

# Concurrency



Nigel Poulton

@nigelpoulton | [www.nigelpoulton.com](http://www.nigelpoulton.com)

# Module Overview



What is *concurrency*?

Go's concurrency model

Concurrency vs Parallelism

Write some concurrent code

## **Module goal:**

Understand enough about  
concurrency and be able to start  
writing simple concurrent code



Mike Van Sickle



## Concurrent Programming with Go

# Module Overview



What is *concurrency*?

Go's concurrency model

Concurrency vs Parallelism

Write some concurrent code

---

# What is Concurrency?

---

# Concurrency:

Creating multiple processes that execute independently

**Concurrency:**  
Creating multiple processes that  
execute *independently*



# This morning...



Deal with emails



Edit podcast

“... concurrency is the *composition* of independently executing processes, while parallelism is the simultaneous *execution* of (possibly related) computations. Concurrency is about *dealing with* lots of things at once. Parallelism is about *doing* lots of things at once.”

— Rob Pike

# This morning...



Deal with emails

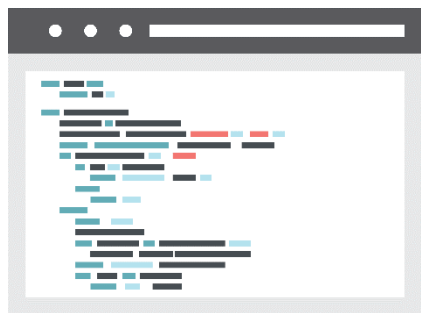


Edit podcast

---

# Concurrent Programs

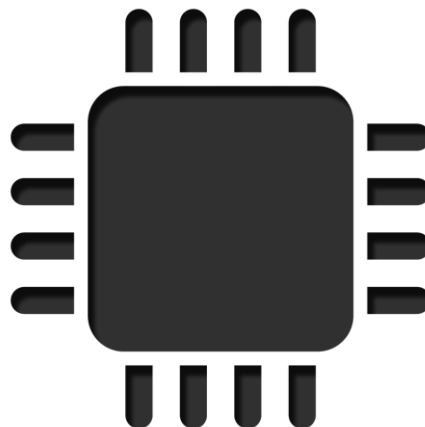
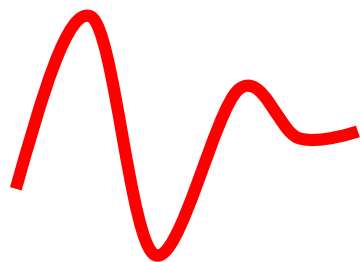
---

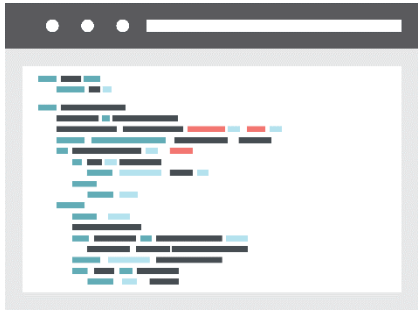


App



Process

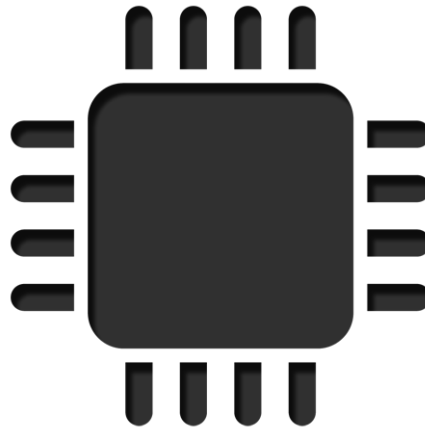
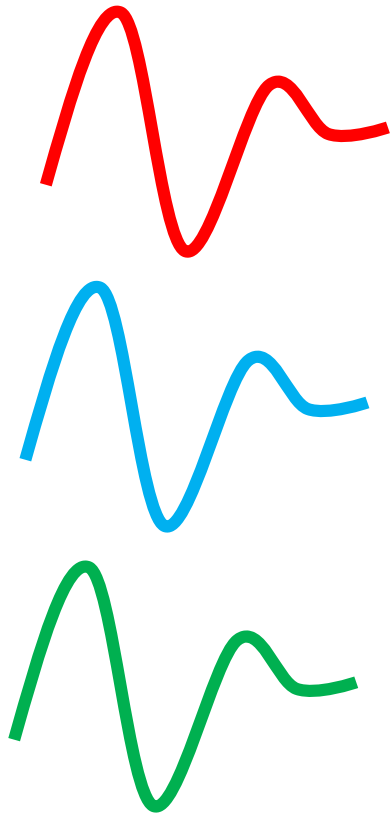




App



Process

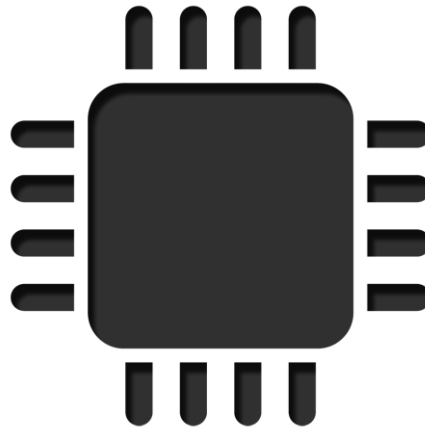
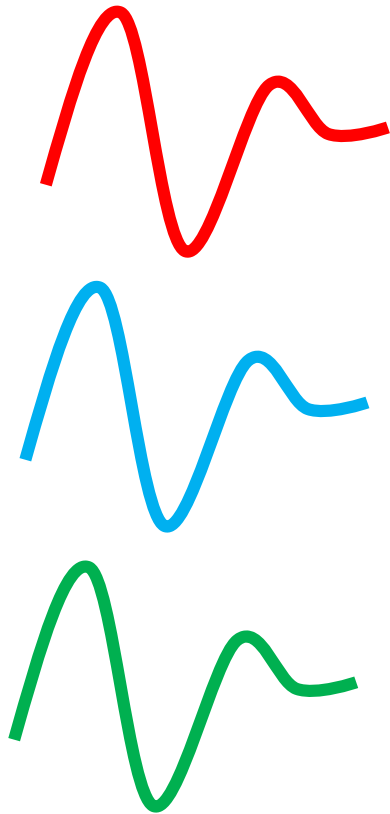




App



Process

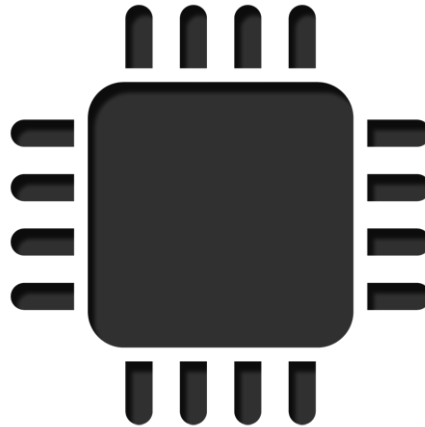
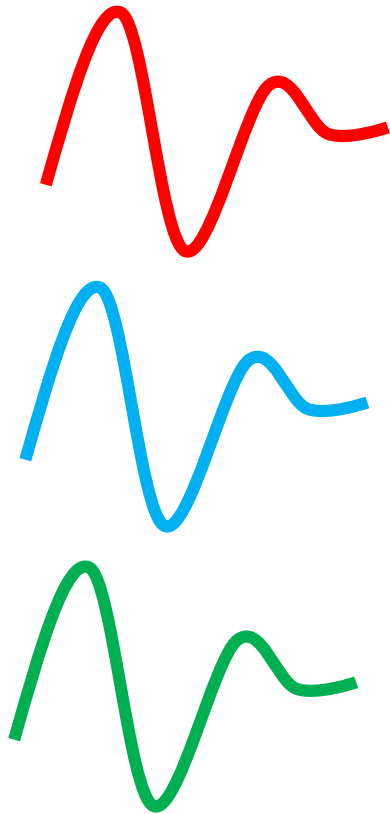




App



Process



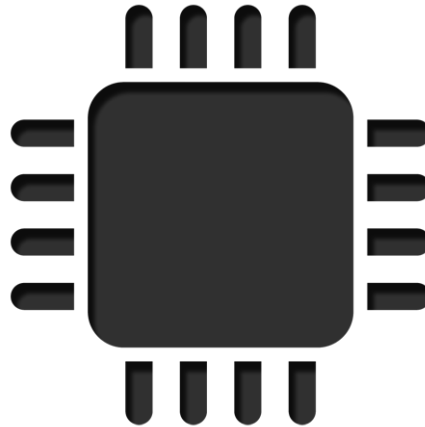
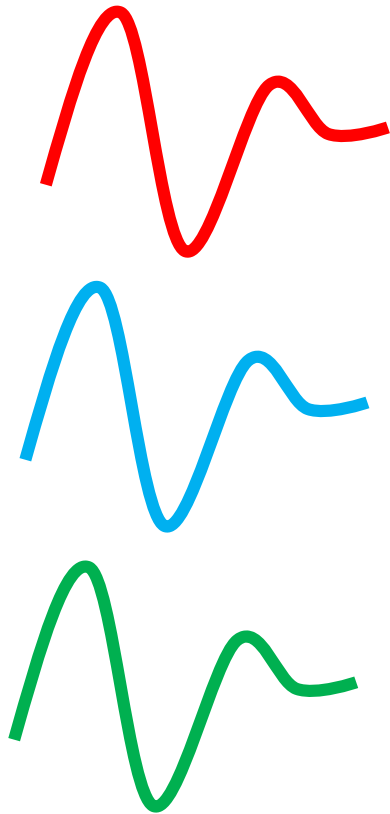


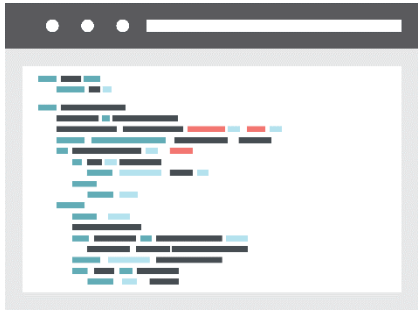


App



Process

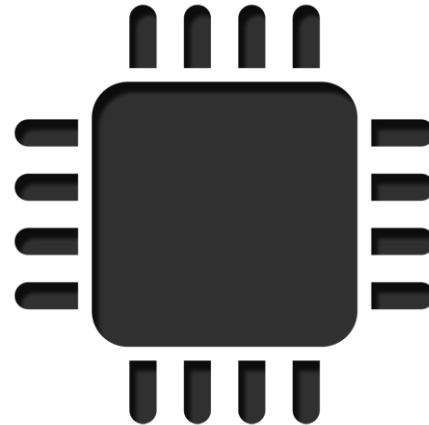
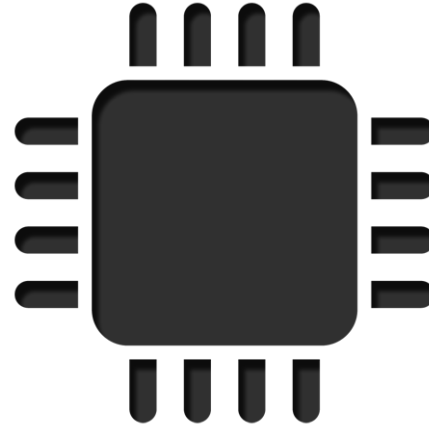
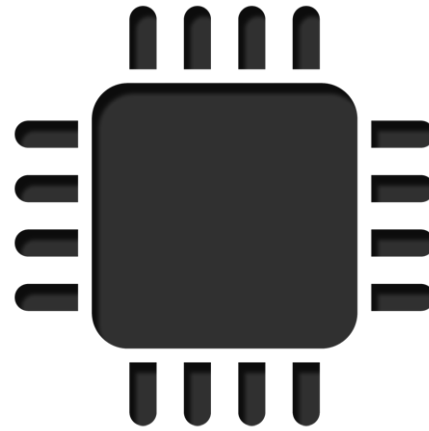
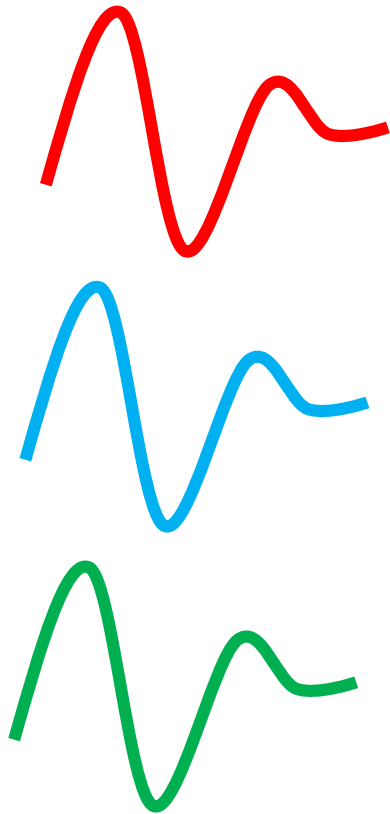




App



Process

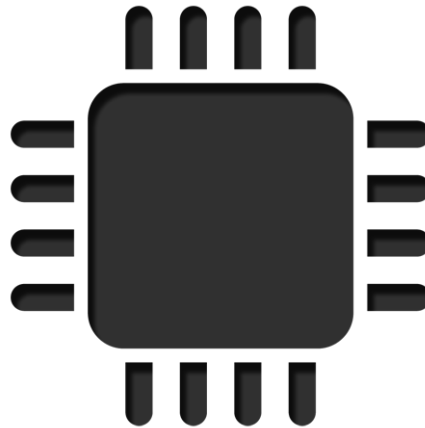
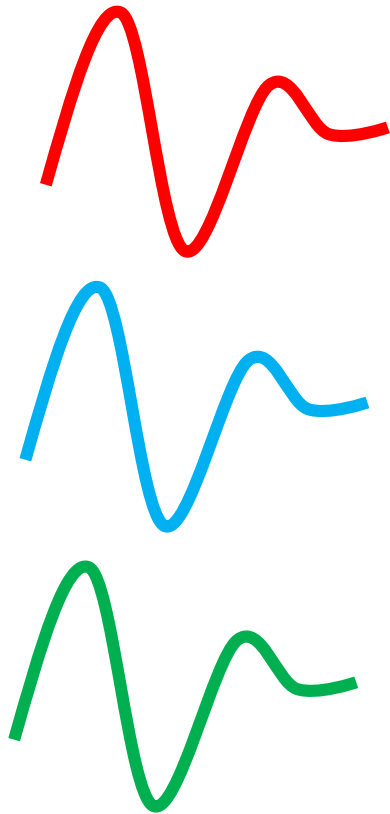




App



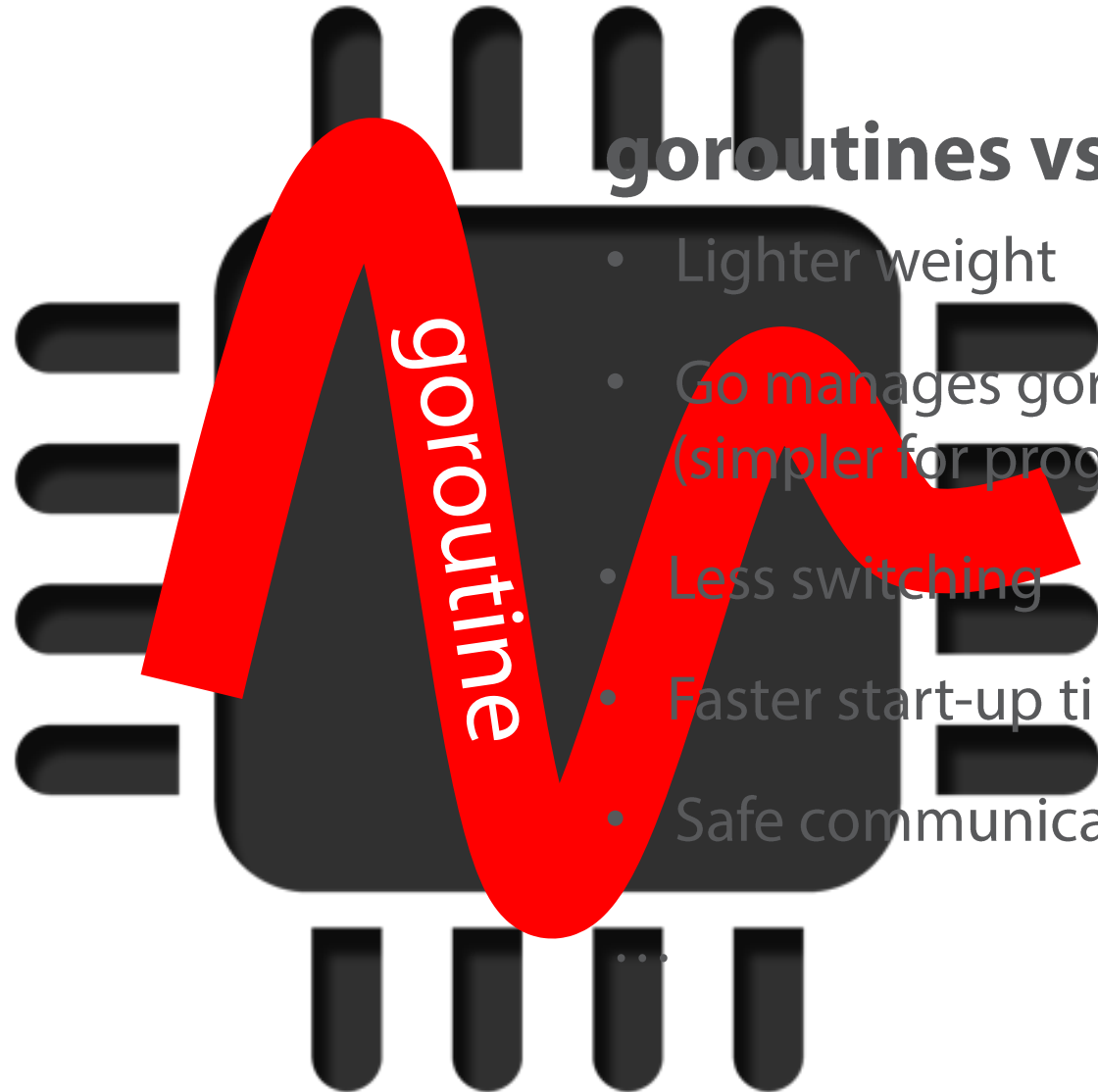
Process



---

# goroutines

---



## goroutines vs OS Threads

- Lighter weight
- Go manages goroutines (simpler for programmers)
- Less switching
- Faster start-up times
- Safe communication

...

# **Go's concurrency model:**

Actor model

Communicating Sequential Processes  
(CSP)



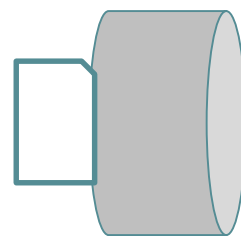
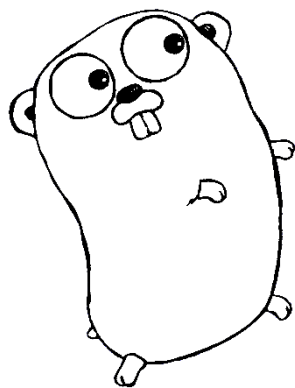
---

# Channels

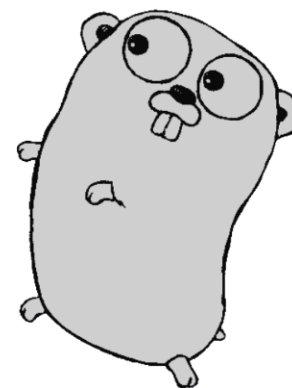
---



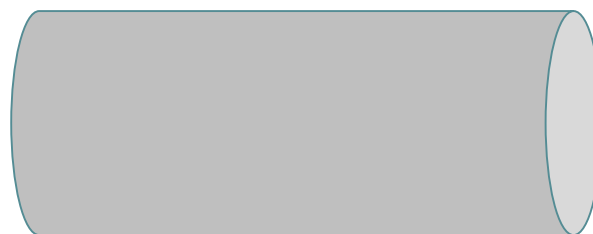
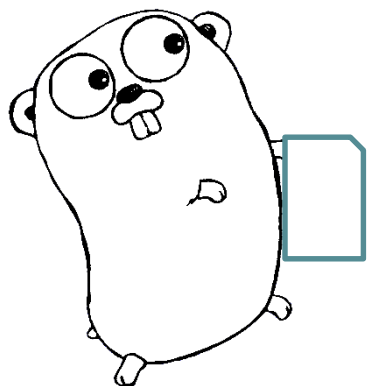




Channel  
(un-buffered)

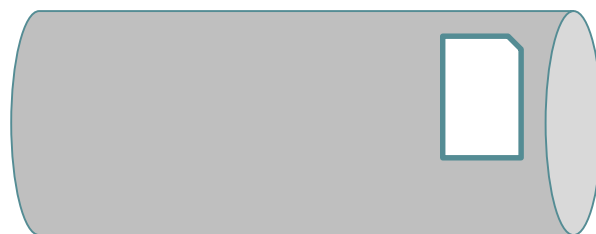
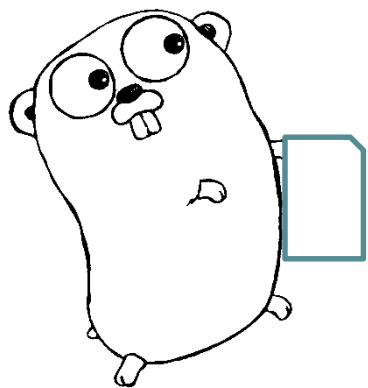


```
myChannel := make(chan int)
```



Channel  
(buffers \* 5)

```
myChannel := make(chan int, 5)
```



Channel  
(buffers \* 5)

```
myChannel := make(chan int, 5)
```

# Summary

## Concurrency



Concurrency = independently executing processes

## goroutines



## The **go** keyword

```
go sender(myChan, waitGrp)
```

## Channels

