

Go

Linguagem de programação da
Google

Introdução

Go é uma linguagem de programação de código aberto (open source) desenvolvida pela Google, anunciada em 2009, projetada para programação de sistemas.

Atualmente, há implementações para Windows, Linux, Mac OS X e FreeBSD.

Origens e Influências

Go teve sua origem como um projeto de uma nova linguagem de programação de sistemas, de modo que ela:

- Fosse estaticamente tipada, escalável para grandes sistemas
- Fosse produtiva e legível (readability), sem palavras-chave obrigatórias e repetições demais
- Não requeresse ferramentas, embora suportasse bem elas
- Desse suporte a redes e multiprocessamentos

Com isso, Go teve influências de muitas linguagens, como Alef, APL, BCPL, C, CSP, Limbo, Modula, Newsqueak, Oberon, occam, Pascal, Python e Smalltalk.

Classificação

- Go é uma linguagem de paradigmas imperativa e estruturada.
- Ela é compilada estaticamente e é concorrente.
- Go possui tipagem forte e estática, porém possui também inferência de tipo.

Avaliação Comparativa

Go é no estilo de C, mas têm muitas mudanças para melhorar a concisão, simplicidade e segurança.

Um exemplo de como fazer uma função para "girar" os valores de 3 variáveis poderia ser implementado da seguinte forma:

Avaliação Comparativa

Em C:

```
#include <stdio.h>
void turn(int*,int*,int*);
int main(void) {
    int a = 1;
    int b = 2;
    int c = 3;
    printf("%d %d %d\n", a, b, c);
    turn(&a, &b, &c);
    printf("%d %d %d\n", a, b, c);
    return 0;
}
void turn(int *a, int *b, int *c) {
    int temp = *a;
    *a = *b;
    *b = *c;
    *c = temp;
}
```

Em Go:

```
package main
import "fmt"
func main() {
    a, b, c := 1, 2, 3
    fmt.Println(a,b,c)
    turn(&a, &b, &c)
    fmt.Println(a,b,c)
}
func turn(a *int, b *int, c *int) {
    *a, *b, *c = *b, *c, *a
}
```

Avaliação Comparativa

- Qual linguagem fica mais claro o algoritmo que está sendo usado (mesmo em um exemplo simples desse)?
- Qual linguagem você conseguiria escrever suas lógicas sem necessitar de escrever muitos detalhes da linguagem?
- Qual linguagem você teria mais facilidade em escrever um algoritmo?

É fácil perceber que a resposta às 3 perguntas é "Go", ou seja, Go tem mais readability, writability e expressiveness, respectivamente à ordem das perguntas, que C.

Avaliação Comparativa

Além disso, vale saber que ela possui algumas características muito importantes não contidos na linguagem C, como:

- Sensação de uma linguagem dinâmica
- Possui Garbage Collection (eficiente)
- Concorrência primitiva embutida: "light-weight processes" (goroutines), "channels", e o "select statement"
- Uma interface de sistema no lugar da herança virtual, tipo e embutido em vez de herança não virtual
- Um conjunto de ferramentas que, por padrão, produz estaticamente binários nativos já "linkados" (linked), sem dependências externas

Avaliação Comparativa

Um exemplo de concorrência em Go que mostra sua goroutine com o uso de channel, um novo recurso que a linguagem trouxe:

```
package main
import "fmt"
func main() {
    ch := make(chan int)
    go func(c chan int) { c <- 1; fmt.Println(2); }(ch)
    go func(ch chan int) { ch <- 3; fmt.Println(4); }(ch)
    fmt.Println(<- ch)
    fmt.Println(<- ch)
}
```

Avaliação Comparativa

O resultado desse programa será:

4
3
2
1

Isso ocorre pois esse programa usa um channel (chan) em cada goroutines (funções chamadas após o comando "go"). Nesse caso, temos 2 funções anônimas que simplesmente imprimem 2 e 4, respectivamente. No entanto, elas recebem por parâmetro um channel e um valor (1 e 3) é recebido (canal<-) por cada um dos channels, o que quer dizer que as funções estão bloqueadas para rodar até esses channels serem enviados (<-canal), liberando sua execução. Um detalhe aqui é que os channels funcionam como uma pilha, empilhando assim seus processos, e por isso a ordem dos números saiu invertida.

Esse exemplo, embora simples, mostra que esse é um recurso muito importante, por exemplo, para trabalhar com deadlocks, dentre outros problemas envolvendo concorrência.

Conclusão

Go é uma linguagem nova (moderna), que ainda está em desenvolvimento, na versão atual 1.6.

Por suas características, ela não é muito produtivo para projetos em que o nível de abstração é prioridade, como por exemplo, sistemas web, mobile ou até desktop, embora existam frameworks open source para facilitar essas tarefas.

No entanto, a linguagem é fortemente projetada e é muito eficiente para programação de sistemas, em opção à C.

Pode-se dizer que Go é uma extensão de C, ou até uma versão melhorada de C, embora tenham algumas diferenças e Go tenha aspectos de outras linguagens.

Go tem a potência de C com a simplicidade de python.