Le Langage Go

Goroutines & Channels

Déclarer une Goroutine

Robin Penea

Définition

Une Goroutine est un thread léger

Syntaxe

Robin Penea

Syntaxe

```
func LongOperation(i int) {
    // takes many seconds...
    fmt.Println("Done", i)
}
```

Syntaxe

```
func LongOperation(i int) {
    // takes many seconds...
    fmt.Println("Done", i)
}

LongOperation(1)
LongOperation(2)
LongOperation(3)
```

Syntaxe

```
func LongOperation(i int) {
    // takes many seconds...
    fmt.Println("Done", i)
}

LongOperation(1)
LongOperation(2)
LongOperation(3)
// output: Done 1, Done 2, Done 3
```

Syntaxe

```
func LongOperation(i int) {
    // takes many seconds...
    fmt.Println("Done", i)
}

go LongOperation(1)
go LongOperation(2)
go LongOperation(3)
```

Syntaxe

```
func LongOperation(i int) {
    // takes many seconds...
    fmt.Println("Done", i)
}

go LongOperation(1)
go LongOperation(2)
go LongOperation(3)
// output: Done 2, Done 3, Done 1
```

Léger ?

Ne s'appuie pas sur le mécanisme des threads de l'OS !

Les Goroutines sont gérées par le moteur de Go

Conséquences

On peut exécuter plus de Goroutines

Pour 1000 threads, des millions de Goroutines

Les Goroutines démarrent plus rapidement

Limitations

Capacités réduites

L'OS garanti le démarrage d'un thread, pas d'une Goroutine

"Sécurité"

L'OS est méfiant d'un thread, pas d'une Goroutine