PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA

INF239 SISTEMAS OPERATIVOS

Semestre 2024-1 Laboratorio 4

1) (12 puntos – nombre del programa: quintetos.go) Se tiene el siguiente programa en Go:

```
// main.go
    package main
4 4 import (
5
         "fmt"
6
         "sync"
8
9d func task(n int) {
        for x := 0; x < 50; x++ {
104
11
            fmt.Printf("%d", n)
12
13
        wg.Done()
14
   }
15
16 var wg sync.WaitGroup
17
18 ⁴ func main() {
19
        for x := 0; x < 5; x++ \{
204
21
             wg.Add(1)
22
             go task(x)
23
24
        wg.Wait()
25
   }
26
```

Que al ser ejecutado varias veces, muestra el siguiente resultado:

Se le solicita modificar el programa de forma que se imprima quintetos de números diferentes. Cada quinteto debe estar separado por un cambio de línea. Para este problema solo puede hacer uso de canales, ya sea asíncronos como síncronos. Ningún otro tipo de variable está permitida. Esto implica que no puede hacer uso de variables tipo sync. Mutex (Lock, Unlock).

La imagen de la derecha muestra un segmento de una salida esperada.

Se sugiere construir una gorutina que administre la sincronización de las otras gorutinas.

```
01234
41230
30412
24031
30142
24310
21430
30412
23401
04312
24310
42130
01243
31024
42310
32014
41023
34102
20341
12043
34120
12034
41023
```

2) (8 puntos – nombre del programa: secuencia.go)

Se tiene el siguiente programa en Go:

```
package main
 3
    import (
             "fmt"
            "sync"
8
    var wg sync.WaitGroup
10
    func worker1() {
            for {
11
12
                     fmt.Printf("A")
13
14
15
            wg.Done()
    }
16
17
    func worker2() {
18
                     fmt.Printf("B")
19
20
21
            wg.Done()
   }
22
23
    func worker3() {
24
25
              for {
26
                       fmt.Printf("C")
27
28
              wg.Done()
29
    }
30
    func worker4() {
31
32
             for {
3.3
                       fmt.Printf("D")
34
35
             wg.Done()
36
    }
37
    func worker5() {
39
              for {
40
                       fmt.Printf("E")
41
             wg.Done()
42
    }
43
44
45
    func main() {
46
             wg.Add(5)
47
             go worker1()
48
             go worker2()
49
             go worker3()
50
             go worker4()
51
             go worker5()
52
             wg.Wait()
              fmt.Printf("\n")
53
54
```

Este programa al ser ejecutado obtiene la siguiente salida:

```
CCABBBBBBBBBBBBBBBBBBBBBBBBEEEEEEEEE
```

Restricciones:

- 1. Cada gorutina imprime solo una letra.
- 2. Solo puede hacer uso de canales, ya sea asíncronos como síncronos.
- 3. Ningún otro tipo de variable está permitida. Esto implica que no puede hacer uso de variables tipo sync. Mutex (Lock, Unlock).
- 4. No puede emplear variables locales dentro de las gorutinas.

Porf. Alejandro T. Bello Ruiz