Go Reverse

Bruno Pujos

September 3, 2014

Golang

- Idea of the Go language began in 2007, became a public open source project into 2009,
- Compiled, concurrent, imperative, structured,
- Developed by Google.

> *Go is an attempt to combine the ease of programming of an interpreted, dynamically typed language with the efficiency and safety of a statically typed, compiled language.*

Go Reverse

- Compiled language
- 2 main compilers: `gc` and `gccgo`
- Everything in this talk is about `gc`-compiled 64bits ELFs

Debug

- By default, `go build` will compile with DWARFv3 debug information
- Should work with `gdb`, not so well sadly

Identify a Go program

Section to Segment mapping:
Segment Sections...
00
01 .text
02 .rodata .typelink .gosymtab .gopclntab
03 .noptrdata .data .bss .noptrbss
04
05

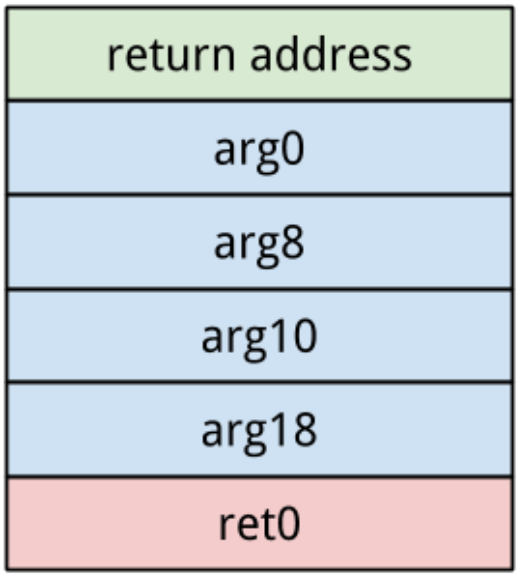
Entry Point

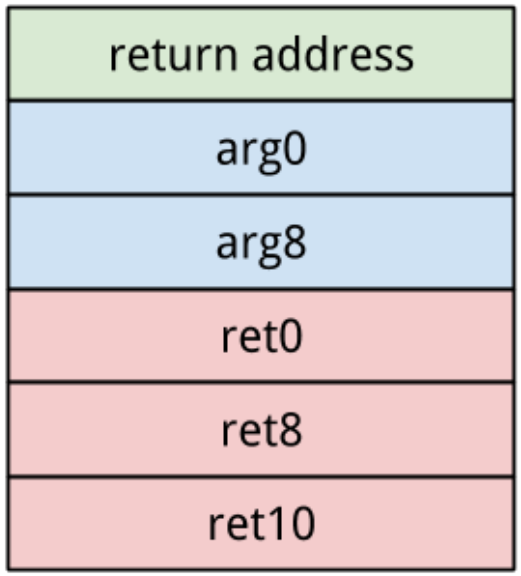
- The start of a Go program is `main.main`
- `main.main` is called by `runtime.main`
- The first byte of the `enoptrbss` section will be update during one of the module initialisation
- From there, we can find `main.main`

Calling convention (1/2)

- In Go, each function can take several arguments and have several return values
- The arguments are passed on the stack and the return values too

Calling convention (2/2)

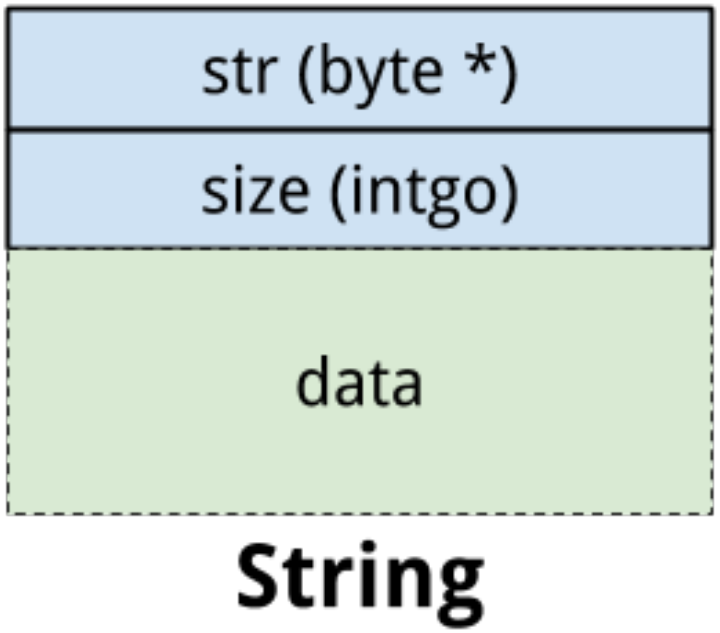




Strings

Go Reverse

Bruno Pujos



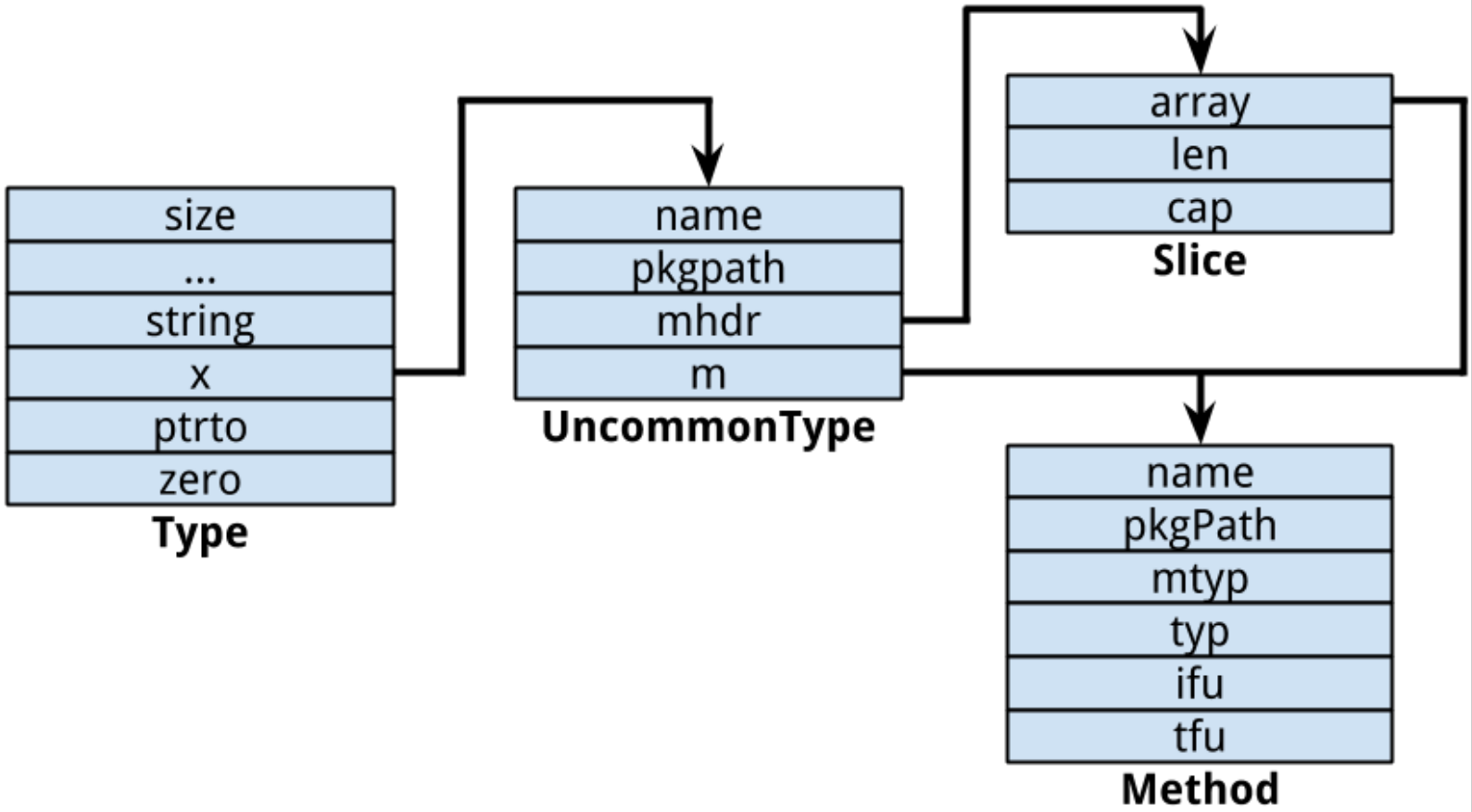
Structures and methods

- Nothing really new(here)
- Structures are the same as C
- Structures are passed in the first argument of methods

Allocation with `new`

- `new` == `runtime.new`
- If the compiler detects that it doesn't need to call `new`, it will use a local variable
- `runtime.new(Type *typ, uint8 *ret)`

Structures


size
...
string
x
ptrto
zero
Type
name
pkgpath
mhdr
m
UncommonType
array
len
cap
Slice
name
pkgPath
mtyp
typ
ifu
tfu
Method

Conclusion

- The assembler is not simple
- The reflective package in Go ”leak”s a lot of information about the program
- Interesting files:
  - pkg/runtime/type.h
  - pkg/runtime/runtime.h
  - pkg/reflect/value.go
  - pkg/reflect/type.go