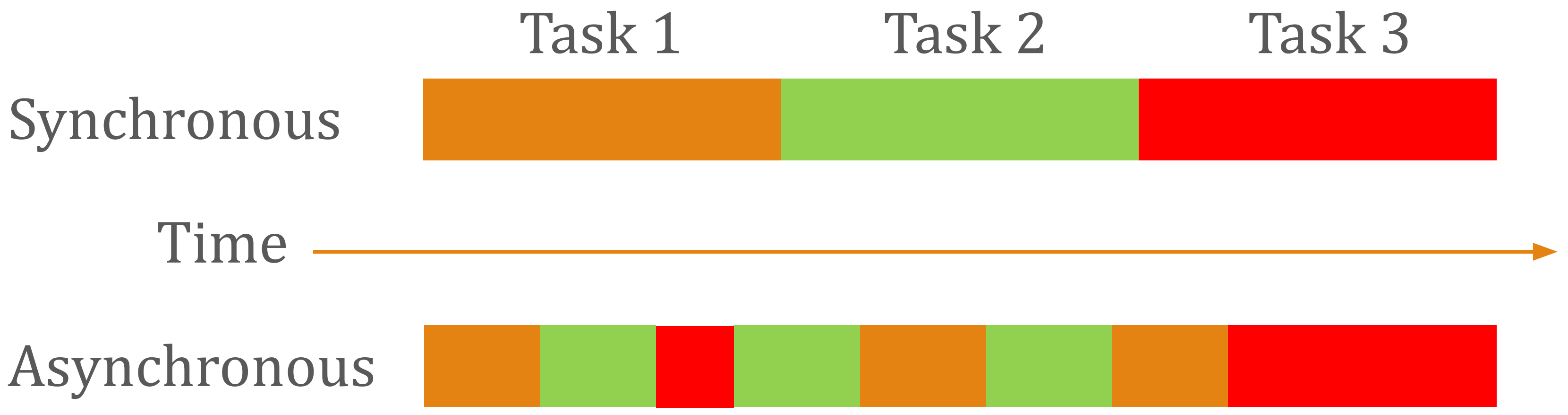
Lab7

NYCU Go Programming 2024

2024/11/26

Sync vs Async



Goroutines

- GoStmt = "go" Expression .
  - A "go" statement starts the execution of a function call as an independent concurrent thread of control, or goroutine, within the same address space.
- type WaitGroup
  - A WaitGroup waits for a collection of goroutines to finish. The main goroutine calls Add to set the number of goroutines to wait for. Then each of the goroutines runs and calls Done when finished. At the same time, Wait can be used to block until all goroutines have finished.
  - func (wg *WaitGroup) Add(delta int)
  - func (wg *WaitGroup) Done()
  - func (wg *WaitGroup) Wait()
- Reference:Program Execution

Critical section

The protected section cannot be entered by more than one process or thread at a time.

Process A:

```
  x = x + 1
```

Process B:

```
  x = x + 1
```

type Mutex

- func (*Mutex) Lock
  - Lock locks m. If the lock is already in use, the calling goroutine blocks until the mutex is available.
- func (*Mutex) TryLock
  - TryLock tries to lock m and reports whether it succeeded.
- func (*Mutex) Unlock
  - Unlock unlocks m. It is a run-time error if m is not locked on entry to Unlock.

Lab7: 夾到手了

程式有兩個 goroutines, door() 和 hand(),

hand() 會把手伸進去再拿出來, door() 會把門關起來再打開,

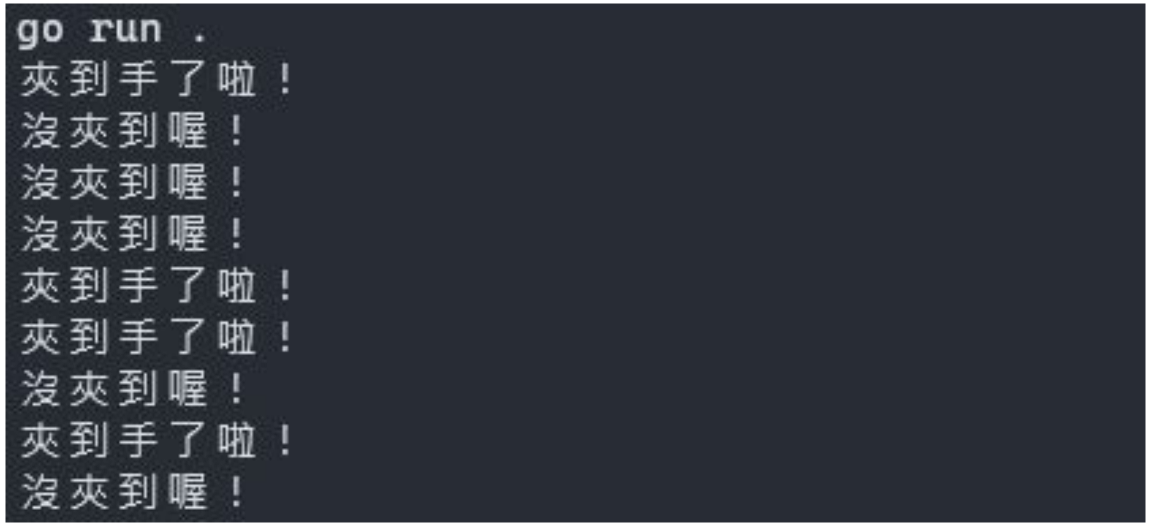
如果 hand() 手伸進去後, 換成 door() 把門關起來, 就會夾到手,

請加入 critical sections 來避免此狀況發生。

Lab7: 夾到手了

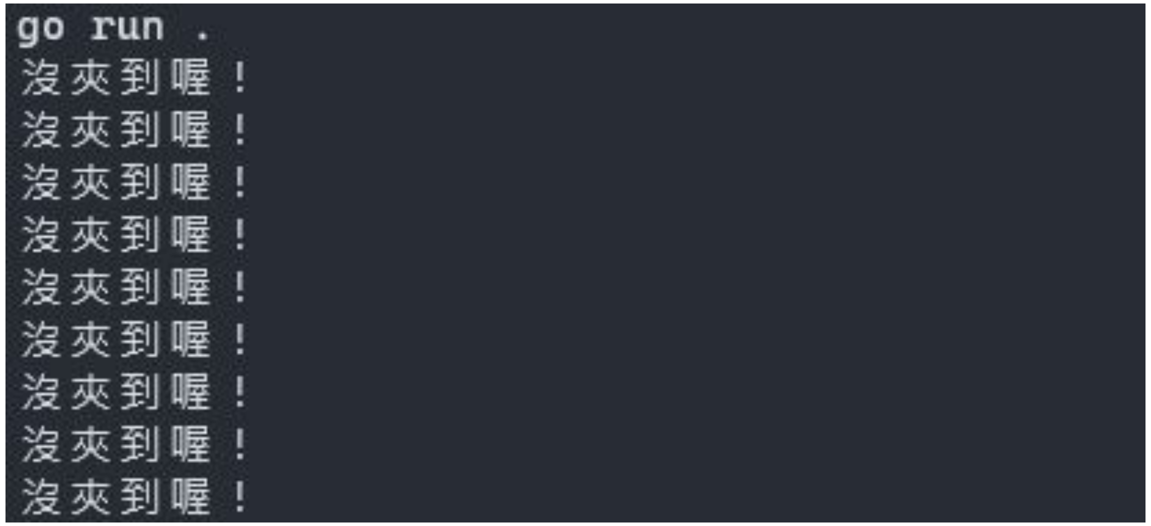
請達成連續 50 次沒有夾到手的紀錄

Before



```
go run .
夾到手了啦！
沒夾到喔！
沒夾到喔！
沒夾到喔！
夾到手了啦！
夾到手了啦！
沒夾到喔！
夾到手了啦！
沒夾到喔！
```

After



```
go run .
沒夾到喔！
沒夾到喔！
沒夾到喔！
沒夾到喔！
沒夾到喔！
沒夾到喔！
沒夾到喔！
沒夾到喔！
沒夾到喔！
```

Lab7: 夾到手了

- .github
  - workflows
    - lab7.yml
- Lab7
  - go.mod
  - go.sum
  - lab7.go
  - lab7_test.go

```
$ go test
```