

☰ Interoperate Swift with C

◀ 服务端的Socket demo - II

返回视频 ▶

(<https://www.boxueio.com/series/interoperate-swift-with-c/ebook/254>)

(</series/interoperate-swift-with-c>)

客户端的Socket demo

⌂ Back to series (</series/interoperate-swift-with-c>)

我们继续补充上一节实现的Socket类，完成客户端的实现。大家可以在这里(<https://github.com/puretears/socket-demo-in-swift/tree/master/SocketClient>)找到客户端的完整实现。

首先，还是先来想象一下，我们希望如何使用呢？其实，对客户端来说，创建完 Socket 对象之后，我们就不需要任何额外的设置工作了，因此，直接连接服务器就好了：

```
do {
    let socket = try Socket(
        socketFilePath: "/tmp/swift_sock_demo",
        type: .active)

    try socket.connect {

    }
}
catch {
    print(error.localizedDescription)
    exit(EXIT_FAILURE)
}
```

这次，我们使用 .active 创建了 Socket 对象，然后，假设存在一个 connect 方法，可以让我们指定连接到服务器之后的行为。接下来，我们就回到Socket.swift来实现这个 connect 方法：

```
func connect(action: (CInt) -> Void) throws {
    let rawPointer = UnsafeMutableRawPointer(&sockAddrUn)
    let generalSockAddr = rawPointer.bindMemory(
        to: sockaddr.self,
        capacity: MemoryLayout<sockaddr>.size)

    #if os(Linux)
        let isConnected = Glibc.connect(socketFd,
            generalSockAddr,
            socklen_t(MemoryLayout<sockaddr_un>.size))
    #else
        let isConnected = Darwin.connect(socketFd,
            generalSockAddr,
            socklen_t(MemoryLayout<sockaddr_un>.size))
    #endif

    if isConnected == -1 {
        throw SocketException.cannotConnectToSocket
    }

    action(socketFd)
}
```

可以看到，和服务端 bind 的实现是类似的，唯一要注意的，就是 UnsafePointer<sockaddr> 和 UnsafeMutablePointer<sockaddr_un> 类型的转换。如果连接失败了，我们就返回特定的异常，否则，就把Socket文件句柄传递给它的closure参数供客户端代码使用。

最后，我们回到main.swift，在 connect 的closure里向服务器发送内容就好了：

⊕ 字号

● 字号

🖌 默认主题

🖌 金色主题

🖌 暗色主题

```
try socket.connect {
    var buffer: [CChar] = Array<CChar>(repeating: 0, count: 256)
    var numRead = read(STDIN_FILENO, &buffer, 256)

    while numRead > 0 {
        if write($0, &buffer, numRead) != numRead {
            fatalError("Partial write")
        }

        numRead = read(STDIN_FILENO, &buffer, 256)
    }

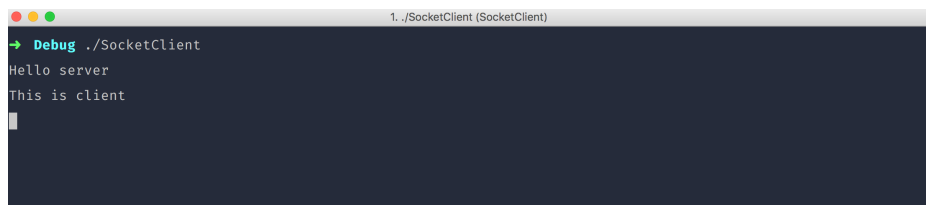
    if numRead == -1 {
        fatalError("Cannot read from stdin")
    }

    exit(EXIT_SUCCESS)
}
```

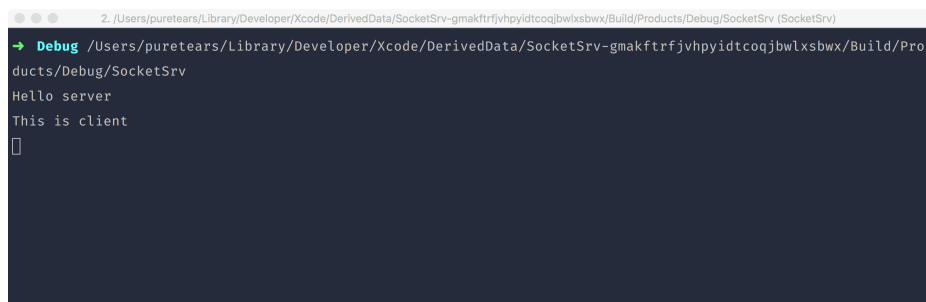
逻辑很简单，都是标准的read和write系统调用，我们就不再重复了。客户端和服务端都实现时候，我们来试一下。先把服务端在终端启动，这时它会被listen阻断，等待来自客户端的连接：



然后，我们启动客户端程序，它也会停在终端等待我们输入内容，我们可以陆续输入"Hello server"和"This is client"：



这样，就可以在服务端看到接收到的结果了：



Conclusion

至此，关于Swift和C交互的话题，我们就告一个段落了。希望这个系列的视频，可以帮你打消一些Swift和C交互很麻烦的顾虑。实际上，把Swift作为一种更现代化的系统编程语言，也是一个不错的选择，它会给我们带来更好的开发体验。



职场漂泊的你，每天多学一点。

从开发、测试到运维，让技术不再成为你成长的绊脚石。我们用打磨产品的精神去传播知识，把最新的移动开发技术，通过简单的图表，清晰的视频，简明的文字和切实可行的例子一一向你呈现。让学习不仅是一种需求，也是一种享受。

泊学动态

一个工作十年PM终创业的故事（二） (<https://www.boxueio.com/after-the-full-upgrade-to-swift3>)
Mar 4, 2017

人生中第一次创业的"10有" (<https://www.boxueio.com/founder-chat>)
Jan 9, 2016

猎云网采访报道泊学 (<http://www.lieyunwang.com/archives/144329>)
Dec 31, 2015

What most schools do not teach (<https://www.boxueio.com/what-most-schools-do-not-teach>)
Dec 21, 2015

一个工作十年PM终创业的故事（一） (<https://www.boxueio.com/founder-story>)
May 8, 2015

泊学相关

关于泊学 >

加入泊学 >

泊学用户隐私及服务条款 ([HTTPS://WWW.BOXUEIO.COM/TERMS-OF-SERVICE](https://www.boxueio.com/terms-of-service))

版权声明 ([HTTPS://WWW.BOXUEIO.COM/COPYRIGHT-STATEMENT](https://www.boxueio.com/copyright-statement))

联系泊学

Email: 10@boxue.io (<mailto:10@boxue.io>)

QQ: 2085489246

2017 © Boxue, All Rights Reserved. 京ICP备15057653号-1 (<http://www.miibeian.gov.cn/>) 京公网安备 11010802020752号 (<http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11010802020752>)

友情链接 [SwiftV](http://www.swiftv.cn/) (<http://www.swiftv.cn/>) | [Seay信息安全博客](http://www.cnseay.com/) (<http://www.cnseay.com/>) | [Swift.gg](http://swift.gg/) (<http://swift.gg/>) | [Laravist](http://laravist.com/) (<http://laravist.com/>) | [SegmentFault](https://segmentfault.com/) (<https://segmentfault.com/>) | [戴青K的博客](http://blog.dianqk.org/) (<http://blog.dianqk.org/>)