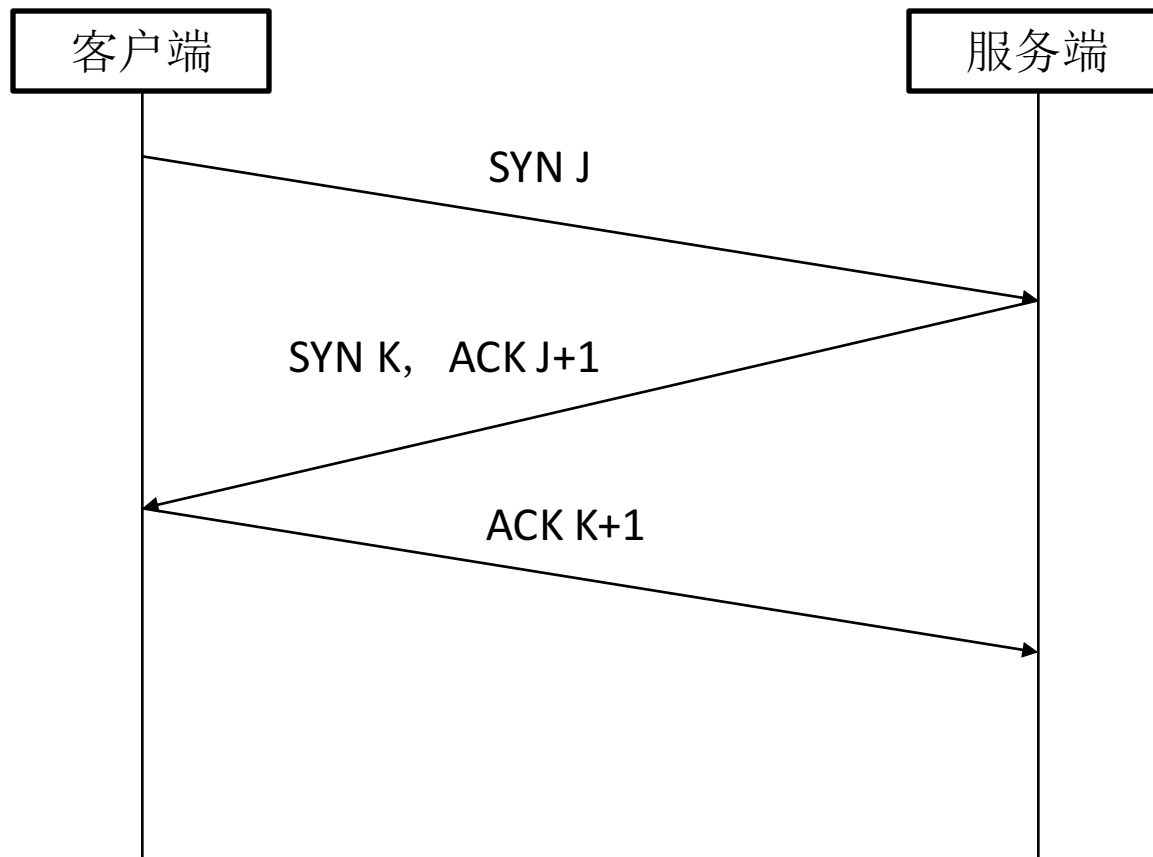
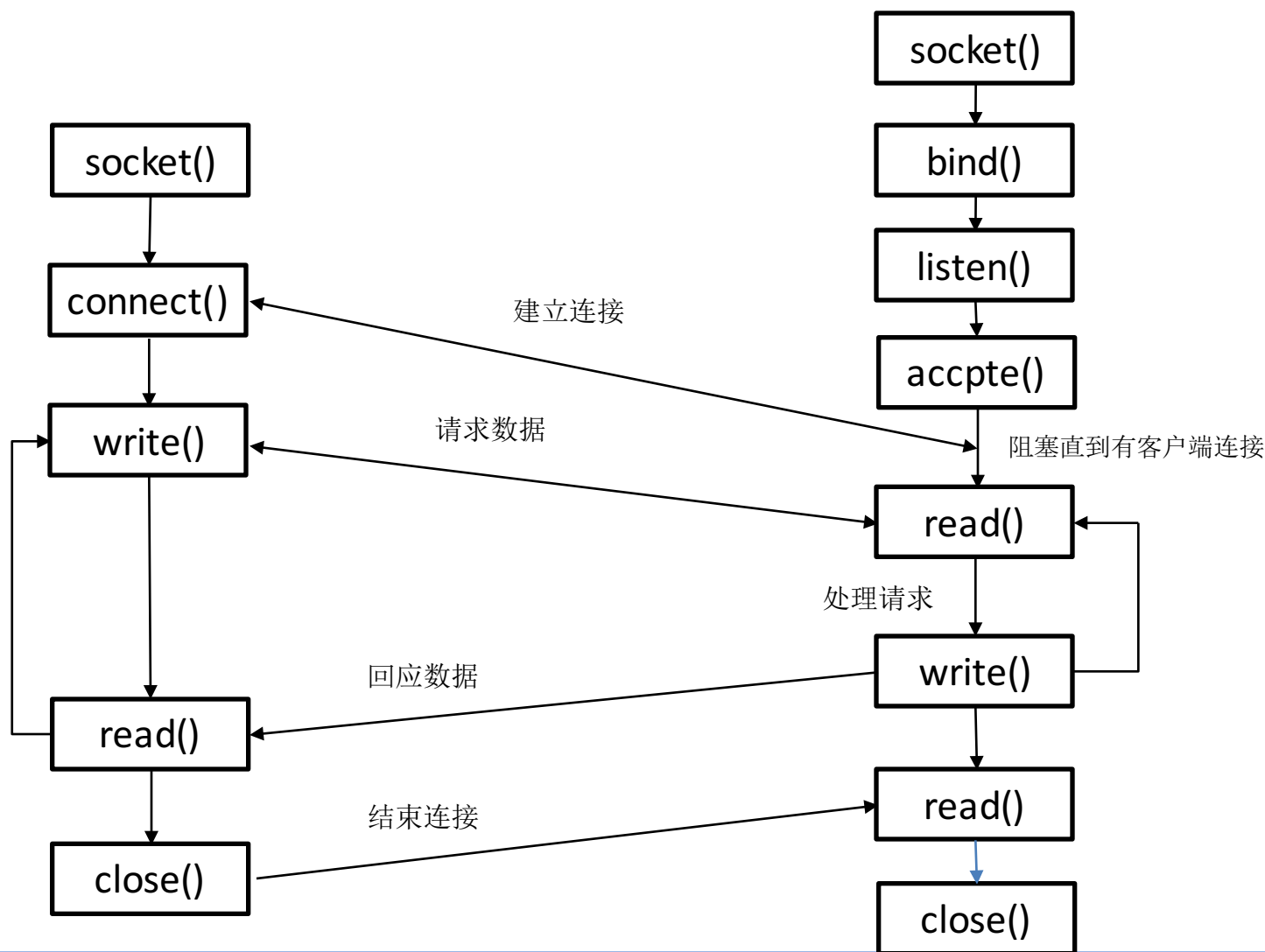


socket编程

1、TCP概述



2、基本TCP客户/服务器程序的套接口函数



3、linux C socket编程API

- `int socket(int domain,int type,int protocol);`
- `int bind(int socket sockfd,const struct sockaddr *addr,socklen_t addrlen);`
- `int listen(int sockfd,int backlog);`
- `int connect(int sockfd,const struct sockaddr *addr,socklen_t addrlen);`
- `int accept(int sockfd,struct sockaddr *addr,socklen_t *addrlen);`
- `int close(int fd);`

4、Java socket编程

- 服务端

```
ServerSocket ss = null;  
Socket s = null;  
try {  
    ss = new ServerSocket(5858);  
    for(;;) {  
        s = ss.accept();  
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));  
        PrintWriter out = new PrintWriter(s.getOutputStream(),true);  
        out.println("Received");  
        if(s != null) s.close();  
    }  
} catch (IOException e) {  
    e.printStackTrace();  
}  
finally {  
    try {  
        if(ss != null) ss.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

4、Java socket编程

- 客户端

```
Socket s = null;
```

```
try {
```

```
    s = new Socket("127.0.0.1",5858);
```

```
    PrintWriter out = new PrintWriter(s.getOutputStream(),true);
```

```
    out.println("hello socket");
```

```
    BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
```

```
    String str = br.readLine();
```

```
} catch (IOException e) {
```

```
    e.printStackTrace();
```

```
} finally {
```

```
    try {
```

```
        if(s != null) s.close();
```

```
    } catch (IOException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

5、作业

- 设计一个具体的协议（建议是应用层协议），采用标准 **Socket API** 编程来实现协议的功能
 - 协议的设计可以参考 **http** 等，语法规义相似
 - 头部域自己设计，需要有完备的功能，服务器端要能理解客户端的各种请求，并有一定的错误处理机制
 - 不要求使用额外封装的库

基本要求

- 编程语言不限，C，C++，Java，Python，Scala...
- 独立完成
- 建议有图形界面

参考题目1

- 设计一个类**http**协议
- 服务器端保存一份学生名单，包括学号、照片、姓名等。名单的存放方式随意。
- 客户端针对学生名单进行各类请求，如增加，删除，查看等，每种请求通过头部字段进行具体的要求。
- **Socket** 编程实现，推荐**TCP**

参考题目2

- 设计一个简单的文件传输协议
- 实现客户与服务器之间简单的文件传递，如**get/put**等
- 客户可以查询服务器存放文件的目录，自定义文件存放的目录等

参考题目3

- 简单的小说阅读器的设计
- 服务器端保存小说文本（**txt**格式的即可）
- 客户可以打开对应的文本，翻页，翻章，跳页，书签，下载，关闭等
- 建议最好有图形界面，因为是**txt**格式，所谓的“页”可以通过规定每次内容包含的字节来规定

参考题目4

- 简单的聊天系统
- 建议最好有图形界面

自己的创意

- 按照规定的要求设计出协议即可，最后提交的文档需要写清楚协议的功能
- 不要用其他的库！
- 使用**Socket** 编程

提交说明

- 使用**Socket** 编程开发文档（包括协议的设计说明，程序的开发）
- 源程序（包括**Client**程序，**Server**程序）
- 发送到助教邮箱中：
- 提交时间：期末考试之前

Q & A



致谢 !