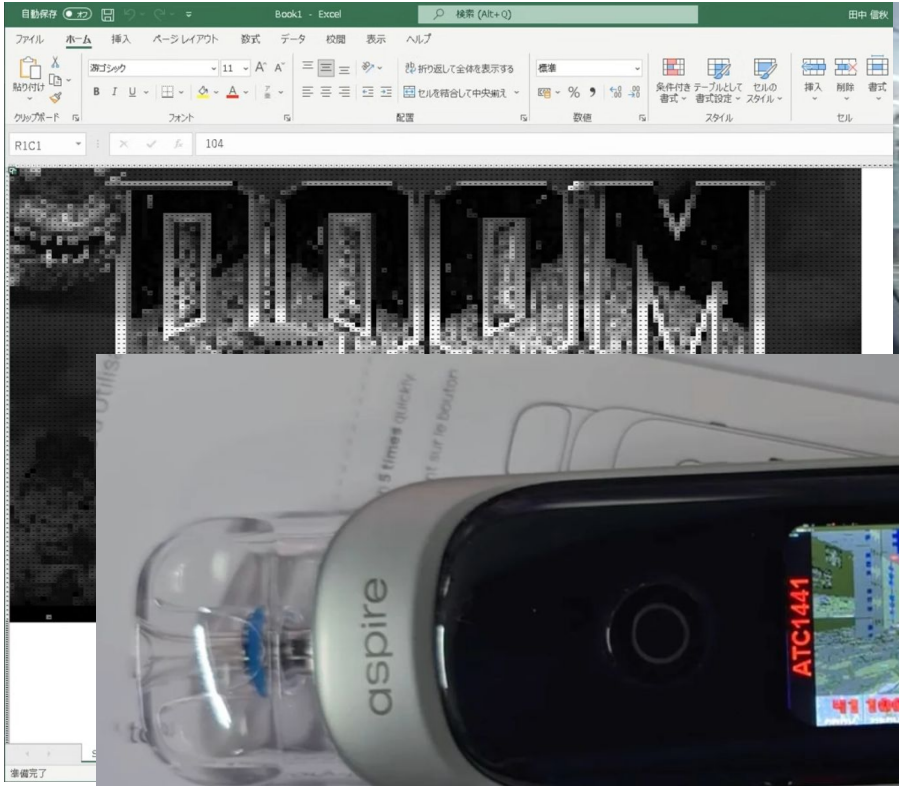
Yes, it runs **Doom**:

Building a **WebAssembly** VM in **Go** From Scratch


自動保存
Book1 - Excel
検索 (Alt+Q)
田中 信秋
ファイル
ホーム
挿入
ページ レイアウト
数式
データ
校閲
表示
ヘルプ
游ゴシック
11
折り返して全体を表示する
標準
セルを結合して中央揃え
条件付き書式
テーブルとして書式設定
セルのスタイル
挿入
削除
書式
クリップボード
フォント
配置
数値
スタイル
セル
R1C1
104
準備完了

DOOM




aspire
ATC1441
Serial S
Built-in factory-tune
tor
PLL supports CPU oper
depende
comparison,
support an encoder interface

$ whoami

Tarcisio Ferraz

Sr Software Engineer at Chainlink

Currently working at Chainlink Runtime Environment (CRE)



Overview

- Intro to WebAssembly
- Parsing
- Runtime
- Demo
- Optimizations

Building a WebAssembly app

```
1 package main
2
3 func main() {
4     println("Hello, World!")
5 }
```

```
go build -o app.wasm main.go
```

WebAssembly (Wasm)

- Run compiled code in the browser
- Reuse the same core logic everywhere
- Secure sandboxing and lightweight isolation

What is a Virtual Machine (VM)?

A VM is a computer implemented in software.

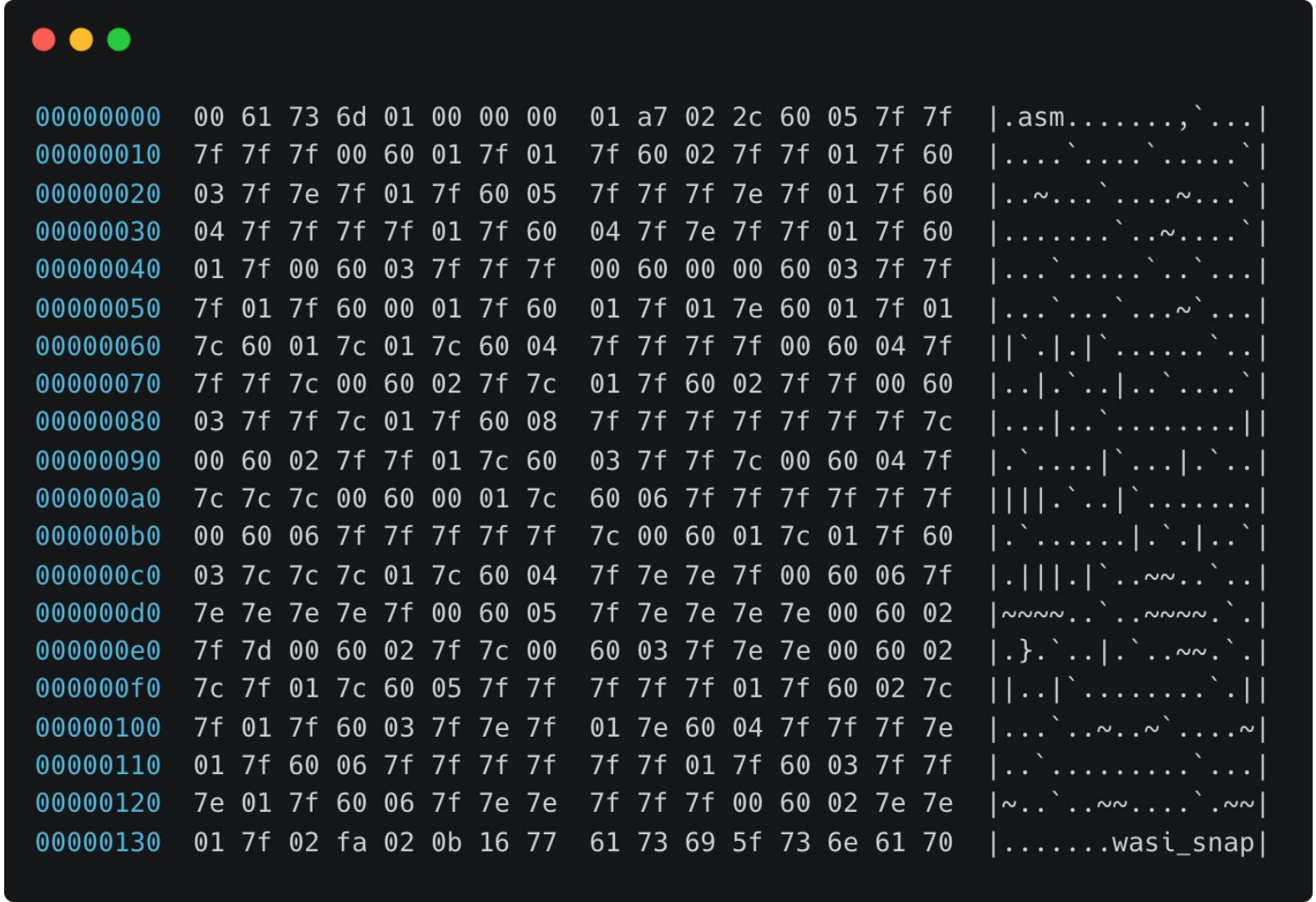


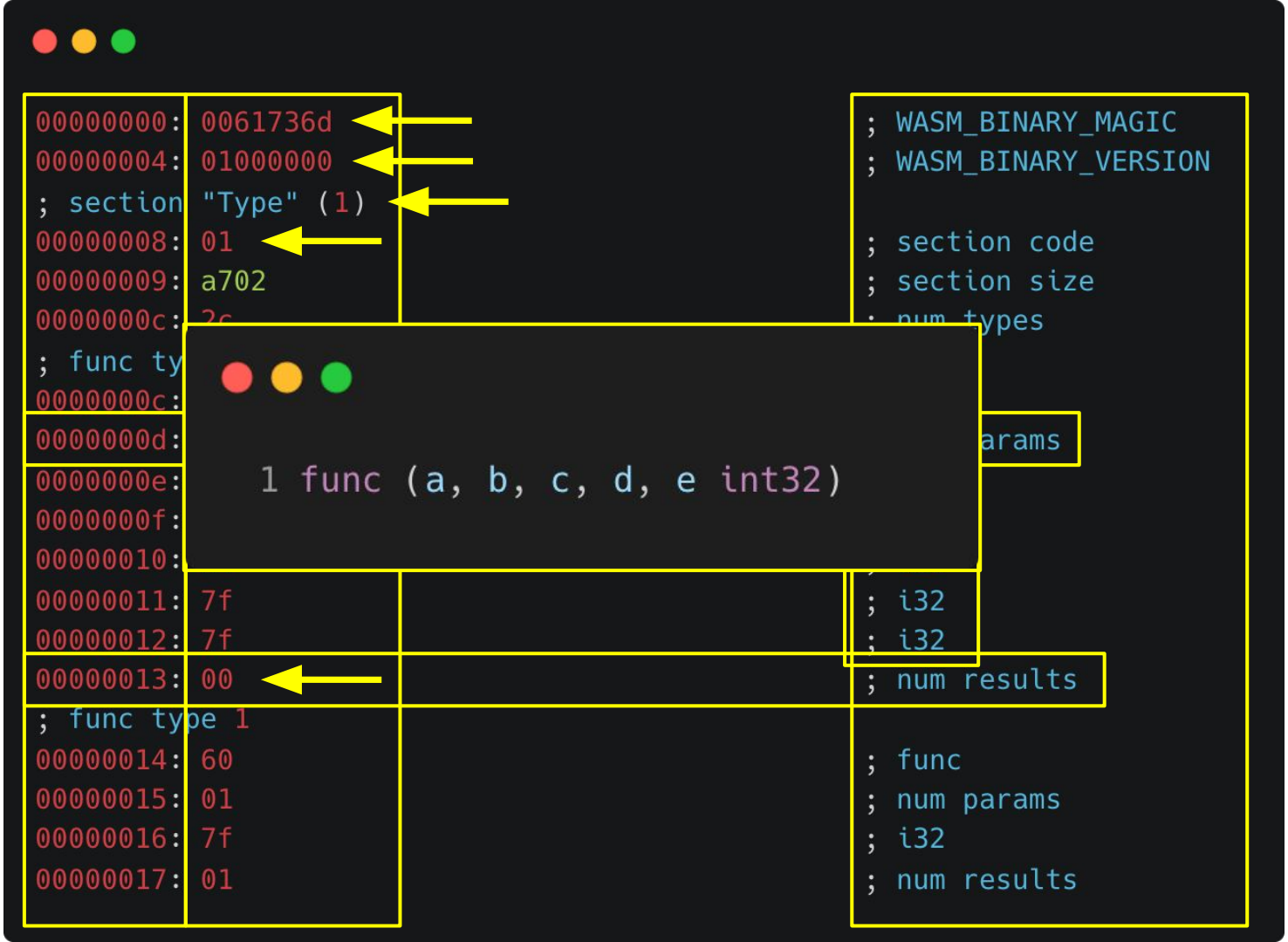
WebAssembly Specification

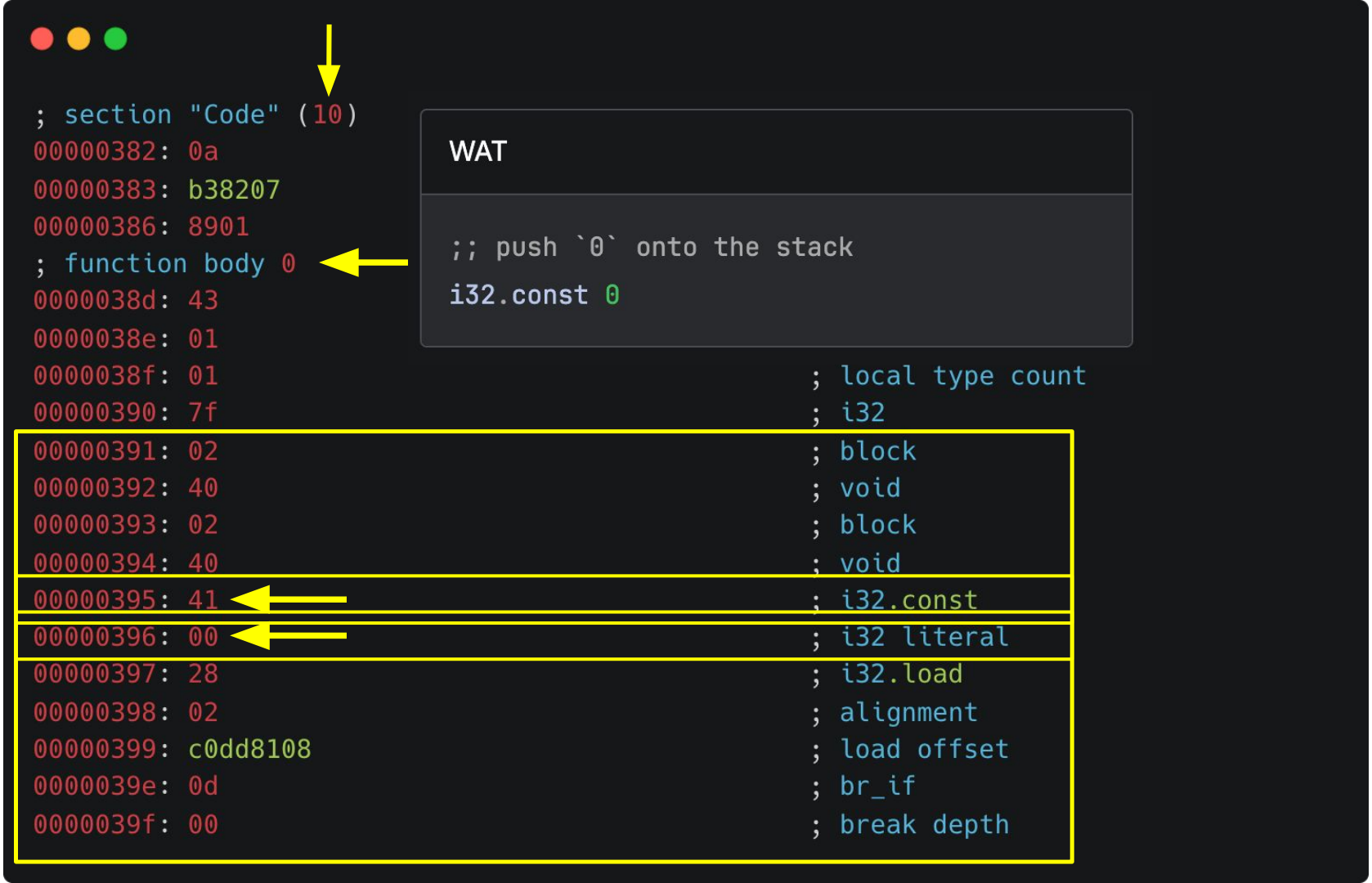
Release 3.0 (2026-02-21)


GOOM


hexdump -C app.wasm


00000000 00 61 73 6d 01 00 00 00 01 a7 02 2c 60 05 7f 7f |.asm.......,`...|
00000010 7f 7f 7f 00 60 01 7f 01 7f 60 02 7f 7f 01 7f 60 |....`....`.....`|
00000020 03 7f 7e 7f 01 7f 60 05 7f 7f 7f 7e 7f 01 7f 60 |..~...`....~...`|
00000030 04 7f 7f 7f 7f 01 7f 60 04 7f 7e 7f 7f 01 7f 60 |.......`..~....`|
00000040 01 7f 00 60 03 7f 7f 7f 00 60 00 00 60 03 7f 7f |...`.....`..`...|
00000050 7f 01 7f 60 00 01 7f 60 01 7f 01 7e 60 01 7f 01 |...`...`...~`...|
00000060 7c 60 01 7c 01 7c 60 04 7f 7f 7f 7f 00 60 04 7f ||`.|.|`......`..|
00000070 7f 7f 7c 00 60 02 7f 7c 01 7f 60 02 7f 7f 00 60 |..|.`..|..`....`|
00000080 03 7f 7f 7c 01 7f 60 08 7f 7f 7f 7f 7f 7f 7f 7c |...|..`........||
00000090 00 60 02 7f 7f 01 7c 60 03 7f 7f 7c 00 60 04 7f |.`....|`...|.`..|
000000a0 7c 7c 7c 00 60 00 01 7c 60 06 7f 7f 7f 7f 7f 7f ||||.`..|`.......|
000000b0 00 60 06 7f 7f 7f 7f 7f 7c 00 60 01 7c 01 7f 60 |.`......|.`.|..`|
000000c0 03 7c 7c 7c 01 7c 60 04 7f 7e 7e 7f 00 60 06 7f |.|||.|`..~~..`..|
000000d0 7e 7e 7e 7e 7f 00 60 05 7f 7e 7e 7e 7e 00 60 02 |~~~~..`..~~~~.`.|
000000e0 7f 7d 00 60 02 7f 7c 00 60 03 7f 7e 7e 00 60 02 |.}.`..|.`..~~.`.|
000000f0 7c 7f 01 7c 60 05 7f 7f 7f 7f 7f 01 7f 60 02 7c ||..|`........`.||
00000100 7f 01 7f 60 03 7f 7e 7f 01 7e 60 04 7f 7f 7f 7e |...`..~..~`....~|
00000110 01 7f 60 06 7f 7f 7f 7f 7f 7f 01 7f 60 03 7f 7f |..`.........`...|
00000120 7e 01 7f 60 06 7f 7e 7e 7f 7f 7f 00 60 02 7e 7e |~..`..~~....`.~~|
00000130 01 7f 02 fa 02 0b 16 77 61 73 69 5f 73 6e 61 70 |.......wasi_snap|


00000000: 0061736d ; WASM_BINARY_MAGIC
00000004: 01000000 ; WASM_BINARY_VERSION
; section "Type" (1)
00000008: 01 ; section code
00000009: a702 ; section size
; num types
; func ty
0000000c:
0000000d: arams
0000000e:
0000000f:
00000010:
1 func (a, b, c, d, e int32)
00000011: 7f ; i32
00000012: 7f ; i32
00000013: 00 ; num results
; func type 1
00000014: 60 ; func
00000015: 01 ; num params
00000016: 7f ; i32
00000017: 01 ; num results

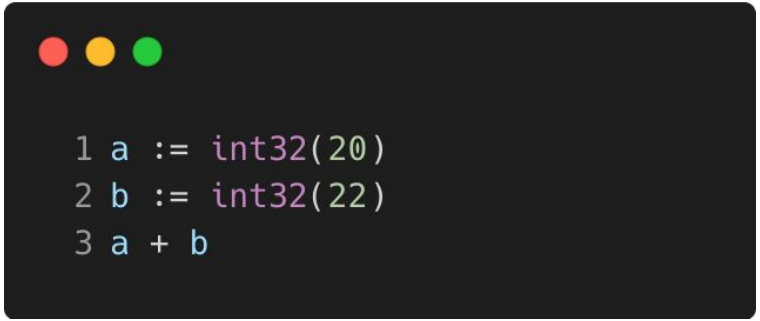

; section "Code" (10)
00000382: 0a
00000383: b38207
00000386: 8901
; function body 0
0000038d: 43
0000038e: 01
0000038f: 01 ; local type count
00000390: 7f ; i32
00000391: 02 ; block
00000392: 40 ; void
00000393: 02 ; block
00000394: 40 ; void
00000395: 41 ; i32.const
00000396: 00 ; i32 literal
00000397: 28 ; i32.load
00000398: 02 ; alignment
00000399: c0dd8108 ; load offset
0000039e: 0d ; br_if
0000039f: 00 ; break depth
WAT
;; push `0` onto the stack
i32.const 0

RUNTIME

A Go program that:

- Opens the binary file
- Parses all sections
- Finds the starting function
- Starts running the instructions of the function

```
body := []byte{
    0x41, // i32.const
    0x14, // i32 literal
    0x41, // i32.const
    0x16, // i32 literal
    0x6a, // i32.add
}
```



```
a := int32(20)
b := int32(22)
a + b
```

How to draw an owl

1.



Draw some circles

2.



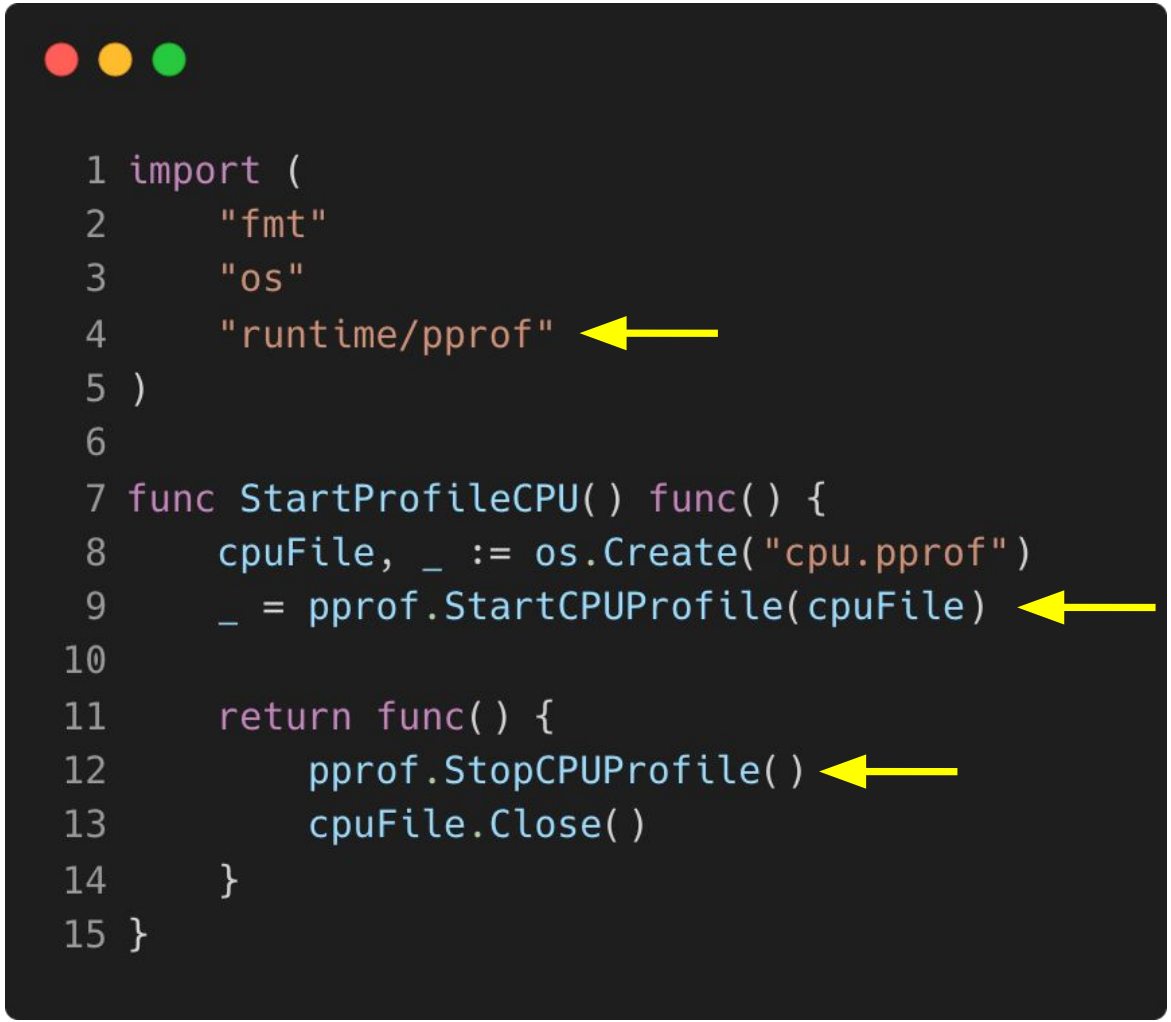
Draw the rest of the owl

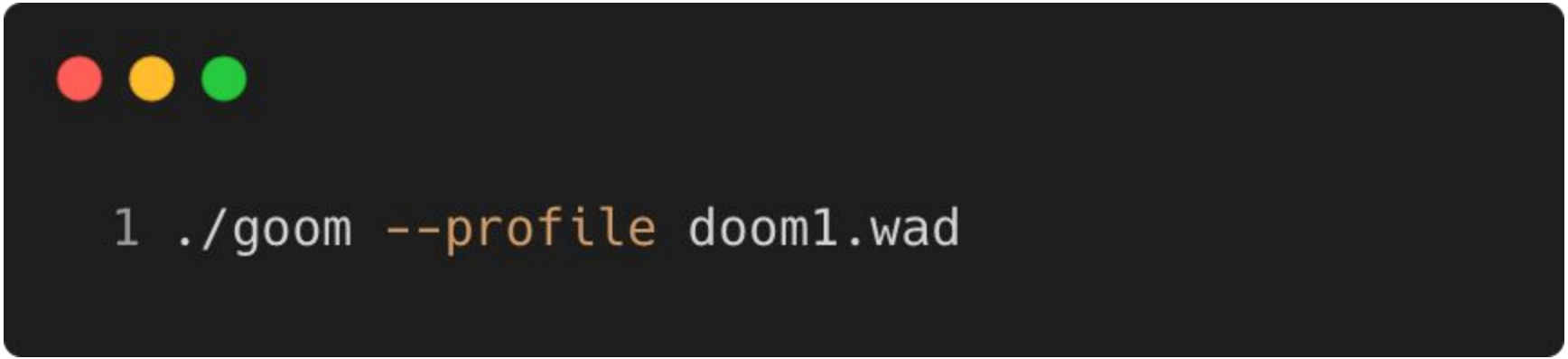
~~Demo Time~~

Demon Time

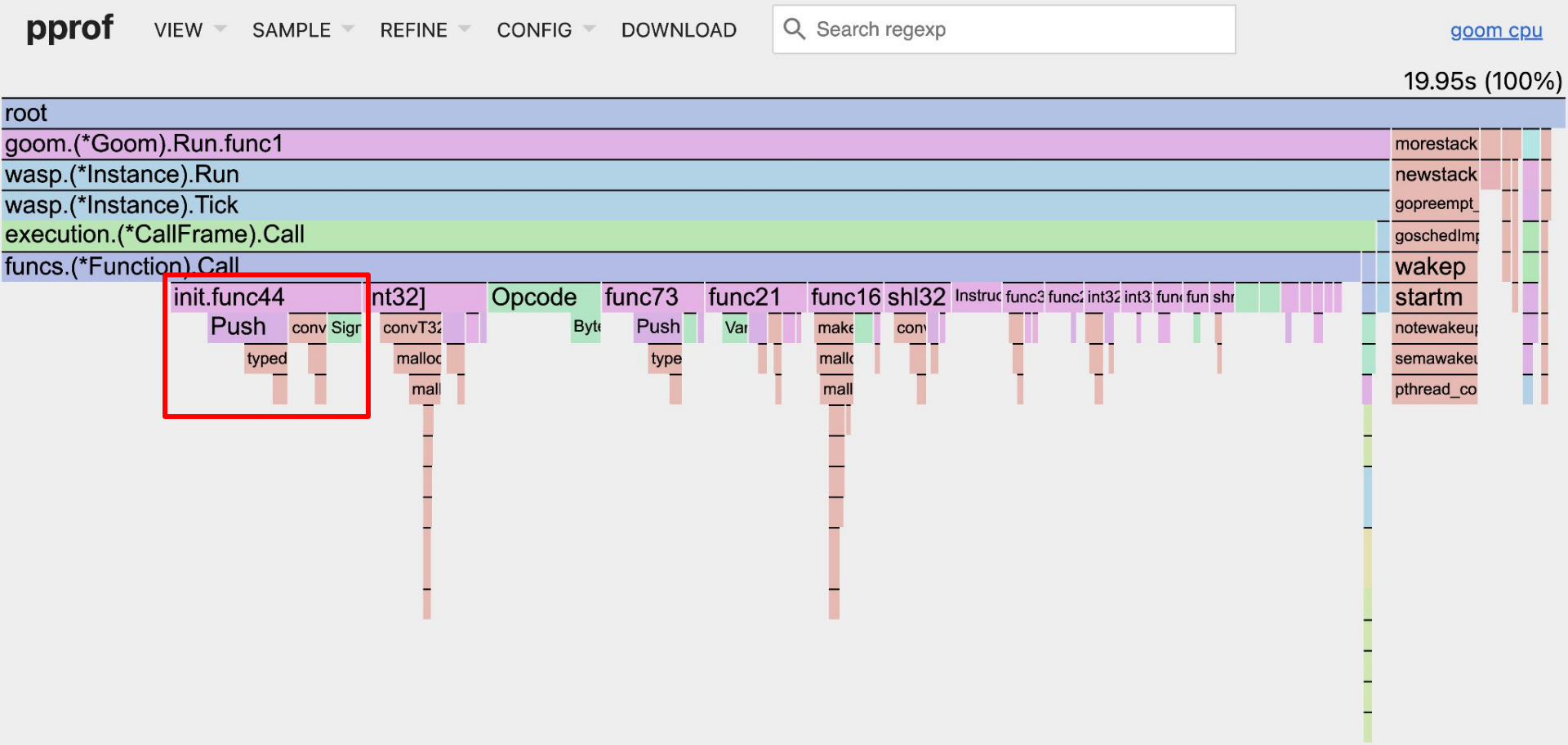

DOOM
id

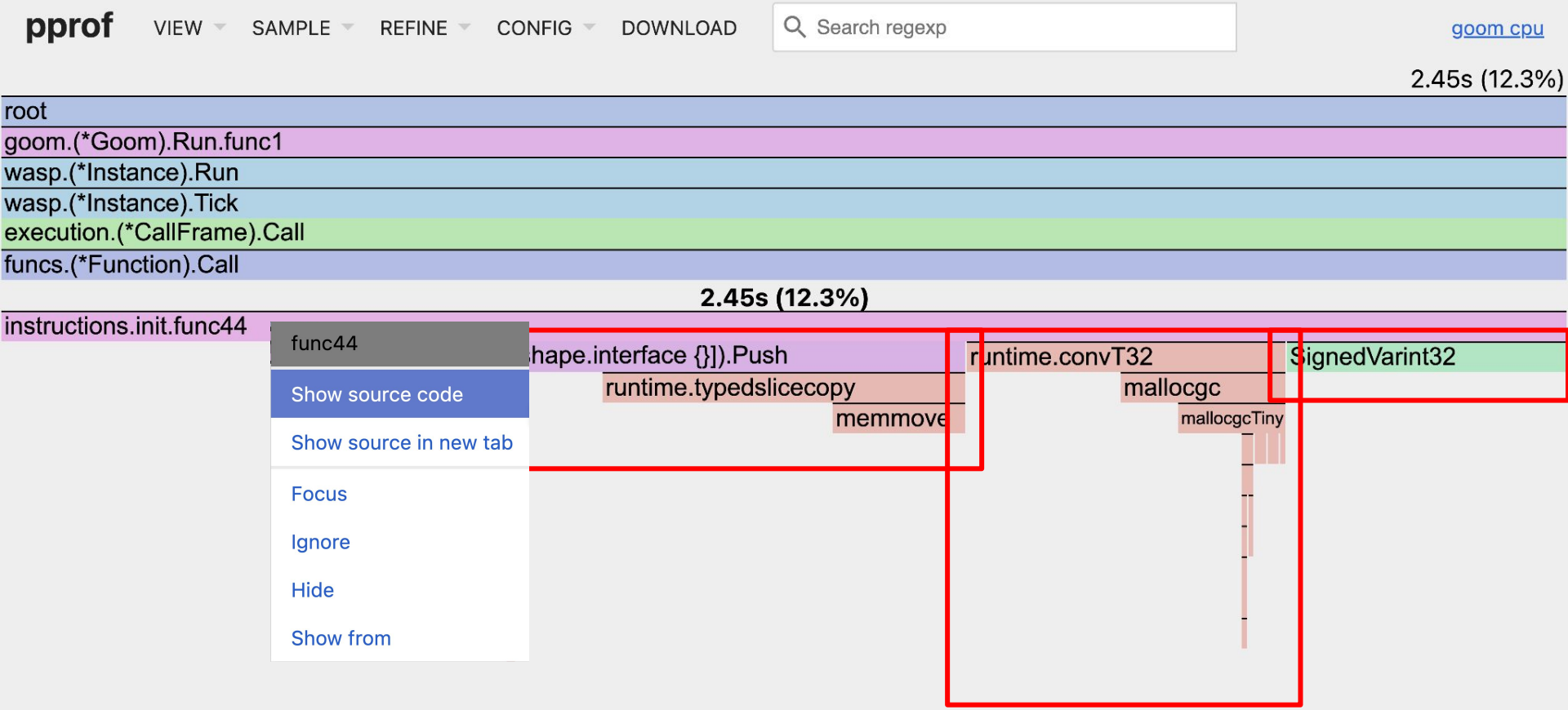
OPTIMIZATIONS


import (
"fmt"
"os"
"runtime/pprof"
)
func StartProfileCPU() func() {
cpuFile, _ := os.Create("cpu.pprof")
_ = pprof.StartCPUProfile(cpuFile)
return func() {
pprof.StopCPUProfile()
cpuFile.Close()
}
}


1 ./goom --profile doom1.wad


1 go tool pprof -http=:8080 cpu.pprof

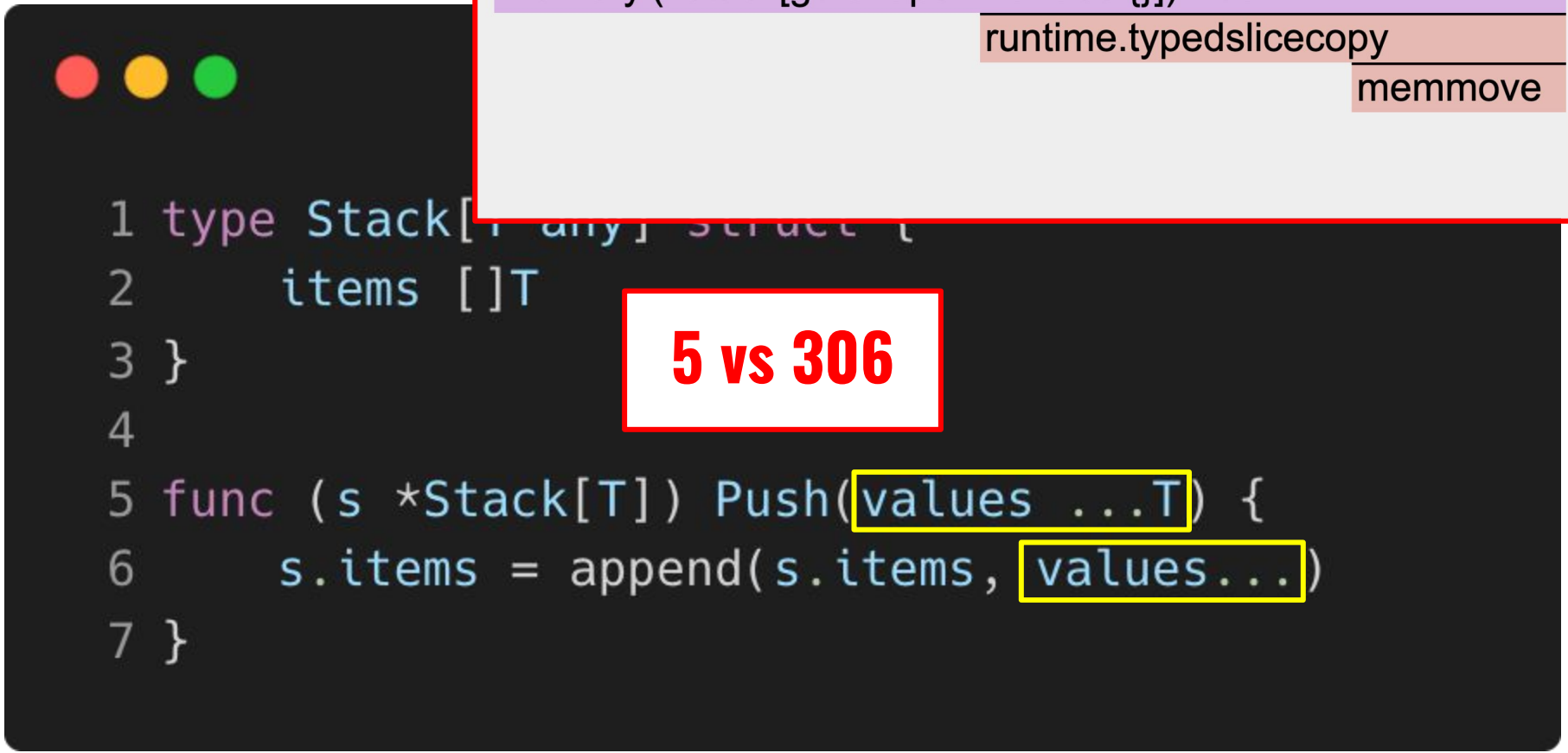

pprof
VIEW
SAMPLE
REFINE
CONFIG
DOWNLOAD
Search regexp
goom cpu
19.95s (100%)
root
goom.(*Goom).Run.func1
wasp.(*Instance).Run
wasp.(*Instance).Tick
execution.(*CallFrame).Call
funcs.(*Function).Call
init.func44
Push
conv
Sign
typed
nt32]
convT32
malloc
mall
Opcode
Byte
func73
Push
type
func21
Var
func16
make
mallo
mall
shl32
conv
Instruc
func3
func2
int32
int3
fun
fun
shr
morestack
newstack
gopreempt_
goschedImp
wakep
startm
notewakeup
semawakeu
pthread_co

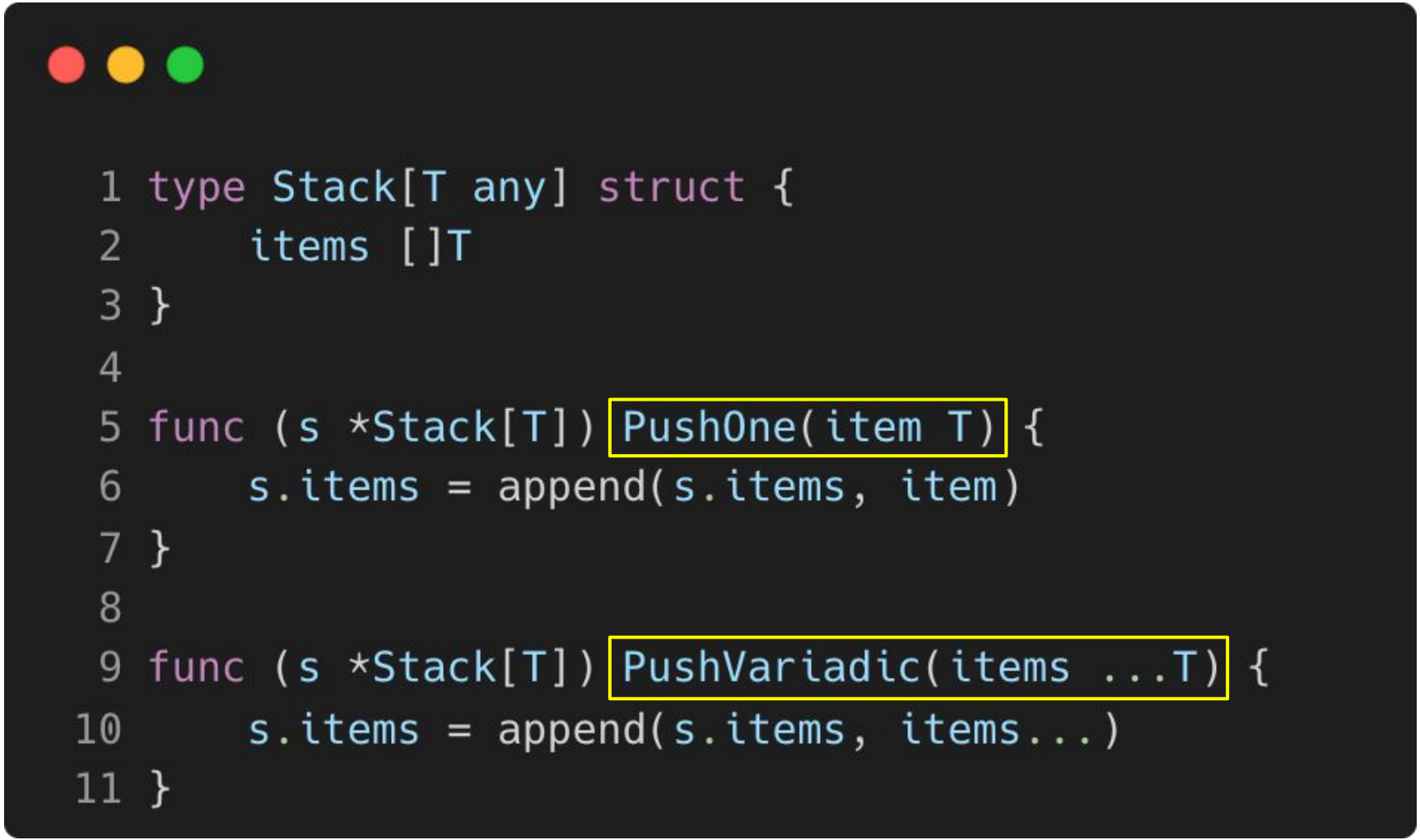

pprof
VIEW
SAMPLE
REFINE
CONFIG
DOWNLOAD
Search regexp
goom cpu
2.45s (12.3%)
root
goom.(*Goom).Run.func1
wasp.(*Instance).Run
wasp.(*Instance).Tick
execution.(*CallFrame).Call
funcs.(*Function).Call
2.45s (12.3%)
instructions.init.func44
func44
Show source code
Show source in new tab
Focus
Ignore
Hide
Show from
hape.interface {}]).Push
runtime.typedslicecopy
memmove
runtime.convT32
mallocgc
mallocgcTiny
SignedVarint32

github.com/tarcisiozf/wasp/internal/instructions.init.func44

/Users/tarcisiozotelliferraz/go/pkg/mod/github.com/tarcisiozf/wasp@v0.0.0-20260307162049-b3e550d4646f/internal/instructions/numeric.go

```
Total:      470ms      1.41s (flat, cum)  7.07%
   683         .          .                }
   684         .          .                return 0, fmt.Errorf("%w, got %T", ErrTypeMismatch, item)
   685         .          .            }
   686         .          .
   687         .          .            var (
   688      390ms      390ms                I32Const = addInstruction(opcodes.I32Const, func(ctx *execution.Context) error {
   689         .      440ms                    value := ctx.Body.SignedVarint32()
   690       40ms      540ms                    ctx.Stack.Push(value)
   691       40ms       40ms                    return nil
   692         .          .                })
   693         .          .                I64Const = addInstruction(opcodes.I64Const, func(ctx *execution.Context) error {
   694         .          .                    value := ctx.Body.SignedVarint64()
   695         .          .                    ctx.Stack.Push(value)
   696         .          .                    return nil
```


memory.(*Stack[go.shape.interface {}]).Push
runtime.typedslicecopy
memmove
type Stack[
items []T
}
func (s *Stack[T]) Push(values ...T) {
s.items = append(s.items, values...)
}
5 vs 306



```go
type Stack[T any] struct {
    items []T
}

func (s *Stack[T]) PushOne(item T) {
    s.items = append(s.items, item)
}

func (s *Stack[T]) PushVariadic(items ...T) {
    s.items = append(s.items, items...)
}
```



```go
type Stack[T any] struct {
    items []T
}

var stack Stack[any]
```

goom cpu

2.11s (9.8%)

```
I32Const = addInstruction(opcodes.I32Const, func(ctx *execution.Context) error {
        value := ctx.Body.SignedVarint32()
        ctx.Stack.Push(value)
        return nil
})
```

instructions.init.func44

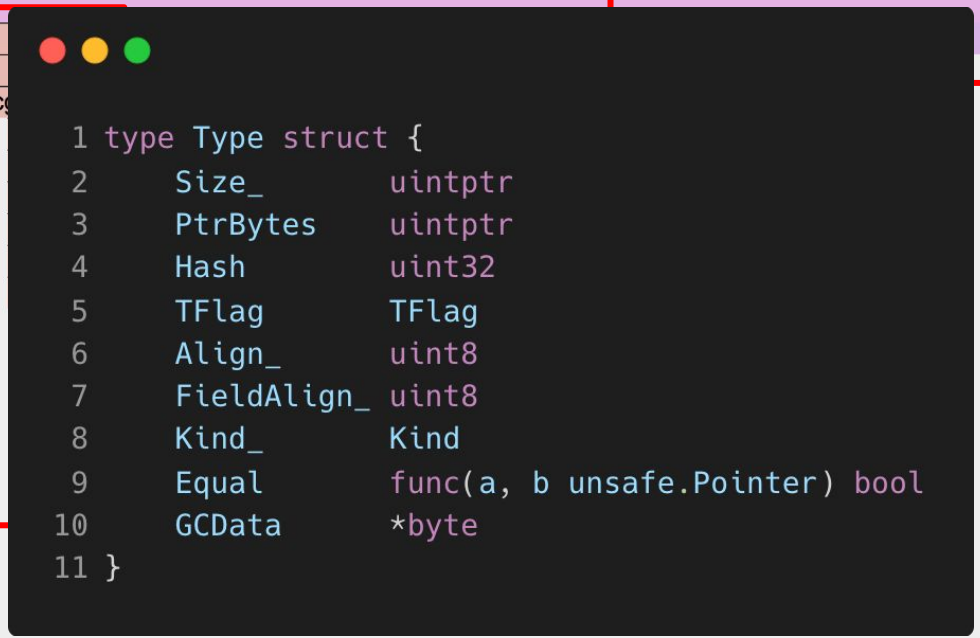
runtime.convT32

runtime.mallocgc

runtime.mallocg



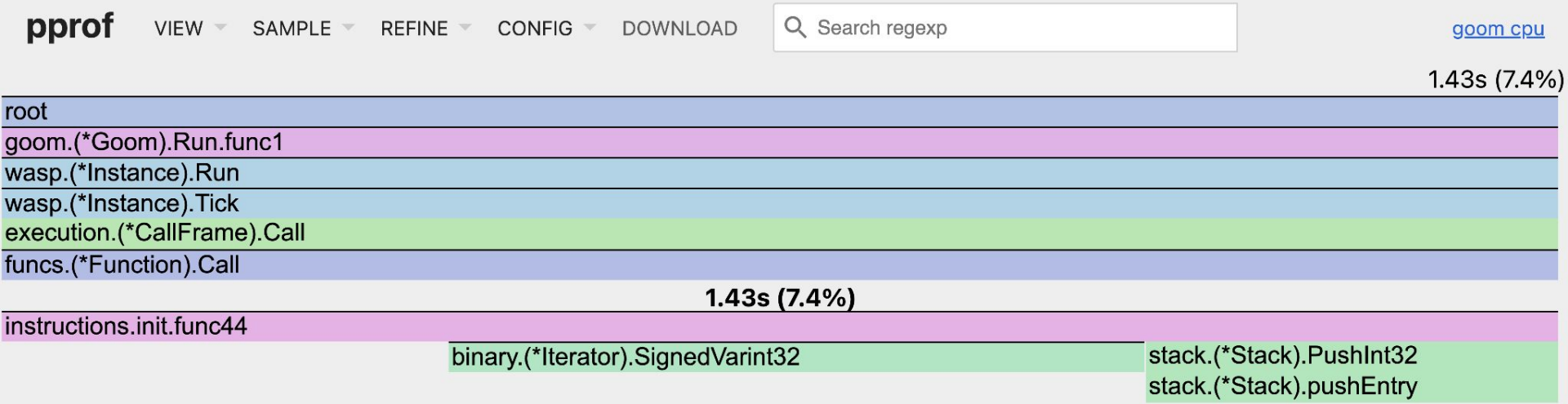
```go
type eface struct {
    _type *abi.Type
    data  unsafe.Pointer
}
```



```go
type Type struct {
    Size_       uintptr
    PtrBytes    uintptr
    Hash        uint32
    TFlag       TFlag
    Align_      uint8
    FieldAlign_ uint8
    Kind_       Kind
    Equal       func(a, b unsafe.Pointer) bool
    GCData      *byte
}
```

```
type TypedStack struct {
    Stack[Value]
}
```

```
type Value struct {
    // 0 - int32 | 1 - int64
    kind  byte
    value uint64
}
```


pprof
VIEW
SAMPLE
REFINE
CONFIG
DOWNLOAD
Search regexp
goom cpu
1.43s (7.4%)
root
goom.(*Goom).Run.func1
wasp.(*Instance).Run
wasp.(*Instance).Tick
execution.(*CallFrame).Call
funcs.(*Function).Call
1.43s (7.4%)
instructions.init.func44
binary.(*Iterator).SignedVarint32
stack.(*Stack).PushInt32
stack.(*Stack).pushEntry


Frame Count: 13, FPS: 6.81
Frame Count: 14, FPS: 6.90
Frame Count: 15, FPS: 6.97
Frame Count: 16, FPS: 7.04
Frame Count: 17, FPS: 7.10
Frame Count: 18, FPS: 7.15
Frame Count: 19, FPS: 7.21
Frame Count: 20, FPS: 7.25
Frame Count: 21, FPS: 7.30
Frame Count: 22, FPS: 7.34
Frame Count: 23, FPS: 7.38
Frame Count: 24, FPS: 7.42
Frame Count: 25, FPS: 7.44
Frame Count: 26, FPS: 7.46
Frame Count: 27, FPS: 7.48
Frame Count: 28, FPS: 7.51
Frame Count: 29, FPS: 7.53
Frame Count: 30, FPS: 7.55
Frame Count: 31, FPS: 7.57
Frame Count: 32, FPS: 7.59
Frame Count: 33, FPS: 7.61
Frame Count: 34, FPS: 7.63
DOOM

Takeaways

- Don't be afraid of specs!
- Go is practical and has great tooling
- VMs have the same shared concept

Thank you!

WE ARE HIRING!







tarcisiozf/goom

Grazie!



tarcisiozf/goom

