

Estruturas de Linguagem

Linguagem de programação Go

Alunos:

Eduardo Arthur
Leonardo Andrade

Professor:

Francisco Sant'Anna

Origens e Influências

- **Go** é uma linguagem de programação criada pela Google e lançada em 2009 com programa de código aberto
- Inspirada em trabalhos do sistema operacional inferno e baseada em C com várias otimizações
- Foi criada em 2007 para resolver problemas de performance que apareciam quando se construía algoritmos complexos em linguagens já existentes como C++, python ou java

Origens e influências

- Foi criada inicialmente com o objetivo de ser uma linguagem contendo as funcionalidades consideradas mais importantes nativamente
- Algumas características:
 - Programação concorrente e paralela nativa
 - Coletor de Lixo nativo
 - Gestão própria de memória e threads de forma transparente (evitando invasão de memória)
 - Simples e otimizada

Classificação

- Compilada
- Multiplataforma
- Paradigmas
- Concorrente
- Imperativa
- Estruturada
- Open Source
- Tipagem estática e inferência de tipos

Avaliação Comparativa

Go é uma linguagem que tenta combinar simplicidade e performance, assim juntando a facilidade de uma linguagem interpretada e dinamicamente tipada, um exemplo disso é o caso da inferência:

Go

C++

```
var (  
    name    string  
    age     int  
    location string  
)
```

Sem inferência

```
var (  
    name    = "Bruce Wayne"  
    age     = 32  
    location = "Gotham"  
)
```

Com inferência

```
string name;  
int age;  
string location;
```

```
string name = "Bruce Wayne";  
int age = 32;  
string location = "Gotham";
```

Figura 2 — Inferência de tipos

Avaliação Comparativa

- Goroutines
 - É uma thread mais “leve” administrada pelo runtime do Go
 - São executadas no mesmo endereço, dessa forma precisando estar sincronizadas para acessar a memória compartilhada
- Canais
 - É um “condutor” de informações que manda e recebe dados
 - Por padrão ele manda e recebe dados até que o outro lado esteja pronto, assim sincronizando as goroutines

Avaliação Comparativa

Goroutine

```
1 package main
2
3 import (
4     "fmt"
5     "time"
6 )
7
8 func say(s string, done chan string) {
9     for i := 0; i < 5; i++ {
10         time.Sleep(100 * time.Millisecond)
11         fmt.Println(s)
12     }
13     done <- "Terminei"
14 }
15
16
17 func main() {
18     done := make(chan string)
19     go say("world", done)
20     fmt.Println(<-done)
21 }
```

```
world
world
world
world
world
Terminei
```

Program exited.

Thread

```
from threading import Thread

class Th(Thread):

    def __init__(self, num):
        Thread.__init__(self)
        self.num = num

    def run(self):

        print "Hello "
        print self.num

a = Th(1)
a.start()
```

```
>>> Hello
1
```

Avaliação Comparativa

Go

Lisp

```
1 package main
2 import (
3     "fmt"
4     "os"
5     "strconv"
6 )
7 func fib(n int) int {
8     if n==0 { return 0
9     }else if n==1 {return 1
10    }else {return fib(n-1)+ fib(n-2)
11    }
12 }
13 func main() {
14     f, err := os.Create("text.txt")
15     if err != nil {
16         panic("cannot create file")
17     }
18     defer f.Close()
19     for i:=0; i <= 40 ; i++ {
20         fmt.Fprintln(f, strconv.Itoa(fib(i)))
21     }
22 }
```

```
1
2 (defun fibonacci(n)
3   (cond
4     ((eq n 0) 0)
5     ((eq n 1) 1)
6     ( (+ (fibonacci (- n 1)) (fibonacci (- n 2))))))
7
8 (with-open-file (str "text0.txt"
9   :direction :output
10  :if-exists :supersede
11  :if-does-not-exist :create)
12   (loop for a from 0 to 40
13     do (format str (write-to-string(fibonacci a))
14       (format str "~%" ) )
15
```