

iOS-Socket网络通信-框架与API



DiamondsAndRust (/u/72195d6d1106) [+ 关注](#)
2016.04.22 19:35* 字数 3680 阅读 2286 评论 2 喜欢 43 阅读 2286 评论 2 喜欢 43
(/u/72195d6d1106)

Socket基础概念

网络中进程之间如何通信？

网络中进程之间如何通信？首要解决的问题是**如何唯一标识一个进程**，否则通信无从谈起，在本地可以通过进程PID来唯一标识一个进程，但是在网络中这是行不通的。其实TCP/IP协议族已经帮我们解决了这个问题，网络层的“IP地址”可以唯一标识网络中的主机，而传输层的“协议+端口”可以唯一标识主机中的应用程序（进程）。这样利用三元组（ip地址，协议，端口）就可以标识网络的进程了，网络中的进程通信就可以利用这个标志与其它进程进行交互。

使用TCP/IP协议的应用程序通常采用应用编程接口：UNIX BSD的套接字（socket）和UNIX System V的TLI（已经被淘汰），来实现网络进程之间的通信。就目前而言，几乎所有的应用程序都是采用socket，而现在又是网络时代，网络中进程通信是无处不在，这就是我为什么说“一切皆socket”。

什么是Socket？

socket起源于Unix，而Unix/Linux基本哲学之一就是“一切皆文件”，都可以用 打开open -> 读写write/read -> 关闭close 模式来操作。我的理解就是Socket就是该模式的一个实现，socket即是一种特殊的文件，一些socket函数就是对其进行的操作（读/写IO、打开、关闭），这些函数在后面进行介绍。

Socket是应用层与TCP/IP协议族通信的中间软件抽象层，它是一组接口。



网络层次结构中的Socket.jpg

iOS网络编程层次结构



iOS网络编程层次结构分为三层，从上往下依次为：

Cocoa层：NSURL，Bonjour，Game Kit，WebKit

Core Foundation层：基于 C 的 CFNetwork 和 CFNetServices

OS层：基于 C 的 BSD Socket

- Cocoa层：是最上层的基于 Objective-C 的 API，比如 URL访问，NSStream, Bonjour，GameKit等，这是大多数情况下我们常用的 API。Cocoa 层是基于 Core Foundation 实现的。
- Core Foundation层：因为直接使用 socket 需要更多的编程工作，所以苹果对 OS 层的 socket 进行简单的封装以简化编程任务。该层提供了 CFNetwork 和 CFNetServices，其中 CFNetwork 又是基于 CFStream 和 CFSocket。
- OS层：最底层的 BSD Socket 提供了对网络编程最大程度的控制，但是编程工作也是最多的。因此，苹果建议我们使用 Core Foundation 及以上层的 API 进行编程。

Socket网络基本框架

TCP-C/S架构程序设计基本框架



TCP.jpg

UDP-C/S架构程序设计基本框架



UDP.jpg

Socket API

-----socket()-----

```
int socket(int domain, int type, int protocol);
```

- 用来创建一个socket描述字，可以理解为打开了一个socket。

socket函数对应于普通文件的打开操作。普通文件的打开操作返回一个文件描述字，而socket()用于创建一个socket描述符（socket descriptor），它唯一标识一个socket。这个socket描述字跟文件描述字一样，后续的操作都有用到它，把它作为参数，通过它来进行一些读写操作。

这个函数会返回一个int值，也就是socket描述字。

正如可以给fopen的传入不同参数值，以打开不同的文件。创建socket的时候，也可以指定不同的参数创建不同的socket描述符，socket函数的三个参数分别为：

- domain：即协议域，又称为协议族（family）。常用的协议族有，AF_INET、AF_INET6、AF_LOCAL（或称AF_UNIX，Unix域socket）、AF_ROUTE等等。协议族决定了socket的地址类型，在通信中必须采用对应的地址，如AF_INET决定了要用ipv4地址（32位的）与端口号（16位的）的组合、AF_UNIX决定了一个绝对路径名作为地址。这里我们只用AF_INET。
- type：指定socket类型。常用的socket类有：SOCK_STREAM（TCP用到）、SOCK_DGRAM（UDP用到）、SOCK_RAW、SOCK_PACKET、SOCK_SEQPACKET。
- protocol：顾名思义，就是指定协议。常用的协议有，IPPROTO_TCP、IPPROTO_UDP、IPPROTO_SCTP、IPPROTO_TIPC等，它们分别对应TCP传输协议、UDP传输协议、STCP传输协议、TIPC传输协议。

注意：并不是上面的type和protocol可以随意组合的，如SOCK_STREAM不可以跟IPPROTO_UDP组合。当protocol为0时，会自动选择type类型对应的默认协议。

当我们调用socket()创建一个socket时，返回的socket描述字它存在于协议族（address family, AF_XXX）空间中，但没有一个具体的地址。如果想要给它赋值一个地址，就必须调用bind()函数，否则就当调用connect()、listen()时系统会自动随机分配一个端口。

-----bind()-----

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

用来将socket()创建的socket描述字同{IP地址-协议号-端口号}绑定起来

当用 socket() 创建套接口后，它便存在于一个名字空间（地址族）中，但并未命名。bind() 函数通过给一个未命名套接口分配一个本地名字来为套接口建立本地捆绑（主机地址/端口号）。

在Internet地址族中，一个名字包括几个组成部分，对于 SOCK_DGRAM 和 SOCK_STREAM 类套接口，名字由三部分组成：主机地址，协议号（显式设置为UDP和TCP）和用以区分应用的端口号。如果一个应用并不关心分配给它的地址，则可将Internet地址设置为 INADDR_ANY，或将端口号置为0。如果Internet地址段为 INADDR_ANY，则可使用任意网络接口，且在有多种主机环境下可简化编程。（关于INADDR_ANY更多的事项在后面说明）

函数的三个参数分别为：

- sockfd：即 socket 描述字，它是通过 socket() 函数创建的，唯一标识一个 socket。bind() 函数就是将给这个描述字绑定一个名字。
- addr：一个 const struct sockaddr * 指针，指向要绑定给 sockfd 的协议地址。这个地址结构根据地址创建 socket 时的地址协议族的不同而不同，如 ipv4 对应的是：

```
struct sockaddr_in {
    sa_family_t sin_family; /* address family: AF_INET /
    in_port_t sin_port; / port in network byte order /
    struct in_addr sin_addr; / internet address */
};

/* Internet address. */
struct in_addr {
    uint32_t s_addr;
    /* address in network byte order */
};
```

- addrlen：对应的是地址的长度。

通常服务器在启动的时候都会绑定一个众所周知的地址（如ip地址+端口号），用于提供服务，客户就可以通过它来接连服务器；而客户端就不用指定，有系统自动分配一个端口号和自身的IP地址组合。这就是为什么通常服务器端在listen之前会调用bind()，而客户端就不会调用，而是在connect()时由系统随机生成一个。

**** 注意：**这里的sockaddr_in结构体中，有一个sin_port，存储的是端口号，这个端口号是在网络字节顺序下的，所以在给它赋值的时候要记得做一下转化。（关于字节顺序在后面会介绍。）

-----listen()-----

```
int listen(int sockfd, int backlog);
```

用来监听目标socket描述字

listen() 函数的第一个参数即为要监听的 socket描述字， 第二个参数为相应 socket 可以排队的最大连接个数（最多和多少个客户端通信？）。socket() 函数创建的 socket 默认是一个主动类型的，listen() 函数将 socket 变为被动类型的，等待客户的连接请求。

-----connect()-----

```
int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

通过两个socket描述字，在客户端和服务端之间建立连接

connect函数的第一个参数即为客户端的socket描述字，第二参数为服务器的socket地址，第三个参数为socket地址的长度。客户端通过调用connect函数来建立与TCP服务器的连接。

-----accept()-----

```
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

接收客户端请求，表明连接建立成功

accept函数的第一个参数为服务器的socket描述字，第二个参数为指向 struct sockaddr * 的指针，用于返回客户端的协议地址，第三个参数为协议地址的长度。如果accpet成功，那么其返回值是由内核自动生成的一个全新的描述字，代表与返回客户的TCP连接。

TCP服务器端依次调用 socket()、bind()、listen() 之后，就会监听指定的 socket 地址了。TCP客户端依次调用 socket()、connect() 之后就想TCP服务器发送了一个连接请求。TCP服务器监听到这个请求之后，就会调用 accept() 函数取接收请求，这样连接就建立好了。之后就可以开始网络I/O操作了，即类同于普通文件的读写I/O操作。

注意: accept 的第一个参数为服务器的 socket 描述字，是服务器开始调用socket()函数生成的，称为监听 socket 描述字；而 accept 函数返回的是已连接的 socket 描述字，是对面的 socket 描述字。一个服务器通常通常仅仅只创建一个监听**socket描述字**，它在该服务器的生命周期内一直存在。内核为每个由服务器进程接受的客户连接创建了一个已连接**socket描述字**，当服务器完成了对某个客户的服务，相应的已连接**socket描述字**就被关闭。

-----read()-----

-----write()-----

至此服务器与客户已经建立好连接了。可以调用网络I/O进行读写操作了，即实现了网咯中不同进程之间的通信！网络I/O操作有下面几组：

- read()/write()
- recv()/send()
- readv()/writev()
- recvmsg()/sendmsg()
- recvfrom()/sendto()

它们的声明如下：

```
#include <unistd.h>

ssize_t read(int fd, void *buf, size_t count);

ssize_t write(int fd, const void *buf, size_t count);
```

```
#include <sys/types.h>
#include <sys/socket.h>
ssize_t send(int sockfd, const void *buf, size_t len, int flags);
ssize_t recv(int sockfd, void *buf, size_t len, int flags);
ssize_t sendto(int sockfd, const void *buf, size_t len, int flags, const struct
sockaddr *dest_addr, socklen_t addrlen);
ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr
*src_addr, socklen_t *addrlen);
ssize_t sendmsg(int sockfd, const struct msghdr *msg, int flags);
ssize_t recvmsg(int sockfd, struct msghdr *msg, int flags);
```

read 函数是负责从 fd 中读取内容.当读成功时， read 返回实际所读的字节数， 如果返回的值是0表示已经读到文件的结束了， 小于0表示出现了错误。如果错误为EINTR说明读是由中断引起的， 如果是ECONNREST表示网络连接出了问题。

write 函数将 buf 中的 nbytes 字节内容写入文件描述符fd.成功时返回写的字节数。失败时返回-1， 并设置 errno 变量。 在网络程序中， 当我们向套接字文件描述符写时有俩种可能。1) write 的返回值大于0， 表示写了部分或者是全部的数据。2)返回的值小于0， 此时出现了错误。我们要根据错误类型来处理。如果错误为EINTR表示在写的时候出现了中断错误。如果为EPIPE表示网络连接出了问题(对方已经关闭了连接)。

-----close()-----

```
int close(int fd);
```

close一个TCP socket的缺省行为时把该socket标记为以关闭， 然后立即返回到调用进程。该描述字不能再由调用进程使用， 也就是说不能再作为read或write的第一个参数。

注意： close 操作只是使相应 socket 描述字的引用计数-1， 只有当引用计数为0的时候， 才会触发TCP客户端向服务器发送终止连接请求。

示例代码

socket

```
//调用socket(), 返回的socket描述符存放在fd中
int fd = socket(AF_INET, SOCK_STREAM, IPPROTO_IP);
```

bind

```
//新建一个地址结构体，用来bind()
struct sockaddr_in addr;
// 内存大小
addr.sin_len=sizeof(addr);
// 地址族，在socket编程中只能是AF_INET
addr.sin_family=AF_INET;
// 端口号
addr.sin_port=htons(1024);
// 按照网络字节顺序存储IP地址
addr.sin_addr.s_addr=INADDR_ANY;


//绑定socket套接字和端口
bind(fd, (const struct sockaddr *)&addr, sizeof(addr));
```

listen

```
//监听fd端口的消息
listen(fd, 10);
```

connect

```
//先将服务器端信息保存在一个sockaddr结构体中
struct sockaddr_in serveraddr;


serveraddr.sin_len=sizeof(serveraddr);
//协议族
serveraddr.sin_family=AF_INET;
// 服务器端口
serveraddr.sin_port=htons(1024);
// 服务器的地址
serveraddr.sin_addr.s_addr=inet_addr("192.168.2.5");
socklen_t addrLen;
addrLen =sizeof(serveraddr);


//连接服务器端
connect(fd, (struct sockaddr *)&serveraddr, addrLen);
```

accept

```
//接收到客户端请求，把得到的对端socket存储在peerfd中等待后续通信
peerfd = accept(fd, (struct sockaddr *)&peeraddr, &addrLen);
```

send

```
char buf[1024];
//向对端发送数据
send(peerfd, buf, 1024, 0);
```

recv

```
char buf[1024]
//接收数据
recv(fd, buf, len, 0);
```

TCP/IP的三次握手与四次挥手



补充内容

INADDR_ANY

在用bind()绑定一个socket的时候，地址中的

```
struct in_addr sin_addr;          /* internet address */
```

可能被绑定为INADDR_ANY，这表示指定为0.0.0.0的地址，这其实表示“任意地址”。

- 一般情况下，如果你要建立网络服务器应用程序，则你要通知服务器操作系统：请在某地址 xxx.xxx.xxx.xxx上的某端口 yyyy上进行侦听，并且把侦听到的数据包发送给我。这个过程，你是通过bind()系统调用完成的。也就是说，你的程序要绑定服务器的某地址，或者说：把服务器的某地址上的某端口占为己用。服务器操作系统可以给你这个指定的地址，也可以不给你。

- 如果你的服务器有多个网卡（每个网卡上有不同的IP地址），而你的服务（不管是在udp端口上侦听，还是在tcp端口上侦听），出于某种原因：可能是你的服务器操作系统可能随时增减IP地址，也有可能是为了省去确定服务器上有什么网络端口（网卡）的麻烦 —— 可以要在调用bind()的时候，告诉操作系统：“我需要在 yyyy 端口上侦听，所有发送到服务器的这个端口，不管是哪个网卡/哪个IP地址接收到的数据，都是我处理的。”这时候，服务器程序则在0.0.0.0这个地址上进行侦听。

- 在使用了INADDR_ANY以后，如果需要传输数据，则必须获取准确的地址，要用到

```
getsockname();
```

这个方法必须要在建立了连接之后调用，可以获得跟某一socket相关的特定IP地址，这个方法返回的IP地址直接存储在参数中。如果没有建立连接就调用，则会返回0.0.0.0。

获取地址

用getsockname获得本地ip和port

用getpeername获得对端ip和port

套接字socket必须是已连接套接字描述符。

字节顺序

网络字节顺序与本地字节顺序之间的转换函数：

htonl()--"Host to Network Long"

ntohl()--"Network to Host Long"

htons()--"Host to Network Short"

ntohs()--"Network to Host Short"

字节顺序


网络字节顺序与本地字节顺序之间的转换函数：

```
htonl()--"Host to Network Long"
ntohl()--"Network to Host Long"
htons()--"Host to Network Short"
ntohs()--"Network to Host Short"
```

本文参考

小礼物走一走，来简书关注我

赞赏支持



DiamondsAndRust (/u/72195d6d1106)

写了 5112 字，被 20 人关注，获得了 43 个喜欢
(/u/72195d6d1106) 写了 5112 字，被 20 人关注，获得了 43 个喜欢

+ 关注

喜欢 | 43



更多分享

(http://cwb.assets.jianshu.io/notes/images/3680764




写下你的评论...

2条评论

只看作者

按喜欢排序 按时间正序 按时间倒序




小小零号 (/u/d26158d9d727)


2楼 · 2016.09.05 20:53

(/u/d26158d9d727)

好评。 怎么不继续写了呢？

赞

 回复




geshen0523 (/u/5ff36ee1ef70)

3楼 · 2016.11.11 17:18

(/u/5ff36ee1ef70)

不错


赞


 回复

被以下专题收入，发现更多相似内容


+ 收入我的专题


 iOS 开发 (/c/2ffaa203eb6a?utm_source=desktop&utm_medium=notes-included-collection)


 iOS Dev... (/c/3233d1a249ca?utm_source=desktop&utm_medium=notes-included-collection)

 iOS进阶指南 (/c/49bfa36740cc?utm_source=desktop&utm_medium=notes-included-collection)

 iOS (/c/6cf13c5c8c9e?utm_source=desktop&utm_medium=notes-included-collection)

 iOS学习笔记 (/c/ac70aca89c4e?utm_source=desktop&utm_medium=notes-included-collection)

 网络编程 (/c/b1a93b07ce41?utm_source=desktop&utm_medium=notes-included-collection)

 iOS零碎知识 (/c/f20839660e8d?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ▾

推荐阅读 更多精彩内容 > (/)

React.js 实现动画效果 (/p/e3565b2f4856?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation) (/p/e3565b2f4856?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)
React.js中的动画效果 写在前面：目前网上搜到的有很多是讲ReactCSSTransitionGroup 的，现在它已经被 CSSTransitionGroup 取代了，DiamondsAndRust (/u/72195d6d1106?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

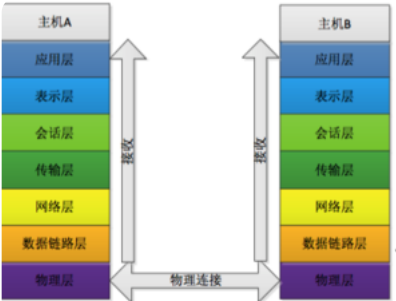
每个迷茫的年轻人，都该去看看《无问西东》 (/p/67b866ffadf6?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation) (/p/67b866ffadf6?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)
被《无问西东》感动了。不是因为它人物塑造得有多好，它讲故事的方式有多好，它的剪辑手法有多厉害。事实上有很多地方人物的动机不明确，讲故事的手吕白Alex (/u/4b6bdf4f9b22?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

你用过的社交软件，都将成为呈堂证供 (/p/d74e5e91a355?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation) (/p/d74e5e91a355?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)
01 有人说互联网时代最大的特点，是鱼龙混杂泥沙俱下，你永远不知道屏幕的背后是人还是狗。这句话的前半句是对的，不过后半句已经错了。现在不管你是 一只西城 (/u/7b834ba8c35a?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

毫无准备就长大了的人，只能挨生活留下的耳光 (/p/285e46a2a9f9?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation) (/p/285e46a2a9f9?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)
01. 前几天看了张艺兴在《偶像练习生》担当全民制作人的宣传片，他很严肃的说，“没准备好，你凭什么上舞台？”播到这句话时，身边一个号称“张艺兴迷诗音姑娘 (/u/2b472123e23e?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

十款冷门而逆天的软件，带你领略软件界的“黑马”们 (/p/156696ed5de2?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation) (/p/156696ed5de2?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)
01 Cold Turkey 老规矩，第一款推荐的软件必须超强是实力派！天降大任于斯人也，必先关其手机，收其平板，拔其网线，这也道出了不少朋友的痛处，当我科技毒瘤 (/u/4f6ed4921847?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/03bdec58a591?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

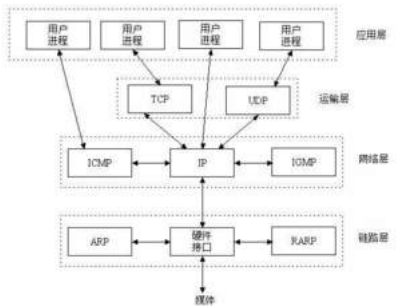
iOS socket应用 (/p/03bdec58a591?utm_campaign=maleskine&utm_c...

一、网络各个协议：TCP/IP、SOCKET、HTTP等 网络七层由下往上分别为物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。 其中物理层、数据链路层和网络层通常被称作媒体层，是网络工程师所

 只是前行 (/u/fcac5c36f59a?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/740596d7ecff?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

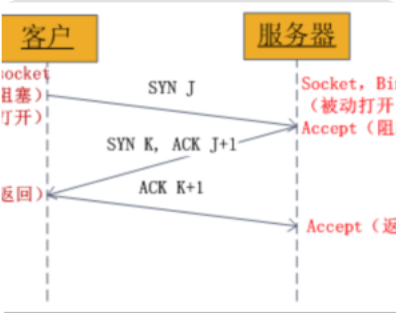
【转发】Linux的SOCKET编程详解 (/p/740596d7ecff?utm_campaign=...

网络中进程之间如何通信 进程通信的概念最初来源于单机系统。由于每个进程都在自己的地址范围内运行，为保证两个相互通信的进程之间既互不干扰又协调一致工作，操作系统为进程通信提供了相应设施，如UNIX

 大雁儿 (/u/af00eec6cd64?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/d6fdfac5e97d?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

Socket通信简介 (/p/d6fdfac5e97d?utm_campaign=maleskine&utm_co...


前言 我们深谙信息交流的价值，那网络中进程之间如何通信，如我们每天打开浏览器浏览网页时，浏览器的进程怎么与web服务器通信的？当你用QQ聊天时，QQ进程怎么与服务器或你好友所在的QQ进程通信？这些

 Chars (/u/bd2336d8f946?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

socket编程及API简介 (/p/f8183982244f?utm_campaign=maleskine&u...


什么是TCP/IP、UDP？ TCP/IP（Transmission Control Protocol/Internet Protocol）即传输控制协议/网间协议，是一个工业标准的协议集，它是为广域网（WANs）设计的。UDP（User Data Protocol，...

 liuboxx1 (/u/d1d0446f40cb?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

网络 socket 编程 端口 大全 (/p/2ea2e060a818?utm_campaign=maleski...

2端口：管理实用程序3端口：压缩进程5端口：远程作业登录7端口：回显9端口：丢弃11端口：在线用户13端口：时间17端口：每日引用18端口：消息发送协议19端口：字符发生器20端口：FTP文件传输协议(默认数

 twistfate33 (/u/026fa917c13a?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/cb81041e6b7f?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

清平教育每日分享——我们要向孩子学习 (/p/cb81041e6b7f?utm_campai...

12月2日，在广西玉林市，一辆三轮车在撞倒一名小孩后逃逸，孩子躺在马路中间，挣扎地想爬起来却发现起

12月2日，在广西玉林市，一辆二轮车撞倒一名小孩后逃逸，孩子躺在马路中间，挣扎地想爬起来却发现起不来了。而肇事的三轮车主选择驾车逃走。就在旁边的大人正观望犹豫时，一名身穿红色上衣的男孩已经在

 武际金 (/u/752dc2022167?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/db5057ec66f4?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
被时光灼伤的人 南歌子 (/p/db5057ec66f4?utm_campaign=maleskine&...

点点星辰 是否愿意听我讲一个关于此时的悲剧 曾经深夜仰望你的那个孩子 长大了 在感知 时间就是失去的那
一刻起 便远离了 童年那无虑的暂停 生活 成了日月交替后的动词 快乐 成为哀伤缝隙中的奢侈 此后 生活的忧

 斗漣 (/u/9bf05d5f2588?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

河南省出台方案助力大众创业 (/p/2b84026f340c?utm_campaign=males...

今年培训16万人次 发放创业贷款80亿元 5月10日，记者从省人社厅获悉，省政府办公厅出台了“2016年河南省助力大众创业专项工作方案”，将进一步落实就业创业扶持政策，完善就业创业服务体系，提升全省创业活

 创业说事 (/u/66aa7c96fbbf?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/3883eb28b98c?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
探班王华峰坏蛋星球乐队排练高校巡演11月19日启动 (/p/3883eb28b98c?...

音乐人王华峰创作专辑《跋涉千里来看你》发布已一个多月，《破冬》、《樱花东街》、《等候》、《跋涉千里来看你》等作品成为这张唱片里深受听众喜爱的作品。9月22的北京首发演出之后乐队一直在为高校巡

 彩妆掩饰苍白 (/u/fda364811c4d?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

错误日志 (/p/e84e6c4e17ff?utm_campaign=maleskine&utm_content=...

html 里的 shadow-root 是什么? https://www.zhihu.com/question/38765618

 小白古 (/u/6426cb2b0105?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)