

Go

T&S Hackthology



What is Go?

A New Systems Programming Language



Systems Programming, What?

- Systems programming means interacting with the computer system at the lowest level to create a component of the system itself
- Application programming means making full use of abstractions provided by the computer system to create a useful tool for users of the system



Classic Systems Languages

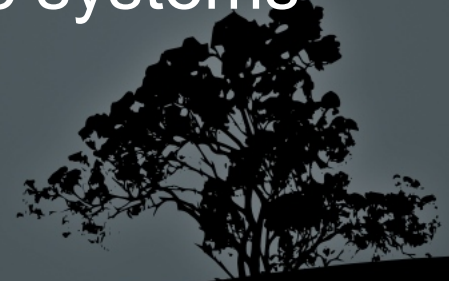
- x86, MIPS, other assembly languages
- C
- Forth
- C++, Objective-C



Language Innovation

- Garbage collection
- Concurrency primitives
- Coroutines
- Object orientation
- Namespaces
- Type safety
- Closures
- Syntactic sugar
- Etc.

These innovations are not present in classic systems languages



Reasons for Missing Innovations

- Standardization, backwards compatibility
 - Unix is C, C is Unix
 - And Windows
- Control
 - I want to be able to write memory leaks
- Speed
 - Bare metal
- Etc.



What does Go offer?

- Type safety
- Garbage collection
- Object system
- Namespaces
- Concurrency primitives
- Closures
- Duck-typed interfaces
- Nicer syntax



Simple syntax example

```
package main
```

```
import (  
    "os"  
    "fmt"  
)
```

```
func GetInput(size int) ([]byte, bool) {  
    var input []byte = make([]byte, size)  
    os.Stdin.Read(input)  
    return input, true  
}
```

```
func main() {  
    text, success := GetInput(10)  
    fmt.Printf("%s (%v)", text, success)  
}
```



Python Style Generator Function

```
package main
import "fmt"
```

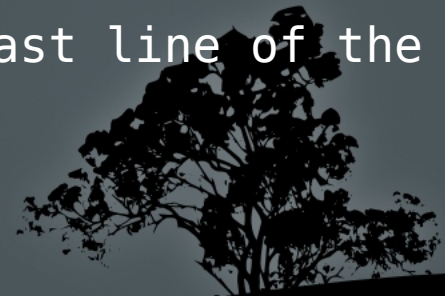
```
func main() {
    results := make(chan int)

    // ack acknowledge receipt and processing of last result
    ack := make(chan bool)

    // Calculating function
    go func(yield chan<- int, ack <-chan bool) {
        for i := 0; i < 10; i++ {
            yield<-i
            <-ack
        }
        close(yield) // causes the processing loop to quit
    }(results, ack)

    // Processing Loop
    for result := range results {
        fmt.Println(result);
        ack<-true; // ack<-true must be the last line of the loop.
    }
}
```

Emulated Using Channels!



But...but D!

- As big as C++
- Less well supported
- Compiler schisms
- Library schisms
- Not fully OSS

Hasn't taken off because of confusion in the community



Q?

