Code Explanation

package main

import "fmt"

func main() {

cnp := make(chan func(), 10)

for i := 0; i < 4; i++ {

go func() {

for f := range cnp {

f()

}

}()

}

cnp <- func() {

fmt.Println("HERE1")

}

fmt.Println("Hello")

}

1. Explanation of Constructs

Channel of Functions: cnp := make(chan func(), 10)

Goroutine: go func() {

for f := range cnp {

f()

}

}()

This launches a new goroutine. A goroutine is a lightweight thread managed by the Go runtime. The anonymous function inside the goroutine continuously reads from the channel cnp. For each function received from the channel, it calls that function.

Buffered Channel: A buffered channel allows sending func values to the channel without an immediate corresponding receive. The channel can hold up to 10 functions before any sender will block.

1. Use-cases of Constructs

**Channel of Functions:** This construct is useful for scenarios where you need to distribute work (in the form of functions) across multiple workers. For example, in a task scheduling system where tasks are represented as functions and dispatched to available workers.

**Goroutines:** Goroutines are used for concurrent execution. They are suitable for performing multiple tasks simultaneously, such as handling multiple requests in a web server or performing background tasks without blocking the main program execution.

1. Significance of the For Loop with 4 Iterations

for i := 0; i < 4; i++ {

go func() {

for f := range cnp {

f()

}

}()

}

This loop launches 4 separate goroutines, each of which continuously reads from the cnp channel. The significance is that it creates a pool of 4 workers that can concurrently process functions received from the channel. This is useful for parallel processing, where multiple tasks can be handled concurrently to improve performance and throughput.

1. Significance of make(chan func(), 10)

cnp <- func() {

fmt.Println("HERE1")

}

The function is enqueued into the channel, but it is never executed. The main function exits immediately after printing "Hello". The program does not wait for the goroutines to process the function from the channel, causing the program to terminate before the goroutines get a chance to execute the function.