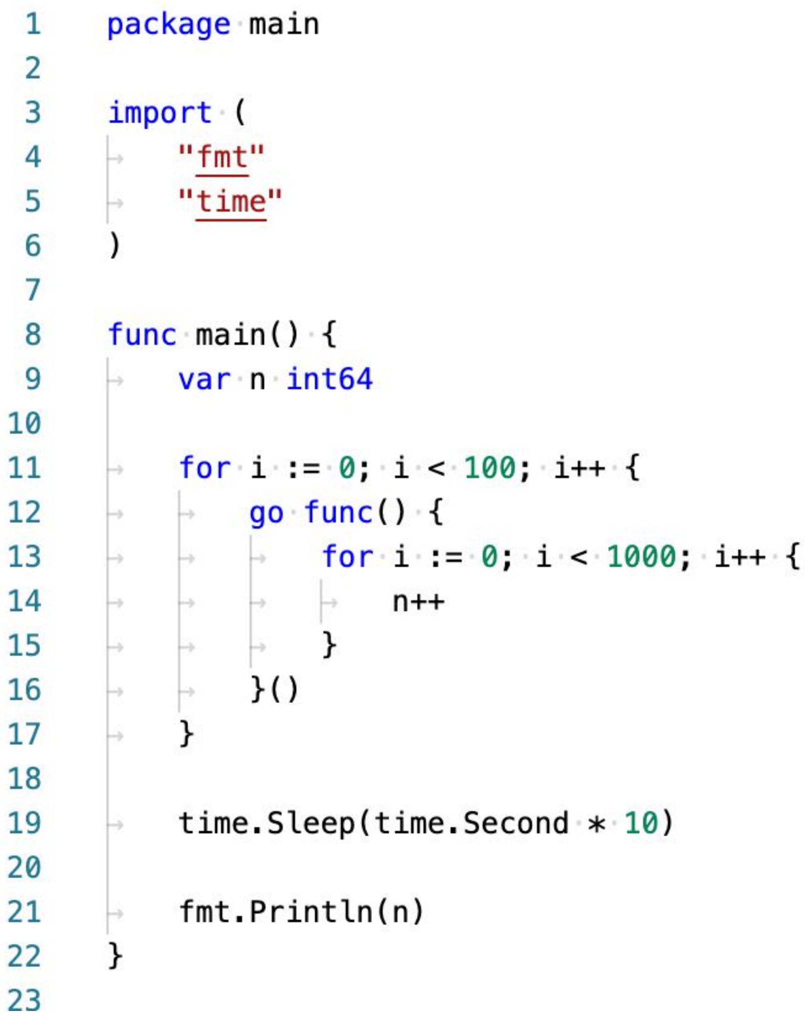
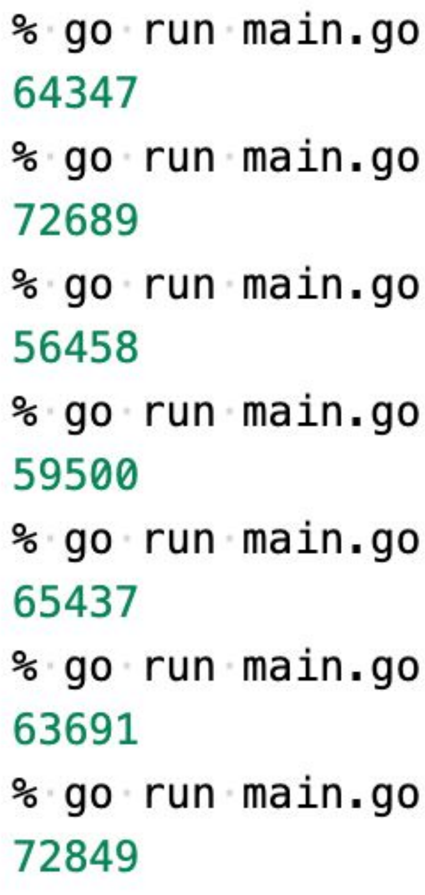
sync/atomic

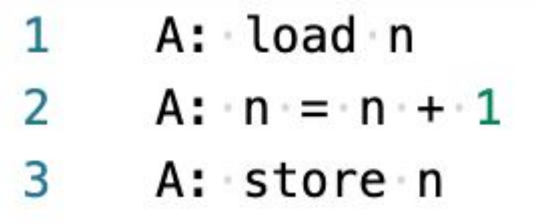
Erik Dubbelboer
github.com/erikdubbelboer



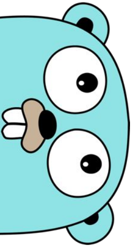
```go
package main

import (
    "fmt"
    "time"
)

func main() {
    var n int64

    for i := 0; i < 100; i++ {
        go func() {
            for i := 0; i < 1000; i++ {
                n++
            }
        }()
    }

    time.Sleep(time.Second * 10)

    fmt.Println(n)
}
```

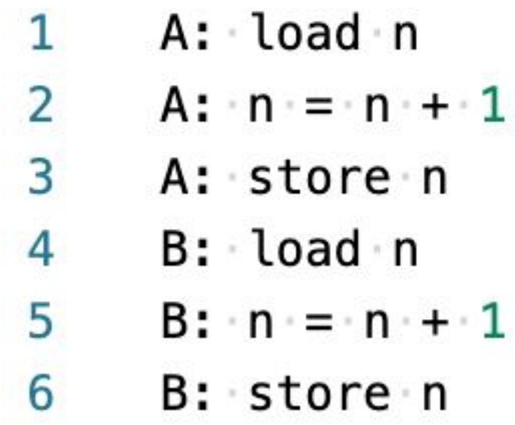


```
% go run main.go
64347
% go run main.go
72689
% go run main.go
56458
% go run main.go
59500
% go run main.go
65437
% go run main.go
63691
% go run main.go
72849
```

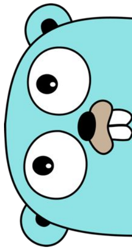


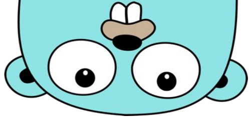
```
A: load n
A: n = n + 1
A: store n
```

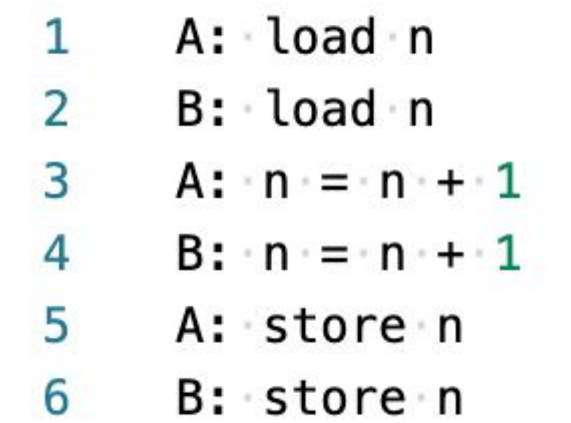




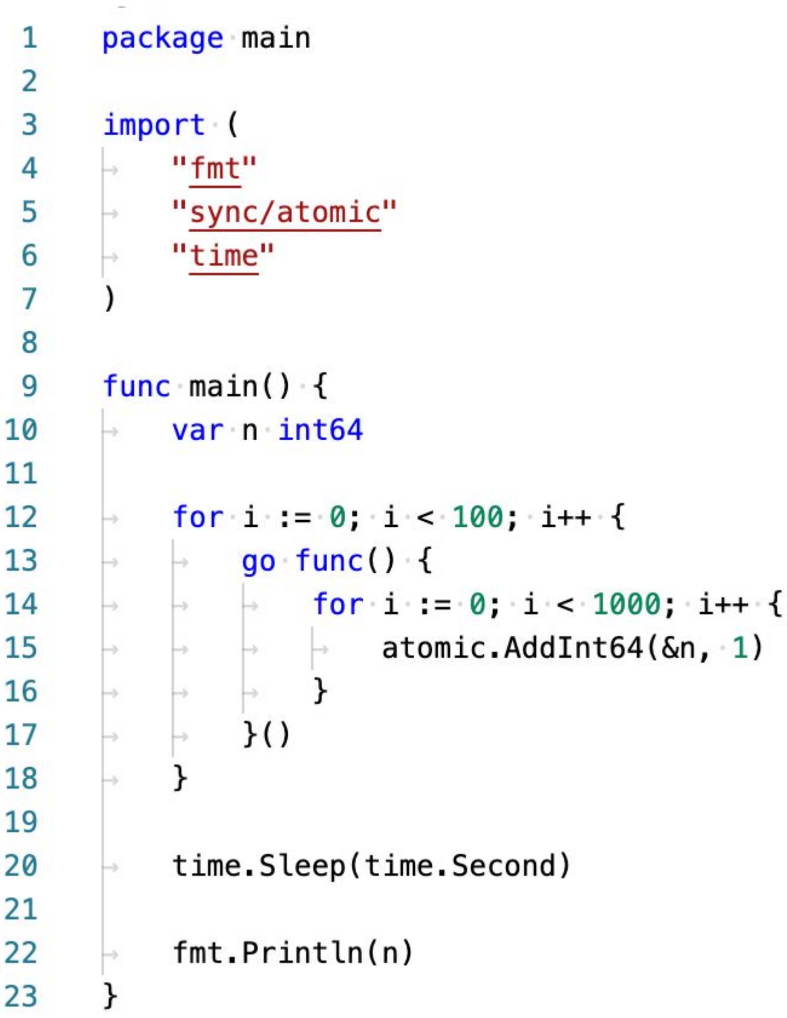
```
A: load n
A: n = n + 1
A: store n
B: load n
B: n = n + 1
B: store n
```



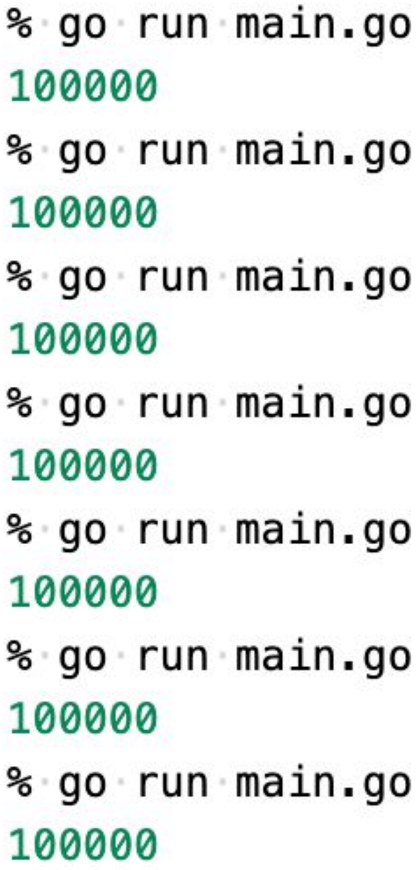




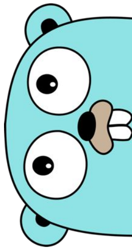
```
A: load n
B: load n
A: n = n + 1
B: n = n + 1
A: store n
B: store n
```

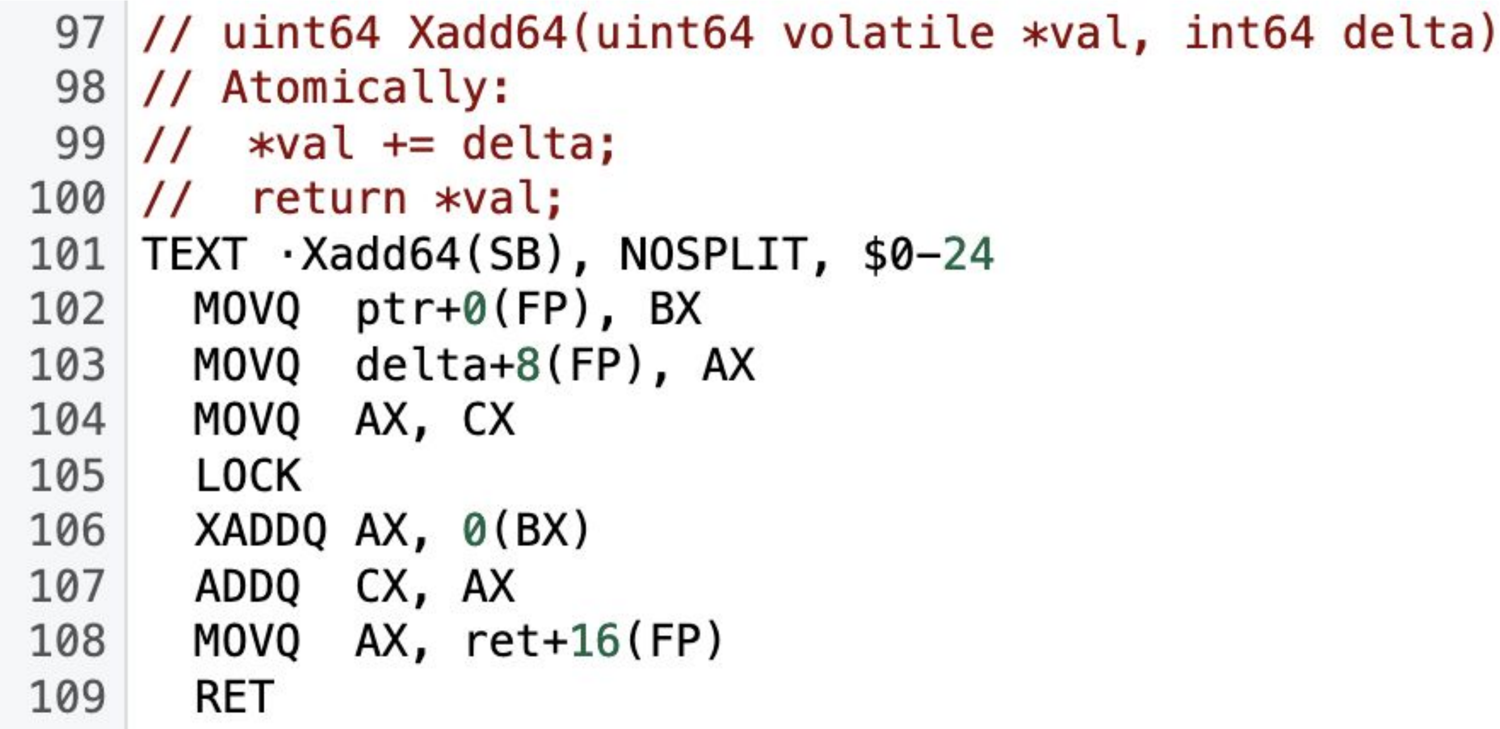


```go
package main

import (
    "fmt"
    "sync/atomic"
    "time"
)

func main() {
    var n int64

    for i := 0; i < 100; i++ {
        go func() {
            for i := 0; i < 1000; i++ {
                atomic.AddInt64(&n, 1)
            }
        }()
    }

    time.Sleep(time.Second)

    fmt.Println(n)
}
```

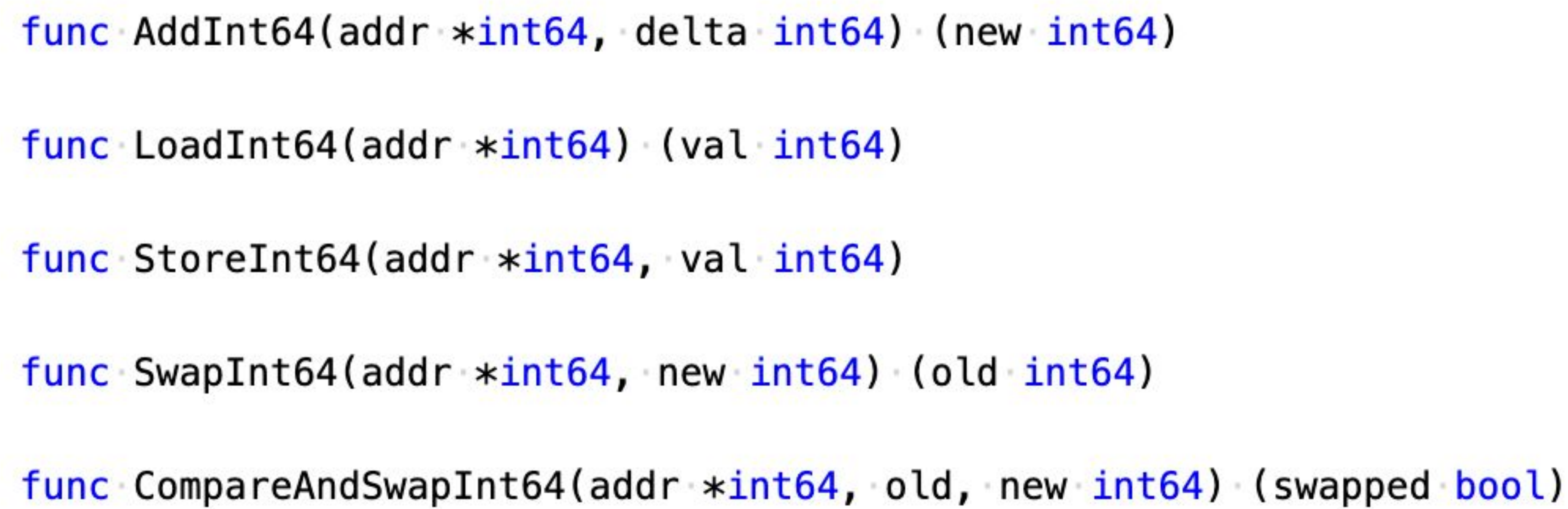


```
% go run main.go
100000
% go run main.go
100000
% go run main.go
100000
% go run main.go
100000
% go run main.go
100000
% go run main.go
100000
% go run main.go
100000
```

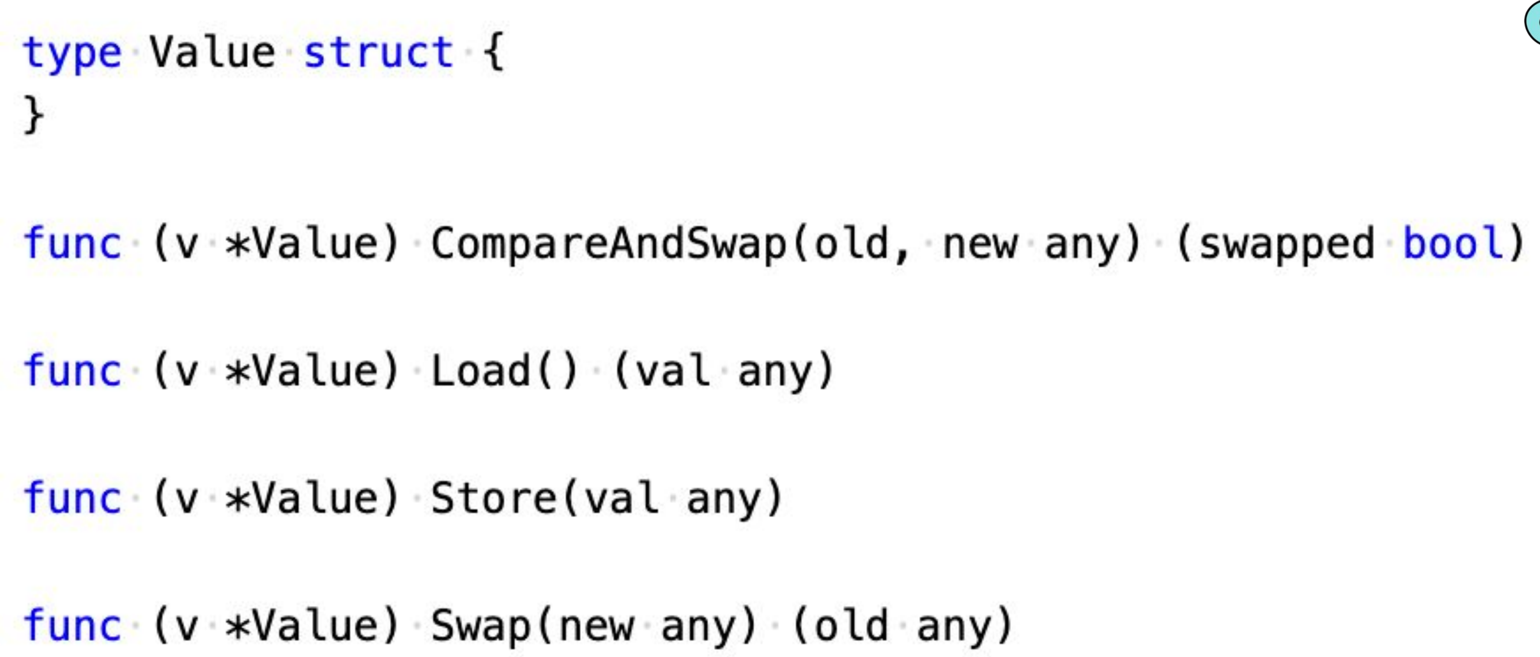




```
// uint64 Xadd64(uint64 volatile *val, int64 delta)
// Atomically:
//  *val += delta;
//  return *val;
TEXT ·Xadd64(SB), NOSPLIT, $0-24
  MOVQ  ptr+0(FP), BX
  MOVQ  delta+8(FP), AX
  MOVQ  AX, CX
  LOCK
  XADDQ AX, 0(BX)
  ADDQ  CX, AX
  MOVQ  AX, ret+16(FP)
  RET
```



```go
func AddInt64(addr *int64, delta int64) (new int64)

func LoadInt64(addr *int64) (val int64)

func StoreInt64(addr *int64, val int64)

func SwapInt64(addr *int64, new int64) (old int64)

func CompareAndSwapInt64(addr *int64, old, new int64) (swapped bool)
```



```
type Value struct {
}

func (v *Value) CompareAndSwap(old, new any) (swapped bool)

func (v *Value) Load() (val any)

func (v *Value) Store(val any)

func (v *Value) Swap(new any) (old any)
```

Questions?