

2017

实验分析

1. 解析函数功能

据说是对一个error的处理的代码，白给

2. 在TCP Echo程序中，由服务器进程终止引发的错误会发生什么？从应用层与传输层解释客户行为

书上的TCP echo程序

- 服务器TCP向客户发送一个FIN，客户TCP对它响应一个ACK。
- 客户进程此时处于Fgets调用的阻塞中，等待从终端接收一行文本。
- 此时如果从终端再键入一行文本，会导致str_cli调用writen，客户TCP从而把数据发送给服务器，因为先前打开与服务器通信的套接字的进程已经终止，服务器TCP响应一个RST。
- 但客户进程收不到这个RST。在调用writen以后，客户进程立刻调用Readline，由于之前接收到的FIN，readline立即返回0，于是以出错信息“server terminated prematurely”退出。

简答题

1. 解释UDP socket如何实现并发服务器设计

服务器启动后，等待下一个客户的到来。当一个客户到来时，记下其IP和port，并fork一个子进程，新建一个socket，bind一个随机端口，然后connect建立与这个客户的连接，在子进程中处理客户的请求，父进程继续循环等待下一个客户的到来。

2. TCP三次握手

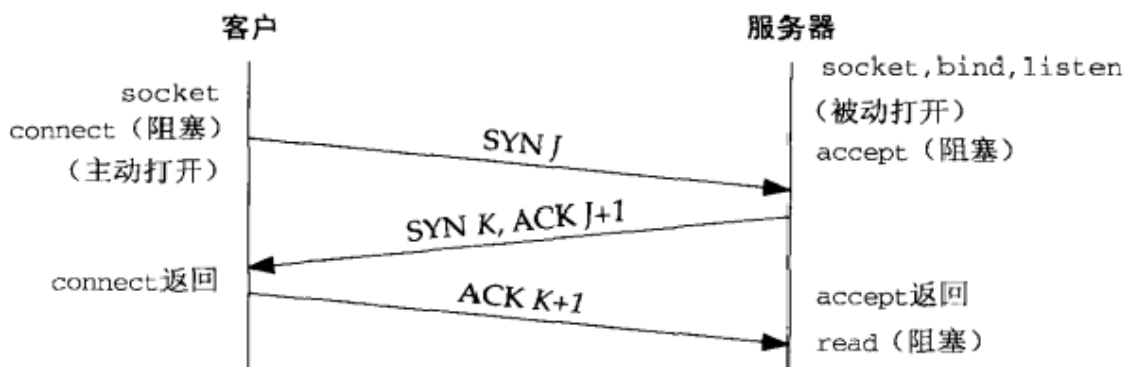


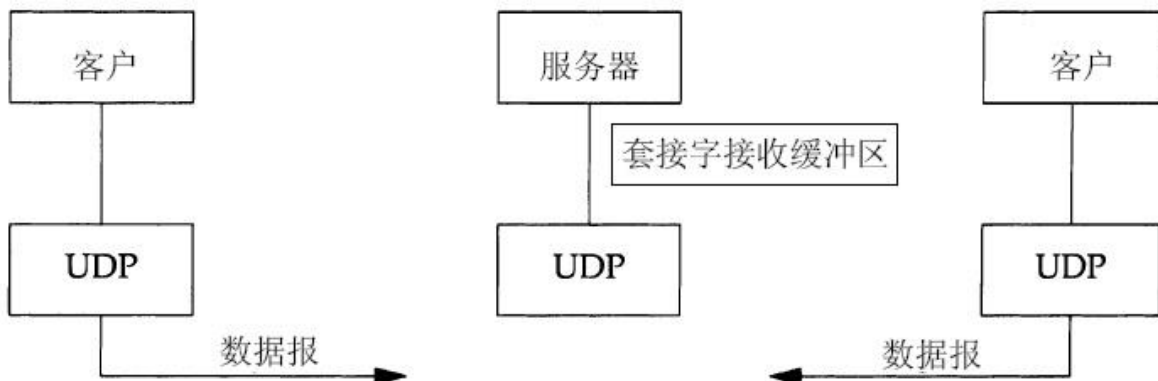
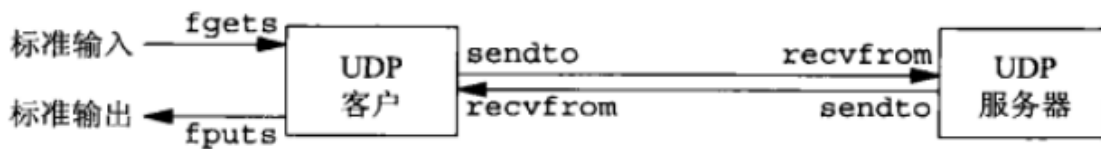
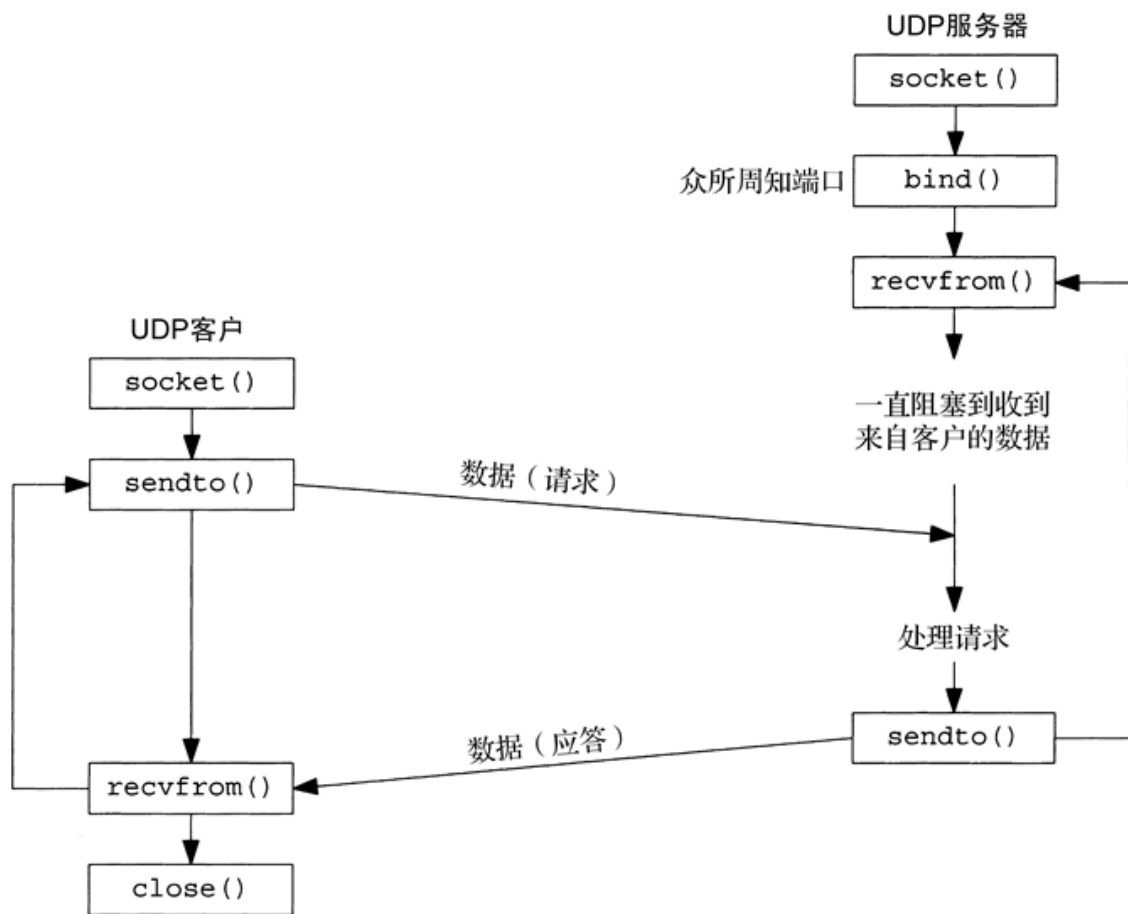
图2-2 TCP的三路握手

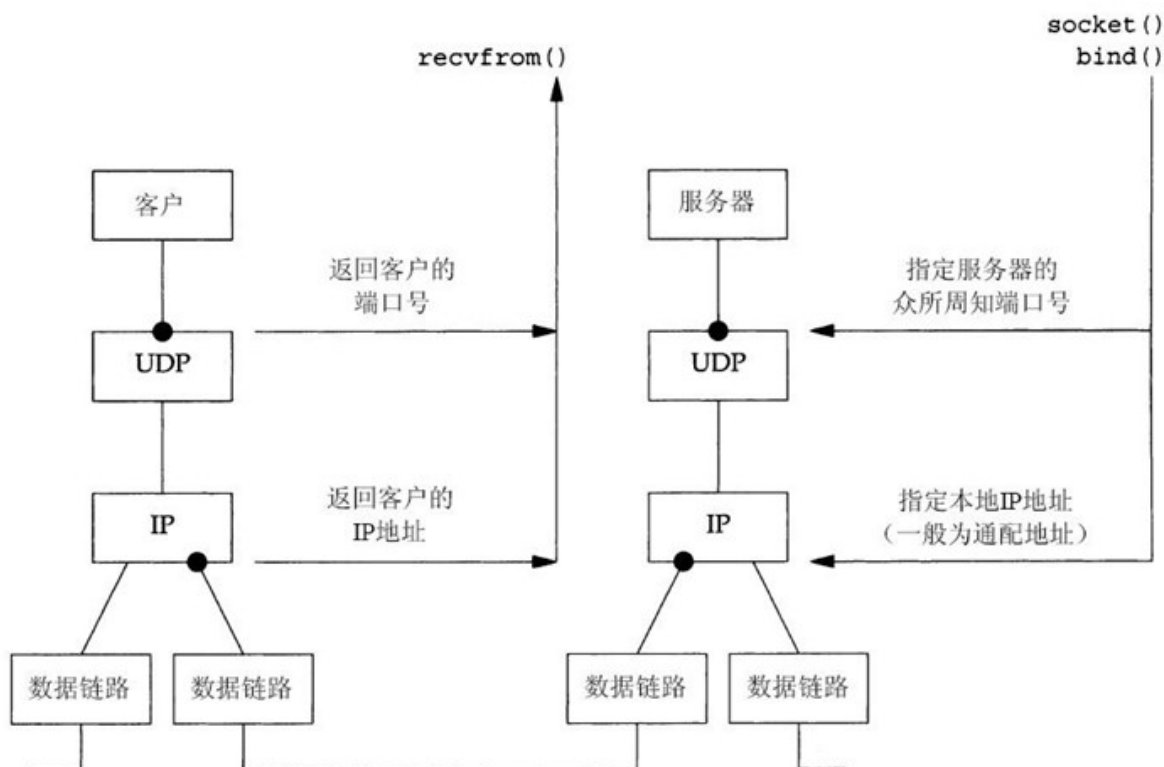
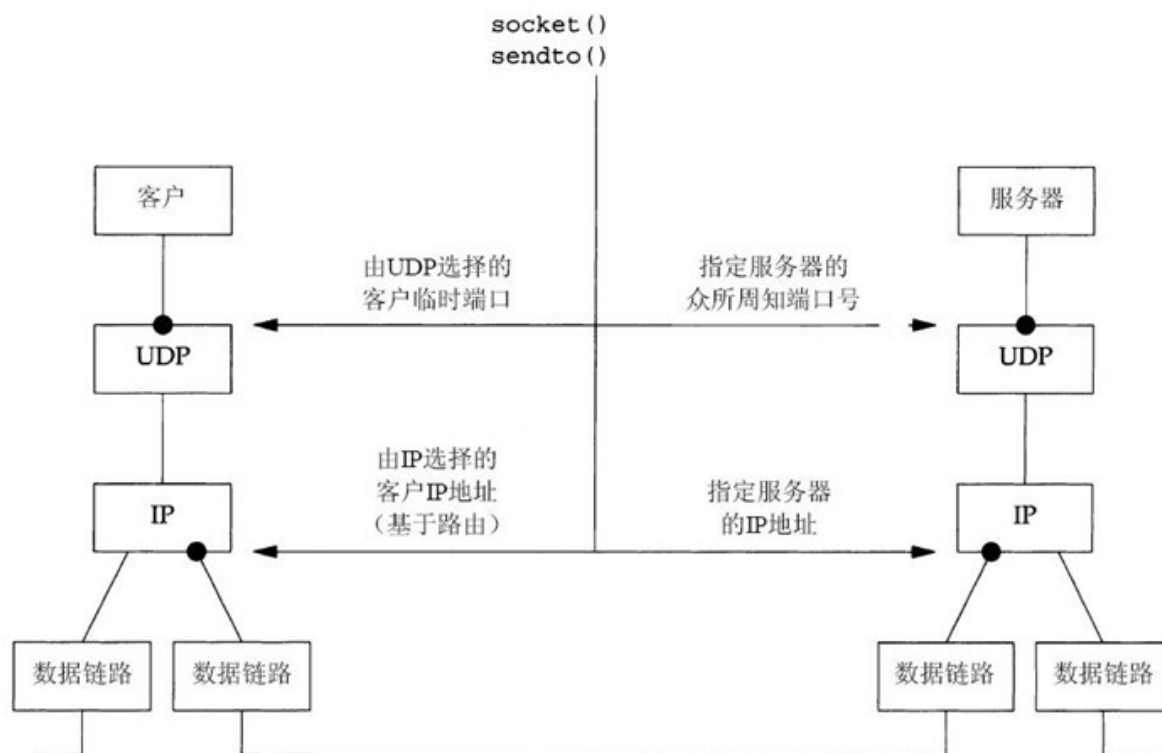
- 服务器通过socket,bind,listen创建一个监听socket，调用并阻塞在accept准备接受外来的连接
- 客户端通过socket函数创建一个未连接的socket，调用并阻塞在connect，尝试连接到服务器。客户TCP向服务器TCP发送一个SYN包，表示客户端向服务器申请服务，等待服务器的确认
- 服务器收到SYN，向客户发送一个ACK，同时发送一个SYN通知客户端服务器需要知道客户端是否收到回复
- 客户端收到ACK和SYN，connect函数返回，连接成功，并向服务器回复一个ACK。

- 服务器收到ACK，accept函数返回，连接正式建立。

3. 基于UDP的echo图

不知道echo图是啥，所有图都放了





4. 比较exec()与fork()的不同

概括：fork的原进程还在，而exec替代了原进程

fork():

- 子进程复制父进程的所有进程内存到其内存地址空间。父，子进程的数据段，堆栈段和代码段完全相同
- 子进程拥有自己的进程ID

exec():

- 进程调用exec()后，将在同一块进程内存里用一个新程序来代替调用 exec()的那个进程，当前进程的“数据段”，“堆栈段”和“代码段”被新程序改写。

- 新程序会保持调用exec()进程的ID不变

5. I/O几种模型 select属于哪种模式?

6. 原始套接字提供的三种UDP/TCP不具备的特点

- 读写ICMPv4, ICMPv6, IGMPv4分组
- 读写内核不处理其协议字段的IPv4数据报
- 使用IP_HDRINCL套接字选项自行构造IPv4首部

概括的说, 利用原始套接字可以处理内核不处理的协议、字段等, 而无需往内核中额外添加编码。

7. defunct的含义

进程所谓的“僵死”状态。一个子进程结束了, 但是它的父进程并没有wait/waitpid它, 则它的状态是defunct, 进程早已死亡, 但仍占有一个pid。

编程题

大小端判断

```
#include "stdio.h"
int main()
{
    union un
    {
        short a;
        char b[sizeof(short)];
    }u;
    u.a=0x0102;
    if(u.b[0]==1&&u.b[1]==2)
        printf("big endian");
    else if(u.b[0]==2&&u.b[1]==1)
        printf("small endian");
    else
        printf("unknown");
    return 0;
}
```

写一个handler function

列举wait()与waitpid()的区别, 并写一个处理所有子进程存在的handler function

知识点: POSIX信号处理

信号就是告知某个进程发生了某个事件的通知, 有时也称为软件中断, 通常是异步的。

每个信号关联一个deposition(或称action), 在信号发生时执行。

- waitpid可以通过指定pid, 指定终止某个子进程, wait不能
- 在没有已经终止的子进程时, waitpid可以不阻塞, 但wait会阻塞, 直到正在执行的子进程终止。

```
//调用: signal(SIGCHLD,sig_chld);
void sig_chld(int signo)
{
    pid_t pid;
    int stat;//value-result para
    while((pid=waitpid(-1,&stat,WNOHANG))>0)
        printf("child %d terminated\n",pid);
    return;
}
```

socket()的函数原型

```
#include<sys/socket.h>
int socket(int domain,int type,int protocol);
/*@param domain 协议族/地址族
 *@param type 套接字类型
 *@param protocol 指定协议
 *@return 返回一个socket描述符 sockfd
 *      sockfd<0创建失败
 *      sockfd>0创建成功，之后用这个sockfd进行I/O操作
 */
```

PF_KEY怎么用

```
int sockfd;
sockfd=socket(PF_KEY,SOCK_RAW,0);
```

使用哪个系统函数修改socket option

题目貌似给了7-2表格里，修改Maximum segment size的name，让你写函数调用，了解setsockopt怎么用就好

```
int sockfd;
setsockopt(sockfd,IPPROTO_TCP,TCP_MAXSEG,&mss,&len);
```