic.AddInt64(&p.newConnsCount, 1)
193             connConfig := p.config.ConnConfig.Copy()
194
195             // Connection will continue in background even if Acquire is canceled. Ensure that a connect won't hang forever.
196             if connConfig.ConnectTimeout <= 0 {
197                 connConfig.ConnectTimeout = 2 * time.Minute
198             }
199
200             if p.beforeConnect != nil {
201                 if err := p.beforeConnect(ctx, connConfig); err != nil {
202                     return nil, err
203                 }
204             }
205
206             conn, err := pgx.ConnectConfig(ctx, connConfig)
207             if err != nil {
208                 return nil, err
209             }
210
211             if p.afterConnect != nil {
212                 err = p.afterConnect(ctx, conn)
213                 if err != nil {
214                     conn.Close(ctx)
215                     return nil, err
216                 }
217             }
218         },
219     },
220 )
221
222
223
224
225
226
227
228

```

Опять ты?!

Кого патчить будем?

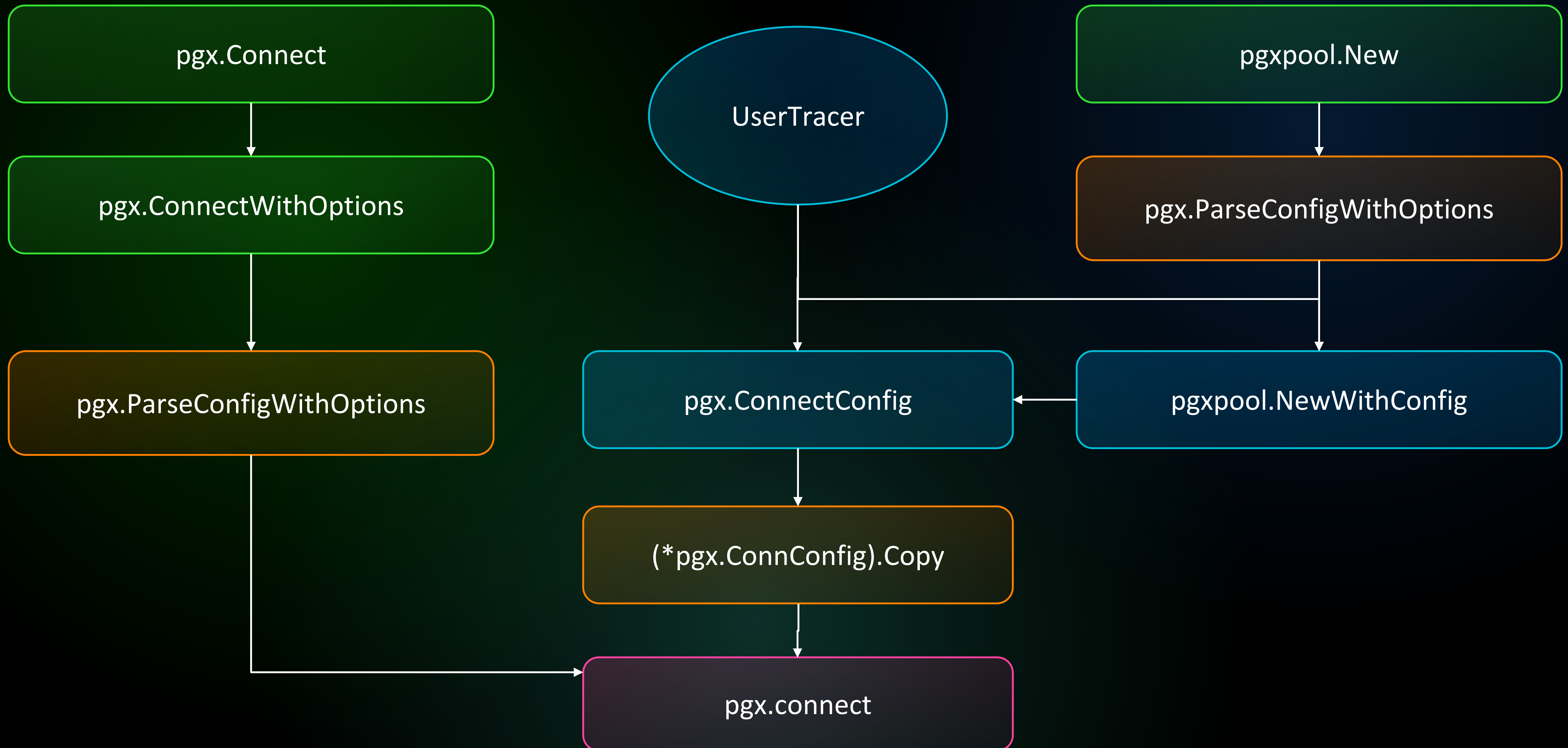


Таблица символов исполняемого файла

ОС	Команда вывода	Парсер в Go sdk
Linux	<code>readelf -St -W file</code>	<code>pkg.go.dev/debug/elf</code>
MacOS	<code>gobjdump -t file</code>	<code>pkg.go.dev/debug/macho</code>
Windows	<code>dumpbin /symbols file</code>	<code>pkg.go.dev/debug/pe</code>

Какое у Вас полное имя?

```
20047779@cab-wsm-0092006 examples % go test -c .
20047779@cab-wsm-0092006 examples % ls
examples.test  main_test.go  pgxtracer      querylogger
20047779@cab-wsm-0092006 examples % gobldump -t examples.test | grep connect
000000000124f500 l      0e SECT    01 0000 [.text] _crypto/tls.(*Conn).connectionStateLocked
00000000013975a0 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5.connect
0000000001397cc0 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5.connect.func1
0000000001280e20 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.(*connectError).Error
0000000001281000 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.(*connectError).Unwrap
00000000012831a0 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.connect
0000000001284c00 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.connect.func5
0000000001284d60 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.connect.newContextWatcher.func1
0000000001284d20 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.connect.newContextWatcher.func2
0000000001284ca0 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.connect.newContextWatcher.func3
0000000001284c60 l      0e SECT    01 0000 [.text] _github.com/jackc/pgx/v5/pgconn.connect.newContextWatcher.func4
0000000001535420 l      0e SECT    03 0000 [__TEXT:__rodata] _go:itab.*github.com/jackc/pgx/v5/pgconn.connectError,error
0000000001201640 l      0e SECT    01 0000 [.text] _net.(*netFD).connect
00000000012022c0 l      0e SECT    01 0000 [.text] _net.(*netFD).connect.func1
00000000012021a0 l      0e SECT    01 0000 [.text] _net.(*netFD).connect.func2
0000000001202400 l      0e SECT    01 0000 [.text] _net.(*netFD).connect.func3
000000000182d880 l      0e SECT    0b 0000 [.data] _net.connectFunc
0000000001078720 l      0e SECT    01 0000 [.text] _syscall.connect
000000000107a100 l      0e SECT    01 0000 [.text] _syscall.libc_connect_trampoline.abi0
000000000128bf40 l      0e SECT    01 0000 [.text] _type:.eq.github.com/jackc/pgx/v5/pgconn.connectError
0000000000000000 g      01 UND     00 0000 _connect
```



```

1 package pgxtracer
2
3 import (
4     "context"
5     "sync/atomic"
6
7     "github.com/jackc/pgx/v5"
8
9     _ "unsafe"
10
11     "gitlab.ocp.delta.sbrf.ru/godev/core.git/function"
12 )
13
14 var (
15     globalTracer atomic.Pointer[pgx.QueryTracer]
16     originConnect func(ctx context.Context, config *pgx.ConnConfig) (c *pgx.Conn, err error)
17 )
18
19 //go:linkname pgxConnect github.com/jackc/pgx/v5.connect
20 func pgxConnect(ctx context.Context, config *pgx.ConnConfig) (c *pgx.Conn, err error)
21
22 func newConnect(ctx context.Context, config *pgx.ConnConfig) (c *pgx.Conn, err error) {
23     if pTracer := globalTracer.Load(); pTracer != nil {
24         if config.Tracer == nil {
25             config.Tracer = *pTracer
26         } else {
27             config.Tracer = QueryTracerList{*pTracer, config.Tracer}
28         }
29     }
30
31     return originConnect(ctx, config)
32 }
33
34 func init() {
35     function.Replace(pgxConnect, newConnect, &originConnect)
36 }
37
38 func SetGlobalTracer(tracer pgx.QueryTracer) {
39     globalTracer.Store(&tracer)
40 }

```

Знакомьтесь,
это Ваша новая
реализация

```

// QueryTracer traces Query, QueryRow, and Exec.
type QueryTracer interface {
    // TraceQueryStart is called at the beginning of Query, QueryRow, and Exec calls. The returned context is used for the
    // rest of the call and will be passed to TraceQueryEnd.
    TraceQueryStart(ctx context.Context, conn *Conn, data TraceQueryStartData) context.Context

    TraceQueryEnd(ctx context.Context, conn *Conn, data TraceQueryEndData)
}

1 package querylogger
2
3 import (
4     "context"
5     "log"
6     "sync/atomic"
7
8     "github.com/jackc/pgx/v5"
9 )
10
11 type QueryLogger struct {
12     QueryNumber uint64
13 }
14
15 const queryNumberKey = "query-number"
16
17 func (s *QueryLogger) TraceQueryStart(ctx context.Context, conn *pgx.Conn, data pgx.TraceQueryStartData) context.Context {
18     number := atomic.AddUint64(&s.QueryNumber, 1)
19
20     log.Println("start", number, "query:", data.SQL)
21
22     return context.WithValue(ctx, queryNumberKey, number)
23 }
24
25 func (s *QueryLogger) TraceQueryEnd(ctx context.Context, conn *pgx.Conn, data pgx.TraceQueryEndData) {
26     log.Println("end", ctx.Value(queryNumberKey), "query")
27 }

```

Тре́йсы будем писать в лог

Подключаем наш трейс-логгер

```
1 package examples
2
3 import (
4     "context"
5     "sync"
6     "testing"
7
8     "github.com/jackc/pgx/v5"
9     "github.com/jackc/pgx/v5/pgxpool"
10    "github.com/stretchr/testify/require"
11
12    "gitlab.ocp.delta.sbrf.ru/godev/grpcmock.git/examples/pgxtracer"
13    "gitlab.ocp.delta.sbrf.ru/godev/grpcmock.git/examples/querylogger"
14 )
15
16 var ctx = context.Background()
17
18 const connStr = "postgresql://postgres:*****@tvlds-nimbs0107.delta.sbrf.ru"
19
20 func init() {
21     pgxtracer.SetGlobalTracer(&querylogger.QueryLogger{})
22 }
23
24 run test | debug test
25 func TestConnect(t *testing.T) {
26     conn, err := pgx.Connect(ctx, connStr)
27     require.NoError(t, err)
28
29     defer conn.Close(ctx)
30
31     rows, err := conn.Query(ctx, "SELECT 1")
32     require.NoError(t, err)
33     rows.Close()
34 }
```

```
2023/10/02 14:49:40 start 1 query: SELECT 1
2023/10/02 14:49:40 end 1 query
PASS
Process 37714 has exited with status 0
```

```

35 func TestPool(t *testing.T) {
36     pool, err := pgxpool.New(ctx, connStr)
37     require.NoError(t, err)
38
39     defer pool.Close()
40
41     var wg sync.WaitGroup
42
43     wg.Add(100)
44     for i := 0; i < 100; i++ {
45         go func() {
46             defer wg.Done()
47             rows, err := pool.Query(ctx, "SELECT 2")
48             require.NoError(t, err)
49             rows.Close()
50         }()
51     }
52
53     wg.Wait()
54 }

```

Победа!

```

2023/10/02 14:50:48 end 79 query
2023/10/02 14:50:48 end 80 query
2023/10/02 14:50:48 start 91 query: SELECT 2
2023/10/02 14:50:48 start 92 query: SELECT 2
2023/10/02 14:50:48 end 82 query
2023/10/02 14:50:48 end 81 query
2023/10/02 14:50:48 start 93 query: SELECT 2
2023/10/02 14:50:48 start 94 query: SELECT 2
2023/10/02 14:50:48 end 84 query
2023/10/02 14:50:48 start 95 query: SELECT 2
2023/10/02 14:50:48 end 86 query
2023/10/02 14:50:48 end 83 query
2023/10/02 14:50:48 start 97 query: SELECT 2
2023/10/02 14:50:48 start 96 query: SELECT 2
2023/10/02 14:50:48 end 85 query
2023/10/02 14:50:48 end 87 query
2023/10/02 14:50:48 start 98 query: SELECT 2
2023/10/02 14:50:48 end 88 query
2023/10/02 14:50:48 end 91 query
2023/10/02 14:50:48 start 99 query: SELECT 2
2023/10/02 14:50:48 start 100 query: SELECT 2
2023/10/02 14:50:48 end 89 query
2023/10/02 14:50:48 end 90 query
2023/10/02 14:50:48 end 92 query
2023/10/02 14:50:48 end 94 query
2023/10/02 14:50:48 end 93 query
2023/10/02 14:50:48 end 95 query
2023/10/02 14:50:48 end 96 query
2023/10/02 14:50:48 end 99 query
2023/10/02 14:50:48 end 97 query
2023/10/02 14:50:48 end 100 query
2023/10/02 14:50:48 end 98 query
PASS
Process 37815 has exited with status 0

```



```

57 func TestConnectConfig(t *testing.T) {
58
59     config := pgx.ConnConfig{
60         Config: pgconn.Config{
61             Host:      "tvlds-nimbs0107.delta.sbrf.ru",
62             Port:     5432,
63             Database: "postgres",
64             User:      "postgres",
65             Password: "*****",
66         },
67         Tracer: nil,
68     }
69
70     conn, err := pgx.ConnectConfig(ctx, &config)
71     require.NoError(t, err)
72
73     defer conn.Close(ctx)
74
75     rows, err := conn.Query(ctx, "SELECT 1")
76     require.NoError(t, err)
77     rows.Close()
78 }

```

Попробуем
передать свой
конфиг

```

--- FAIL: TestConnectConfig (0.00s)
panic: config must be created by ParseConfig [recovered]
      panic: config must be created by ParseConfig

```



```
// connect connects to a database. connect takes ownership of config. The caller must not use
func connect(ctx context.Context, config *ConnConfig) (c *Conn, err error) {
    if connectTracer, ok := config.Tracer.(ConnectTracer); ok {
        ctx = connectTracer.TraceConnectStart(ctx, TraceConnectStartData{ConnConfig: config})
        defer func() {
            connectTracer.TraceConnectEnd(ctx, TraceConnectEndData{Conn: c, Err: err})
        }()
    }

    // Default values are set in ParseConfig. Enforce initial creation by ParseConfig rather than
    // zero values.
```

```
    if !config.createdByParseConfig {
        panic("config must be created by ParseConfig")
    }
```

```
func ParseConfig(connString string) (*Config, error) {
    connConfig, err := pgx.ParseConfig(connString)
    if err != nil {
        return nil, err
    }
```

```
    config := &Config{
        ConnConfig:      connConfig,
        createdByParseConfig: true,
    }
```

Преступление
раскрыто!

Да нас же предупреждали...

```
type ConnConfig struct {
    pgconn.Config

    Tracer QueryTracer

    // Original connection string that was parsed into config.
    connString string

    // StatementCacheCapacity is maximum size of the statement cache used when executing a query with "cache_statement"
    // query exec mode.
    StatementCacheCapacity int

    // DescriptionCacheCapacity is the maximum size of the description cache used when executing a query with
    // "cache_describe" query exec mode.
    DescriptionCacheCapacity int

    // DefaultQueryExecMode controls the default mode for executing queries. By default pgx uses the extended protocol
    // and automatically prepares and caches prepared statements. However, this may be incompatible with proxies such as
    // PGBouncer. In this case it may be preferable to use QueryExecModeExec or QueryExecModeSimpleProtocol. The same
    // functionality can be controlled on a per query basis by passing a QueryExecMode as the first query argument.
    DefaultQueryExecMode QueryExecMode

    createdByParseConfig bool // Used to enforce created by ParseConfig rule.
}
```



```

57 func TestConnectConfig(t *testing.T) {
58
59     config := pgx.ConnConfig{
60         Config: pgconn.Config{
61             Host:      "tvlds-nimbs0107.delta.sbrf.ru",
62             Port:      5432,
63             Database:  "postgres",
64             User:      "postgres",
65             Password:  "*****",
66         },
67         Tra
68     }
69
70     config.createdByParseConfig = true
71
72     conn, err := pgx.ConnectConfig(ctx, &config)
73     require.NoError(t, err)
74
75     defer conn.Close(ctx)
76
77     rows, err := conn.Query(ctx, "SELECT 1")
78     require.NoError(t, err)
79     rows.Close()
80 }
81

```

config.createdByParseConfig undefined (type pgx.ConnConfig has no field or method createdByParseConfig) compiler([MissingFieldOrMethod](#))

[View Problem \(F8\)](#) No quick fixes available

И что, прям вообще
никак?

И так тоже нельзя?!

```
reflect.ValueOf(config).FieldByName("createdByParseConfig").Set(reflect.ValueOf(true))
```

```
panic: reflect: reflect.Value.Set using value obtained using unexported field [recovered]  
      panic: reflect: reflect.Value.Set using value obtained using unexported field
```



```

61 config := pgx.ConnConfig{
62     Config:          pgconn.Config{
63         Host: "tvlds-nimbs0107.delta.sbrf.ru",
64         Port: 5432,
65         Database: "postgres",
66         User: "postgres",
67         Password: "*****"},
68     Tracer:          nil,
69     StatementCacheCapacity: 0,
70     DescriptionCacheCapacity: 0,
71     DefaultQueryExecMode: 0,
72 }
73
74 tp := reflect.ValueOf(config).Type()
75 for i := 0; i < tp.NumField(); i++ {
76     fmt.Println(tp.Field(i).Name, tp.Field(i).Offset)
77 }

```

```

Config 0
Tracer 216
connString 232
StatementCacheCapacity 248
DescriptionCacheCapacity 256
DefaultQueryExecMode 264
createdByParseConfig 268

```

А где
Вы живете?

```

20 type ConnConfig struct {
21     pgconn.Config
22
23     Tracer QueryTracer
24
42     createdByParseConfig bool // Used to enforce created by ParseConfig rule.

```

```

75     tp := reflect.TypeOf(config.Config)
76     for i := 0; i < tp.NumField(); i++ {
77         field := tp.Field(i)
78         fmt.Println(field.Name, field.Offset)
79     }

```

```

31 type Config struct {
32     Host      string // host (e.g. localhost
33     Port      uint16
34     Database  string
35     User      string
36     Password  string

```

```

63     createdByParseConfig bool // Used to e

```

```

Host 0
Port 16
Database 24
User 40
Password 56
TLSConfig 72
ConnectTimeout 80
DialFunc 88
LookupFunc 96
BuildFrontend 104
RuntimeParams 112
KerberosSrvName 120
KerberosSpn 136
Fallbacks 152
ValidateConnect 176
AfterConnect 184
OnNotice 192
OnNotification 200
createdByParseConfig

```

И с Вами приятно познакомиться


```

62 func TestConnectConfig(t *testing.T) {
63     connConfig := pgx.ConnConfig{
64         Config: pgconn.Config{
65             Host:      "tvlds-nimbs0107.delta.sbrf.ru",
66             Port:      5432,
67             Database:  "postgres",
68             User:      "postgres",
69             Password:  "*****",
70             TLSConfig: nil,
71             ConnectTimeout: time.Second,
72             DialFunc:    (&net.Dialer{KeepAlive: 5 * time.Minute}).DialContext,
73             LookupFunc:  net.DefaultResolver.LookupHost,
74             BuildFrontend: func(r io.Reader, w io.Writer) *pgproto3.Frontend {
75                 return pgproto3.NewFrontend(r, w)
76             },
77         },
78         Tracer:          nil,
79         StatementCacheCapacity: 512,
80         DescriptionCacheCapacity: 512,
81         DefaultQueryExecMode:    pgx.QueryExecModeCacheStatement,
82     }
83
84     *(*bool)(unsafe.Pointer(uintptr(unsafe.Pointer(&connConfig)) + 268)) = true
85     *(*bool)(unsafe.Pointer(uintptr(unsafe.Pointer(&connConfig.Config)) + 208)) = true
86
87     conn, err := pgx.ConnectConfig(ctx, &connConfig)
88     require.NoError(t, err)
89
90     defer conn.Close(ctx)
91
92     rows, err := conn.Query(ctx, "SELECT 1")
93     require.NoError(t, err)
94     rows.Close()
95 }

```

А говорили ,
что нельзя....

```

2023/10/03 11:10:06 start 1 query: SELECT 1
2023/10/03 11:10:06 end 1 query
PASS
Process 24267 has exited with status 0

```

И даже так можно!

```
*(*bool)(reflect.ValueOf(&connConfig).Elem().FieldByName("createdByParseConfig").  
    Addr().UnsafePointer()) = true  
  
*(*bool)(reflect.ValueOf(&connConfig.Config).Elem().FieldByName("createdByParseConfig").  
    Addr().UnsafePointer()) = true
```

2023/10/03 11:20:20 start 1 query: SELECT 1

2023/10/03 11:20:20 end 1 query

PASS

Process 28748 has exited with status 0

Смотря, как попросить...

Нельзя	Но можно
Читать изменять/приватные поля объекта	<ol style="list-style-type: none">1. Вычисляем адрес поля в памяти, где хранится это поле2. Кастим полученный указатель в указатель на тип его значения
Читать изменять/приватные поля объекта имеющие приватный тип	<ol style="list-style-type: none">1. Определяем размер приватного типа2. Создаем новый совместимый тип такого же размера3. Вычисляем адрес поля в памяти, где хранится целевое значение4. Кастим полученный указатель в указатель на совместимый-тип
Получить доступ к приватному методу или функции другого пакета	<ol style="list-style-type: none">1. Находим адрес в памяти, где начинается эта функция при помощи любого из вариантов: symbols table, runtime.moduledata , linkname, disasm caller func2. Создаем на основе точки входа в функцию переменную с функциональным типом
Изменять неизменяемое. Например иммутабельный string, либо тело функции.	<ol style="list-style-type: none">1. Получаем адрес в памяти, где хранится константа, либо код2. Снимаем защиту от записи со страниц виртуальной памяти, где расположена эта константа3. Кастим указатель на эту память в массив байт либо любой совместимый тип и изменяем его элементы.

```
func TestString(t *testing.T) {
```

```
    const str = "hello world"
```

cannot assign to str[1] (neither addressable nor a map index expression)

```
    const str untyped string = "hello world"
```

[View Problem \(⌘F8\)](#) No quick fixes available

```
    str[1] = 'X'
```

```
    fmt.Println(str)
```

```
}
```

```
func TestString(t *testing.T) {
```

```
    const str string = "hello world"
```

```
    stringAsBytes := unsafe.Slice(unsafe.StringData(str), len(str))
```

```
    stringAsBytes[1] = byte('X')
```

```
    fmt.Println(str)
```

```
}
```

Ну вот так
ТОЧНО НЕЛЬЗЯ...

unexpected fault address 0x1720b47

fatal error: fault

[signal SIGBUS: bus error code=0x2 addr=0x1720b47 pc=0x163f247]

Произносим заклинание

```
97 //go:linkname setMemProtect gitlab.ocp.delta.sbrf.ru/godev/core.git/function.setMemProtect
98 func setMemProtect(ptr uintptr, size int, allowWrite bool)
run test | debug test
99 func TestString(t *testing.T) {
100     const str string = "hello world"
101
102     ptr := unsafe.StringData(str)
103     stringAsBytes := unsafe.Slice(ptr, len(str))
104
105     setMemProtect(uintptr(unsafe.Pointer(ptr)), len(str), true)
106
107     stringAsBytes[1] = 'X'
108     fmt.Println(str)
109 }
```

```
hXllo world
PASS
Process 41205 has exited with status 0
```

← → ↺

developer.apple.com/documentation/kernel/1402291-mach_vm_protect

🔖 ☆ □ ⓘ

🍏 Developer

News Discover Design Develop Distribute Support Account 🔍

Kernel 📖

🔗 mach_vm_read_list

🔗 mach_vm_read_overwrite

Configuration

🔗 mach_vm_protect

🔗 mach_vm_wire

🔗 mach_vm_inherit

🔗 mach_vm_machine_attribute

🔗 mach_vm_msync

🔗 mach_vm_purgable_control

🔗 mach_vm_behavior_set

🔗 mach_vm_page_info

🔗 mach_vm_page_query

🔗 mach_vm_page_range_query

🔗 mach_vm_round_page_overflow

Regions

🔗 mach_vm_region

🔗 mach_vm_region_info

Documentation / ... / Mach VM / mach_vm_protect

Language: Objective-C API Changes: None

Function

mach_vm_protect

macOS 10.4+

```
kern_return_t mach_vm_protect(vm_map_t target_task, mach_vm_address_t address, mach_vm_size_t size, boolean_t set_maximum, vm_prot_t new_protection);
```

See Also

Configuration

[mach_vm_wire](#)

Подключаем СИ

function/protect_darwin.go

```
1 //go:build darwin
2 // +build darwin
3
4 package function
5
6 /*
7 #include <mach/mach.h>
8 */
9 import "C"
10
11 func setMemProtect(ptr uintptr, size int, allowWrite bool) {
12     prot := C.VM_PROT_READ | C.VM_PROT_EXECUTE | C.VM_PROT_COPY
13
14     if allowWrite {
15         prot |= C.VM_PROT_WRITE
16     }
17
18     if C.vm_protect(C.mach_task_self_, C.ulong(ptr), C.ulong(size), 0, C.int(prot)) != C.KERN_SUCCESS {
19         panic("can't setMemProtect memory protect on macos")
20     }
21 }
```

Заглянем в кроличью нору

```
← → ↻ 🔒 opensource.apple.com/source/xnu/xnu-7195.50.7.100.1/libsyscall/mach/mach_vm.c.auto.html

kern_return_t
mach_vm_protect(
    mach_port_name_t task,
    mach_vm_address_t address,
    mach_vm_size_t size,
    boolean_t set_maximum,
    vm_prot_t new_protection)
{
    kern_return_t rv;

    rv = _kernelrpc_mach_vm_protect_trap(task, address, size, set_maximum,
        new_protection);

    if (rv == MACH_SEND_INVALID_DEST) {
        rv = _kernelrpc_mach_vm_protect(task, address, size,
            set_maximum, new_protection);
    }

    return rv;
}
```

```
← → ↻ 🔒 opensource.apple.com/source/xnu/xnu-6153.11.26/osfmk/mach/mach_traps.h.auto.html

struct _kernelrpc_mach_vm_protect_args {
    PAD_ARG_1(mach_port_name_t, target); /* 1 word */
    PAD_ARG_2(mach_vm_address_t, address); /* 2 words */
    PAD_ARG_2(mach_vm_size_t, size); /* 2 words */
    PAD_ARG_1(boolean_t, set_maximum); /* 1 word */
    PAD_ARG_1(vm_prot_t, new_protection); /* 1 word */
    /* Total: 7 */
};

extern kern_return_t _kernelrpc_mach_vm_protect_trap(
    struct _kernelrpc_mach_vm_protect_args *args);
```



```

< > ↺ opensource.apple.com/source/xnu/xnu-4570.41.2/osfmk/mach/i386/syscall_sw.h.auto.html

/*
#define kernel_trap(trap_name, trap_number, number_args) \
LEAF(##trap_name, 0) ;\
    movq    %rcx, %r10    ;\
    movl    $ SYSCALL_CONSTRUCT_MACH(##trap_number), %eax    ;\
    syscall                ;\
END(##trap_name)

#endif /* !KERNEL */

#endif /* defined(__x86_64__) */

/*
 * Syscall classes for 64-bit system call entry.
 * For 64-bit users, the 32-bit syscall number is partitioned
 * with the high-order bits representing the class and low-order
 * bits being the syscall number within that class.
 * The high-order 32-bits of the 64-bit syscall number are unused.
 * All system classes enter the kernel via the syscall instruction.
 *
 * These are not #ifdef'd for x86-64 because they might be used for
 * 32-bit someday and so the 64-bit comm page in a 32-bit kernel
 * can use them.
 */
#define SYSCALL_CLASS_SHIFT    24
#define SYSCALL_CLASS_MASK    (0xFF << SYSCALL_CLASS_SHIFT)
#define SYSCALL_NUMBER_MASK    (~SYSCALL_CLASS_MASK)

#define I386_SYSCALL_CLASS_MASK    SYSCALL_CLASS_MASK
#define I386_SYSCALL_ARG_BYTES_SHIFT    (16)
#define I386_SYSCALL_ARG_DWORDS_SHIFT    (I386_SYSCALL_ARG_BYTES_SHIFT + 2)
#define I386_SYSCALL_ARG_BYTES_NUM    (64) /* Must be <= sizeof(uu_arg) */
#define I386_SYSCALL_ARG_DWORDS_MASK    ((I386_SYSCALL_ARG_BYTES_NUM >> 2) - 1)
#define I386_SYSCALL_ARG_BYTES_MASK    (((I386_SYSCALL_ARG_BYTES_NUM - 1) & ~0x3) << I386_SYSCALL_ARG_BYTES_SHIFT)
#define I386_SYSCALL_NUMBER_MASK    (0xFFFF)

#define SYSCALL_CLASS_NONE    0    /* Invalid */
#define SYSCALL_CLASS_MACH    1    /* Mach */
#define SYSCALL_CLASS_UNIX    2    /* Unix/BSD */
#define SYSCALL_CLASS_MDEP    3    /* Machine-dependent */
#define SYSCALL_CLASS_DIAG    4    /* Diagnostics */
#define SYSCALL_CLASS_IPC    5    /* Mach IPC */

/* Macros to simplify constructing syscall numbers. */
#define SYSCALL_CONSTRUCT_MACH(syscall_number) \
    ((SYSCALL_CLASS_MACH << SYSCALL_CLASS_SHIFT) | \
     (SYSCALL_NUMBER_MASK & (syscall_number)))

```

В поисках СИСТЕМНОГО ВЪ

`syscall_number = (1 << 24) | (~(1
<< 24) & (-trap_number))`

`syscall_number = 0x1000000 -
trap_number`

Ах вот же ты!

```
← → ↻ 🔒 opensource.apple.com/source/xnu/xnu-6153.11.26/osfmk/mach/syscall_sw.h.auto.html

/*
 * i386 and x86_64 just load of the stack or use
 * registers in order; no munging is required,
 * and number of args is ignored.  ARM loads args
 * into registers beyond r3, unlike the normal
 * procedure call standard; we pad for 64-bit args.
 */
kernel_trap(_kernelrpc_mach_vm_allocate_trap,-10,5) /* 4 args, +1 for mach_vm_size_t */
kernel_trap(_kernelrpc_mach_vm_purgable_control_trap,-11,5) /* 4 args, +1 for mach_vm_offset_t */
kernel_trap(_kernelrpc_mach_vm_deallocate_trap,-12,5) /* 3 args, +2 for mach_vm_size_t and mach_vm_address_t */
kernel_trap(_kernelrpc_mach_vm_protect_trap,-14,7) /* 5 args, +2 for mach_vm_address_t and mach_vm_size_t */
kernel_trap(_kernelrpc_mach_vm_map_trap,-15,9)
kernel_trap(_kernelrpc_mach_port_allocate_trap,-16,3)
kernel_trap(_kernelrpc_mach_port_destroy_trap,-17,2)
kernel_trap(_kernelrpc_mach_port_deallocate_trap,-18,2)
kernel_trap(_kernelrpc_mach_port_mod_refs_trap,-19,4)
kernel_trap(_kernelrpc_mach_port_move_member_trap,-20,3)
kernel_trap(_kernelrpc_mach_port_insert_right_trap,-21,4)
kernel_trap(_kernelrpc_mach_port_insert_member_trap,-22,3)
kernel_trap(_kernelrpc_mach_port_extract_member_trap,-23,3)
kernel_trap(_kernelrpc_mach_port_construct_trap,-24,5)
kernel_trap(_kernelrpc_mach_port_destruct_trap,-25,5)
```

syscall_number = 0x10000000 -(-14) = 0x10000000 + 14 = 0x1000000E

```

1 //go:build darwin && amd64
2 // +build darwin,amd64
3
4 #include "go_asm.h"
5 #include "textflag.h"
6
7 // func taskSelfTrap() (ret uint32)
8 TEXT ·taskSelfTrap(SB), $8-0
9     PUSHQ AX
10    MOVL $(0x1000000+28), AX // task_self_trap
11    SYSCALL
12    MOVL AX, ret+0(FP)
13    POPQ AX
14    RET
15
16 // func vmProtect(targetTask uint32, address uintptr, size int, setMaximum, newProtection uint32) (ret uint32)
17 TEXT ·vmProtect(SB), $56-40
18     PUSHQ AX
19     PUSHQ DI
20     PUSHQ SI
21     PUSHQ DX
22     PUSHQ R10
23     PUSHQ R8
24     PUSHQ R9
25     MOVL targetTask+0(FP), DI
26     MOVQ address+8(FP), SI
27     MOVQ size+16(FP), DX
28     MOVL set_maximum+24(FP), R10
29     MOVL new_protection+28(FP), R8
30     XORQ R9, R9
31     MOVQ $(0x1000000+14), AX // mach_vm_protect
32     SYSCALL
33     MOVL AX, ret+32(FP)
34     POPQ R9
35     POPQ R8
36     POPQ R10
37     POPQ DX
38     POPQ SI
39     POPQ DI
40     POPQ AX
41     RET

```

Слезает
с зависимостей от СИ

function/protect_darwin_amd64.s

Мы больше не на СИ!

function/protect_darwin.go

```
1 //go:build darwin
2 // +build darwin
3
4 package function
5
6 import "fmt"
7
8 const (
9     vmProtRead uint32 = 1 << iota
10    vmProtWrite
11    vmProtExecute
12    _ // vmProtNoChange
13    vmProtCopy
14 )
15
16 //go:noescape
17 func taskSelfTrap() (id uint32)
18
19 //go:noescape
20 func vmProtect(targetTask uint32, address uintptr, size int, setMaximum, newProtection uint32) (ret uint32)
21
22 func setMemProtect(ptr uintptr, size int, allowWrite bool) {
23     prot := vmProtRead | vmProtExecute | vmProtCopy
24
25     if allowWrite {
26         prot |= vmProtWrite
27     }
28
29     if retCode := vmProtect(taskSelfTrap(), ptr, size, 0, prot); retCode != 0 {
30         panic(fmt.Sprintf("can't setMemProtect memory protect on macos, mach_vm_protect return ", retCode))
31     }
32 }
```

```
1  package main
2
3  import "fmt"
4
5  func main() {
6      var (
7          a = 5
8          b = 6
9      )
10
11     c := a + b
12
13     fmt.Println(c)
14 }
```

А теперь серьез
задача...

```
20047779@cab-wsm-0092006 demo % go build .
20047779@cab-wsm-0092006 demo % ls
demo      demo.go
20047779@cab-wsm-0092006 demo % ./demo
```

Знакомьтесь, Ида



IDA - The Interactive Disassembler

Version 8.2.230124 macOS x86_64 (64-bit address size)

(c) 2023 Hex-Rays SA

Freeware version with the following limitations:

1. Only for non-commercial use
2. Without technical support
3. Only supports x86/x64 code
4. Only PE/ELF/Mach-O files are supported
5. IDAPython is not available

For commercial use please acquire the full version

www.hex-rays.com

Перейдем сразу к делу

Load a new file

Load file /Users/20047779/Documents/work/grpctest/cmd/demo/demo as

Mach-O file (EXECUTE). X86_64 [macho64.dylib]

Binary file

Processor type (double-click to set)

Intel Pentium Pro (P6) with MMX	80686p
Intel Pentium protected with MMX	80586p
Intel Pentium real with MMX	80586r
MetaPC (disassemble all opcodes)	metapc

Analysis

Kernel options 1Kernel options 2Kernel options 3

Processor options

Options

☐ Loading options

☒ Fill segment gaps

☐ Load as code segment

☒ Create segments

☐ Create FLAT group

☐ Create imports segment

☐ Load resources

☒ Rename DLL entries

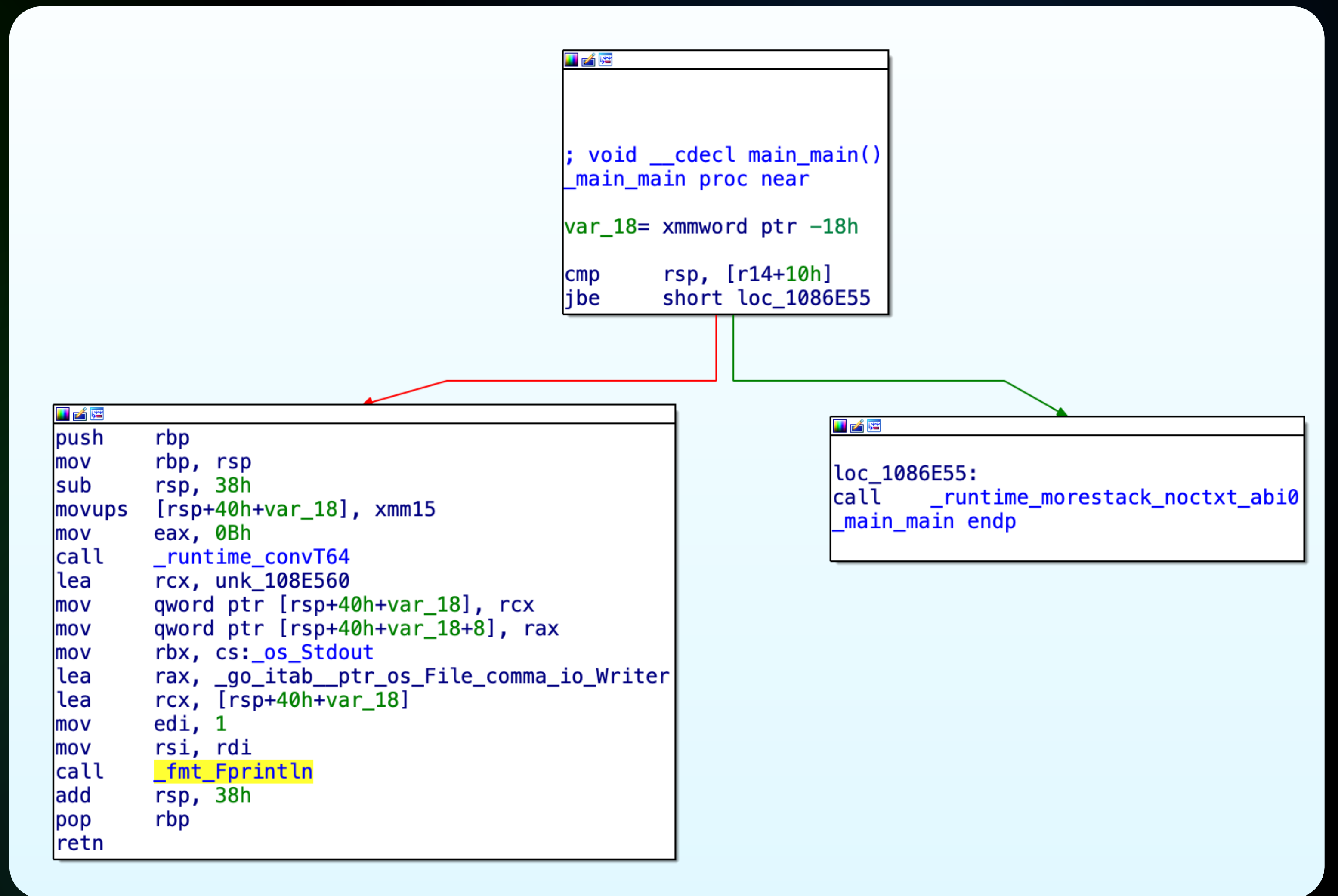
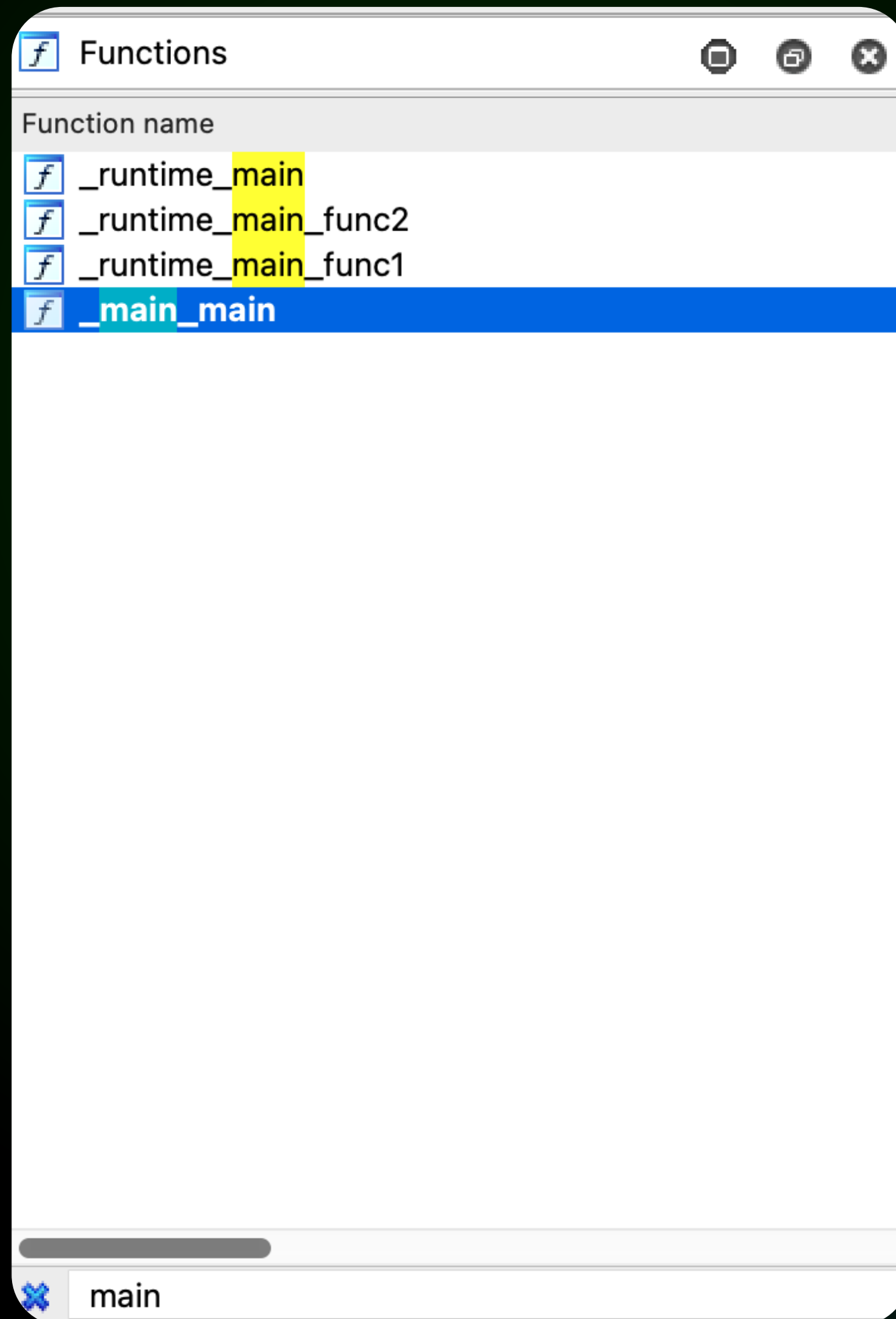
☐ Manual load

Help

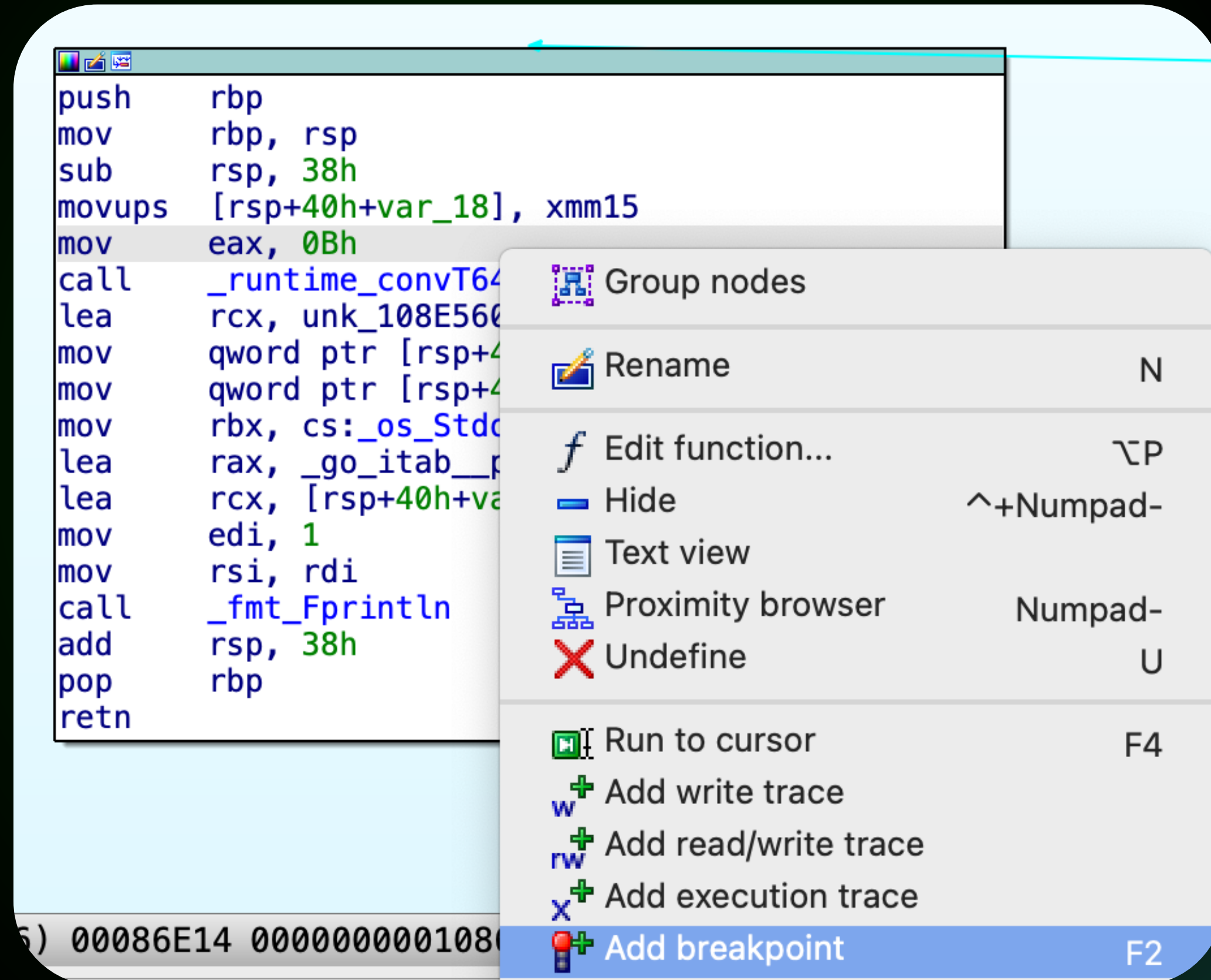
Cancel

OK

Знакомый код



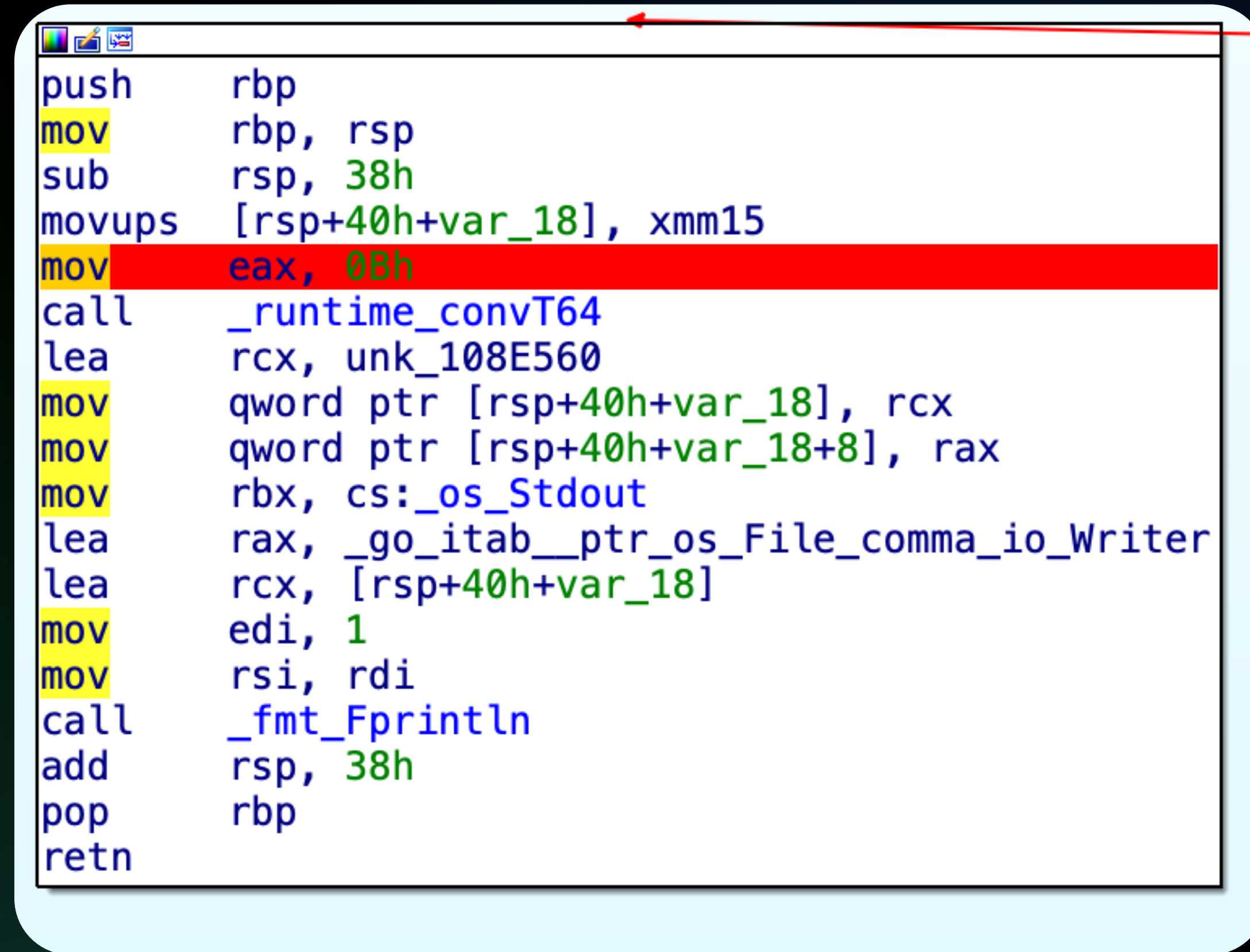
На следующей остановите



The screenshot shows a debugger window with assembly code. A context menu is open over the code, listing various actions. The assembly code includes instructions like `push rbp`, `mov rbp, rsp`, `sub rsp, 38h`, `movups [rsp+40h+var_18], xmm15`, `mov eax, 0Bh`, `call _runtime_convT64`, `lea rcx, unk_108E560`, `mov qword ptr [rsp+40h+var_18], rcx`, `mov qword ptr [rsp+40h+var_18+8], rax`, `mov rbx, cs:_os_Stdout`, `lea rax, _go_itab_ptr_os_File_comma_io_Writer`, `lea rcx, [rsp+40h+var_18]`, `mov edi, 1`, `mov rsi, rdi`, `call _fmt_Fprintln`, `add rsp, 38h`, `pop rbp`, and `retn`. The context menu options include: Group nodes, Rename (N), Edit function... (\P), Hide (^+Numpad-), Text view, Proximity browser (Numpad-), Undefine (U), Run to cursor (F4), Add write trace (w), Add read/write trace (rw), Add execution trace (x), and Add breakpoint (F2).

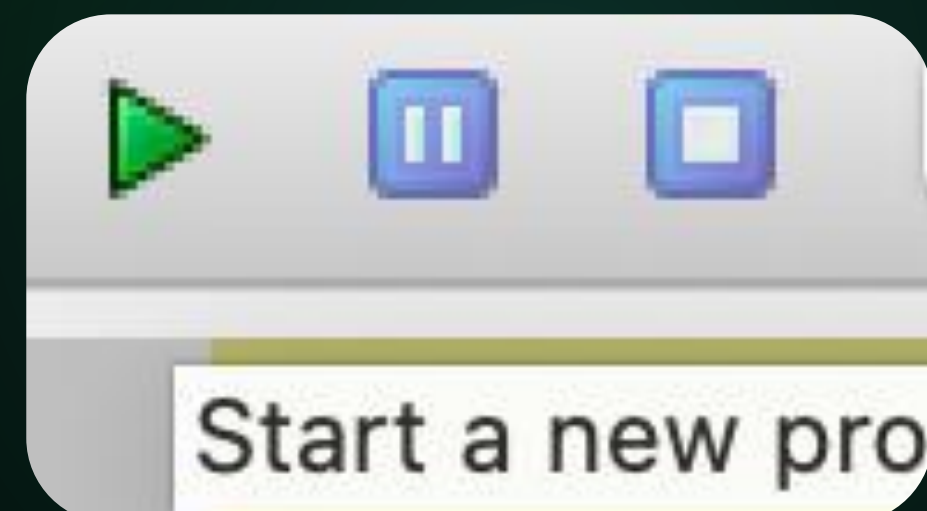
```
push    rbp
mov     rbp, rsp
sub     rsp, 38h
movups  [rsp+40h+var_18], xmm15
mov     eax, 0Bh
call    _runtime_convT64
lea     rcx, unk_108E560
mov     qword ptr [rsp+40h+var_18], rcx
mov     qword ptr [rsp+40h+var_18+8], rax
mov     rbx, cs:_os_Stdout
lea     rax, _go_itab_ptr_os_File_comma_io_Writer
lea     rcx, [rsp+40h+var_18]
mov     edi, 1
mov     rsi, rdi
call    _fmt_Fprintln
add     rsp, 38h
pop     rbp
retn
```

5) 00086E14 000000000108



The screenshot shows a debugger window with assembly code. The instruction `mov eax, 0Bh` is highlighted in red. The assembly code includes instructions like `push rbp`, `mov rbp, rsp`, `sub rsp, 38h`, `movups [rsp+40h+var_18], xmm15`, `mov eax, 0Bh`, `call _runtime_convT64`, `lea rcx, unk_108E560`, `mov qword ptr [rsp+40h+var_18], rcx`, `mov qword ptr [rsp+40h+var_18+8], rax`, `mov rbx, cs:_os_Stdout`, `lea rax, _go_itab_ptr_os_File_comma_io_Writer`, `lea rcx, [rsp+40h+var_18]`, `mov edi, 1`, `mov rsi, rdi`, `call _fmt_Fprintln`, `add rsp, 38h`, `pop rbp`, and `retn`.

```
push    rbp
mov     rbp, rsp
sub     rsp, 38h
movups  [rsp+40h+var_18], xmm15
mov     eax, 0Bh
call    _runtime_convT64
lea     rcx, unk_108E560
mov     qword ptr [rsp+40h+var_18], rcx
mov     qword ptr [rsp+40h+var_18+8], rax
mov     rbx, cs:_os_Stdout
lea     rax, _go_itab_ptr_os_File_comma_io_Writer
lea     rcx, [rsp+40h+var_18]
mov     edi, 1
mov     rsi, rdi
call    _fmt_Fprintln
add     rsp, 38h
pop     rbp
retn
```



Чего желаете?

Local Mac OS X debugger

Library function Regular function Instruction Data Unexplored External symbol Lumina function

Debug View Structures Enums

IDA View-RIP

```
__text:000000001086E04 jbe short loc_1086E55
__text:000000001086E06 push rbp
__text:000000001086E07 mov rbp, rsp
__text:000000001086E0A sub rsp, 38h
__text:000000001086E0E movups [rsp+40h+var_18], xmm15
__text:000000001086E14 mov eax, 08h
__text:000000001086E19 call _runtime_convT64
__text:000000001086E1E lea rcx, unk_108E560
__text:000000001086E25 mov qword ptr [rsp+40h+var_18], rcx
__text:000000001086E2A mov qword ptr [rsp+40h+var_18+8], rax
__text:000000001086E2F mov rbx, cs:_os_Stdout
__text:000000001086E36 lea rax, _go_itab_ptr_os_File_comma_io_Writer
__text:000000001086E3D lea rcx, [rsp+40h+var_18]
__text:000000001086E42 mov edi, 1
__text:000000001086E47 mov rsi, rdi
__text:000000001086E4A call _fmt_Fprintln
__text:000000001086E4F add rsp, 38h
__text:000000001086E53 pop rbp
__text:000000001086E54 retn
__text:000000001086E55 ;
__text:000000001086E55 loc_1086E55: ; CODE XREF: _main_main+4+j
__text:000000001086E55 call _runtime_morestack_noctxt_abi0
__text:000000001086E55 _main_main endp
__text:000000001086E55
```

00086E14 0000000001086E14: _main_main+14 (Synchronized with RIP, Hex View-1)

Hex View-1

```
000000001086DC0 54 00 00 00 E8 B7 C1 F7 FF EB 02 31 C0 48 83 C4 T.....1...
000000001086DD0 18 5D C3 48 89 44 24 08 48 89 5C 24 10 0F 1F 00 .]...D$.H.\$....
000000001086DE0 E8 1B 3A FD FF 48 8B 44 24 08 48 8B 5C 24 10 E9 .....H.D$.H.\$....
000000001086DF0 4C FF FF FF CC CC CC CC CC CC CC CC CC CC CC L.....
000000001086E00 49 3B 66 10 76 4F 55 48 89 E5 48 83 EC 38 44 0F I;f.vOUH.....
000000001086E10 11 7C 24 28 B8 0B 00 00 00 E8 42 25 F8 FF 48 8D .|$(.....H.
000000001086E20 0D 3B 77 00 00 48 89 4C 24 28 48 89 44 24 30 48 .;w..H.L$(H.D$0H
000000001086E30 8B 1D 92 C6 0A 00 48 8D 05 EB A2 03 00 48 8D 4C .....H.....H.L
000000001086E40 24 28 BF 01 00 00 00 48 89 FE E8 B1 AE FF FF 48 $(.....H..H
000000001086E50 83 C4 38 5D C3 E8 A6 39 FD FF EB A4 CC 00 00 00 ...]...9.....
000000001086E60 FF 25 AA 33 0A 00 FF 25 AC 33 0A 00 FF 25 AE 33 .%.3...%.3...%.3
000000001086E70 0A 00 FF 25 B0 33 0A 00 FF 25 B2 33 0A 00 FF 25 ...%.3...%.3...%.
000000001086E80 B4 33 0A 00 FF 25 B6 33 0A 00 FF 25 B8 33 0A 00 .3...%.3...%.3..
000000001086E90 FF 25 BA 33 0A 00 FF 25 BC 33 0A 00 FF 25 BE 33 .%.3...%.3...%.3
000000001086EA0 0A 00 FF 25 C0 33 0A 00 FF 25 C2 33 0A 00 FF 25 .%.....%.
000000001086EB0 C4 33 0A 00 FF 25 C6 33 0A 00 FF 25 C8 33 0A 00 .....%.
00086E14 0000000001086E14: _main_main+14 (Synchronized with RIP, IDA View-RIP)
```

Stack view

```
000000C000070EF8 0000000000000000
000000C000070F00 0000000000000060
000000C000070F08 0000000000000000
000000C000070F10 000000C000068058 debug151:000000C000068058
000000C000070F18 000000000112A4E8 __noptrdata:_reflect..inittask+8
000000C000070F20 0000000000000000
000000C000070F28 0000000000000000
000000C000070F30 000000C000070FD0 debug151:000000C000070FD0
000000C000070F38 000000000103245B _runtime_main+2BB
000000C000070F40 000000C000068000 debug151:000000C000068000
000000C000070F48 0000000000000000
000000C000070F50 0000000000000000
000000C000070F58 0000000000000000
000000C000070F60 0100000000000000
000000C000070F68 0000000000000001
000000C000070F70 000000000000000B
UNKNOWN 000000C000070EF8: debug151:000000C000070EF8 (Synchronized with RSP)
```

General registers

RAX	0000000001086E00	↪ _main_main	ID	0
RBX	000000000000000B	↪	VIP	0
RCX	000000C0000061A0	↪ debug151:000000C0000061A0	VIF	0
RDY	00000000010A93F0	↪ __rodata:off_10A93F0	AC	0
RSI	000000000112A801	↪ __noptrdata:_runtime..inittask+21	VM	0
RDI	0000000000000001	↪	RF	0
			NT	0

Modules

Path	Base
/Users/20047779/Documents/work/grpcmock/cmd/demo/demo	000000000
/usr/lib/libobjc.A.dylib	00007FF816

Threads

Decimal	Hex	State	Name
74199	121D7	Ready	demo
40423	9DE7	Ready	FFFFFFFFFFFFFFFF
70471	11347	Ready	FFFFFFFFFFFFFFFF
28915	70E2	Ready	FFFFFFFFFFFFFFFF

Output

```
FFFFFFFFFFFFFFFF: thread has started (tid=40423)
FFFFFFFFFFFFFFFF: thread has started (tid=70471)
FFFFFFFFFFFFFFFF: thread has started (tid=28915)
FFFFFFFFFFFFFFFF: thread has started (tid=90535)
1086E0E: got SIGURG signal (urgent condition present on socket) (exc.code 10, tid 74199)
```

ND

Я хочу...

RIP

__text:000000001086E0E movups [rsp+40h+var_18], xmm15

__text:000000001086E14 mov eax, 0Bh

__text:000000001086E19 call _runtime_convT64

__text:000000001086E1E lea rcx, unk_108E560

__text:000000001086E25 mov qword ptr [rsp+40h+var_18], rcx

__text:000000001086E2A mov qword ptr [rsp+40h+var_18+8], rax

__text:000000001086E2F mov rbx, cs:_os_Stdout

__text:000000001086E36 lea rax, _go_itab_ptr_os_File_comma_io_Writer

__text:000000001086E3D lea rcx, [rsp+40h+var_18]

__text:000000001086E42 mov edi, 1

__text:000000001086E47 mov rsi, rdi

__text:000000001086E4A call _fmt_Fprintln

__text:000000001086E4F add esp, 28h

00086E14 0000000001086E14: _main_main+14 (Synchronized with RIP, Hex View-1)

Hex View-1

0000000001086DC0	54 00 00 00 E8 B7 C1 F7 FF EB 02 31 C0 48 83 C4	T.....1....
0000000001086DD0	18 5D C3 48 89 44 24 08 48 89 5C 24 10 0F 1F 00	.]...D\$.H.\\$....
0000000001086DE0	E8 1B 3A FD FF 48 8B 44 24 08 48 8B 5C 24 10 E9H.D\$.H.\\$....
0000000001086DF0	4C FF FF FF CC CC CC CC CC CC CC CC CC CC CC	L.....
0000000001086E00	49 3B 66 10 76 4F 55 48 89 E5 48 83 EC 38 44 0F	I;f.vOUH.....
0000000001086E10	11 7C 24 28 B8 0B 00 00 00 E8 42 25 F8 FF 48 8D	. \$((.....H.
0000000001086E20	0D 3B 77 00 00 48 89 4C 24 28 48 89 44 24 30 48	.;w..H.L\$(H.D\$0H
0000000001086E30	8B 1D 92 C6 0A 00 48 8D 05 EB A2 03 00 48 8D 4CH.....H.L
0000000001086E40	24 28 BF 01 00 00 00 48 89 FE E8 B1 AE FF FF 48	\$(.....H..猥...H
0000000001086E50	83 C4 38 5D C3 E8 A6 39 FD FF EB A4 CC 00 00 00	...]...9.....
0000000001086E60	FF 25 14 22 04 00 FF 25 14 22 04 00 FF 25 14 22

Изменить ответ!

RIP

__text:000000001086E0E movups [rsp+40h+var_18], xmm15
__text:000000001086E14 mov eax, 0Bh
__text:000000001086E19 call runtime_convT64
00086E14 000000001086E14: _main_main+14 (Synchronized)

Hex View-1

0000000001086E00	49	3B	66	10	76	4F	55	48	89	E5	48	83	EC	38	44
0000000001086E10	11	7C	24	28	B8	0B	00	00	00	F8	42	25	F8	FF	48
0000000001086E20	0D	3B	77	00	00										
0000000001086E30	8B	1D	92	C6	0A										
0000000001086E40	24	28	BF	01	00										
0000000001086E50	83	C4	38	5D	C3										
0000000001086E60	FF	25	AA	33	0A										
0000000001086E70	0A	00	FF	25	B0										
0000000001086E80	B4	33	0A	00	FF										
0000000001086E90	FF	25	BA	33	0A										

0B 00 00 00

- Data format
- Columns
- Text
- Edit... F2
- Synchronize with

76	4F	55	48	89	E5	48	83	EC	38	4
B8	FF	00	00	00	E8	42	25	F8	FF	4
00	48									
0A	00									
00	00									
C3	E8									
0A	00									
B0	33	0A	00	FF	25	B2	33	0A	00	

FF 00 00 00

- Text
- Apply changes F2
- Synchronize with

001086E14 mov eax, 0FFh
001086E19 call _runtime_convT
001086E1E lea rcx, unk_108E5

```
20047779@cab-wsm-0092006 ~ % sudo ida64
Password:
qt.qpa.fonts: Populating font family aliases
st.
2023-10-03 17:30:34.317 ida64[95092:1810035]
11
11
qt.qpa.window: Window position QRect(-3,631 1
11
255
4294967295
11
255
```


Windows

Help

Load desktop...

Save desktop...

Delete desktop...

Reset desktop

Reset hidden messages...

Windows list

Next window

F6

Previous window

⬆ F6

Close window

⌘ F3

Focus command line

⬆ .

Output window

⌘ 0

IDA View-RIP

⌘ 1

Segments

⌘ 2

General registers

⌘ 3

Modules

⌘ 4

Threads

⌘ 5

Hex View-1

⌘ 6

Stack view

⌘ 7

Structures

⌘ 8

Enums

⌘ 9

Name	Start	End
HEADER	0000000001000000	0000000001001000
__text	0000000001001000	0000000001086E60
__symbol_stub1	0000000001086E60	0000000001086FA0
__rodata	0000000001086FA0	00000000010C2CC0
__typelink	00000000010C2CC0	00000000010C3280
__itablink	00000000010C3280	00000000010C32F0
demo: __gosymtab	00000000010C32F0	00000000010C3300
__gopclntab	00000000010C3300	000000000112A000
__go_buildinfo	000000000112A000	000000000112A210

А можно
навсегда?

```
20047779@cab-wsm-0092006 demo % gobjdump -h demo_100

demo_100:      file format mach-o-x86-64

Sections:
Idx Name          Size      VMA      LMA      File off  Algn
  0 .text          00085e5d  0000000001001000  0000000001001000  00001000  2**5
                CONTENTS, ALLOC, LOAD, CODE
  1 __TEXT.__symbol_stub1 00000126  0000000001086e60  0000000001086e60  00086e60  2**5
                CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 __TEXT.__rodata 0003bd17  0000000001086fa0  0000000001086fa0  00086fa0  2**5
                CONTENTS, ALLOC, LOAD, READONLY, CODE
  3 __TEXT.__typelink 000005a4  00000000010c2cc0  00000000010c2cc0  000c2cc0  2**5
```

OffsetInFile = VMAddress - TextVMASStart + TextFileOffset

= 0x1086E15 - 0x1000000

= 0x86E15

```

20047779@cab-wsm-0092006 demo % printf '\x64' | dd of=demo_100 bs=1 seek=$((0x86E15)) conv=notrunc
1+0 records in
1+0 records out
1 bytes transferred in 0.000292 secs (3425 bytes/sec)
20047779@cab-wsm-0092006 demo % ./demo_100
100
20047779@cab-wsm-0092006 demo % printf '\xFF' | dd of=demo_100 bs=1 seek=$((0x86E15)) conv=notrunc
1+0 records in
1+0 records out
1 bytes transferred in 0.000106 secs (9434 bytes/sec)
20047779@cab-wsm-0092006 demo % ./demo_100
255

```

dec 1,234,567,890

hex 00000000 49 96 02 D2

Порядок байт на архитектуре x86/x86_64 - little-endian

```

20047779@cab-wsm-0092006 demo % printf '\xD2\x02\x96\x49' | dd of=demo_100 bs=1 seek=$((0x86E15)) conv=notrunc
4+0 records in
4+0 records out
4 bytes transferred in 0.023235 secs (172 bytes/sec)
20047779@cab-wsm-0092006 demo % ./demo_100
1234567890

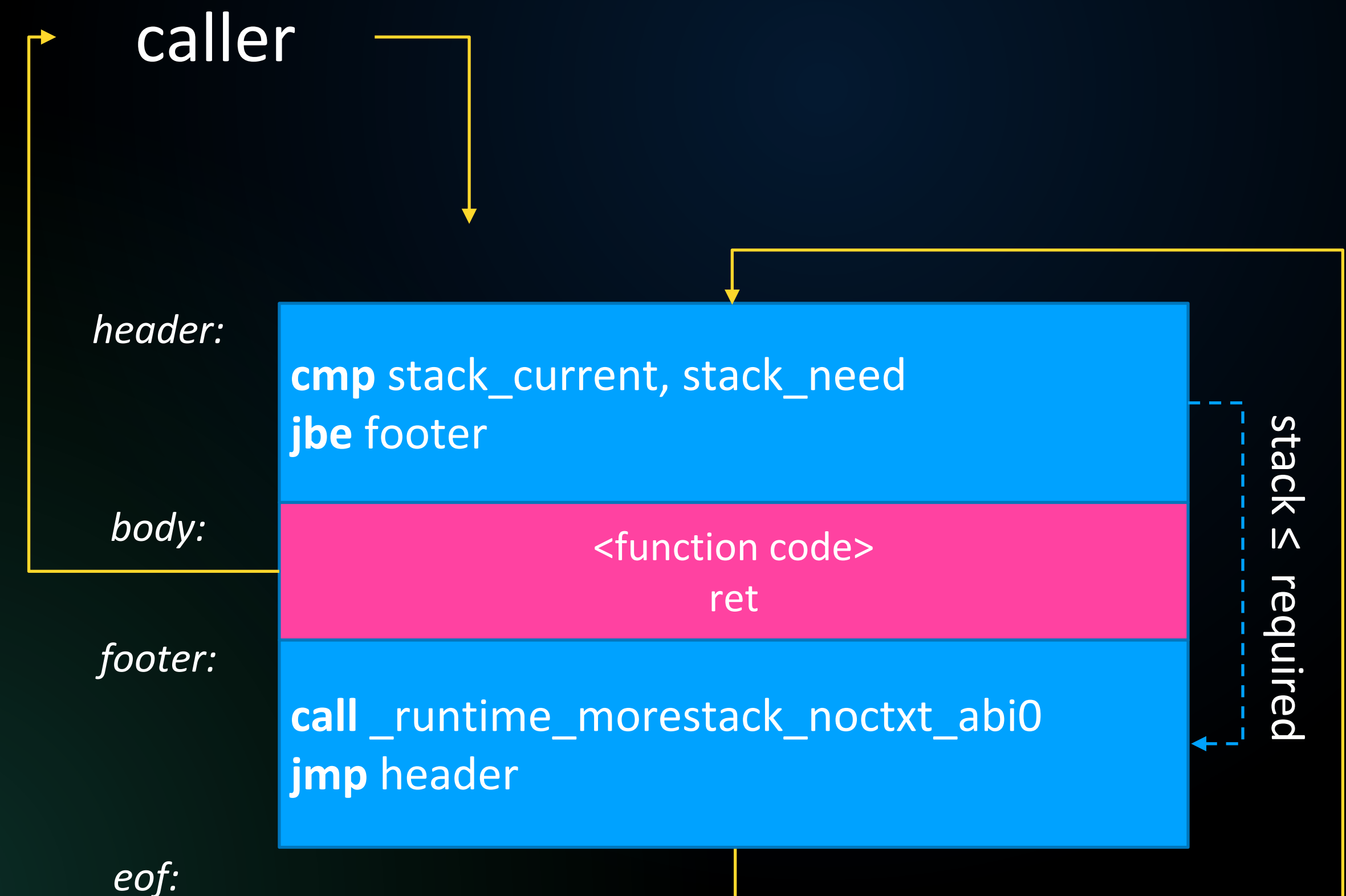
```


Можно!

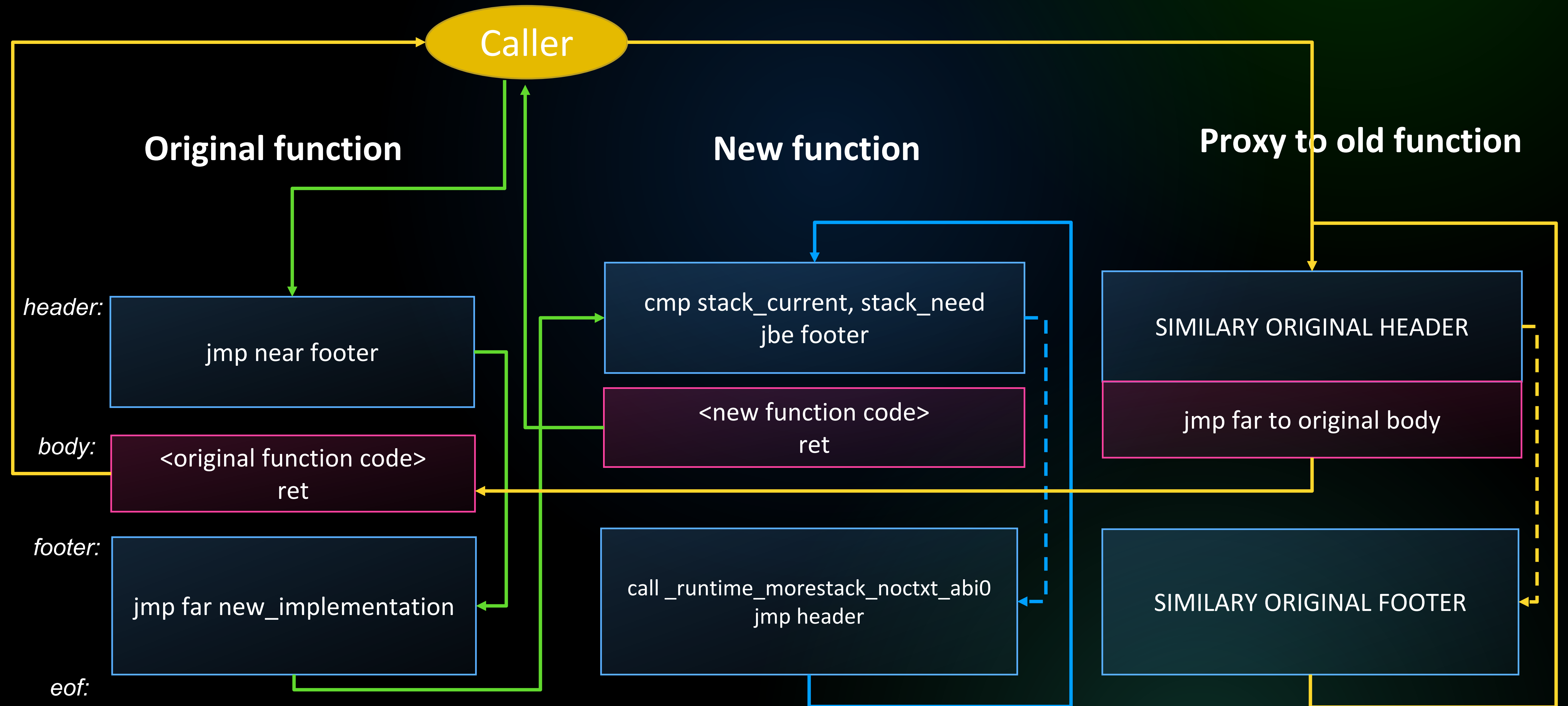
Исследуем устройство Go функции

Ида, подай микроскоп!

```
__text:0000000001086E00 ; void __cdecl main_main()
__text:0000000001086E00 _main_main      proc near                ; CODE XREF: _runtime
__text:0000000001086E00                                     ; sub_1086E5A+j
__text:0000000001086E00                                     ; DATA XREF: ...
__text:0000000001086E00 var_18          = xmmword ptr -18h
__text:0000000001086E00
__text:0000000001086E00 cmp     rsp, [r14+10h]
__text:0000000001086E04 jbe     short loc_1086E55
__text:0000000001086E06 push    rbp
__text:0000000001086E07 mov     rbp, rsp
__text:0000000001086E0A sub     rsp, 38h
__text:0000000001086E0E movups  [rsp+40h+var_18], xmm15
__text:0000000001086E14 mov     eax, 499602D2h
__text:0000000001086E19 call    _runtime_convT64
__text:0000000001086E1E lea     rcx, unk_108E560
__text:0000000001086E25 mov     qword ptr [rsp+40h+var_18], rcx
__text:0000000001086E2A mov     qword ptr [rsp+40h+var_18+8], rax
__text:0000000001086E2F mov     rbx, cs:_os_Stdout
__text:0000000001086E36 lea     rax, _go_itab_ptr_os_File_comma_i
__text:0000000001086E3D lea     rcx, [rsp+40h+var_18]
__text:0000000001086E42 mov     edi, 1
__text:0000000001086E47 mov     rsi, rdi
__text:0000000001086E4A call    _fmt_Fprintln
__text:0000000001086E4F add     rsp, 38h
__text:0000000001086E53 pop     rbp
__text:0000000001086E54 retn
__text:0000000001086E55 ; -----
__text:0000000001086E55 loc_1086E55:                ; CODE XREF: _main
__text:0000000001086E55 call    _runtime_morestack_noctxt_abi0
__text:0000000001086E55 _main_main      endp
__text:0000000001086E5A ; ===== S U B R O U T I N E =====
__text:0000000001086E5A
__text:0000000001086E5A sub_1086E5A      proc near
__text:0000000001086E5A jmp     short _main_main
__text:0000000001086E5A sub_1086E5A      endp
```



Я кажется придумал...



И даже реализовал

```
148 // Replace replace function to new implementation and returns proxy to old implementation.
149 func Replace[T any](tg, rp T, oldTo ...*T) *Patch[T] {
150     sec, err := inspectFunction(**(**uintptr)(unsafe.Pointer(&tg)))
151     if err != nil {
152         panic(fmt.Errorf("%w, maybe function already patched", err))
153     }
154
155     newFnc := *(*uintptr)(unsafe.Pointer(&rp))
156     oldFnc := *(*uintptr)(unsafe.Pointer(&tg))
157
158     beforeJBE := copyBytes(sec.Header, int(sec.JBEAddress-sec.Header))
159     beforeCall := copyBytes(sec.Footer, int(sec.CallAddress-sec.Footer))
160     afterCall := copyBytes(sec.CallAddress+uintptr(sec.CallInstruction.Len),
161         int(sec.JMPAddress-sec.CallAddress-uintptr(sec.CallInstruction.Len)))
162     rel, ok := sec.CallInstruction.Args[0].(x86asm.Rel)
163
164     if !ok {
165         panic("call of split_stack is not relative")
166     }
167
168     newHeader := make([]byte, sec.JBEAddress-sec.Header, sec.Body-sec.Header)
169     You, 2 weeks ago • add vendor
170     for i := range newHeader {
171         newHeader[i] = 0x90
172     }
173
174     switch sec.JBEInstruction.Len {
175     case 2:
176         newHeader = append(newHeader, 0xEB) // JMP SHORT
177     case 6:
178         newHeader = append(newHeader, 0x90, 0xE9) // JMP NEAR
179     default:
180         panic(fmt.Sprintf("unsupported JBE instruction %v", sec.JBEInstruction))
181     }
```

```
46 type sections struct {
47     Header      uintptr
48     Body        uintptr
49     Footer      uintptr
50     EOF         uintptr
51     JBEInstruction x86asm.Inst
52     JBEAddress    uintptr
53     CallAddress   uintptr
54     CallInstruction x86asm.Inst
55     JMPAddress    uintptr
56 }
```



```

89 func inspectFunction(ptr uintptr) (res sections, err error) {
90     ins, addr, err := findInstruction(ptr, 16, x86asm.JBE, x86asm.LEA, x86asm.CMP, x86asm.NOP, x86asm.INT)
91     if err != nil {
92         return res, err
93     }
94
95     res.JBEInstruction = *ins
96     res.JBEAddress = addr
97
98     rel, ok := ins.Args[0].(x86asm.Rel)
99     if !ok {
100         return res, ErrBadJBEEArg
101     }
102
103     res.Body = addr + uintptr(ins.Len)
104     res.Footer = res.Body + uintptr(rel)
105
106     ins, addr, err = findInstruction(res.Footer, 128, x86asm.CALL, x86asm.MOV, x86asm.CMP, x86asm.NOP, x86asm.INT)
107     if err != nil {
108         return res, err
109     }
110
111     res.CallAddress, res.CallInstruction = addr, *ins
112     ins, addr, err = findInstruction(addr+uintptr(ins.Len), 128, x86asm.JMP, x86asm.MOV, x86asm.CMP, x86asm.NOP, x86asm.INT)
113     if err != nil {
114         return res, err
115     }
116
117     if rel, ok := ins.Args[0].(x86asm.Rel); !ok || addr+uintptr(rel)+uintptr(ins.Len) != ptr {
118         return res, ErrBackAddressNoMatch
119     }
120
121     res.Header = ptr
122     res.EOF = addr + uintptr(ins.Len) + 1
123     res.JMPAddress = addr
124
125     return res, nil
126 }

```

Заглянем в тело функции

```

58 func findInstruction(ptr uintptr, size int, desired x86asm.Op, allowed ...x86asm.Op) (instr *x86asm.I
59     code := _asBytes(ptr, size)
60     pos := 0
61 s:
62     for {
63         if pos > (size - 10) {
64             ptr += uintptr(pos)
65             code, pos = _asBytes(ptr, size), 0
66         }
67
68         ins, err := x86asm.Decode(code[pos:], strconv.IntSize)
69         if err != nil {
70             return nil, 0, err
71         }
72
73         if ins.Op == desired {
74             return &ins, ptr + uintptr(pos), nil
75         }
76
77         for _, allow := range allowed {
78             if ins.Op == allow {
79                 pos += ins.Len
80
81                 continue s
82             }
83         }
84
85         return nil, 0, fmt.Errorf("%w %v excepted %v", ErrUnexpectedInstructionFound, ins, desired)
86     }
87 }

```

```

1 package function
2
3 import (
4     "encoding/binary"
5     "fmt"
6     "reflect"
7     "strconv"
8     "sync/atomic"
9     "unsafe"
10
11     "golang.org/x/arch/x86/x86asm"
12 )

```

Дизассемблер уж
есть

Собираем байт-код новой функции

```
183 newFooter := append(moveDXBytes(newFnc), 0xFF, 0x22) // jmp to new implementation
184 oldHeader := copyBytes(sec.Header, len(newHeader))
185 oldFooter := copyBytes(sec.Footer, len(newFooter))
186
187 jmpBytes := append(moveDXBytes(oldFnc), jmpFar(sec.Body)...)
188 callAddr := sec.CallAddress + uintptr(sec.CallInstruction.Len) + uintptr(rel)
189 callBytes := append(moveDXBytes(callAddr), 0xFF, 0xD2) // call morestack_ctx
190 size := len(beforeJBE) + len(beforeCall) + len(callBytes) + len(afterCall) + len(jmpBytes) + 4
191 originTramp := make([]byte, 0, size)
192 originTramp = append(originTramp, beforeJBE...)
193 originTramp = append(originTramp, 0x76, byte(len(jmpBytes))) // JBE SHORT
194 originTramp = append(originTramp, jmpBytes...) // JMP FAR TO ORIGINAL BODY
195 originTramp = append(originTramp, beforeCall...)
196 originTramp = append(originTramp, callBytes...)
197 originTramp = append(originTramp, afterCall...)
198 originTramp = append(originTramp, 0xEB, byte(-size)) // JMP SHORT
199
```



```

200 setMemProtect(uintptr(unsafe.Pointer(&originTramp[0])), size, true)
201
202 entrypoint := &originTramp[0]
203 fnc := &entrypoint
204 proxyFnc := *(*T)(unsafe.Pointer(&fnc))
205
206 for _, ot := range oldTo {
207     *ot = proxyFnc
208 }
209
210 doPatch(sec.Header, sec.Footer, sec.EOF, newHeader, newFooter)
211 You, 2 weeks ago • add vendor
212 res := &Patch[T]{
213     CallOld: proxyFnc,
214     CallNew: rp,
215     Unpatch: nil,
216     unpatched: 0,
217 }
218
219 h, f, e := sec.Header, sec.Footer, sec.EOF // copy for nolink to sec
220 res.Unpatch = func() { //nolint: mustpanic //never panic in run
221     if old := atomic.SwapInt32(&res.unpatched, 1); old != 0 {
222         panic(fmt.Errorf("function %s %w", New(tg).FullName(), ErrAlreadyPatched))
223     }
224     doPatch(h, f, e, oldHeader, oldFooter)
225 }
226
227
228 return res
229

```

```

// Replace replace function to new implementation and
func Replace[T any](tg, rp T, oldTo ...*T) *Patch[T]

```

И скажем компилятору, что
это функция!

```

120 func Example_logs() {
121     log.SetOutput(os.Stdout)
122     log.Println("one")
123
124     time.Sleep(time.Second)
125
126     if runtime.GOMAXPROCS(0) > 1 {
127         log.Println("two")
128         time.Sleep(time.Second)
129     }
130
131     log.Println("three")
132     // output:
133     // 2023/10/03 16:19:48 one
134     // 2023/10/03 16:19:48 two
135     // 2023/10/03 16:19:48 three
136 }

```

--- FAIL: Example_logs (2.00s)

got:

```

2023/10/03 16:20:50 one
2023/10/03 16:20:51 two
2023/10/03 16:20:52 three

```

want:

```

2023/10/03 16:19:48 one
2023/10/03 16:19:48 two
2023/10/03 16:19:48 three

```

FAIL

Process 85906 has exited with status 1

```

121 func Example_logs() {
122     log.SetOutput(os.Stdout)
123     log.Println("one")
124
125     time.Sleep(time.Second)
126
127     if runtime.GOMAXPROCS(0) > 1 {
128         log.Println("two")
129         time.Sleep(time.Second)
130     }
131
132     log.Println("three")
133     // output:
134     // 2023/10/03 16:19:48 one
135     // 2023/10/03 16:19:48 two
136     // 2023/10/03 16:19:48 three
137 }
138 func init() {
139     function.Replace(time.Now, func() time.Time {
140         return time.Date(2023, 10, 03, 16, 19, 48, 0, time.Local)
141     })
142 }

```

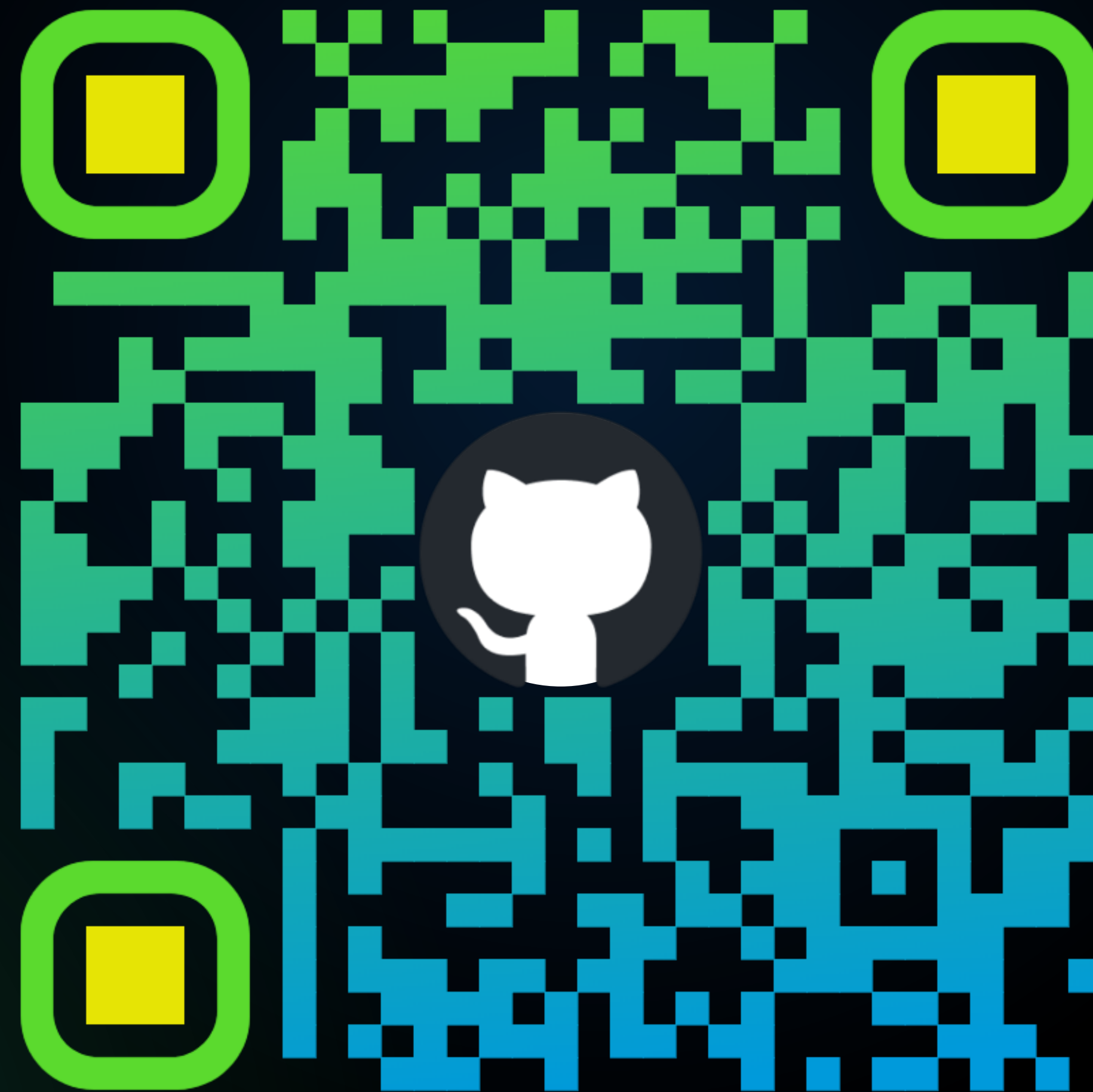
Остановим
время!

PASS

Process 88466 has exited with status 0

Подведем итоги

- Остановили время в тестах
- Обошли все ограничения ООП
- Изменили работу драйвера БД без правки его исходников
- Изменили скомпилированный код без исходников
- Выполнили массив байт как машинный код
- Осознали важность средств защиты от дизассемблирования и отладки нашего кода



Есть вопросы?



Егор Лазаренков

@ellzr



Спасибо