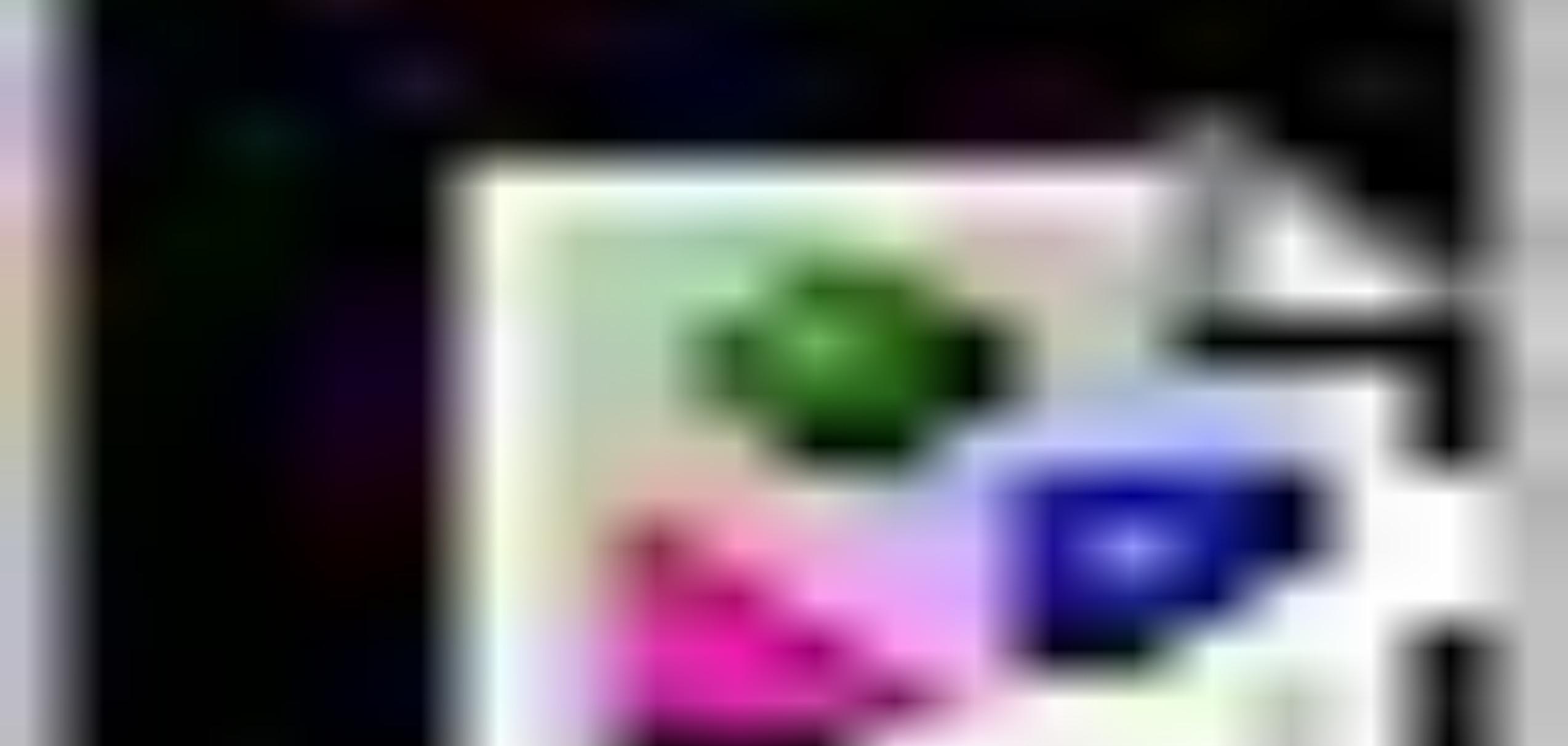




Jesús Espino - Software Engineer @Mattermost

How are goroutines born?





How are goroutines born?

The go scheduler



The pieces of the scheduler

P (the Processor)



- Status (Idle, running, syscall, gcstop)
- Current M
- Runnable goroutines
- Free goroutines
- Other metadata

The pieces of the scheduler

M (the Machine)



- Current goroutine
- Current P
- Other metadata

The pieces of the scheduler

Sched (the Scheduler)



- Idle Ms
- Idle Ps
- Global runnable goroutines
- Global free goroutines
- Other metadata



The pieces of the scheduler

G (the Goroutine)



- Stack (2048 bytes)
- Program counter
- Status
- Current M
- Wait reason
- Other metadata.

The whole picture



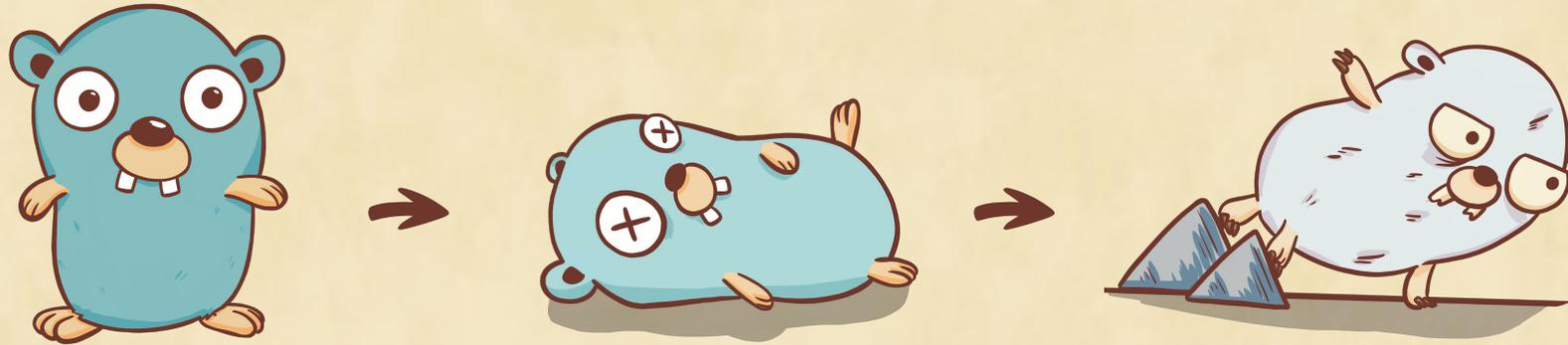
- All Ms
- All Ps
- All Goroutines



The birth of a goroutine



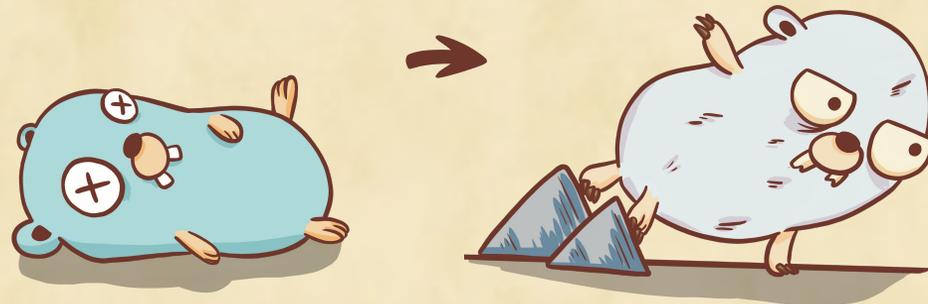
The goroutine lifecycle



Idle

Dead

Runnable



Dead

Runnable



The whole picture



- All Ms
- All Ps
- All Goroutines



The whole picture



- All Ms
- All Ps
- All Goroutines



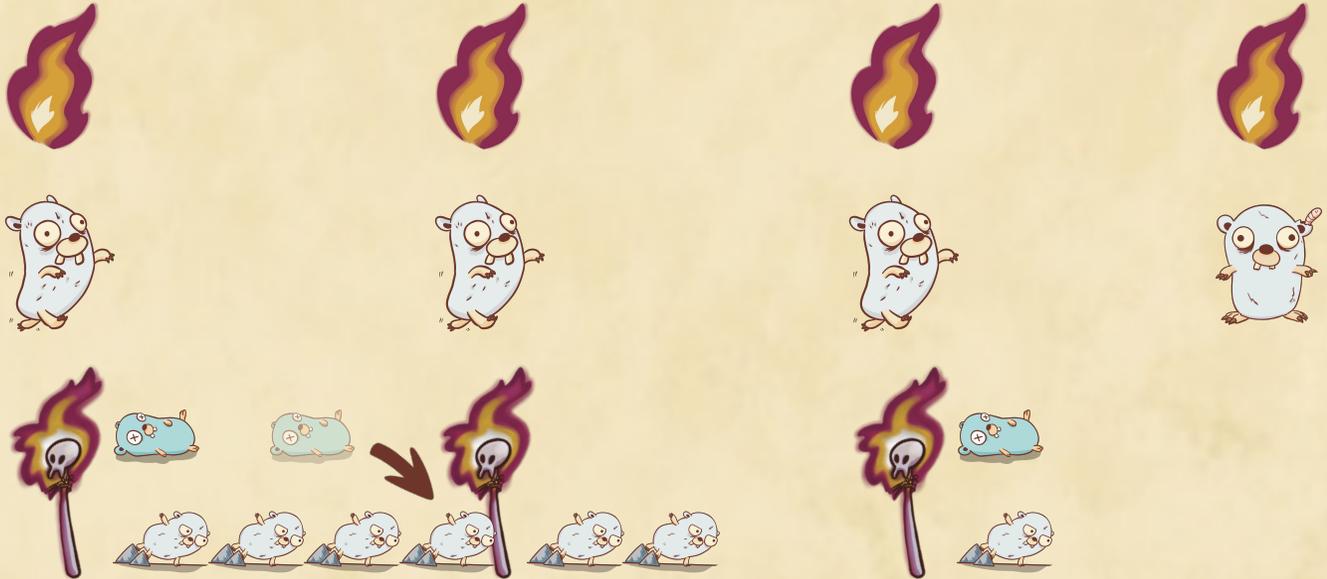
The whole picture



- All Ms
- All Ps
- All Goroutines



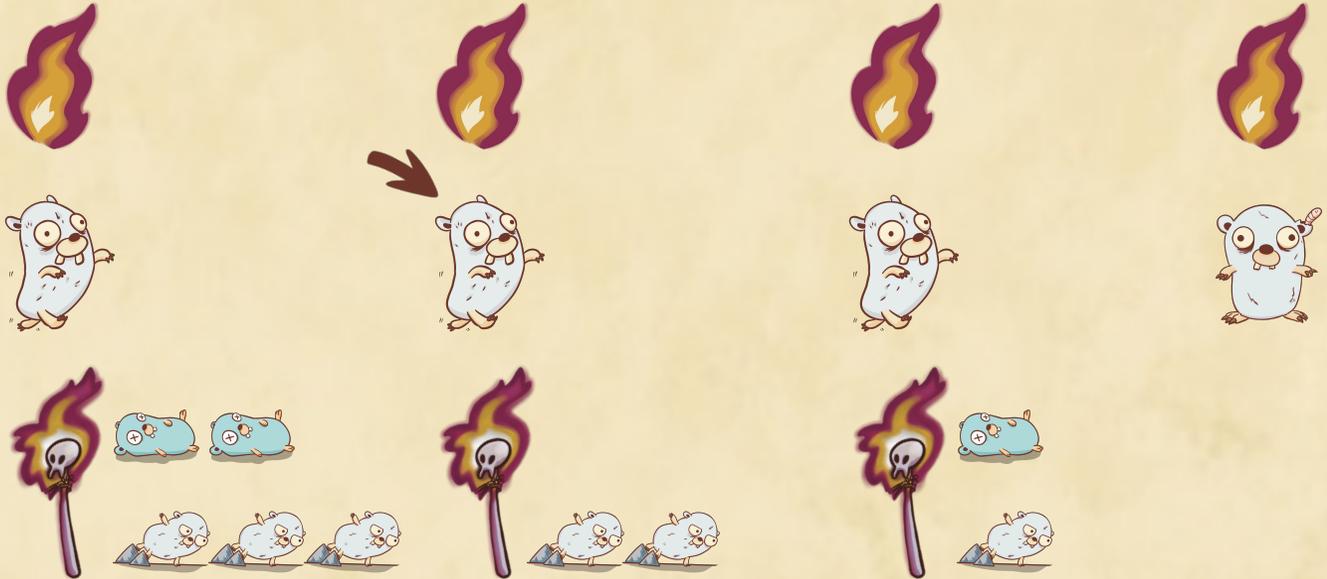
The whole picture



- All Ms
- All Ps
- All Goroutines



The whole picture



- All Ms
- All Ps
- All Goroutines



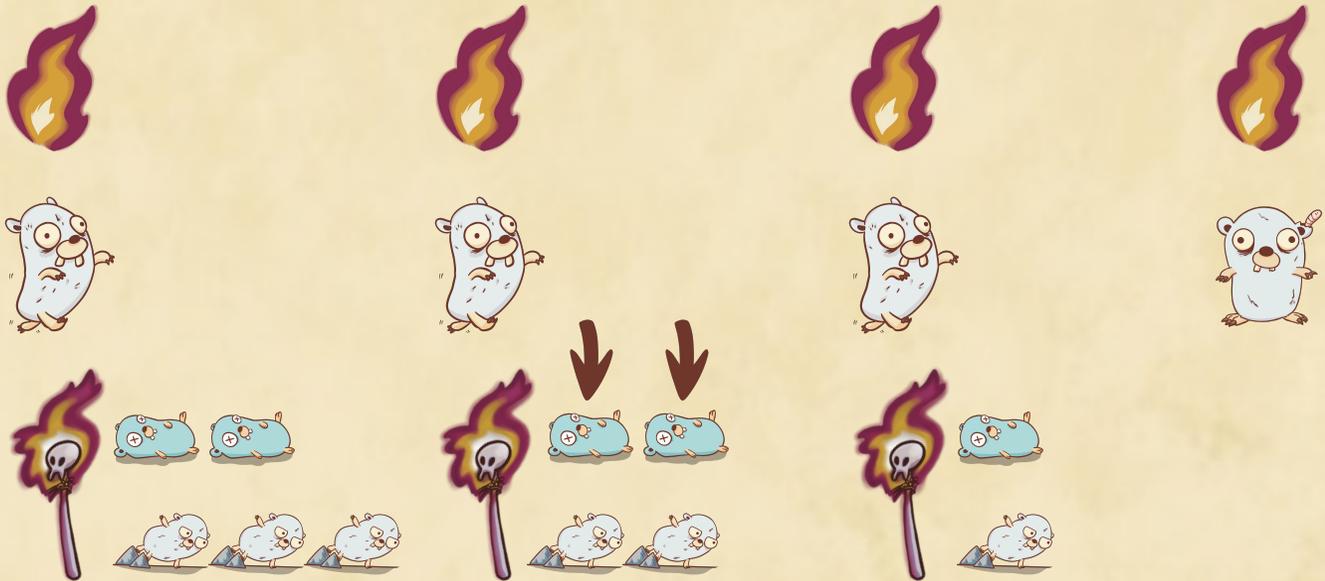
The whole picture



- All Ms
- All Ps
- All Goroutines



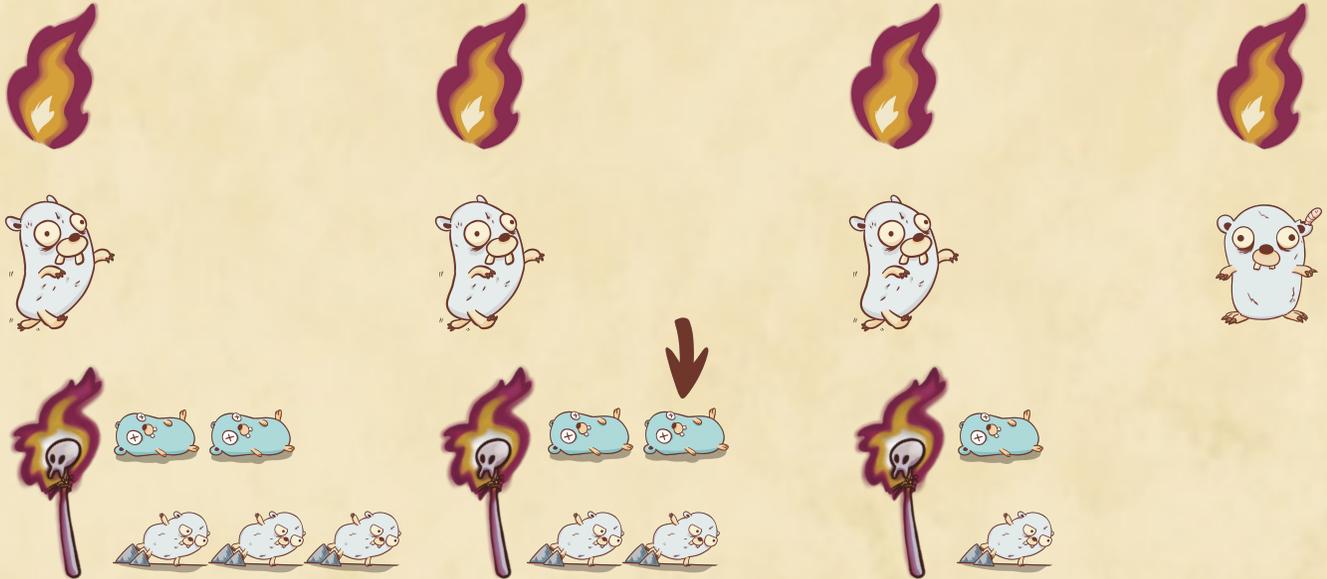
The whole picture



- All Ms
- All Ps
- All Goroutines



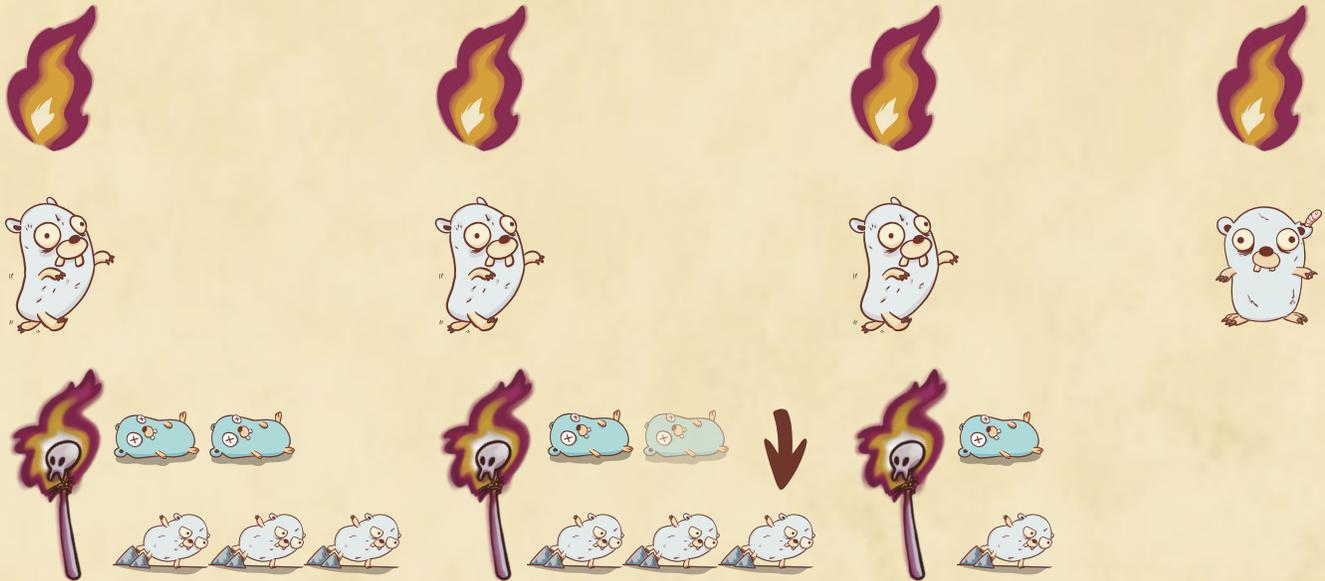
The whole picture



- All Ms
- All Ps
- All Goroutines



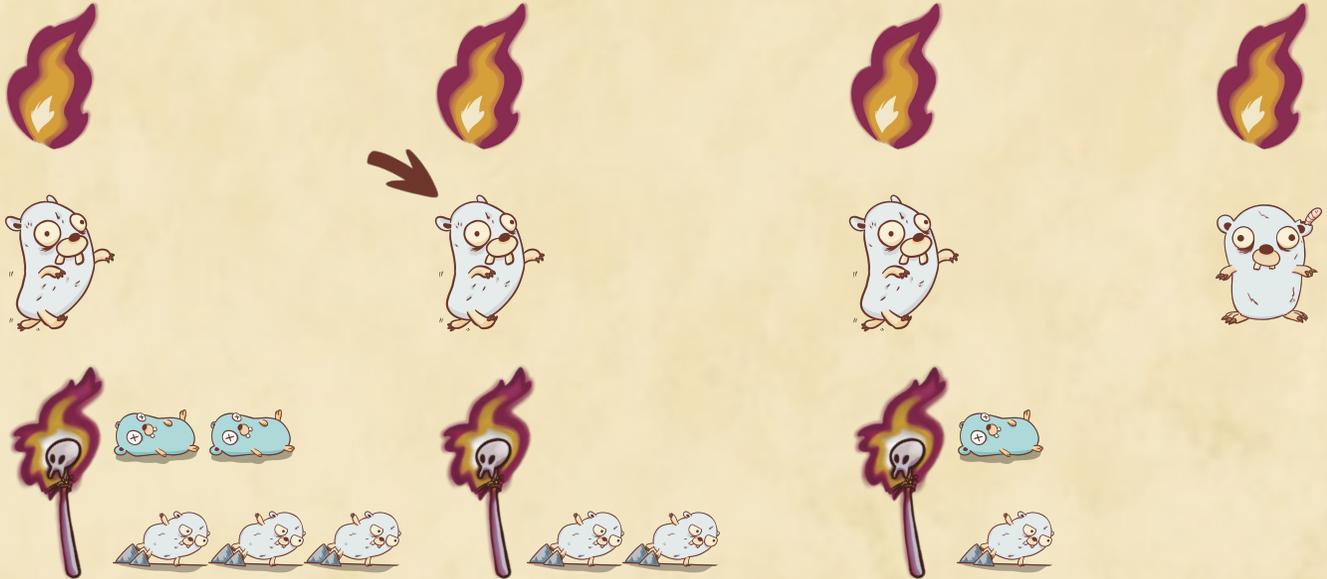
The whole picture



- All Ms
- All Ps
- All Goroutines



The whole picture



- All Ms
- All Ps
- All Goroutines



The whole picture



- All Ms
- All Ps
- All Goroutines



The whole picture



- All Ms
- All Ps
- All Goroutines



The whole picture



- All Ms
- All Ps
- All Goroutines



The life of a goroutine



The life of a goroutine



Runnable



Running



Syscall



Waiting



Preempted



Copystack



Runnable to running



- Scheduler runs
- Find a goroutine to run
- Assigns it to the M
- Mark as Running
- Executes the code



Running to waiting



- Park itself
- Detach from M
- Run the scheduler



Running to waiting



```
waitReasonZero // ""
waitReasonGCAssistMarking // "GC assist marking"
waitReasonIOWait // "IO wait"
waitReasonChanReceiveNilChan // "chan receive (nil chan)"
waitReasonChanSendNilChan // "chan send (nil chan)"
waitReasonDumpingHeap // "dumping heap"
waitReasonGarbageCollection // "garbage collection"
waitReasonGarbageCollectionScan // "garbage collection scan"
waitReasonPanicWait // "panicwait"
waitReasonSelect // "select"
waitReasonSelectNoCases // "select (no cases)"
waitReasonGCAssistWait // "GC assist wait"
waitReasonGCSweepWait // "GC sweep wait"
waitReasonGCScavengeWait // "GC scavenge wait"
waitReasonChanReceive // "chan receive"
waitReasonChanSend // "chan send"
waitReasonFinalizerWait // "finalizer wait"
waitReasonForceGCIdle // "force gc (idle)"
waitReasonSemacquire // "semacquire"
waitReasonSleep // "sleep"
waitReasonSyncCondWait // "sync.Cond.Wait"
waitReasonSyncMutexLock // "sync.Mutex.Lock"
waitReasonSyncRWMutexRLock // "sync.RWMutex.RLock"
waitReasonSyncRWMutexLock // "sync.RWMutex.Lock"
waitReasonTraceReaderBlocked // "trace reader (blocked)"
waitReasonWaitForGCCycle // "wait for GC cycle"
waitReasonGCWorkerIdle // "GC worker (idle)"
waitReasonGCWorkerActive // "GC worker (active)"
waitReasonPreempted // "preempted"
waitReasonDebugCall // "debug call"
waitReasonGCMarkTermination // "GC mark termination"
waitReasonStoppingTheWorld // "stopping the world"
```



Running to waiting



- GC
- Mutex
- Semaphore
- Channel
- Sleep
- IO



Running to syscall and to running or runnable



- On every syscall
- entersyscall is executed (moving to Sycall state)
- The syscall is executed
- exitsyscall is executed (moving back to Running/Runnable)



Running to copystack and back



- More stack needed
- Change Running to Copystack
- Grow the stack
- Change back to Running



Waiting to runnable



- goready is called
- Is added to the queue
- Try to get a P



Waiting to runnable



- Reactivate a list of goroutines
- Mark all of them as runnable
- Wakes up all the Ps needed
- Add them to the queues



Waiting to runnable



- Change to Waiting
- Don't need to wait
- Change to Runnable
- And then to Running rights away



Waiting to runnable



- Finding a goroutine
- Check in the netpoll
- Or wake up tasks for mark assist



Running to preempt, waiting and runnable



- Preempt flag is set
- Change to preempted
- Next GC change to waiting
- Do the GC scan
- Change to Runnable
- Add it back to the queue



The goroutine lifecycle



Runnable



Running



Syscall



Waiting



Preempted



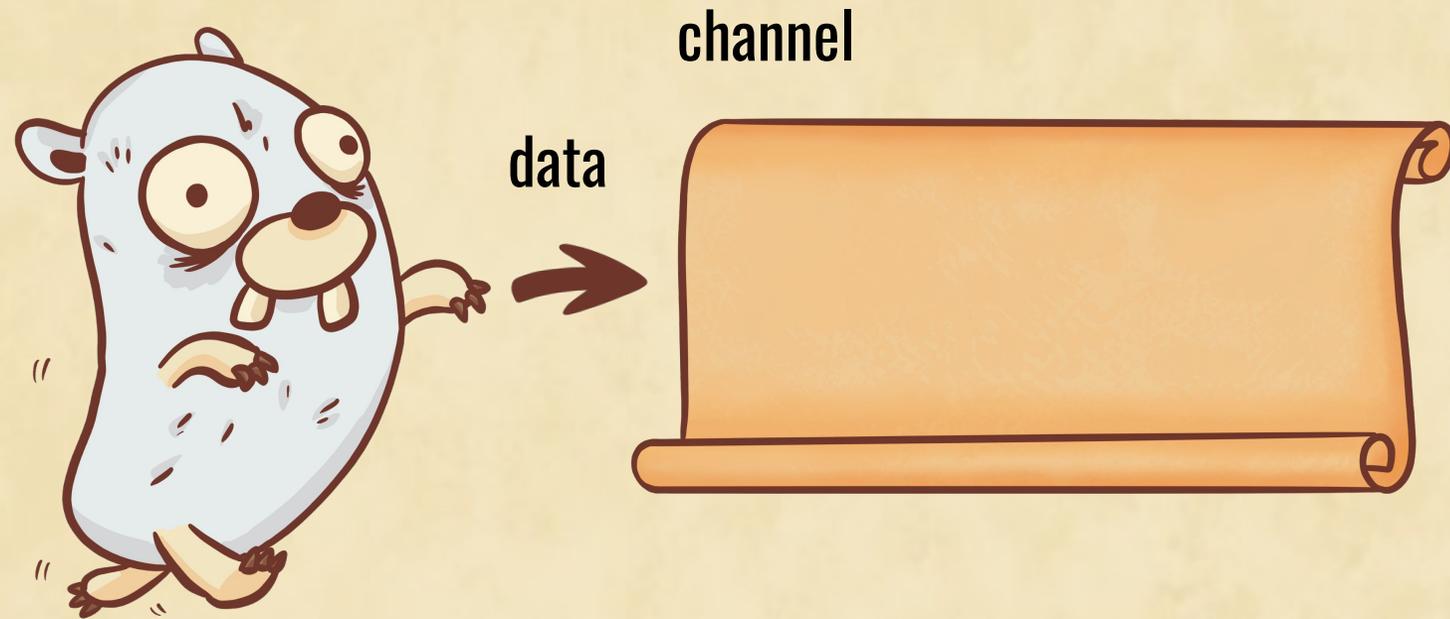
Copystack



Examples

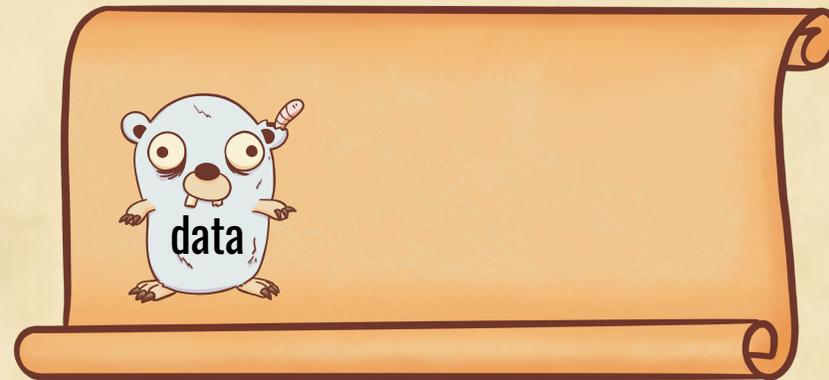


The channel example



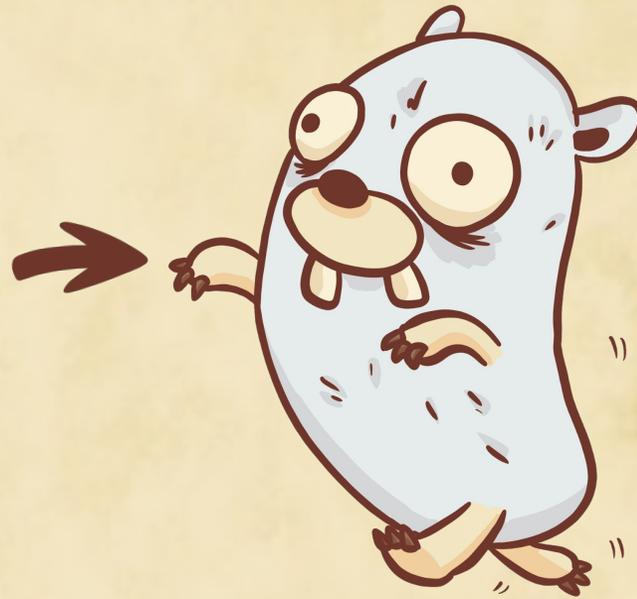
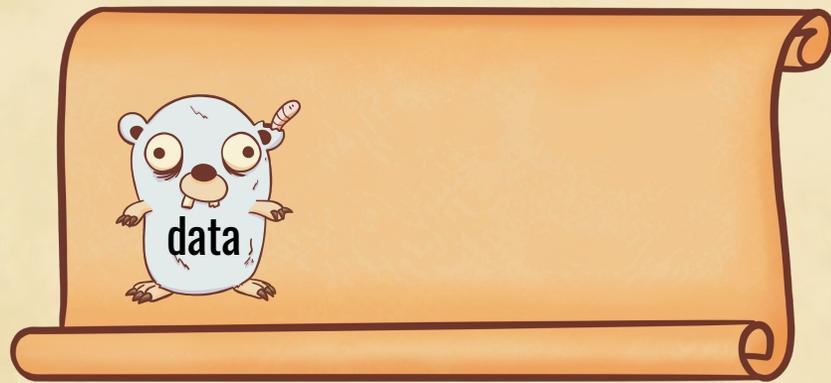
The channel example

channel



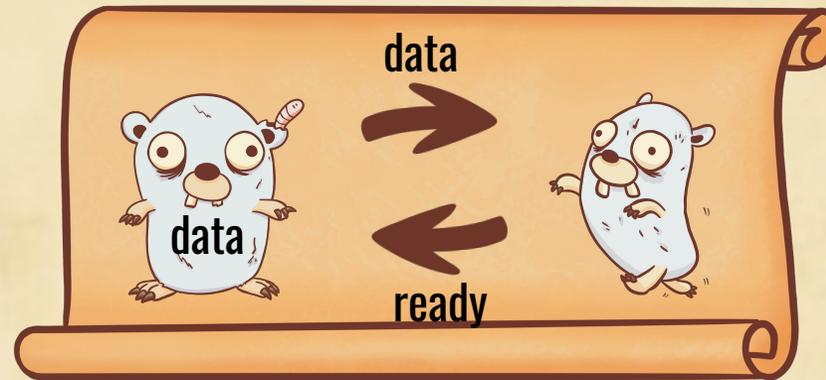
The channel example

channel

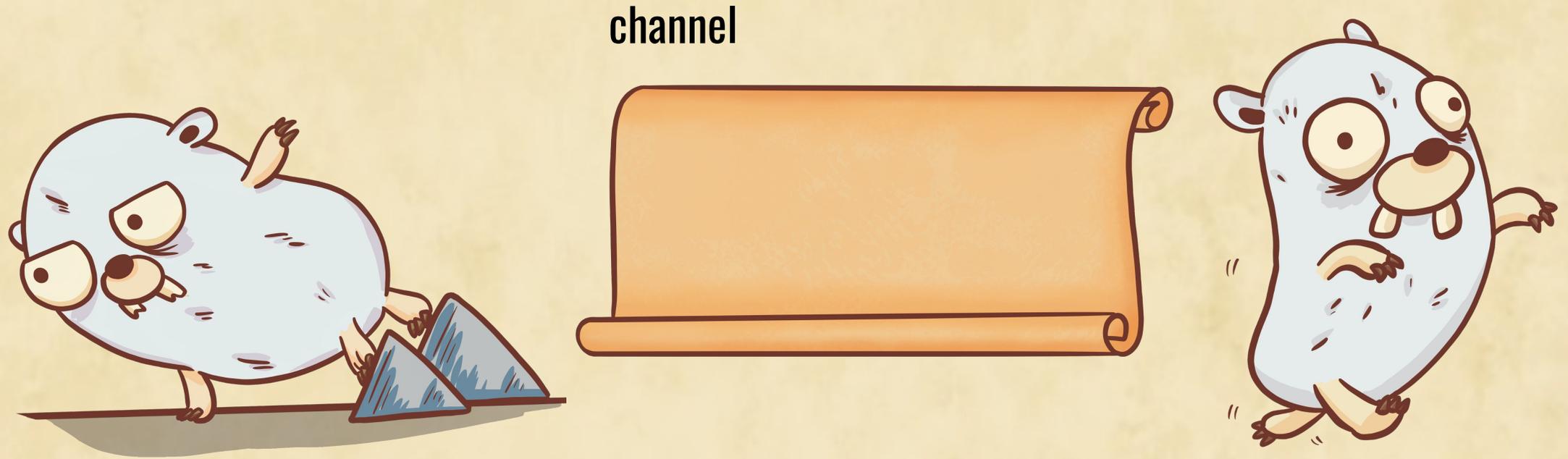


The channel example

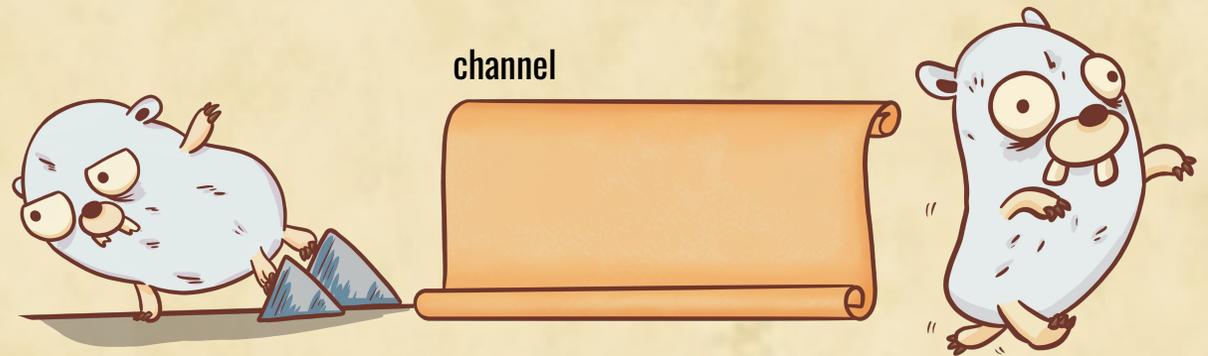
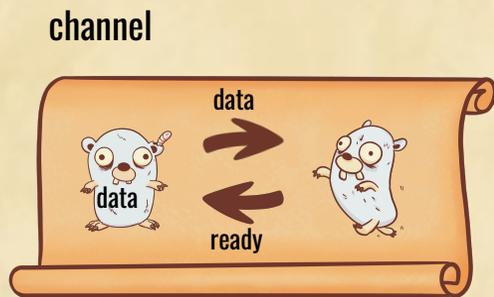
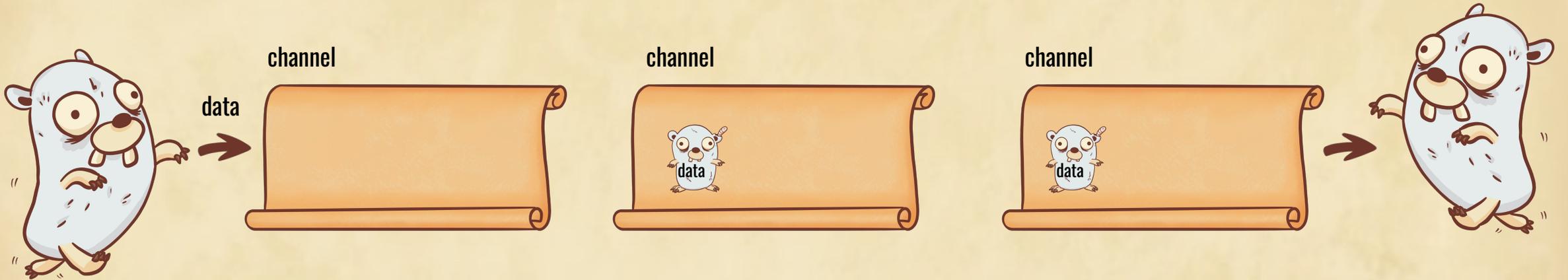
channel



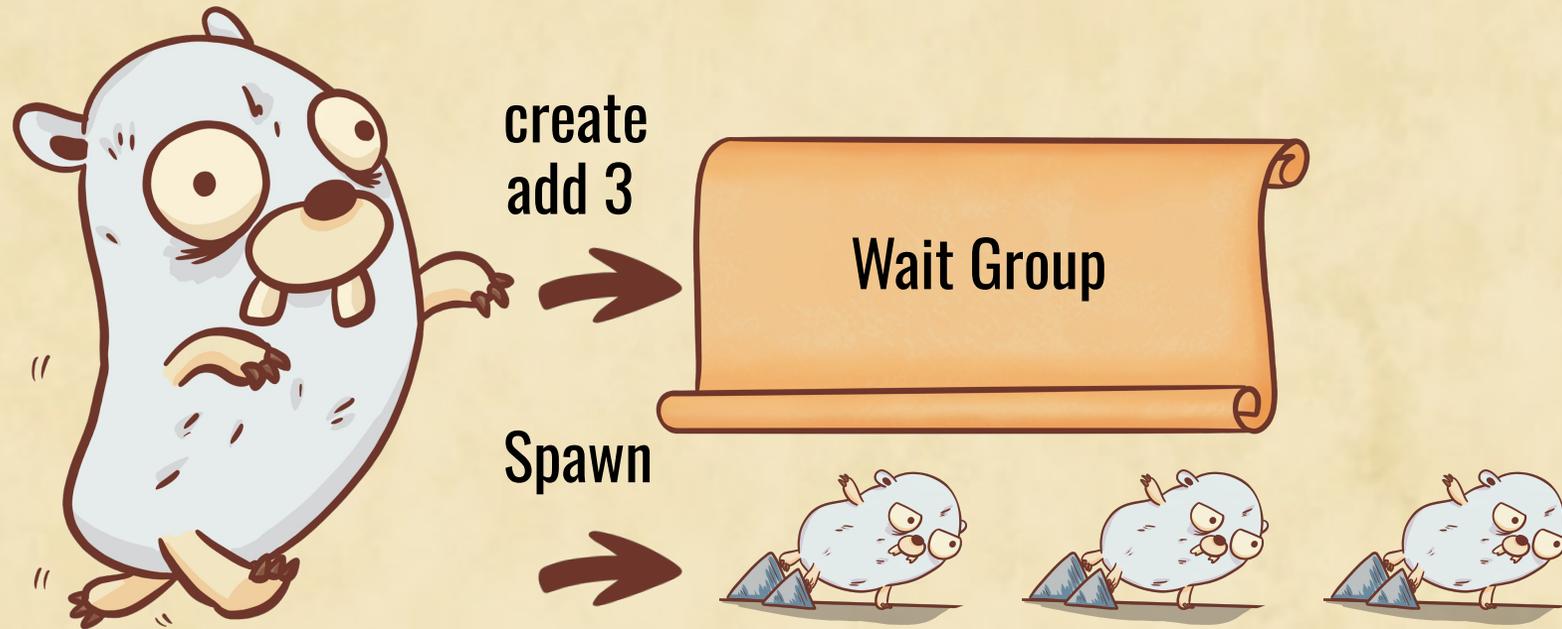
The channel example



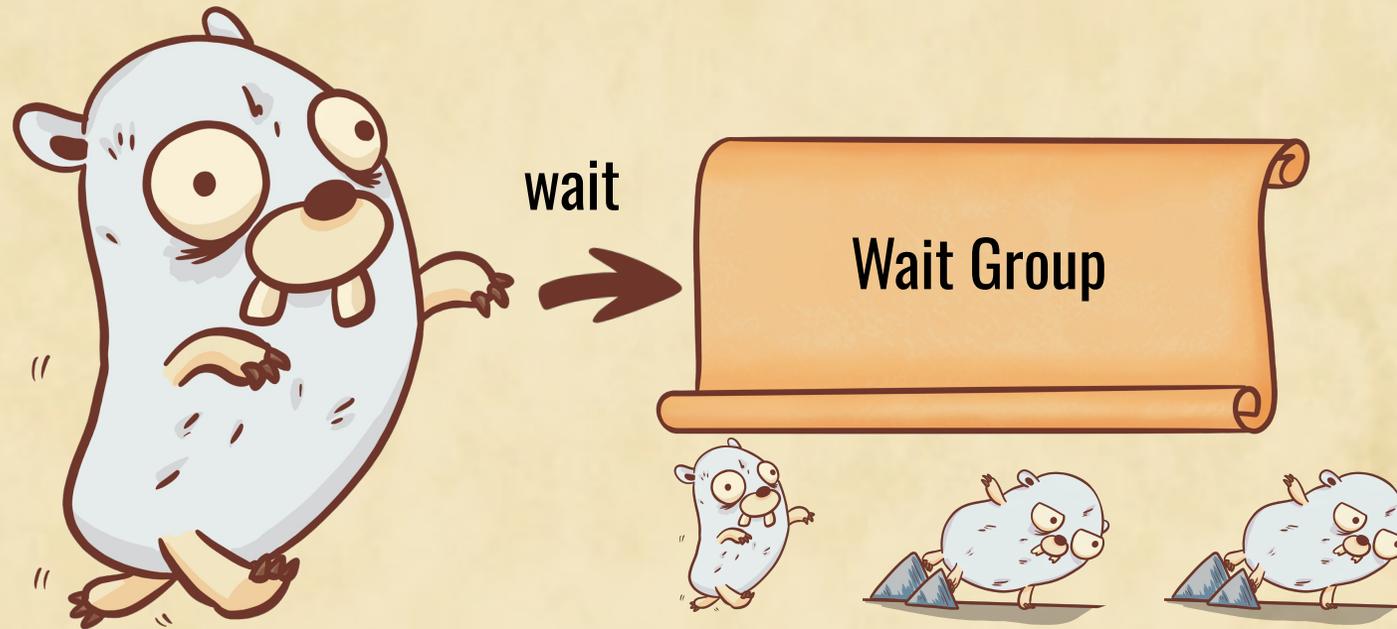
The channel example



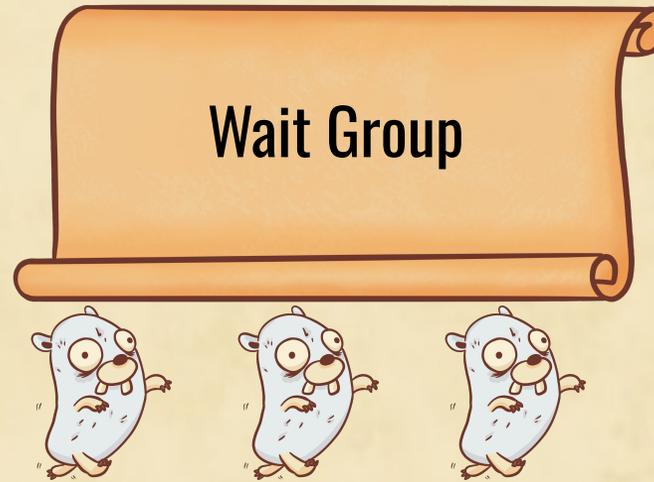
The waitgroup example



The waitgroup example



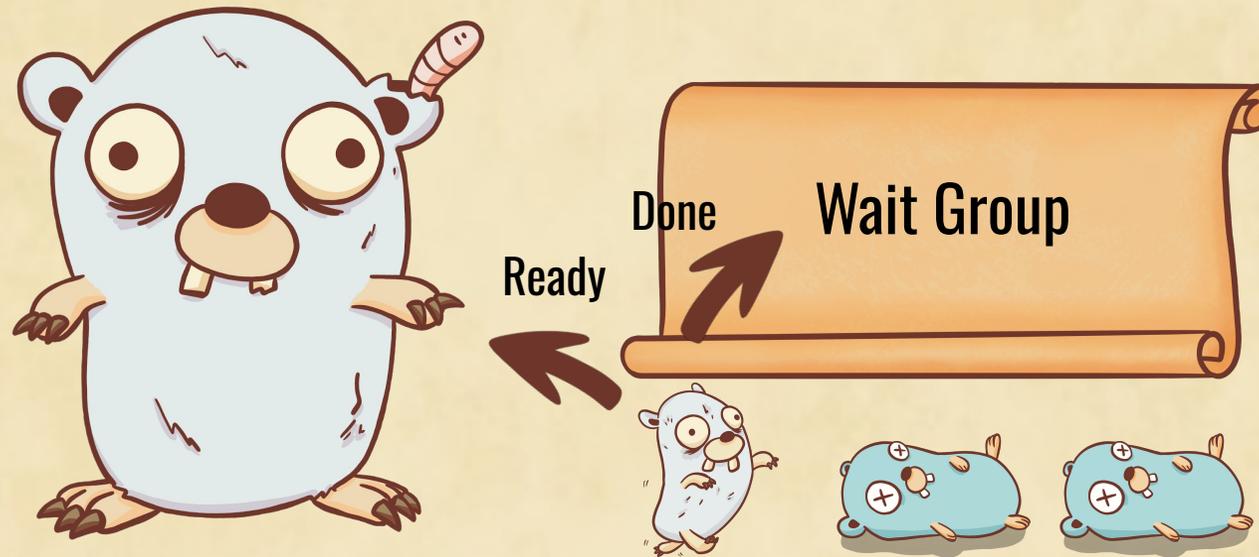
The waitgroup example



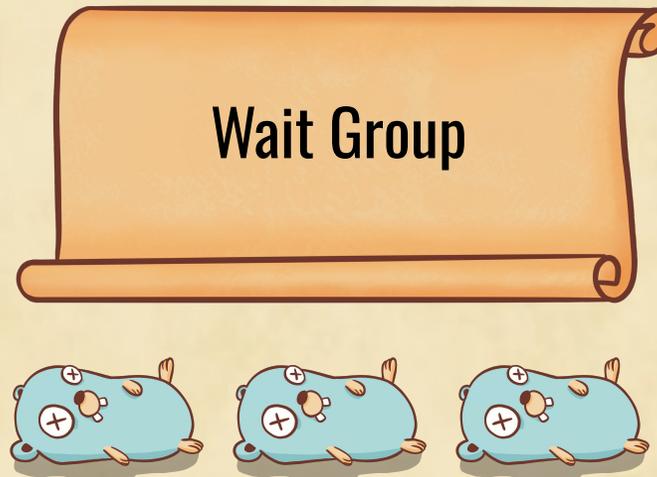
The waitgroup example



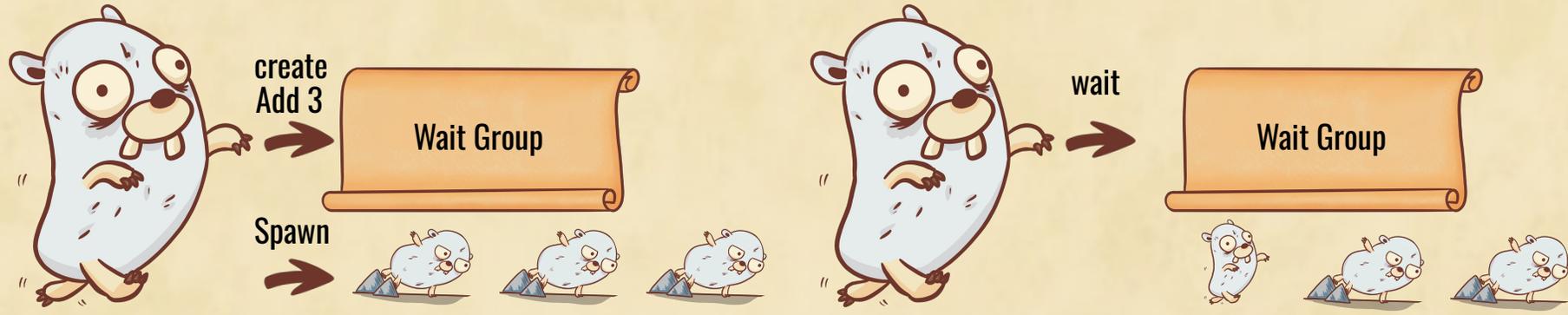
The waitgroup example



The waitgroup example



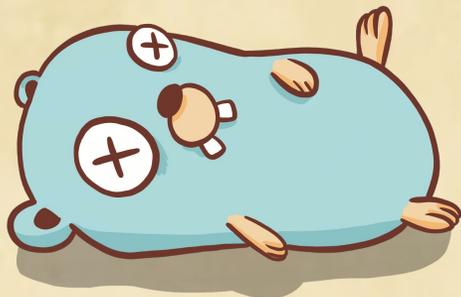
The waitgroup example



The death of a goroutine



The death of a goroutine



- A goroutine finish it work
- Change state to Dead
- Set most of the goroutine values to zero
- Disconnect the goroutine from the M
- Add the goroutine to the free list of the P
- Call the scheduler



The whole life



The whole picture



All Ms



All Ps



All Goroutines



Summary



Illustrations

- CC-BY
- Created by Laura Pareja
- <http://laurapareja.com>



A gift from Mattermost



What is missing?

- The garbage collector
- The netpoll
- Cgo
- Mark assist
- Sysmon

References

- The Go source code
- Illustrated Tales of Go Runtime Scheduler:
<https://www.youtube.com/watch?v=KxOwt6z0FvY>
- Scheduling In Go:
<https://www.ardanlabs.com/blog/2018/08/scheduling-in-go-part1.html>

Conclusions





Let's keep in touch



jespinog



jespino



jesus-espino



Let's keep in touch



<https://forms.gle/rAFKZwVM4U26JPyn6>