



Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

有缓冲的channel

目录



PART 01
有缓冲的channel



PART 02
关闭channel



有缓冲的channel

1.有缓冲的channel



有缓冲的channel

有缓冲的通道（buffered channel）是一种在被接收前能存储一个或者多个数据值的通道。

这种类型的通道并不强制要求 goroutine 之间必须同时完成发送和接收。

通道会阻塞发送和接收动作的条件也不同。

只有通道中没有要接收的值时，接收动作才会阻塞。

只有通道没有可用缓冲区容纳被发送的值时，发送动作才会阻塞。

这导致有缓冲的通道和无缓冲的通道之间的一个很大的不同：无缓冲的通道保证进行发送和接收的 goroutine 会在同一时间进行数据交换；有缓冲的通道没有这种保证。



有缓冲的channel

有缓冲的channel创建格式：

make(chan Type, capacity)

如果给定了一个缓冲区容量，通道就是异步的。

只要缓冲区有未使用空间用于发送数据，或还包含可以接收的数据，那么其通信就会无阻塞地进行。

借助函数 `len(ch)` 求取缓冲区中剩余元素个数， `cap(ch)` 求取缓冲区元素容量大小。



关闭channel

1.关闭channel



关闭channel

如果发送者知道，没有更多的值需要发送到channel的话，那么让接收者也能及时知道没有多余的值可接收将是有用的，因为接收者可以停止不必要的接收等待。

这可以通过内置的close函数来关闭channel实现。

注意：

- channel不像文件一样需要经常去关闭，只有当你确实没有任何发送数据了，或者你想显示结束range循环之类的，才去关闭channel。（可以使用 range 来迭代不断操作channel）
- 关闭channel后，无法向channel 再发送数据(引发 panic 错误后导致接收立即返回零值)。
- 关闭channel后，可以继续从channel接收数据。
- 对于nil channel，无论收发都会被阻塞。



实例演示





Go语言入门到精通

江洲老师云课堂

—— 主讲：江洲老师 ——

感谢您的聆听和观看