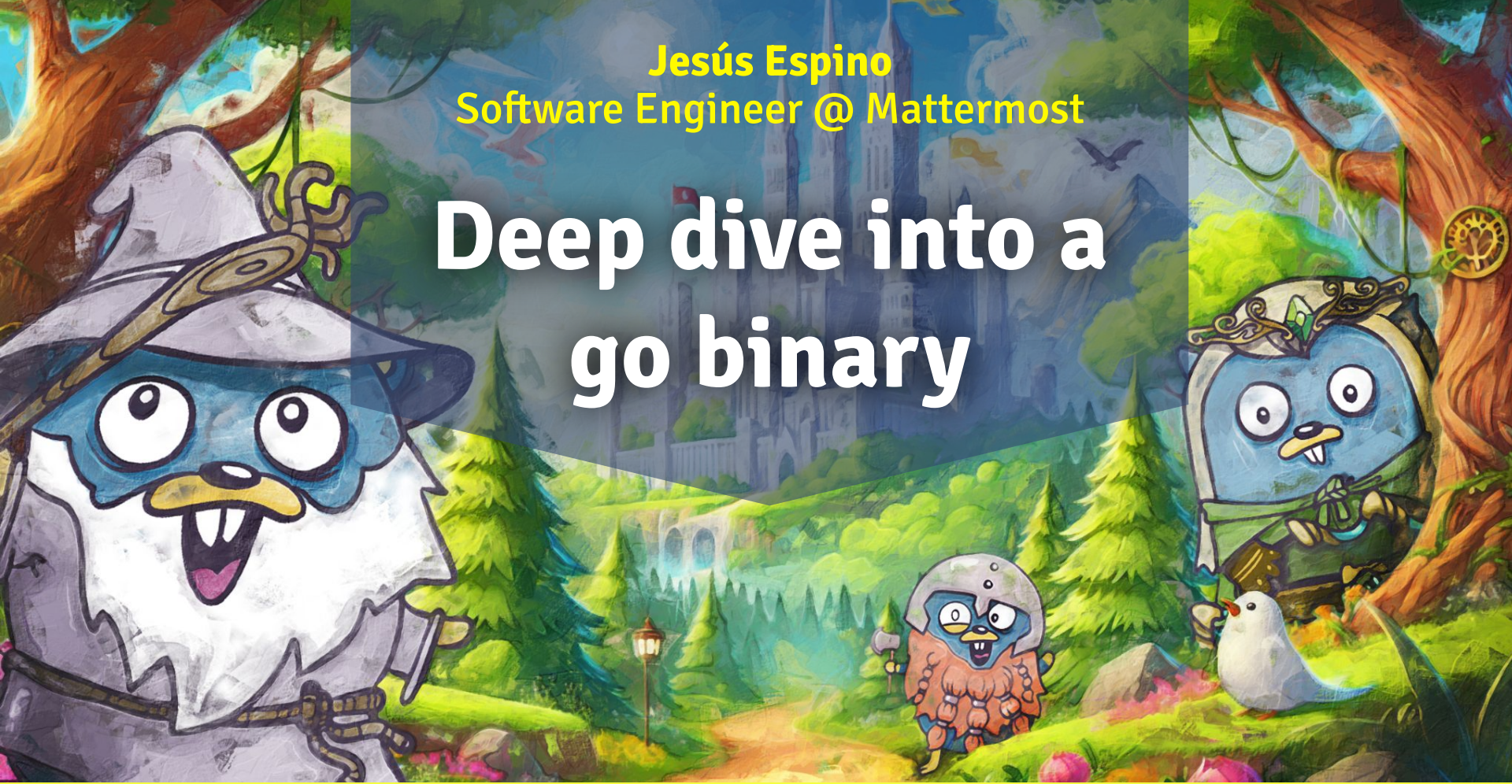
Jesús Espino
Software Engineer @ Mattermost

Deep dive into a go binary




Introduction

My go file

```
package main

import ("fmt")

func main() {
    fmt.Println("Hello, World!")
}
```

Compilation

The go binary

1852 Kilobytes

The go binary

- An ELF file
- Contains:
  - your program
  - your dependencies
  - the go runtime
  - debug information
  - metadata

The ELF format

- A set of headers
- A list of sections
- A list of segments

ELF: Sections

- Used for tools (linker, debugger...)
- Continuous blocks on memory

ELF: Sections

There are 23 section headers, starting at offset 0x190:

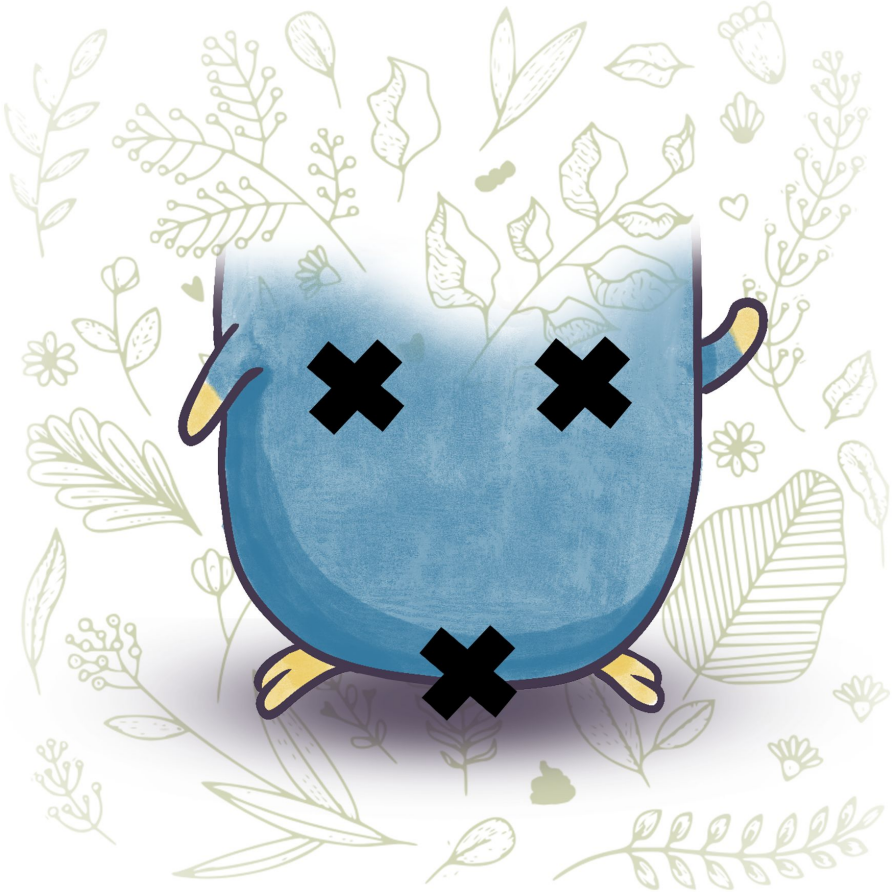
Section Headers:

[Nr]	Name	Type	Address	Off	Size	ES	Flg	Lk	Inf	Al
[0]		NULL	0000000000000000	000000	000000	00		0	0	0
[1]	.text	PROGBITS	0000000000401000	001000	07f9da	00	AX	0	0	32
[2]	.rodata	PROGBITS	0000000000481000	081000	03d2da	00	A	0	0	32
[3]	.typelink	PROGBITS	00000000004be2e0	0be2e0	000590	00	A	0	0	32
[4]	.itablink	PROGBITS	00000000004be880	0be880	000058	00	A	0	0	32
[5]	.gosymtab	PROGBITS	00000000004be8d8	0be8d8	000000	00	A	0	0	1
[6]	.gopclntab	PROGBITS	00000000004be8e0	0be8e0	0643f8	00	A	0	0	32
[7]	.go.buildinfo	PROGBITS	0000000000523000	123000	000130	00	WA	0	0	16
[8]	.noptrdata	PROGBITS	0000000000523140	123140	0054a0	00	WA	0	0	32
[9]	.data	PROGBITS	00000000005285e0	1285e0	004250	00	WA	0	0	32
[10]	.bss	NOBITS	000000000052c840	12c840	05fb30	00	WA	0	0	32
[11]	.noptrbss	NOBITS	000000000058c380	18c380	003a40	00	WA	0	0	32
[12]	.debug_abbrev	PROGBITS	0000000000000000	12d000	000135	00	C	0	0	1
[13]	.debug_line	PROGBITS	0000000000000000	12d135	01f619	00	C	0	0	1
[14]	.debug_frame	PROGBITS	0000000000000000	14c74e	00624f	00	C	0	0	1
[15]	.debug_gdb_scripts	PROGBITS	0000000000000000	15299d	00002a	00		0	0	1
[16]	.debug_info	PROGBITS	0000000000000000	1529c7	03d936	00	C	0	0	1
[17]	.debug_loc	PROGBITS	0000000000000000	1902fd	01c4d6	00	C	0	0	1
[18]	.debug_ranges	PROGBITS	0000000000000000	1ac7d3	00b132	00	C	0	0	1
[19]	.note.go.buildid	NOTE	0000000000400f9c	000f9c	000064	00	A	0	0	4
[20]	.shstrtab	STRTAB	0000000000000000	1cea85	000107	00		0	0	1
[21]	.symtab	SYMTAB	0000000000000000	1b7908	00bca0	18		22	116	8
[22]	.strtab	STRTAB	0000000000000000	1c35a8	00b4dd	00		0	0	1

readelf -SW main

ELF: Segments

- Used at runtime
- Continuous blocks of memory

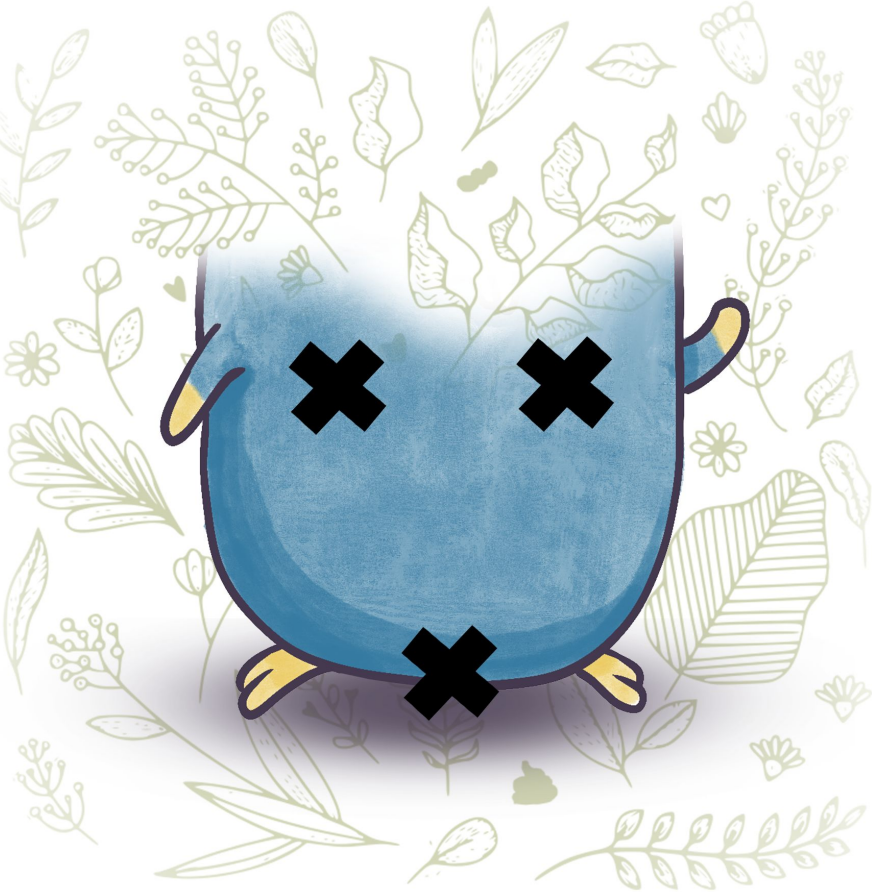


ELF: Segments

```
Elf file type is EXEC (Executable file)
Entry point 0x463940
There are 6 program headers, starting at offset 64

Program Headers:
  Type           Offset   VirtAddr           PhysAddr           FileSiz  MemSiz   Flg Align
  PHDR           0x000040 0x0000000000400040 0x0000000000400040 0x000150 0x000150 R   0x1000
  NOTE           0x000f9c 0x0000000000400f9c 0x0000000000400f9c 0x000064 0x000064 R   0x4
  LOAD           0x000000 0x0000000000400000 0x0000000000400000 0x0809da 0x0809da R E 0x1000
  LOAD           0x081000 0x0000000000481000 0x0000000000481000 0x0a1cd8 0x0a1cd8 R   0x1000
  LOAD           0x123000 0x0000000000523000 0x0000000000523000 0x009840 0x06cdc0 RW  0x1000
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RW  0x8

 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.go.buildid
   02     .text .note.go.buildid
   03     .rodata .typelink .itablink .gosymtab .gopclntab
   04     .go.buildinfo .noptrdata .data .bss .noptrbss
   05
```



readelf -W --segment main

The elf Headers

- Metadata
- Section headers
- Program headers

The elf Headers

1872 Bytes

The elf Headers

```
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x463940
  Start of program headers:          64 (bytes into file)
  Start of section headers:          400 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         6
  Size of section headers:           64 (bytes)
  Number of section headers:         23
  Section header string table index: 20
```

readelf -h main

The elf Headers

```
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x463940
  Start of program headers:          64 (bytes into file)
  Start of section headers:          400 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         6
  Size of section headers:           64 (bytes)
  Number of section headers:         23
  Section header string table index: 20
```

readelf -h main

The elf Headers

```
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x463940
  Start of program headers:          64 (bytes into file)
  Start of section headers:          400 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         6
  Size of section headers:           64 (bytes)
  Number of section headers:         23
  Section header string table index: 20
```

readelf -h main

A segment header

```
Type        Offset   VirtAddr           PhysAddr           FileSiz  MemSiz   Flg Align
PHDR        0x000040 0x0000000000400040 0x0000000000400040 0x000150 0x000150 R   0x1000
```

readelf -W --segment main

The elf Headers

```
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x463940
  Start of program headers:          64 (bytes into file)
  Start of section headers:          400 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         6
  Size of section headers:           64 (bytes)
  Number of section headers:         23
  Section header string table index: 20
```

readelf -h main

A section header

```
[Nr] Name              Type             Address           Off    Size   ES  Flg Lk  Inf Al
[ 1] .text             PROGBITS         0000000000401000 001000 07f9da 00  AX  0   0   32
```

readelf -SW main

ELF common sections

.text: Contains your code

.text

```
print.go:1216       0x480720      493b6610              CMPQ SP, 0x10(R14)
print.go:1216       0x480724      0f8662010000          JBE 0x48088c
print.go:1216       0x48072a      55                    PUSHQ BP
print.go:1216       0x48072b      4889e5                MOVQ SP, BP
print.go:1216       0x48072e      4883ec60              SUBQ $0x60, SP
print.go:1216       0x480732      48895c2478            MOVQ BX, 0x78(SP)
print.go:1217       0x480737      4889442470            MOVQ AX, 0x70(SP)
print.go:1217       0x48073c      48898c2480000000      MOVQ CX, 0x80(SP)
print.go:1217       0x480744      31d2                  XORL DX, DX
print.go:1217       0x480746      eb2e                  JMP 0x480776
print.go:1221       0x480748      4c89c3                MOVQ R8, BX
print.go:1221       0x48074b      4c89c9                MOVQ R9, CX
print.go:1221       0x48074e      bf76000000            MOVL $0x76, DI
print.go:1221       0x480753      e8c8daffff            CALL fmt.(*pp).printArg(SB)
print.go:1217       0x480758      488b5c2458            MOVQ 0x58(SP), BX
print.go:1217       0x48075d      4883c310              ADDQ $0x10, BX
print.go:1217       0x480761      488b542440            MOVQ 0x40(SP), DX
print.go:1217       0x480766      48ffc2                INCQ DX
print.go:1219       0x480769      488b442470            MOVQ 0x70(SP), AX
print.go:1217       0x48076e      488b8c2480000000      MOVQ 0x80(SP), CX
print.go:1217       0x480776      4839d1                CMPQ CX, DX
print.go:1217       0x480779      0f8ea9000000          JLE 0x480828
print.go:1217       0x48077f      4889542440            MOVQ DX, 0x40(SP)
print.go:1217       0x480784      48895c2458            MOVQ BX, 0x58(SP)
print.go:1217       0x480789      4c8b03                MOVQ 0(BX), R8
print.go:1217       0x48078c      4c8b4b08              MOVQ 0x8(BX), R9
print.go:1218       0x480790      4885d2                TESTQ DX, DX
print.go:1218       0x480793      7eb3                  JLE 0x480748
print.go:1219       0x480795      4c8b5010              MOVQ 0x10(AX), R10
print.go:112        0x480799      4c8b5808              MOVQ 0x8(AX), R11
print.go:112        0x48079d      49ffc3                INCQ R11
print.go:112        0x4807a0      4c8b20                MOVQ 0(AX), R12
```

go tool objdump main

.text

```
print.go:1216       0x480720        493b6610            CMPQ SP, 0x10(R14)
print.go:1216       0x480724        0f8662010000        JBE 0x48088c
print.go:1216       0x48072a        55                  PUSHQ BP
print.go:1216       0x48072b        4889e5              MOVQ SP, BP
print.go:1216       0x48072e        4883ec60            SUBQ $0x60, SP
print.go:1216       0x480732        48895c2478          MOVQ BX, 0x78(SP)
print.go:1217       0x480737        4889442470          MOVQ AX, 0x70(SP)
print.go:1217       0x48073c        48898c2480000000    MOVQ CX, 0x80(SP)
print.go:1217       0x480744        31d2                XORL DX, DX
print.go:1217       0x480746        eb2e                JMP 0x480776
print.go:1221       0x480748        4c89c3              MOVQ R8, BX
print.go:1221       0x48074b        4c89c9              MOVQ R9, CX
print.go:1221       0x48074e        bf76000000          MOVL $0x76, DI
print.go:1221       0x480753        e8c8daffff          CALL fmt.(*pp).printArg(SB)
print.go:1217       0x480758        488b5c2458          MOVQ 0x58(SP), BX
print.go:1217       0x48075d        4883c310            ADDQ $0x10, BX
print.go:1217       0x480761        488b542440          MOVQ 0x40(SP), DX
print.go:1217       0x480766        48ffc2              INCQ DX
print.go:1219       0x480769        488b442470          MOVQ 0x70(SP), AX
print.go:1217       0x48076e        488b8c2480000000    MOVQ 0x80(SP), CX
print.go:1217       0x480776        4839d1              CMPQ CX, DX
print.go:1217       0x480779        0f8ea9000000        JLE 0x480828
print.go:1217       0x48077f        4889542440          MOVQ DX, 0x40(SP)
print.go:1217       0x480784        48895c2458          MOVQ BX, 0x58(SP)
print.go:1217       0x480789        4c8b03              MOVQ 0(BX), R8
print.go:1217       0x48078c        4c8b4b08            MOVQ 0x8(BX), R9
print.go:1218       0x480790        4885d2              TESTQ DX, DX
print.go:1218       0x480793        7eb3                JLE 0x480748
print.go:1219       0x480795        4c8b5010            MOVQ 0x10(AX), R10
print.go:112        0x480799        4c8b5808            MOVQ 0x8(AX), R11
print.go:112        0x48079d        49ffc3              INCQ R11
print.go:112        0x4807a0        4c8b20              MOVQ 0(AX), R12
```

go tool objdump main

.text

```
print.go:1216       0x480720        493b6610              CMPQ SP, 0x10(R14)
print.go:1216       0x480724        0f8662010000          JBE 0x48088c
print.go:1216       0x48072a        55                    PUSHQ BP
print.go:1216       0x48072b        4889e5                MOVQ SP, BP
print.go:1216       0x48072e        4883ec60              SUBQ $0x60, SP
print.go:1216       0x480732        48895c2478            MOVQ BX, 0x78(SP)
print.go:1217       0x480737        4889442470            MOVQ AX, 0x70(SP)
print.go:1217       0x48073c        48898c2480000000      MOVQ CX, 0x80(SP)
print.go:1217       0x480744        31d2                  XORL DX, DX
print.go:1217       0x480746        eb2e                  JMP 0x480776
print.go:1221       0x480748        4c89c3                MOVQ R8, BX
print.go:1221       0x48074b        4c89c9                MOVQ R9, CX
print.go:1221       0x48074e        bf76000000            MOVL $0x76, DI
print.go:1221       0x480753        e8c8daffff            CALL fmt.(*pp).printArg(SB)
print.go:1217       0x480758        488b5c2458            MOVQ 0x58(SP), BX
print.go:1217       0x48075d        4883c310              ADDQ $0x10, BX
print.go:1217       0x480761        488b542440            MOVQ 0x40(SP), DX
print.go:1217       0x480766        48ffc2                INCQ DX
print.go:1219       0x480769        488b442470            MOVQ 0x70(SP), AX
print.go:1217       0x48076e        488b8c2480000000      MOVQ 0x80(SP), CX
print.go:1217       0x480776        4839d1                CMPQ CX, DX
print.go:1217       0x480779        0f8ea9000000          JLE 0x480828
print.go:1217       0x48077f        4889542440            MOVQ DX, 0x40(SP)
print.go:1217       0x480784        48895c2458            MOVQ BX, 0x58(SP)
print.go:1217       0x480789        4c8b03                MOVQ 0(BX), R8
print.go:1217       0x48078c        4c8b4b08              MOVQ 0x8(BX), R9
print.go:1218       0x480790        4885d2                TESTQ DX, DX
print.go:1218       0x480793        7eb3                  JLE 0x480748
print.go:1219       0x480795        4c8b5010              MOVQ 0x10(AX), R10
print.go:112        0x480799        4c8b5808              MOVQ 0x8(AX), R11
print.go:112        0x48079d        49ffc3                INCQ R11
print.go:112        0x4807a0        4c8b20                MOVQ 0(AX), R12
```

go tool objdump main

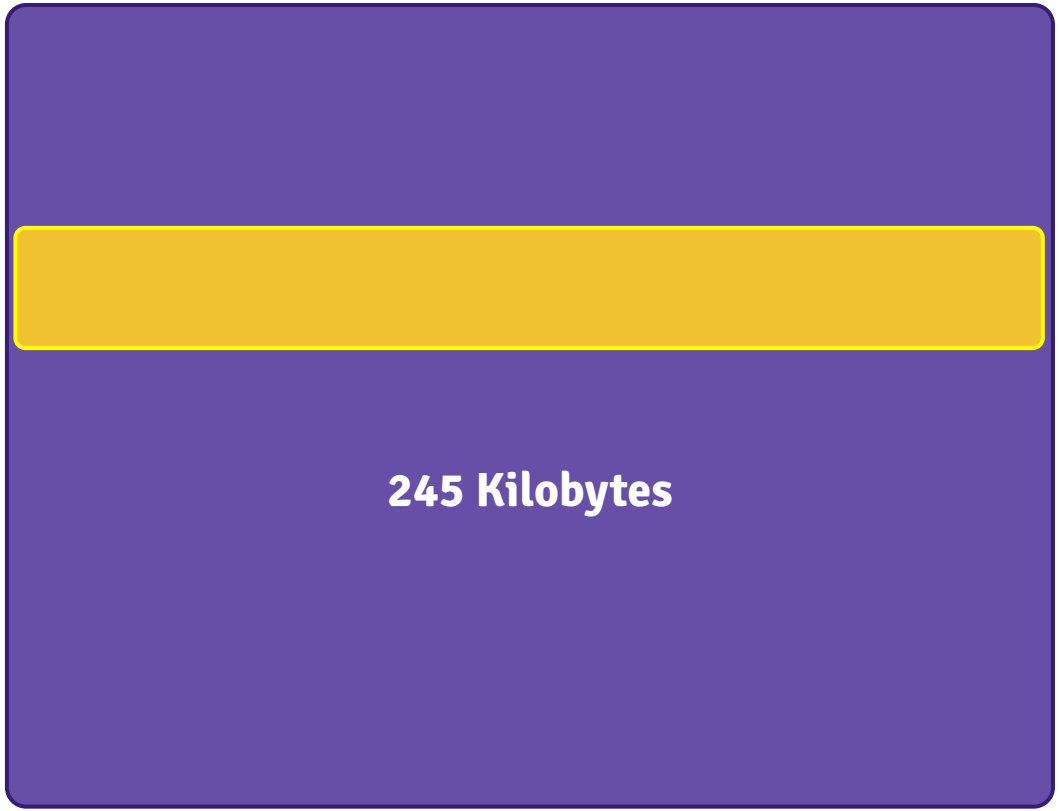
.text

```
[Nr] Name              Type            Address          Off    Size   ES Flg Lk Inf Al
[ 1] .text             PROGBITS        0000000000401000 001000 07f9da 00  AX  0   0 32

print.go:1216          0x480720        493b6610              CMPQ SP, 0x10(R14)
print.go:1216          0x480724        0f8662010000          JBE 0x48088c

$ cat main | xxd | grep 00079720 # 0x480720 - 0x401000 == 0x79720
00079720: 493b 6610 0f86 8402 0000 5548 89e5 4883  I;f.......UH..H.
```

.rodata: Contains read-only data



.rodata: Contains read-only data

- Constants
- Strings
- Types
- Interfaces
- ...

.rodata

```
0x0049bc20 77697265 703a2070 2d3e6d3d 2920702d wirep: p->m=) p-
0x0049bc30 3e737461 7475733d 72656c65 61736570 >status=releasep
0x0049bc40 3a206d3d 20737973 6d6f6e77 6169743d : m= sysmonwait=
0x0049bc50 20707265 656d7074 6f66663d 63617336  preemptoff=cas6
0x0049bc60 34206661 696c6564 206d2d3e 67736967 4 failed m->gsig
0x0049bc70 6e616c3d 2d627974 65206c69 6d69740a nal=-byte limit.
0x0049bc80 72756e74 696d653a 2073703d 61626920 runtime: sp=abi
0x0049bc90 6d69736d 61746368 696c6c65 67616c20 mismatchillegal
0x0049bca0 7365656b 696e7661 6c696420 736c6f74 seekinvalid slot
0x0049bcb0 686f7374 20697320 646f776e 6e6f7420 host is downnot
0x0049bcc0 706f6c6c 61626c65 48656c6c 6f2c2057 pollableHello, W
0x0049bcd0 6f726c64 21333831 34363937 32363536 orld!38146972656
0x0049bce0 32357761 6b656162 6c65536c 65657070 25wakeableSleepp
0x0049bcf0 726f664d 656d4163 74697665 70726f66 rofMemActiveprof
0x0049bd00 4d656d46 75747572 65747261 63655374 MemFuturetraceSt
0x0049bd10 61636b54 61626578 65635249 6e746572 ackTabexecRInter
0x0049bd20 6e616c74 65737452 496e7465 726e616c naltestRInternal
0x0049bd30 47432073 77656570 20776169 74534947 GC sweep waitSIG
0x0049bd40 51554954 3a207175 69745349 474b494c QUIT: quitSIGKIL
0x0049bd50 4c3a206b 696c6c6f 7574206f 66206d65 L: killout of me
0x0049bd60 6d6f7279 20697320 6e696c2c 206e6f74 mory is nil, not
0x0049bd70 2076616c 7565206d 6574686f 64206261  value method ba
0x0049bd80 64206d61 70207374 61746520 7370616e d map state span
```

readelf -x .rodata main

.rodata

```
0x0049bc20 77697265 703a2070 2d3e6d3d 2920702d wirep: p->m=) p-
0x0049bc30 3e737461 7475733d 72656c65 61736570 >status=releasep
0x0049bc40 3a206d3d 20737973 6d6f6e77 6169743d : m= sysmonwait=
0x0049bc50 20707265 656d7074 6f66663d 63617336  preemptoff=cas6
0x0049bc60 34206661 696c6564 206d2d3e 67736967 4 failed m->gsig
0x0049bc70 6e616c3d 2d627974 65206c69 6d69740a nal=-byte limit.
0x0049bc80 72756e74 696d653a 2073703d 61626920 runtime: sp=abi
0x0049bc90 6d69736d 61746368 696c6c65 67616c20 mismatchillegal
0x0049bca0 7365656b 696e7661 6c696420 736c6f74 seekinvalid slot
0x0049bcb0 686f7374 20697320 646f776e 6e6f7420 host is downnot
0x0049bcc0 706f6c6c 61626c65 48656c6c 6f2c2057 pollableHello, W
0x0049bcd0 6f726c64 21333831 34363937 32363536 orld!38146972656
0x0049bce0 32357761 6b656162 6c65536c 65657070 25wakeableSleepp
0x0049bcf0 726f664d 656d4163 74697665 70726f66 rofMemActiveprof
0x0049bd00 4d656d46 75747572 65747261 63655374 MemFuturetraceSt
0x0049bd10 61636b54 61626578 65635249 6e746572 ackTabexecRInter
0x0049bd20 6e616c74 65737452 496e7465 726e616c naltestRInternal
0x0049bd30 47432073 77656570 20776169 74534947 GC sweep waitSIG
0x0049bd40 51554954 3a207175 69745349 474b494c QUIT: quitSIGKIL
0x0049bd50 4c3a206b 696c6c6f 7574206f 66206d65 L: killout of me
0x0049bd60 6d6f7279 20697320 6e696c2c 206e6f74 mory is nil, not
0x0049bd70 2076616c 7565206d 6574686f 64206261  value method ba
0x0049bd80 64206d61 70207374 61746520 7370616e d map state span
```

readelf -x .rodata main

.data: Contains read-write variables

.bss: Contains not-initialized variables

Empty

.symtab: Contains the symbols table

.strtab: Contains the symbol strings

.strtab: Contains the symbol strings

```
Hex dump of section '.strtab':
  0x00000000 00676f2e 676f0072 756e7469 6d652e74 .go.go.runtime.t
  0x00000010 65787400 636d7062 6f647900 6d656d65 ext.cmpbody.meme
  0x00000020 71626f64 7900696e 64657862 79746562 qbody.indexbyteb
  0x00000030 6f647900 676f676f 0063616c 6c526574 ody.gogo.callRet
  0x00000040 00676f73 6176655f 73797374 656d7374 .gosave_systemst
  0x00000050 61636b5f 73776974 63680073 6574675f ack_switch.setg_
  0x00000060 67636300 61657368 61736862 6f647900 gcc.aeshashbody.
  0x00000070 67635772 69746542 61727269 65720064 gcWriteBarrier.d
  0x00000080 65627567 43616c6c 33320064 65627567 ebugCall32.debug
  0x00000090 43616c6c 36340064 65627567 43616c6c Call64.debugCall
  0x000000a0 31323800 64656275 6743616c 6c323536 128.debugCall256
  0x000000b0 00646562 75674361 6c6c3531 32006465 .debugCall512.de
  0x000000c0 62756743 616c6c31 30323400 64656275 bugCall1024.debu
  0x000000d0 6743616c 6c323034 38006465 62756743 gCall2048.debugC
  0x000000e0 616c6c34 30393600 64656275 6743616c all4096.debugCal
  0x000000f0 6c383139 32006465 62756743 616c6c31 l8192.debugCall1
  0x00000100 36333834 00646562 75674361 6c6c3332 6384.debugCall32
  0x00000110 37363800 64656275 6743616c 6c363535 768.debugCall655
  0x00000120 33360072 756e7469 6d652e73 69677072 36.runtime.sigpr
  0x00000130 6f664e6f 6e476f57 72617070 65720072 ofNonGoWrapper.r
  0x00000140 756e7469 6d652e65 74657874 00666d74 untime.etext.fmt
  0x00000150 2e2e7479 70654173 73657274 2e320066 ..typeAssert.2.f
```

.symtab + .strtab

```
Symbol table '.symtab' contains 2012 entries:
   Num: Value                  Size Type      Bind   Vis      Ndx  Name
        0: 0000000000000000    0    NOTYPE    LOCAL  DEFAULT UND
        1: 0000000000000000    0    FILE      LOCAL  DEFAULT ABS  go.go
        2: 0000000000401000    0    FUNC      LOCAL  DEFAULT 1    runtime.text
        3: 0000000000402d20    557  FUNC      LOCAL  DEFAULT 1    cmpbody
        4: 0000000000402f70    339  FUNC      LOCAL  DEFAULT 1    memeqbody
        5: 0000000000403120    297  FUNC      LOCAL  DEFAULT 1    indexbytebody
        6: 000000000045f600    64   FUNC      LOCAL  DEFAULT 1    gogo
        7: 000000000045f640    43   FUNC      LOCAL  DEFAULT 1    callRet
        8: 000000000045f680    47   FUNC      LOCAL  DEFAULT 1    gosave_systemstack_switch
        9: 000000000045f6c0    13   FUNC      LOCAL  DEFAULT 1    setg_gcc
       10: 000000000045f6d0    1380 FUNC      LOCAL  DEFAULT 1    aeshashbody
       11: 000000000045fc40    205  FUNC      LOCAL  DEFAULT 1    gcWriteBarrier
       12: 000000000045fd20    65   FUNC      LOCAL  DEFAULT 1    debugCall32
       13: 000000000045fd80    65   FUNC      LOCAL  DEFAULT 1    debugCall64
       14: 000000000045fde0    86   FUNC      LOCAL  DEFAULT 1    debugCall128
       15: 000000000045fe40    95   FUNC      LOCAL  DEFAULT 1    debugCall256
       16: 000000000045fea0    95   FUNC      LOCAL  DEFAULT 1    debugCall512
       17: 000000000045ff00    95   FUNC      LOCAL  DEFAULT 1    debugCall1024
       18: 000000000045ff60    95   FUNC      LOCAL  DEFAULT 1    debugCall2048
       19: 000000000045ffc0    99   FUNC      LOCAL  DEFAULT 1    debugCall4096
       20: 0000000000460040    99   FUNC      LOCAL  DEFAULT 1    debugCall8192
       21: 00000000004600c0    99   FUNC      LOCAL  DEFAULT 1    debugCall16384
       22: 0000000000460140    99   FUNC      LOCAL  DEFAULT 1    debugCall32768
       23: 00000000004601c0    99   FUNC      LOCAL  DEFAULT 1    debugCall65536
       24: 0000000000463960    108  FUNC      LOCAL  DEFAULT 1    runtime.sigprofNonGoWrapper
       25: 00000000004809da    0    FUNC      LOCAL  DEFAULT 1    runtime.etext
       26: 000000000052b960    24   OBJECT    LOCAL  DEFAULT 9    fmt..typeAssert.2
       27: 000000000052b980    24   OBJECT    LOCAL  DEFAULT 9    fmt..typeAssert.3
       28: 000000000052b9a0    24   OBJECT    LOCAL  DEFAULT 9    fmt..typeAssert.4
       29: 000000000052bc40    32   OBJECT    LOCAL  DEFAULT 9    fmt..interfaceSwitch.0
```

readelf -s main

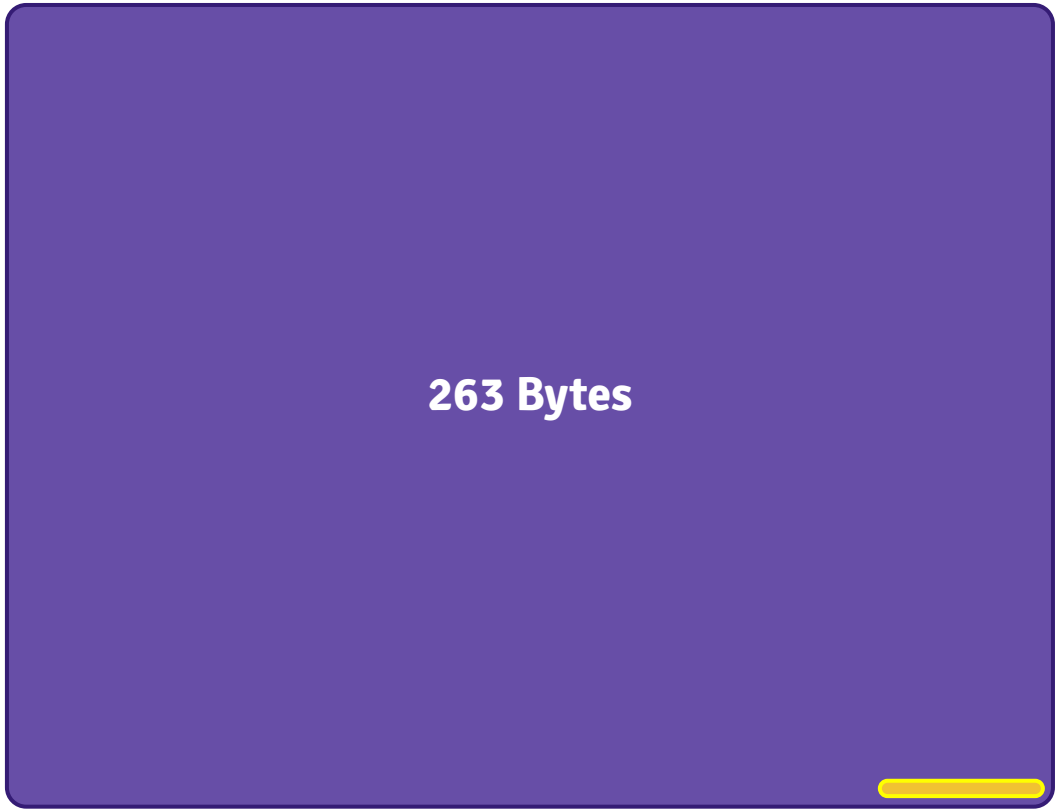
.symtab + .strtab

```
Symbol table '.symtab' contains 2012 entries:
   Num: Value                  Size Type      Bind   Vis      Ndx  Name
        0: 0000000000000000    0    NOTYPE    LOCAL  DEFAULT UND
        1: 0000000000000000    0    FILE      LOCAL  DEFAULT ABS  go.go
        2: 0000000000401000    0    FUNC      LOCAL  DEFAULT 1    runtime.text
        3: 0000000000402d20    557  FUNC      LOCAL  DEFAULT 1    cmpbody
        4: 0000000000402f70    339  FUNC      LOCAL  DEFAULT 1    memeqbody
        5: 0000000000403120    297  FUNC      LOCAL  DEFAULT 1    indexbytebody
        6: 000000000045f600    64   FUNC      LOCAL  DEFAULT 1    gogo
        7: 000000000045f640    43   FUNC      LOCAL  DEFAULT 1    callRet
        8: 000000000045f680    47   FUNC      LOCAL  DEFAULT 1    gosave_systemstack_switch
        9: 000000000045f6c0    13   FUNC      LOCAL  DEFAULT 1    setg_gcc
       10: 000000000045f6d0    1380 FUNC      LOCAL  DEFAULT 1    aeshashbody
       11: 000000000045fc40    205  FUNC      LOCAL  DEFAULT 1    gcWriteBarrier
       12: 000000000045fd20    65   FUNC      LOCAL  DEFAULT 1    debugCall32
       13: 000000000045fd80    65   FUNC      LOCAL  DEFAULT 1    debugCall64
       14: 000000000045fde0    86   FUNC      LOCAL  DEFAULT 1    debugCall128
       15: 000000000045fe40    95   FUNC      LOCAL  DEFAULT 1    debugCall256
       16: 000000000045fea0    95   FUNC      LOCAL  DEFAULT 1    debugCall512
       17: 000000000045ff00    95   FUNC      LOCAL  DEFAULT 1    debugCall1024
       18: 000000000045ff60    95   FUNC      LOCAL  DEFAULT 1    debugCall2048
       19: 000000000045ffc0    99   FUNC      LOCAL  DEFAULT 1    debugCall4096
       20: 0000000000460040    99   FUNC      LOCAL  DEFAULT 1    debugCall8192
       21: 00000000004600c0    99   FUNC      LOCAL  DEFAULT 1    debugCall16384
       22: 0000000000460140    99   FUNC      LOCAL  DEFAULT 1    debugCall32768
       23: 00000000004601c0    99   FUNC      LOCAL  DEFAULT 1    debugCall65536
       24: 0000000000463960    108  FUNC      LOCAL  DEFAULT 1    runtime.sigprofNonGoWrapper
       25: 00000000004809da    0    FUNC      LOCAL  DEFAULT 1    runtime.etext
       26: 000000000052b960    24   OBJECT    LOCAL  DEFAULT 9    fmt..typeAssert.2
       27: 000000000052b980    24   OBJECT    LOCAL  DEFAULT 9    fmt..typeAssert.3
       28: 000000000052b9a0    24   OBJECT    LOCAL  DEFAULT 9    fmt..typeAssert.4
       29: 000000000052bc40    32   OBJECT    LOCAL  DEFAULT 9    fmt..interfaceSwitch.0
```

.text

.data

.shstrtab: Contains the section header strings



.shstrtab

```
Hex dump of section '.shstrtab':
  0x00000000 002e7465 7874002e 6e6f7074 72646174 ..text..noptrdat
  0x00000010 61002e64 61746100 2e627373 002e6e6f a..data..bss..no
  0x00000020 70747262 7373002e 676f2e66 757a7a63 ptrbss..go.fuzzc
  0x00000030 6e747273 002e676f 2e627569 6c64696e ntrs..go.buildin
  0x00000040 666f002e 6e6f7465 2e676f2e 6275696c fo..note.go.buil
  0x00000050 64696400 2e656c66 64617461 002e726f did..elfdata..ro
  0x00000060 64617461 002e7479 70656c69 6e6b002e data..typelink..
  0x00000070 69746162 6c696e6b 002e676f 73796d74 itablink..gosymt
  0x00000080 6162002e 676f7063 6c6e7461 62002e73 ab..gopclntab..s
  0x00000090 796d7461 62002e73 74727461 62002e64 ymtab..strtab..d
  0x000000a0 65627567 5f616262 72657600 2e646562 ebug_abbrev..deb
  0x000000b0 75675f66 72616d65 002e6465 6275675f ug_frame..debug_
  0x000000c0 696e666f 002e6465 6275675f 6c6f6300 info..debug_loc.
  0x000000d0 2e646562 75675f6c 696e6500 2e646562 .debug_line..deb
  0x000000e0 75675f67 64625f73 63726970 7473002e ug_gdb_scripts..
  0x000000f0 64656275 675f7261 6e676573 002e7368 debug_ranges..sh
  0x00000100 73747274 616200                     strtab.
```

readelf -x .shstrtab main

.shstrtab

```
Hex dump of section '.shstrtab':
  0x00000000 002e7465 7874002e 6e6f7074 72646174 ..text..noptrdat
  0x00000010 61002e64 61746100 2e627373 002e6e6f a..data..bss..no
  0x00000020 70747262 7373002e 676f2e66 757a7a63 ptrbss..go.fuzzc
  0x00000030 6e747273 002e676f 2e627569 6c64696e ntrs..go.buildin
  0x00000040 666f002e 6e6f7465 2e676f2e 6275696c fo..note.go.buil
  0x00000050 64696400 2e656c66 64617461 002e726f did..elfdata..ro
  0x00000060 64617461 002e7479 70656c69 6e6b002e data..typelink..
  0x00000070 69746162 6c696e6b 002e676f 73796d74 itablink..gosymt
  0x00000080 6162002e 676f7063 6c6e7461 62002e73 ab..gopclntab..s
  0x00000090 796d7461 62002e73 74727461 62002e64 ymtab..strtab..d
  0x000000a0 65627567 5f616262 72657600 2e646562 ebug_abbrev..deb
  0x000000b0 75675f66 72616d65 002e6465 6275675f ug_frame..debug_
  0x000000c0 696e666f 002e6465 6275675f 6c6f6300 info..debug_loc.
  0x000000d0 2e646562 75675f6c 696e6500 2e646562 .debug_line..deb
  0x000000e0 75675f67 64625f73 63726970 7473002e ug_gdb_scripts..
  0x000000f0 64656275 675f7261 6e676573 002e7368 debug_ranges..sh
  0x00000100 73747274 616200                     strtab.
```

readelf -x .shstrtab main

Dwarf: Debug information

- Debugging With Arbitrary Record Formats
- .debug_abbrev
- .debug_line
- .debug_frame
- .debug_gdb_scripts
- .debug_info
- .debug_loc
- .debug_ranges

Dwarf: Debug information

540 Kilobytes

Dwarf: Debug information

You shall not pass!


Go specific
sections

.go.buildinfo

304 Bytes

.go.buildinfo

```
Hex dump of section '.go.buildinfo':
  0x00523000 ff20476f 20627569 6c64696e 663a0802 . Go buildinf:..
  0x00523010 00000000 00000000 00000000 00000000 ................
  0x00523020 08676f31 2e32322e 30800230 77af0c92 .go1.22.0..0w...
  0x00523030 74080241 e1c107e6 d618e670 61746809 t..A.......path.
  0x00523040 6578616d 706c650a 6d6f6409 6578616d example.mod.exam
  0x00523050 706c6509 28646576 656c2909 0a627569 ple.(devel)..bui
  0x00523060 6c64092d 6275696c 646d6f64 653d6578 ld.-buildmode=ex
  0x00523070 650a6275 696c6409 2d636f6d 70696c65 e.build.-compile
  0x00523080 723d6763 0a627569 6c640943 474f5f45 r=gc.build.CGO_E
  0x00523090 4e41424c 45443d31 0a627569 6c640943 NABLED=1.build.C
  0x005230a0 474f5f43 464c4147 533d0a62 75696c64 GO_CFLAGS=.build
  0x005230b0 0943474f 5f435050 464c4147 533d0a62 .CGO_CPPFLAGS=.b
  0x005230c0 75696c64 0943474f 5f435858 464c4147 uild.CGO_CXXFLAG
  0x005230d0 533d0a62 75696c64 0943474f 5f4c4446 S=.build.CGO_LDF
  0x005230e0 4c414753 3d0a6275 696c6409 474f4152 LAGS=.build.GOAR
  0x005230f0 43483d61 6d643634 0a627569 6c640947 CH=amd64.build.G
  0x00523100 4f4f533d 6c696e75 780a6275 696c6409 OOS=linux.build.
  0x00523110 474f414d 4436343d 76310af9 32433186 GOAMD64=v1..2C1.
  0x00523120 18207200 82421041 16d8f200 00000000 . r..B.A........
```

.go.buildinfo

```
Hex dump of section '.go.buildinfo':
  0x00523000 ff20476f 20627569 6c64696e 663a0802 . Go buildinf:..
  0x00523010 00000000 00000000 00000000 00000000 ................
  0x00523020 08676f31 2e32322e 30800230 77af0c92 .go1.22.0..0w...
  0x00523030 74080241 e1c107e6 d618e670 61746809 t..A.......path.
  0x00523040 6578616d 706c650a 6d6f6409 6578616d example.mod.exam
  0x00523050 706c6509 28646576 656c2909 0a627569 ple.(devel)..bui
  0x00523060 6c64092d 6275696c 646d6f64 653d6578 ld.-buildmode=ex
  0x00523070 650a6275 696c6409 2d636f6d 70696c65 e.build.-compile
  0x00523080 723d6763 0a627569 6c640943 474f5f45 r=gc.build.CGO_E
  0x00523090 4e41424c 45443d31 0a627569 6c640943 NABLED=1.build.C
  0x005230a0 474f5f43 464c4147 533d0a62 75696c64 GO_CFLAGS=.build
  0x005230b0 0943474f 5f435050 464c4147 533d0a62 .CGO_CPPFLAGS=.b
  0x005230c0 75696c64 0943474f 5f435858 464c4147 uild.CGO_CXXFLAG
  0x005230d0 533d0a62 75696c64 0943474f 5f4c4446 S=.build.CGO_LDF
  0x005230e0 4c414753 3d0a6275 696c6409 474f4152 LAGS=.build.GOAR
  0x005230f0 43483d61 6d643634 0a627569 6c640947 CH=amd64.build.G
  0x00523100 4f4f533d 6c696e75 780a6275 696c6409 OOS=linux.build.
  0x00523110 474f414d 4436343d 76310af9 32433186 GOAMD64=v1..2C1.
  0x00523120 18207200 82421041 16d8f200 00000000 . r..B.A........
```

.note.go.buildid

100 Bytes

.note.go.buildid

```
Hex dump of section '.note.go.buildid':
  0x00400f9c 04000000 53000000 04000000 476f0000 ....S.......Go..
  0x00400fac 7170544e 4668695a 506d3335 4d394b43 qpTNFhiZPm35M9KC
  0x00400fbc 7a626563 2f447632 54394c4c 48764943 zbec/Dv2T9LLHvIC
  0x00400fcc 72776c66 736d6363 352f715a 74745a77 rwlfsmcc5/qZttZw
  0x00400fdc 6c2d3449 58554576 624a6367 70712f47 l-4IXUEvbJcgpq/G
  0x00400fec 5637795a 48524377 62316570 68365048 V7yZHRCwb1eph6PH
  0x00400ffc 35677000                            5gp.
```

```
$ go tool buildid main
qpTNFhiZPm35M9KCzbec/Dv2T9LLHvICrwlfsmcc5/qZttZwl-4IXUEvbJcg
pq/GV7yZHRCwb1eph6PH5gp
```

```
$ go build -ldflags=”-buildid=test” main.go
$ go tool buildid main
test
```

.typelink

1204 Bytes

.typelink

- Types table
- Links types to the type definition
- Point to .rodata (by offset)
- The stored data in .rodata will be an abi.Type
- Sorted by name

.typelink

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

.itablink

88 Bytes

.itablink

- Interface Tables
- Links interface tables to the their definition
- Point to .rodata (by address)
- The stored data in .rodata will be an abi.ITab

.itablink

```
type ITab struct {
     Inter *InterfaceType
     Type *Type
     Hash uint32
     Fun [1]uintptr // variable sized.
}
```

.gopclntab

401 Kilobytes

.gopclntab

- Header
- funcNameTab
- CuTab
- FileNameTab
- pctab (pcdata)
- functab
  - PC->func table
  - FuncInfo table

.gopclntab: Header

- Magic: fffffff1 (version)
- Min Instruction Size: 1
- Ptr Size: 8
- Text Start: 401000
- NFunc: 1542
- NFiles: 198
- FuncNameOffset: 0x60
- CuOffset: 0xe0e0
- FiletabOffset: 0xe900
- PctabOffset: 0x10740
- PclnOffset: 0x427c0

.gopclntab: funcNameTab

- List of function names

```
0x004be940 696e7465 726e616c 2f616269 2e4b696e internal/abi.Kin
0x004be950 642e5374 72696e67 00696e74 65726e61 d.String.interna
0x004be960 6c2f6162 692e282a 54797065 292e4c65 l/abi.(*Type).Le
0x004be970 6e00696e 7465726e 616c2f61 62692e28 n.internal/abi.(
0x004be980 2a547970 65292e4b 696e6400 696e7465 *Type).Kind.inte
0x004be990 726e616c 2f616269 2e282a54 79706529 rnal/abi.(*Type)
0x004be9a0 2e457870 6f727465 644d6574 686f6473 .ExportedMethods
0x004be9b0 00696e74 65726e61 6c2f6162 692e282a .internal/abi.(*
0x004be9c0 54797065 292e556e 636f6d6d 6f6e0069 Type).Uncommon.i
0x004be9d0 6e746572 6e616c2f 6162692e 282a556e nternal/abi.(*Un
0x004be9e0 636f6d6d 6f6e5479 7065292e 4578706f commonType).Expo
0x004be9f0 72746564 4d657468 6f647300 696e7465 rtedMethods.inte
0x004bea00 726e616c 2f616269 2e616464 43686563 rnal/abi.addChec
0x004bea10 6b656400 696e7465 726e616c 2f616269 ked.internal/abi
0x004bea20 2e282a54 79706529 2e4e756d 4d657468 .(*Type).NumMeth
```

.gopclntab: CuTab

- Table of Compilation units
- CU -> Filename

```
[39, 0, 39, 55, 93, 135, 39, 39, 176, 39,
4294967295, 4294967295, 4294967295, 4294967295,
4294967295]
```

.gopclntab: fileNameTab

- List of file names

```
0x004cd1e0 2f757372 2f6c6f63 616c2f67 6f2f7372 /usr/local/go/sr
0x004cd1f0 632f696e 7465726e 616c2f61 62692f74 c/internal/abi/t
0x004cd200 7970652e 676f003c 6175746f 67656e65 ype.go.<autogene
0x004cd210 72617465 643e002f 7573722f 6c6f6361 rated>./usr/loca
0x004cd220 6c2f676f 2f737263 2f696e74 65726e61 l/go/src/interna
0x004cd230 6c2f6370 752f6370 752e676f 002f7573 l/cpu/cpu.go./us
0x004cd240 722f6c6f 63616c2f 676f2f73 72632f69 r/local/go/src/i
0x004cd250 6e746572 6e616c2f 6370752f 6370755f nternal/cpu/cpu_
0x004cd260 7838362e 676f002f 7573722f 6c6f6361 x86.go./usr/loca
0x004cd270 6c2f676f 2f737263 2f696e74 65726e61 l/go/src/interna
0x004cd280 6c2f6370 752f6370 755f7838 362e7300 l/cpu/cpu_x86.s.
0x004cd290 2f757372 2f6c6f63 616c2f67 6f2f7372 /usr/local/go/sr
0x004cd2a0 632f7275 6e74696d 652f696e 7465726e c/runtime/intern
0x004cd2b0 616c2f73 79732f69 6e747269 6e736963 al/sys/intrinsic
0x004cd2c0 732e676f 002f7573 722f6c6f 63616c2f s.go./usr/local/
0x004cd2d0 676f2f73 72632f69 6e746572 6e616c2f go/src/internal/
```

.gopclntab: pctab

- Store PC related metadata
- Stack pointer
- File
- Line
- Inlined
- Other data

.gopclntab: functab

- PC to FuncInfo (x num of funcs)
  - PC
  - FuncInfoOffset
- Func Info values (x num of funcs)
  - Func entry offset (.text)
  - Func name offset (.gopclntab: funcnames)
  - FuncArgs
  - DeferReturn
  - PcData
  - CuIndex
  - StartLine
  - FuncId (Used to label special functions)
  - Flags (TopFrame, SPWrite, Asm)
  - FuncDataLength
  - FuncData

.gopclntab: functab

- PC to FuncInfo (x num of funcs)

```
Function 0:
  PC: 0
  FuncInfoOffset: 12344

Function 1:
  PC: 96
  FuncInfoOffset: 12432
```

.gopclntab: functab

- Func Info values (x num of funcs)

Function Info:
  FuncEntryOffset: 0
  FuncNameOffset: 0
  FuncArgs: 8
  DeferReturn: 0
  PcData: [1, 14, 17, 4]
  CuIndex: 0
  StartLine: 84
  FuncId: 0
  Flag: 0
  FuncDataLen: 7
  FuncData: [35 0 0 0 40 0 0]

Other sections

- .noptrdata: read-write variables without pointers
- .noptrbss: Not initialized variables without pointers
- .gosymtab: Empty, replaced by .gopclntab

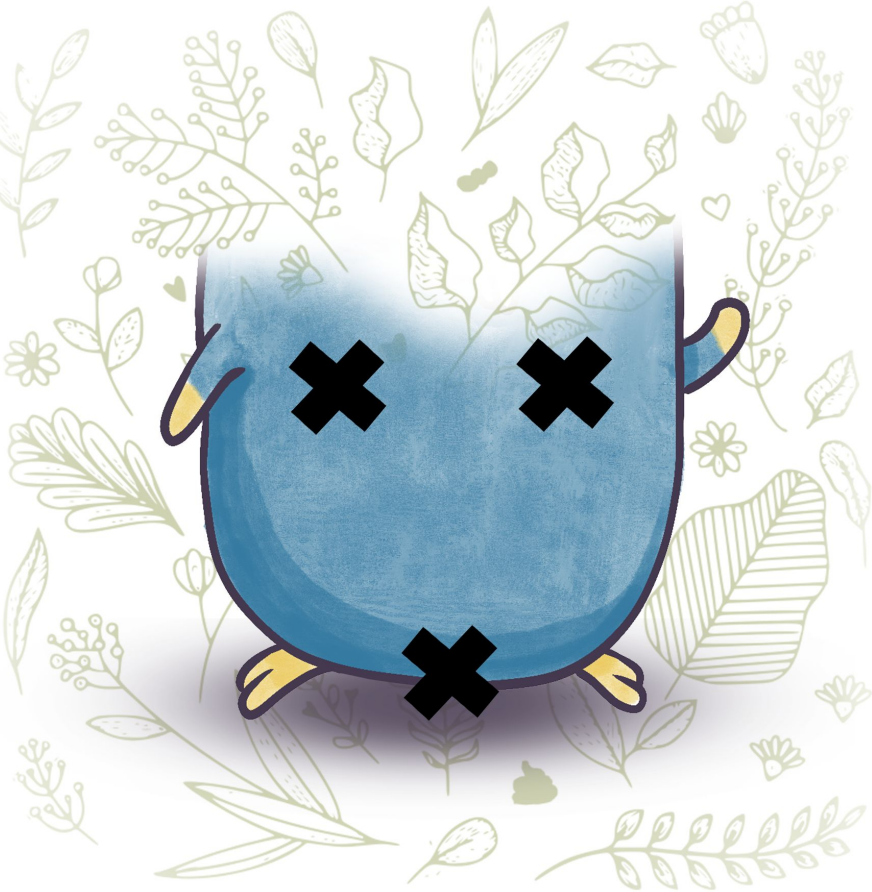
Segments

ELF: Segments

```
Elf file type is EXEC (Executable file)
Entry point 0x463940
There are 6 program headers, starting at offset 64

Program Headers:
  Type           Offset   VirtAddr           PhysAddr           FileSiz  MemSiz   Flg Align
  PHDR           0x000040 0x0000000000400040 0x0000000000400040 0x000150 0x000150 R   0x1000
  NOTE           0x000f9c 0x0000000000400f9c 0x0000000000400f9c 0x000064 0x000064 R   0x4
  LOAD           0x000000 0x0000000000400000 0x0000000000400000 0x0809da 0x0809da R E 0x1000
  LOAD           0x081000 0x0000000000481000 0x0000000000481000 0x0a1cd8 0x0a1cd8 R   0x1000
  LOAD           0x123000 0x0000000000523000 0x0000000000523000 0x009840 0x06cdc0 RW  0x1000
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RW  0x8

 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.go.buildid
   02     .text .note.go.buildid
   03     .rodata .typelink .itablink .gosymtab .gopclntab
   04     .go.buildinfo .noptrdata .data .bss .noptrbss
   05
```



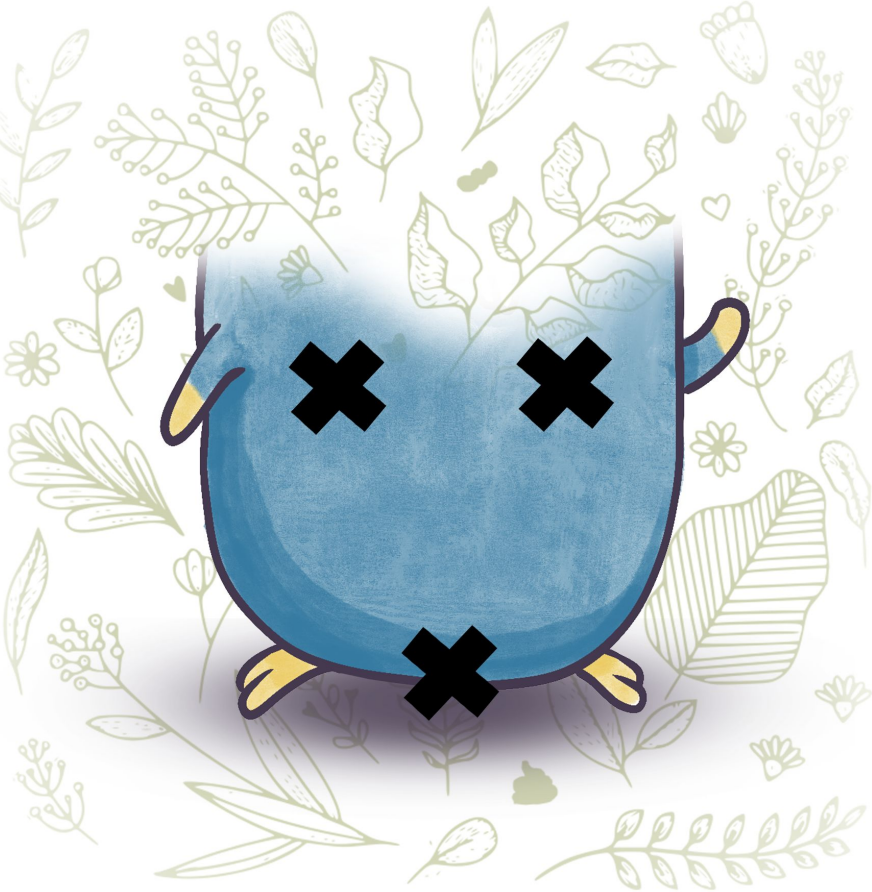
readelf -W --segment main

ELF: Segments

```
Elf file type is EXEC (Executable file)
Entry point 0x463940
There are 6 program headers, starting at offset 64

Program Headers:
  Type           Offset   VirtAddr           PhysAddr           FileSiz  MemSiz   Flg Align
  PHDR           0x000040 0x0000000000400040 0x0000000000400040 0x000150 0x000150 R   0x1000
  NOTE           0x000f9c 0x0000000000400f9c 0x0000000000400f9c 0x000064 0x000064 R   0x4
  LOAD           0x000000 0x0000000000400000 0x0000000000400000 0x0809da 0x0809da R E 0x1000
  LOAD           0x081000 0x0000000000481000 0x0000000000481000 0x0a1cd8 0x0a1cd8 R   0x1000
  LOAD           0x123000 0x0000000000523000 0x0000000000523000 0x009840 0x06cdc0 RW  0x1000
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RW  0x8

 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.go.buildid
   02     .text .note.go.buildid
   03     .rodata .typelink .itablink .gosymtab .gopclntab
   04     .go.buildinfo .noptrdata .data .bss .noptrbss
   05
```



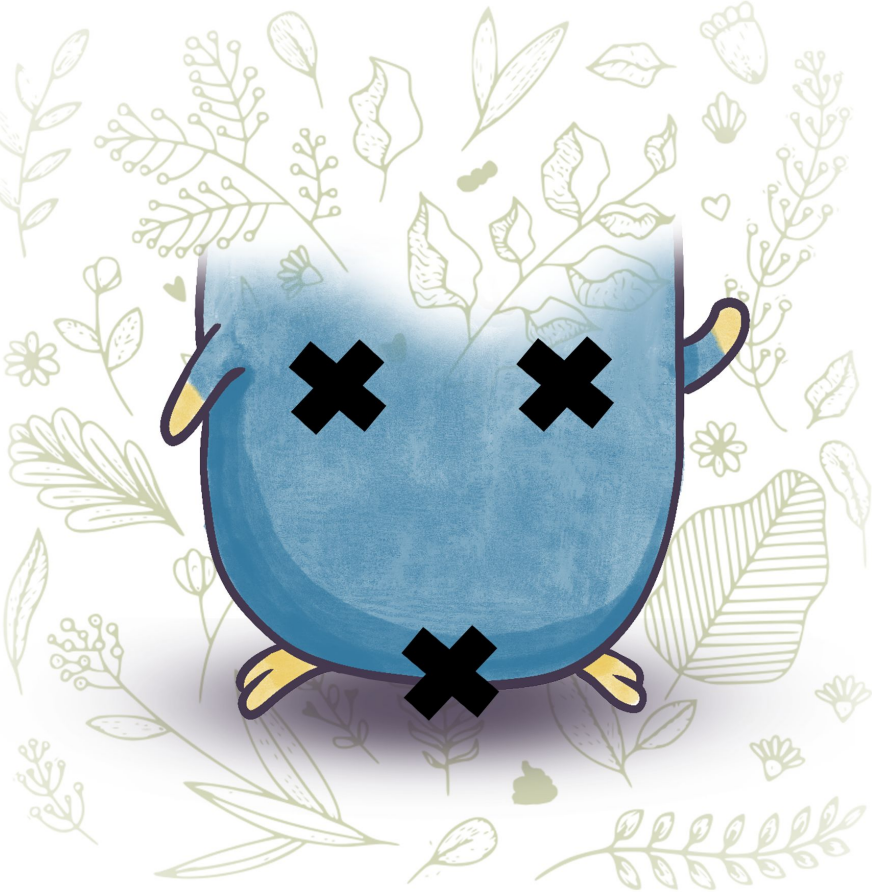
readelf -W --segment main

ELF: Segments

```
Elf file type is EXEC (Executable file)
Entry point 0x463940
There are 6 program headers, starting at offset 64

Program Headers:
  Type           Offset   VirtAddr           PhysAddr           FileSiz  MemSiz   Flg Align
  PHDR           0x000040 0x0000000000400040 0x0000000000400040 0x000150 0x000150 R   0x1000
  NOTE           0x000f9c 0x0000000000400f9c 0x0000000000400f9c 0x000064 0x000064 R   0x4
  LOAD           0x000000 0x0000000000400000 0x0000000000400000 0x0809da 0x0809da R E 0x1000
  LOAD           0x081000 0x0000000000481000 0x0000000000481000 0x0a1cd8 0x0a1cd8 R   0x1000
  LOAD           0x123000 0x0000000000523000 0x0000000000523000 0x009840 0x06cdc0 RW  0x1000
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RW  0x8

 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.go.buildid
   02     .text .note.go.buildid
   03     .rodata .typelink .itablink .gosymtab .gopclntab
   04     .go.buildinfo .noptrdata .data .bss .noptrbss
   05
```



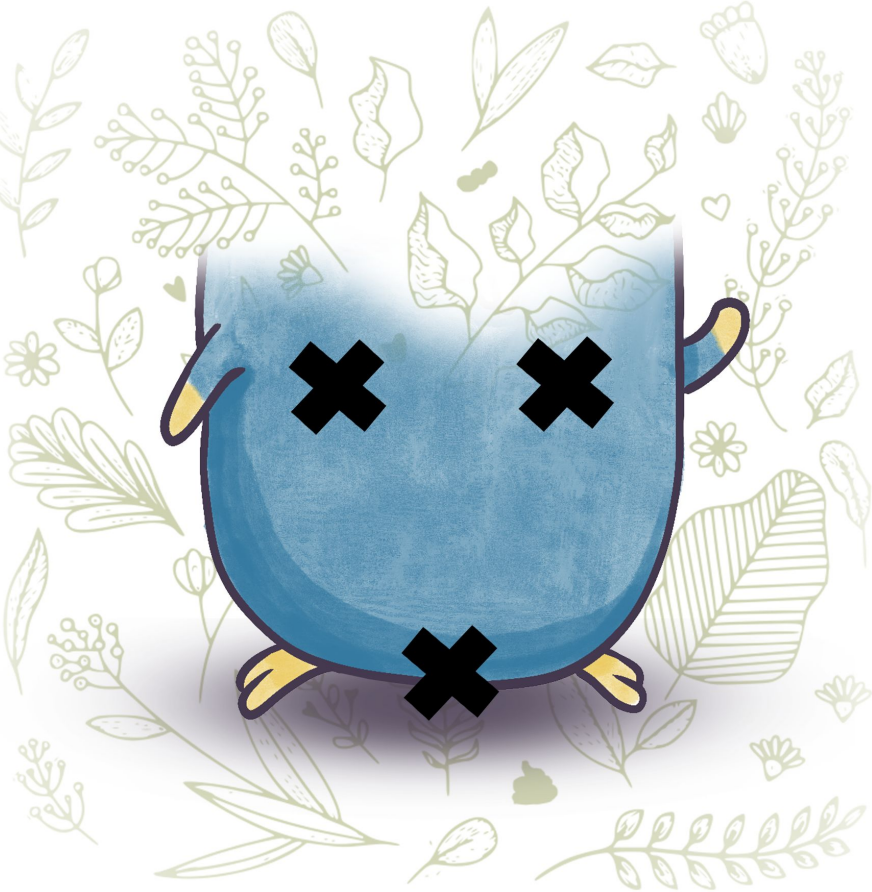
readelf -W --segment main

ELF: Segments

```
Elf file type is EXEC (Executable file)
Entry point 0x463940
There are 6 program headers, starting at offset 64

Program Headers:
  Type           Offset   VirtAddr           PhysAddr           FileSiz  MemSiz   Flg Align
  PHDR           0x000040 0x0000000000400040 0x0000000000400040 0x000150 0x000150 R   0x1000
  NOTE           0x000f9c 0x0000000000400f9c 0x0000000000400f9c 0x000064 0x000064 R   0x4
  LOAD           0x000000 0x0000000000400000 0x0000000000400000 0x0809da 0x0809da R E 0x1000
  LOAD           0x081000 0x0000000000481000 0x0000000000481000 0x0a1cd8 0x0a1cd8 R   0x1000
  LOAD           0x123000 0x0000000000523000 0x0000000000523000 0x009840 0x06cdc0 RW  0x1000
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RW  0x8

 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.go.buildid
   02     .text .note.go.buildid
   03     .rodata .typelink .itablink .gosymtab .gopclntab
   04     .go.buildinfo .noptrdata .data .bss .noptrbss
   05
```



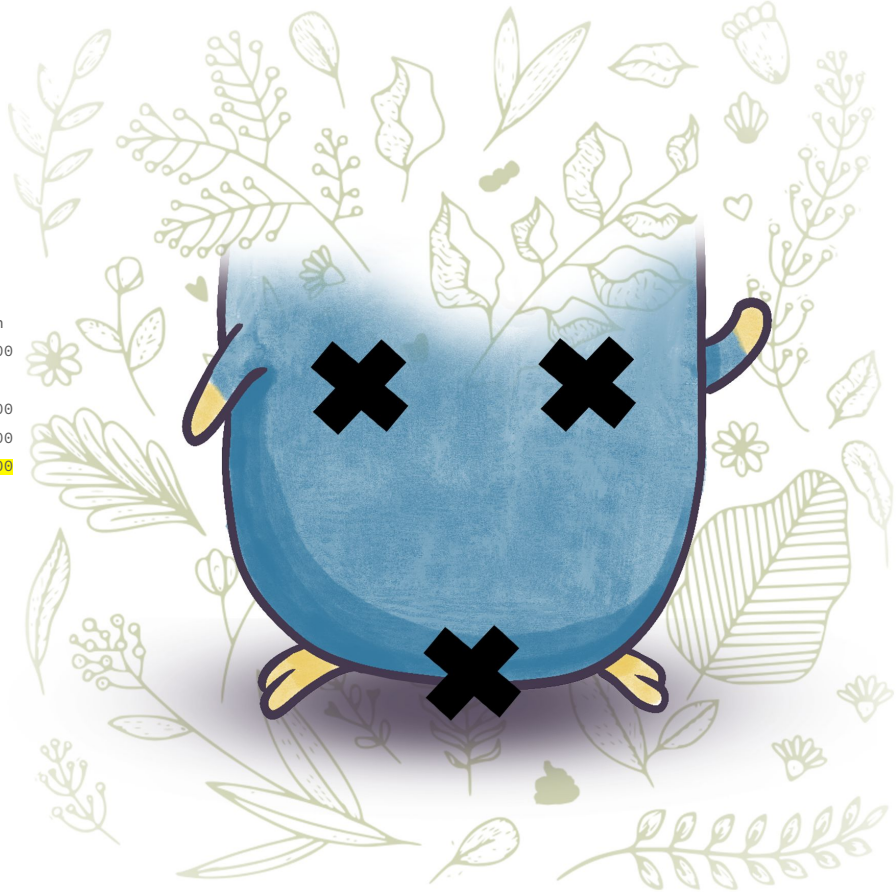
readelf -W --segment main

ELF: Segments

```
Elf file type is EXEC (Executable file)
Entry point 0x463940
There are 6 program headers, starting at offset 64

Program Headers:
  Type           Offset   VirtAddr           PhysAddr           FileSiz  MemSiz   Flg Align
  PHDR           0x000040 0x0000000000400040 0x0000000000400040 0x000150 0x000150 R   0x1000
  NOTE           0x000f9c 0x0000000000400f9c 0x0000000000400f9c 0x000064 0x000064 R   0x4
  LOAD           0x000000 0x0000000000400000 0x0000000000400000 0x0809da 0x0809da R E 0x1000
  LOAD           0x081000 0x0000000000481000 0x0000000000481000 0x0a1cd8 0x0a1cd8 R   0x1000
  LOAD           0x123000 0x0000000000523000 0x0000000000523000 0x009840 0x06cdc0 RW  0x1000
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RW  0x8

 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.go.buildid
   02     .text .note.go.buildid
   03     .rodata .typelink .itablink .gosymtab .gopclntab
   04     .go.buildinfo .noptrdata .data .bss .noptrbss
   05
```



readelf -W --segment main


Interesting
things

You don’t need all your sections

```
$ ls -s main
1,9M main
$ strip main
$ ls main
1,2M main
$ readelf -SW main
There are 14 section headers, starting at offset 0x12c8b8:

Section Headers:
  [Nr] Name              Type            Address          Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            0000000000000000 000000 000000 00      0   0  0
  [ 1] .text             PROGBITS        0000000000401000 001000 07f9da 00  AX  0   0 32
  [ 2] .rodata           PROGBITS        0000000000481000 081000 03d2da 00   A  0   0 32
  [ 3] .typelink         PROGBITS        00000000004be2e0 0be2e0 000590 00   A  0   0 32
  [ 4] .itablink         PROGBITS        00000000004be880 0be880 000058 00   A  0   0 32
  [ 5] .gosymtab         PROGBITS        00000000004be8d8 0be8d8 000000 00   A  0   0  1
  [ 6] .gopclntab        PROGBITS        00000000004be8e0 0be8e0 0643f8 00   A  0   0 32
  [ 7] .go.buildinfo     PROGBITS        0000000000523000 123000 000130 00  WA  0   0 16
  [ 8] .noptrdata        PROGBITS        0000000000523140 123140 0054a0 00  WA  0   0 32
  [ 9] .data             PROGBITS        00000000005285e0 1285e0 004250 00  WA  0   0 32
  [10] .bss              NOBITS          000000000052c840 12c830 05fb30 00  WA  0   0 32
  [11] .noptrbss         NOBITS          000000000058c380 12c830 003a40 00  WA  0   0 32
  [12] .note.go.buildid  NOTE            0000000000400f9c 000f9c 000064 00   A  0   0  4
  [13] .shstrtab         STRTAB          0000000000000000 12c830 000081 00      0   0  1
```

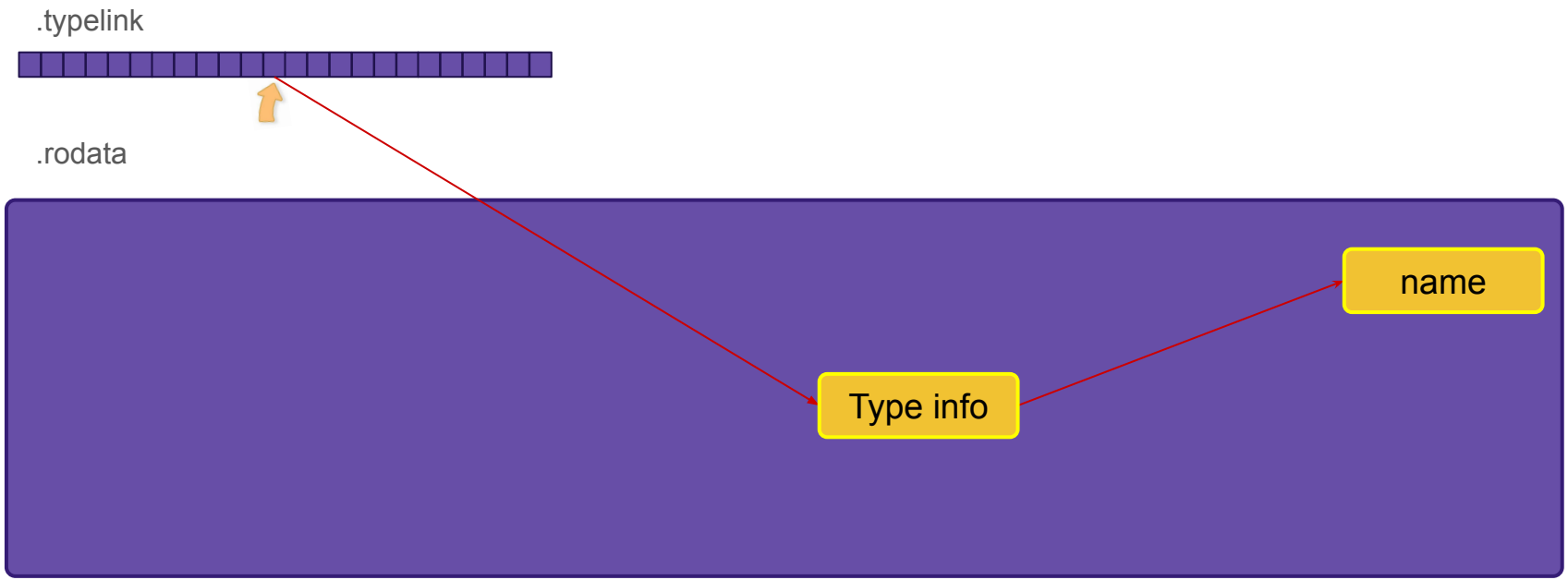
You can be leaking info

```
$ readelf -p .gopclntab main | grep home
  [ 106ed]  /home/jespino/Projects/talks/understanding-a-go-binary/example/main.go
$ readelf -p .gopclntab main | grep main.main
  [  e0ad]  main.main
```

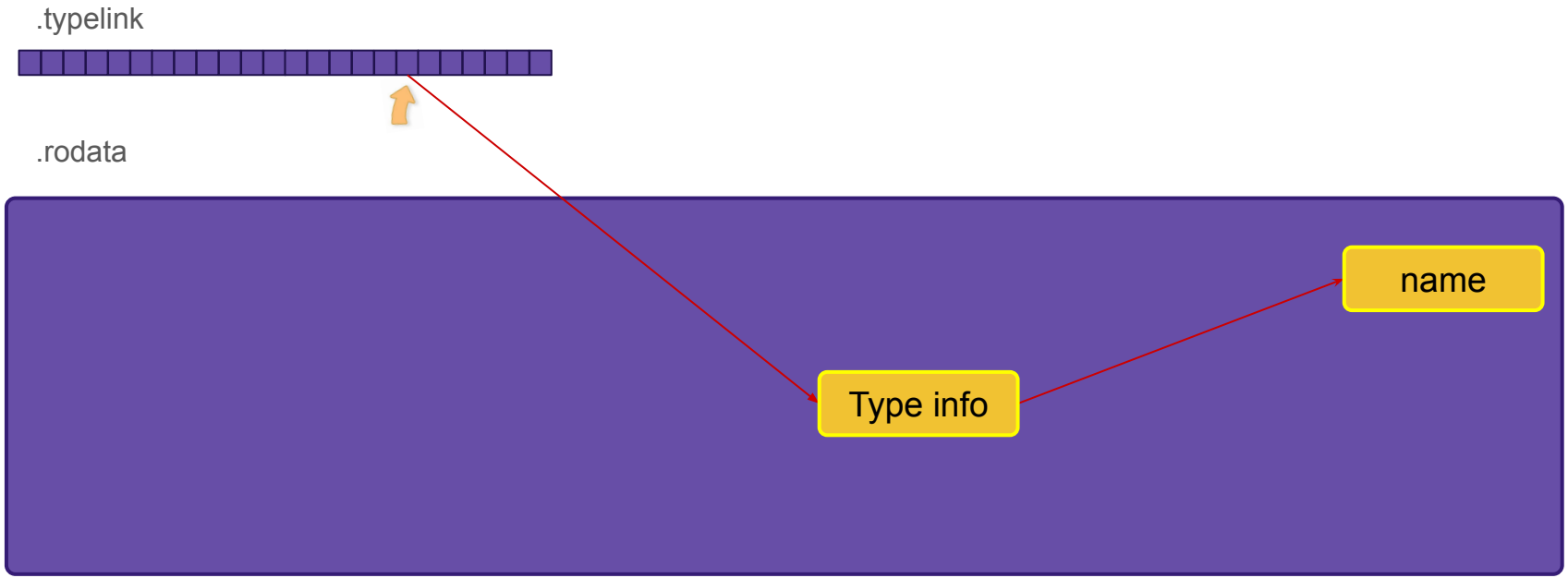
Your tracebacks use the .gopclntab info

- Tracebacks are built with the .gopclntab info
- It use the current PC to start the process

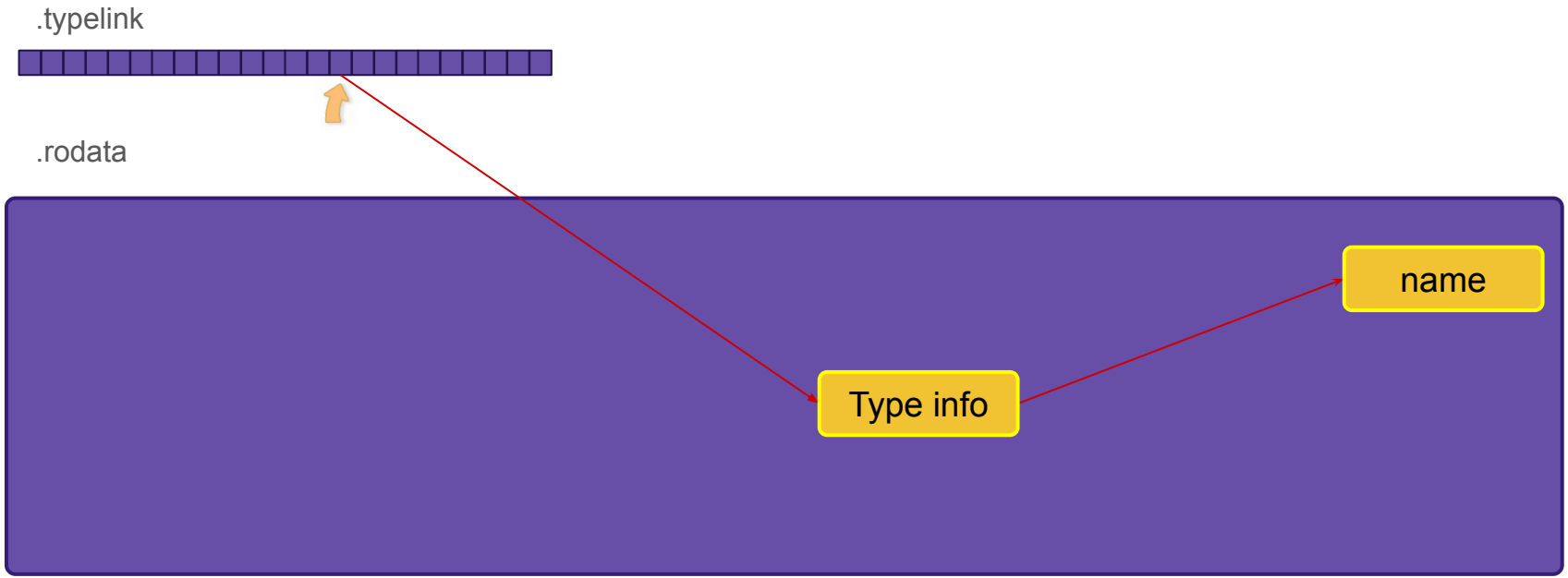
reflect use .typelink info



reflect use .typelink info

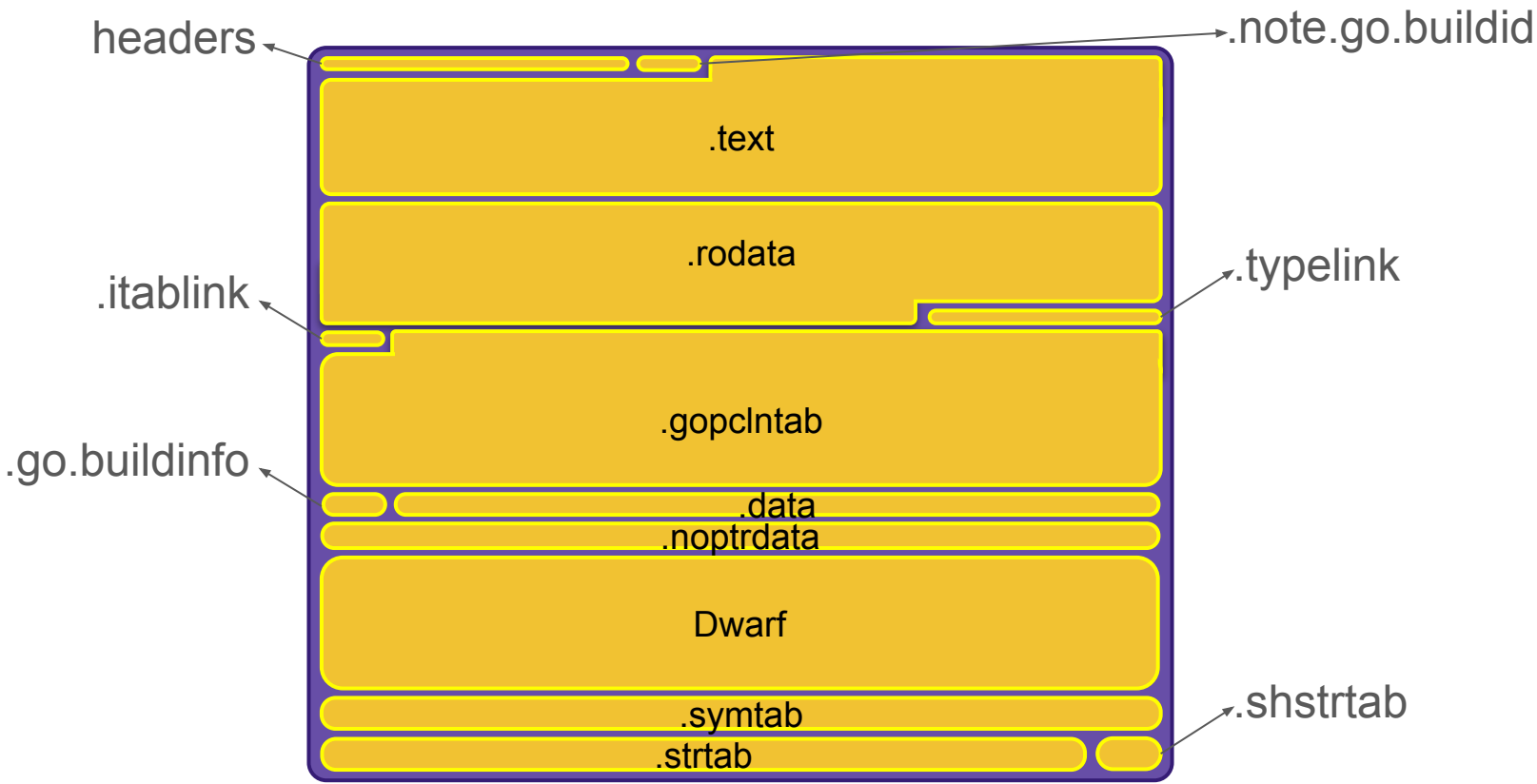


reflect use .typelink info



Summary

Summary


headers
.note.go.buildid
.text
.rodata
.itablink
.typelink
.gopclntab
.go.buildinfo
.data
.noptrdata
Dwarf
.symtab
.shstrtab
.strtab

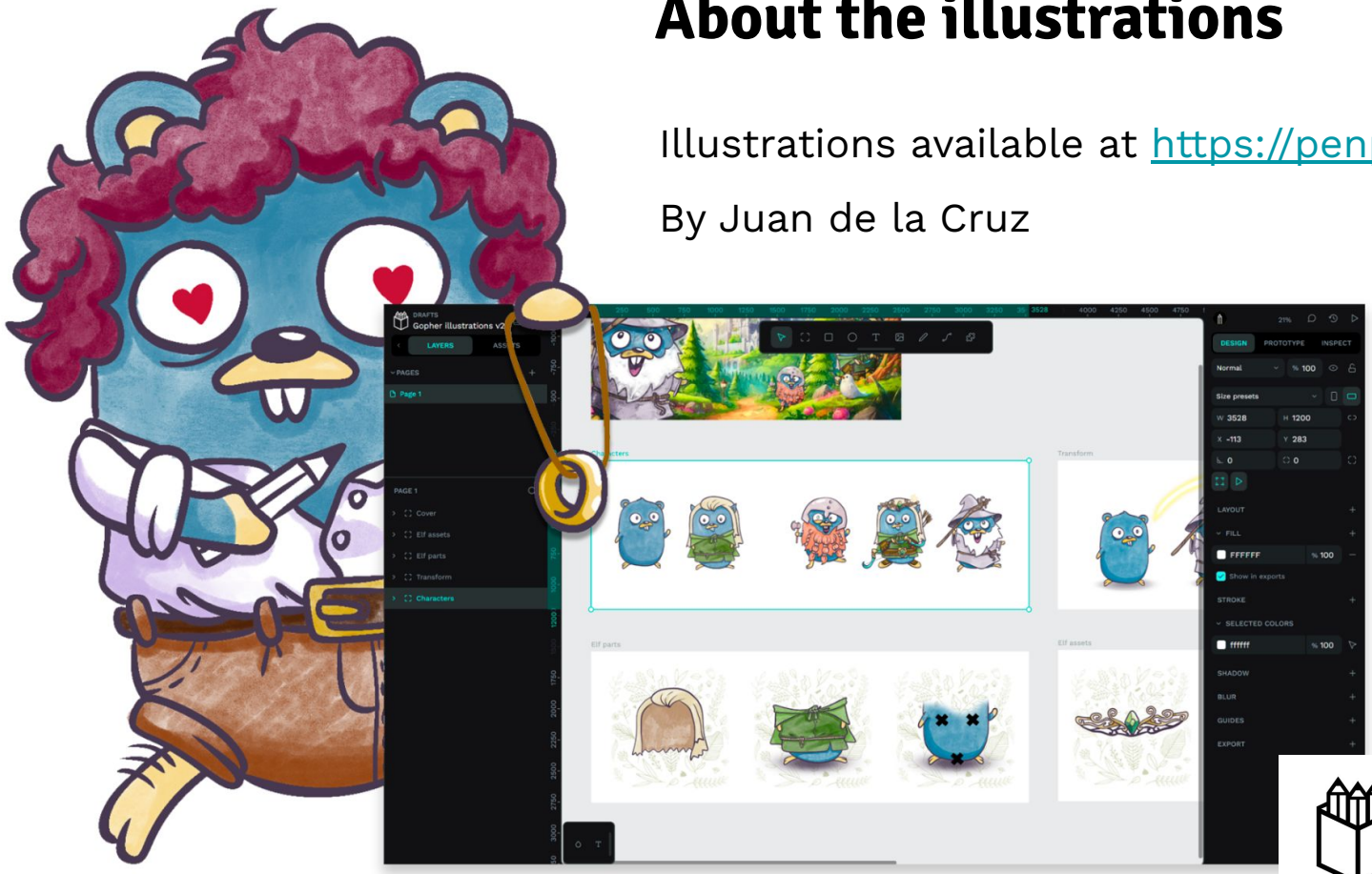
References

- In-depth ELF: https://youtu.be/nC1U1LJQL8o?si=O-X8EnOUVgbmCkPp
- gopclntab design document: https://docs.google.com/document/d/1lyPIbmsYbXnpNj57a261hgOYVpNRcgydurVQIyZOz_o/pub
- GoReSym: https://github.com/mandiant/GoReSym
- Garble: https://github.com/burrowers/garble
- The go source code (specifically the linker code)

About the illustrations

Illustrations available at https://penpot.app

By Juan de la Cruz



penpot

A gift from Mattermost

Let’s keep in touch

jespinog

jespino

jesus-espino


Thank you