链滴

# select + time.After 导致内存暴涨

作者：ghqemperor
原文链接：https://ld246.com/article/1572421370256
来源网站：链滴

# select + time.After 导致内存暴涨

```
func FFFF(notRun notRun) {
    for {
        after.Reset(afterTime)
        select {
        case <-notRun.notRun:
        // 大多数情况都是有notRun的输入
        case <-time.After(time.Minute):
            log.Printf("[ ConversationDeadline dead ]")
            notRun.Close()
            return
        case <-notRun.run:
            log.Printf("[ ConversationDeadline run ]")
            notRun.Close()
            return
        }
    }
}
```

- notRun 发送消息频率过快，而每次select都会调用到time.After，而time.After又会NewTimer，每次NewTimer都必须在1分钟后才能释放。
- 当notRun的频率很高时，会在内存中堆积非常多的无用的Timer。导致内存暴涨。

# 解决方法

```
func FFFF(notRun notRun) {
    afterTime := time.Minute
    after := time.NewTimer(afterTime)
    defer after.Stop()
    for {
        after.Reset(afterTime)
        select {
        case <-notRun.notRun:
        case <-after.C:
            log.Printf("[ ConversationDeadline dead ]")
            notRun.Close()
            return
        case <-notRun.run:
            log.Printf("[ ConversationDeadline run ]")
            notRun.Close()
            return
        }
    }
}
```

- 自己 NewTimer 在每次 select 之前 reset，使 timer 重新计时，从而避免每次都 new timer。