电子科技大学信息与软件工程学院

实验指导书

（实验）课程名称：嵌入式网络编程

电子科技大学教务处制表

实验 1 Linux 平台下的基于 TCP 的文本聊天程序

实验所属系列： 《嵌入式网络编程》课内实验　　实验对象： 本科

相关课程及专业：嵌入式软件　　实验时数（学分）：4 学时

实验类别　课内上机

实验开发教师： 计算机网络课程组

【实验目的】

掌握 Linux 下的 Socket 编程相关的基本数据结构和 TCP 数据通信的函数使用，实现基于 TCP 数据传输的文本聊天程序。

【实验内容】

文本聊天程序的双方分别是服务器方和客户方。

（1）、服务器方：

服务器程序等待客户的连接，客户连接成功后可以与客户进行文字通信。服务器如果收到客户方的 exit 文本，则退出服务器程序。

（2）客户方：

建立一个用 TCP 连接的客户，以命令行的方式启动，命令行要求有服务器地址，服务器端口号。连接建立成功后可以输入文本与服务器进行通信。客户方通过输入 exit 文本结束语服务器的通信并退出程序。

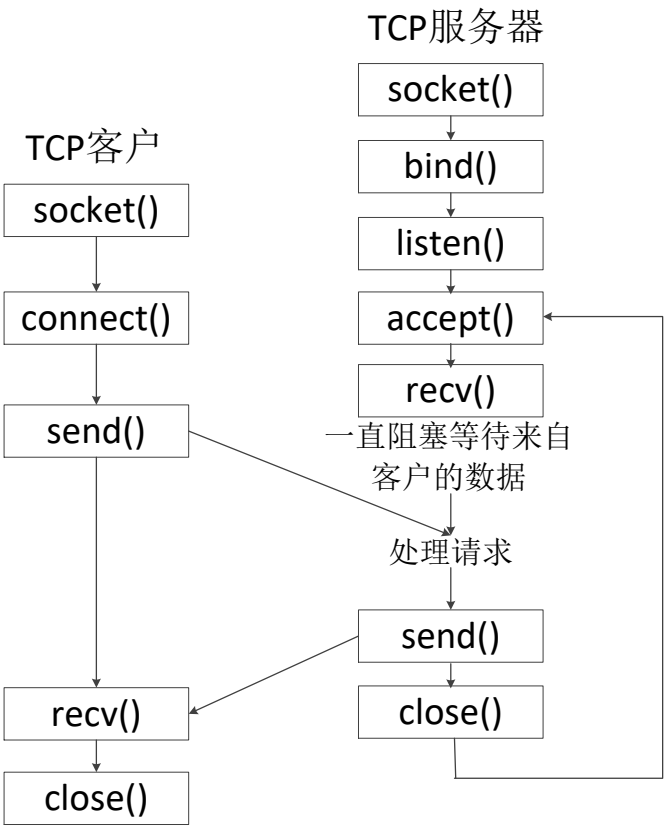
【实验环境】

Linux 操作系统、带有 C 语言编译插件的 Eclipse 开发工具。

【实验设备】

具有以太网通信接口的 PC 机一台。

【实验原理】



TCP 套接字编程基本流程描述

【实验核心函数说明】

套接字地址结构类型

```
struct sockaddr {
    unsigned short sa_family;/*地址族：AF_INET*/
    char sa_data[14];/*协议地址：IP 地址和端口号*/
};
```

套接字地址结构类型

```
struct sockaddr_in {
    short int sin_family; /*地址族：AF_INET*/
    unsigned short int sin_port;/*端口号*/
    struct in_addr sin_addr;/*IP 地址*/
    unsigned char sin_zero[8];/*零数据（bzero 或 memset 设置）*/
};
```

套接字地址结构类型

```
struct in_addr {
```

in_addr_t s_addr;/*IP 地址,in_addr_t 一般为 32 位的 unsigned long*/

};

字节序转换函数

主机转网络：

uint16_t htons(uint16_t hostshort)

uint32_t htonl(uint32_t hostlong)

网络转主机：

uint16_t ntohs(uint16_t netshort)

uint32_t ntohl(uint32_t netlong)

IP 地址网络字节和字符串转换函数

inet_addr()、inet_aton 或 inet_pton();/*将 IP 字符串转换成网络字节序的二进制*/

inet_ntoa()或 inet_ntop();/*将网络地址转换成 IP 字符串*/

创建套接字函数

格式：

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

功能：根据给定的参数创建一个套接字描述子，成功返回一个套接字描述符，不成功返回-1。

参数：

domain 指示网络协议族，AF_INET 或 AF_INET6 标识 TCP/IP 协议；

type 指示协议类型，如 SOCK_STREAM（TCP）、SOCK_DRAM（UDP）和 SOCK_RAW；

protocol 指示特定的协议类型，type 为 SOCK_STREAM 和 SOCK_DGRAM 时 protocol 设置为 0，type 为 SOCK_RAW 时可设置不同的值（IPPROTO_IP、IPPROTO_ICMP 等）

绑定套接字函数

格式：

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

功能：绑定 IP 地址和端口号到套接字 sockfd。成功返回 0，否则返回

-1。

参数：

sockfd 指示套接字描述符；

addr 指示本地的套接字地址；

addrlen 指示套接字地址长度；

建立连接套接字函数

格式：

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

功能：建立套接字与给定服务器地址 addr 的连接。如果 sockfd 是 TCP 描述符，则建立一条连接；如果是 UDP 描述符，则只建立 sockfd 和 addr 的绑定关系。成功返回 0，否则返回-1。

参数：

sockfd 指示套接字描述符；

addr 指示服务器的套接字地址；

addrlen 指示套接字地址长度；

监听套接字函数

格式：

int listen（int sockfd, int backlog）

功能：设置未被接收的 TCP 连接请求的最大数量，并监听连接请求。成功返回 0，否则返回-1。

参数：

sockfd 指示套接字描述符；

addr 指示套接字地址；

addrlen 指示套接字地址长度；

接收请求套接字函数

格式：

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

功能：设置未被接收的 TCP 连接请求的最大数量，并监听连接请求。成功返回 0，否则返回-1。

参数：

sockfd 指示套接字描述符；

addr 获得对方的套接字地址；

addrlen 指示套接字地址长度；

发送数据套接字函数

格式：

int send（int sockfd, const void *buf, size_t len, int flags）

功能：通过已经连接的套接字发送数据。成功返回发送的字符数，否则返回-1。

参数：

sockfd 指示套接字描述符；

buf 要发送的数据缓冲区；

len 指示要发送数据的长度；

flags 指示发送数据的属性，典型的 MSG_OOB 指示带外数据发送；

接收数据套接字函数

格式：

ssize_t recvfrom（int sockfd, void *buf, size_t len, int flags, struct sockaddr *src_addr, socklen_t *addrlen）

功能：接收指定套接字的数据。成功返回接收的字节数，否则返回-1。

参数：

sockfd 指示套接字描述符；

buf 用于存放接收数据的缓冲区；

len 指示接收数据的长度；

flags 指示接收数据的属性，典型的 MSG_OOB 指示接收带外数据；

src_addr 和 addrlen 分别指示接收数据的来源地址及其地址长度

关闭套接字函数

格式：

int close（int sockfd）

功能：关闭网络套接字。成功返回 0，否则返回-1。

参数：

sockfd 指示套接字描述符；

【实验报告】

1．撰写设计过程并展示源代码，代码需要进行注释。

2．展示运行结果并论述。

实验 2 Linux 平台下的基于 UDP 的文本聊天程序

实验所属系列： 《嵌入式网络编程》课内实验　　实验对象： 本科

相关课程及专业：嵌入式软件　　实验时数（学分）：4 学时

实验类别　课内上机

实验开发教师： 计算机网络课程组

【实验目的】

掌握 Linux 下的 Socket 编程相关的数据结构和 UDP 数据通信的函数使用，实现基于 UDP 数据传输的文本聊天程序。

【实验内容】

文本聊天程序的双方分别是服务器方和客户方。

（1）、服务器方：

服务器程序建立等待接收数据的套接字，收到客户的数据后通过键盘输入回复信息并通过网络发送给客户端。服务器如果收到客户方的 exit 文本，则退出服务器程序。

（2）客户方：

以命令行的方式启动，命令行要求有服务器地址，服务器端口号。通过键盘输入聊天文本信息并发送到服务器方。客户方通过输入 exit 文本结束语服务器的通信并退出程序。

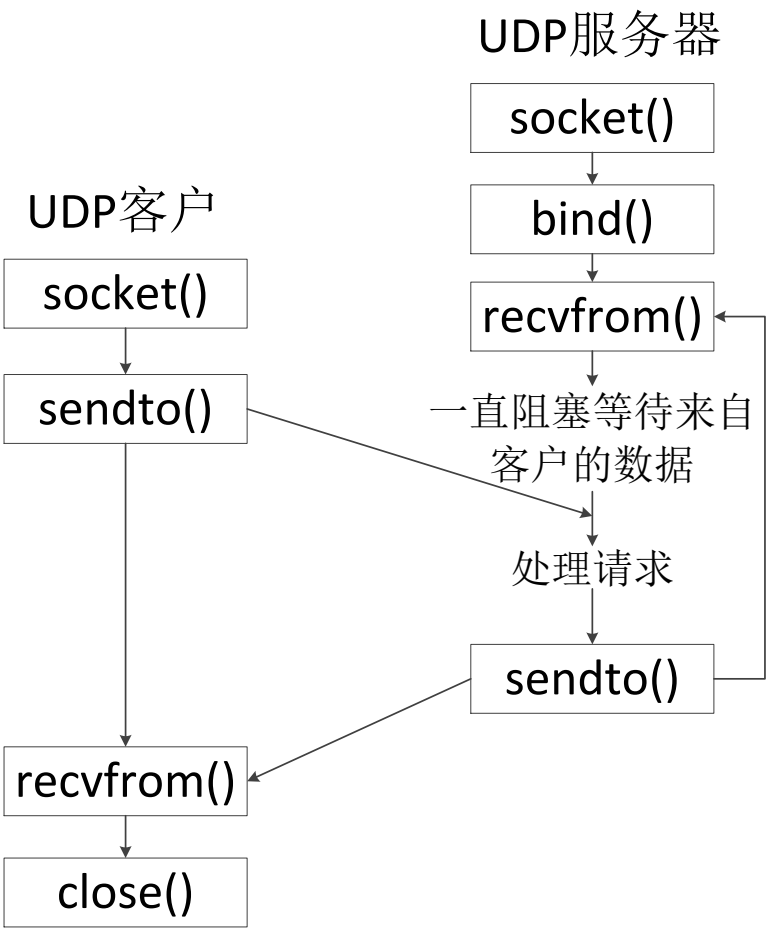
【实验环境】

Linux 操作系统、带有 C 语言编译插件的 Eclipse 开发工具。

【实验设备】

具有以太网通信接口的 PC 机一台。

【实验原理】



UDP 套接字编程基本流程描述

【实验核心函数说明】

套接字地址结构类型

```
struct sockaddr {
    unsigned short sa_family;/*地址族：AF_INET*/
    char sa_data[14];/*协议地址：IP 地址和端口号*/
};
```

套接字地址结构类型

```
struct sockaddr_in {
    short int sin_family; /*地址族：AF_INET*/
    unsigned short int sin_port;/*端口号*/
    struct in_addr sin_addr;/*IP 地址*/
    unsigned char sin_zero[8];/*零数据（bzero 或 memset 设置）*/
};
```

套接字地址结构类型

struct in_addr {

in_addr_t s_addr;/*IP 地址,in_addr_t 一般为 32 位的 unsigned long*/

};

字节序转换函数

主机转网络：

uint16_t htons(uint16_t hostshort)

uint32_t htonl(uint32_t hostlong)

网络转主机：

uint16_t ntohs(uint16_t netshort)

uint32_t ntohl(uint32_t netlong)

IP 地址网络字节和字符串转换函数

inet_addr()、inet_aton 或 inet_pton();/*将 IP 字符串转换成网络字节序的二进制*/

inet_ntoa()或 inet_ntop();/*将网络地址转换成 IP 字符串*/

创建套接字函数

格式：

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

功能：根据给定的参数创建一个套接字描述子，成功返回一个套接字描述符，不成功返回-1。

参数：

domain 指示网络协议族，AF_INET 或 AF_INET6 标识 TCP/IP 协议；

type 指示协议类型，如 SOCK_STREAM（TCP）、SOCK_DRAM（UDP）和 SOCK_RAW；

protocol 指示特定的协议类型，type 为 SOCK_STREAM 和 SOCK_DGRAM 时 protocol 设置为 0，type 为 SOCK_RAW 时可设置不同的值（IPPROTO_IP、IPPROTO_ICMP 等）

绑定套接字函数

格式：

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

功能：绑定 IP 地址和端口号到套接字 sockfd。成功返回 0，否则返回 -1。

参数：

sockfd 指示套接字描述符；

addr 指示本地的套接字地址；

addrlen 指示套接字地址长度；

发送数据套接字函数

格式：

int sendto（int sockfd, const void *buf, size_t len, int flags, const struct sockaddr *dest_addr, socklen_t addrlen）

功能：向指定的目的地址发送数据。成功返回发送的字符数，否则返回-1。

参数：

sockfd 指示套接字描述符；

buf 要发送的数据缓冲区；

len 指示要发送数据的长度；

flags 指示发送数据的属性，典型的 MSG_OOB 指示带外数据发送；

dest_addr 和 addrlen 分别指示目的地址和目的地址的长度。

接收数据套接字函数

格式：

ssize_t recv（int sockfd, void *buf, size_t len, int flags）

- 功能：接收指定套接字的数据。成功返回接收的字节数，否则返回-1。

参数：

sockfd 指示套接字描述符；

buf 用于存放接收数据的缓冲区；

len 指示接收数据的长度；

flags 指示接收数据的属性，典型的 MSG_OOB 指示接收带外数据；

接收数据套接字函数

格式：

ssize_t recvfrom（int sockfd, void *buf, size_t len, int flags, struct sockaddr *src_addr, socklen_t *addrlen）

- 功能：接收指定套接字的数据。成功返回接收的字节数，否则返回-1。

参数：

sockfd 指示套接字描述符；

buf 用于存放接收数据的缓冲区；

len 指示接收数据的长度；

flags 指示接收数据的属性，典型的 MSG_OOB 指示接收带外数据；

src_addr 和 addrlen 分别指示接收数据的来源地址及其地址长度

关闭套接字函数

格式：

int close（int sockfd）

功能：关闭网络套接字。成功返回 0，否则返回-1。

参数：

sockfd 指示套接字描述符；

【实验报告】

1. 撰写设计过程并展示源代码，代码需要进行注释。
2. 展示运行结果并论述。

实验 3 Linux 平台下基于 select 的文本回显程序

实验所属系列： 《嵌入式网络编程》课内实验　　实验对象： 本科

相关课程及专业：嵌入式软件　　实验时数（学分）：4 学时

实验类别　课内上机

实验开发教师： 计算机网络课程组

【实验目的】

掌握 Linux 平台下的 select 编程相关的数据结构和 TCP 数据通信的相关函数，实现文本回显程序。

【实验内容】

文本回显程序包括服务器方和客户方，客户方输入文本发送给服务器，服务器回显该文本给客户，客户在显示器上显示。

（1）、服务器方：

服务器等待客户的连接，客户连接成功后服务器为客户创建一个新的套接字与客户进行回显通信，服务器可以同时与多个客户进行回显通信。服务器如果收到客户方的 exit 文本，则只关闭对应客户的套接字连接。服务器收到键盘输入的 exit 才会退出程序。

（2）客户方：

建立一个用 TCP 连接的客户，以命令行的方式启动，命令行要求有服务器地址，服务器端口号。连接建立成功后可以输入文本与服务器进行回显通信。客户方通过输入 exit 文本结束语服务器的通信并退出程序。

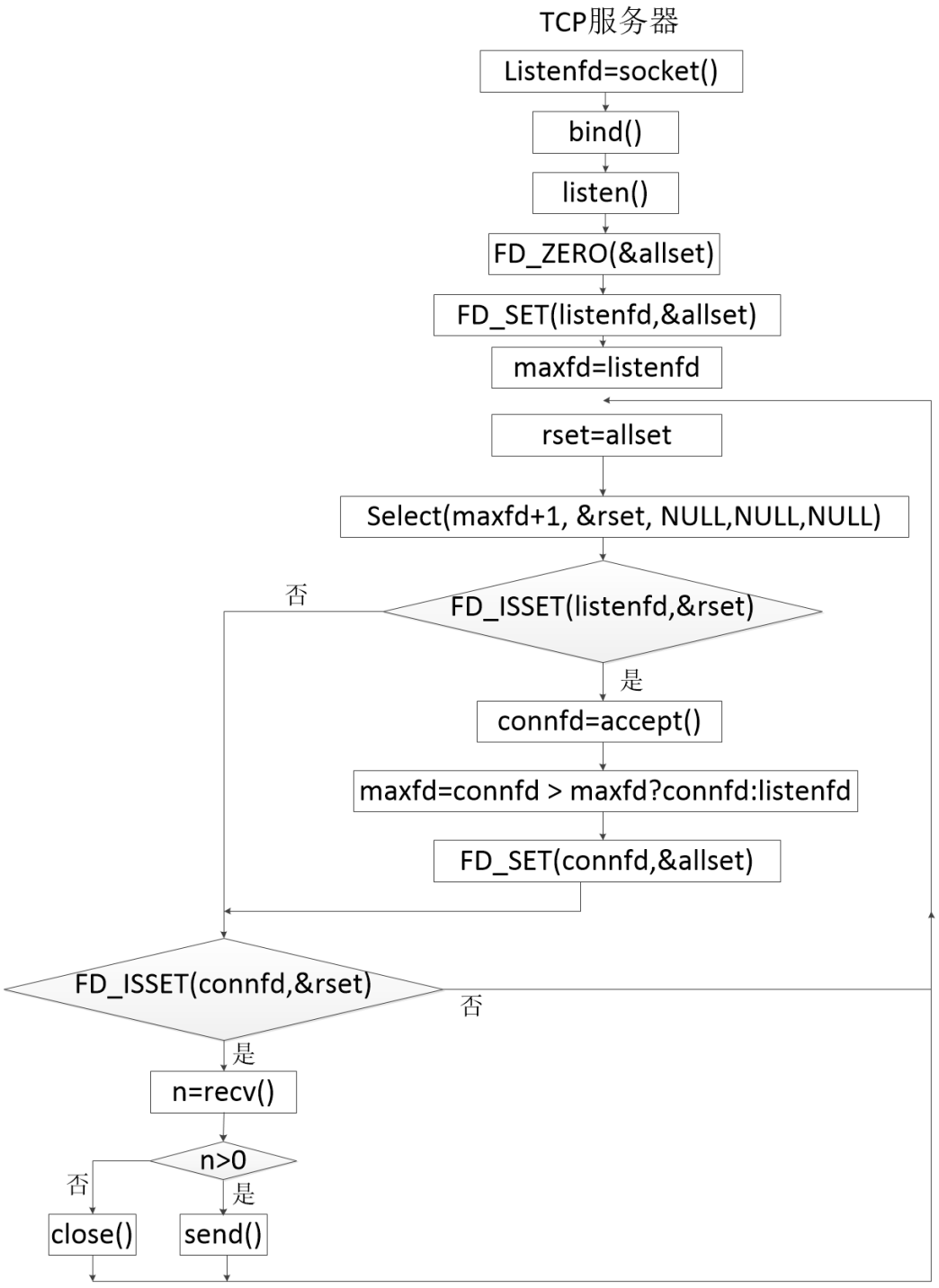
【实验环境】

Linux 操作系统、带有 C 语言编译插件的 Eclipse 开发工具。

【实验设备】

具有以太网通信接口的 PC 机一台。

【实验原理】



select 流程描述

【实验核心函数说明】

select 函数允许进程指示内核等待多个事件中的任何一个发生，并只在有一个或多个事件发生或经历一段指定的时间后才唤醒它。

格式：

Int select(int nfds, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct

timeval *timeout)

参数：

nfds 指定待测的描述符个数，它的值是待测的最大描述符数加 1。

readfds、writefds 和 exceptfds 分别是可读、可写和异常的描述符集。

timeout 表示 select 阻塞等待的最大时间。

timeval 结构格式：

```
struct timeval {
    long tv_sec;
    long tv_usec;
};
```

fd_set 集合处理宏：

void FD_ZERO(fd_set *set);

清除 set 集合中的内容；

void FD_CLR(int fd, fd_set *set);

删除 set 集中的 fd 描述符；

void FD_SET(int fd, fd_set *set);

增加 fd 到描述符集 set；

int FD_ISSET(int fd, fd_set *set);

判断 set 中满足条件的描述符是否是 fd。

【实验报告】

1. 撰写设计过程并展示源代码，代码需要进行注释。
2. 展示运行结果并论述。

实验 4 Linux 平台下文本回显并发服务器

实验所属系列： 《嵌入式网络编程》课内实验　　实验对象： 本科

相关课程及专业：嵌入式软件　　实验时数（学分）：4 学时

实验类别　课内上机

实验开发教师： 计算机网络课程组

【实验目的】

掌握 Linux 平台下的并发网络编程和 TCP 数据通信的相关函数，实现文本回显程序。

【实验内容】

文本回显程序包括服务器方和客户方，客户方输入文本发送给服务器，服务器回显该文本给客户，客户在显示器上显示。

（1）、服务器方：

服务器主进程等待客户的连接，客户连接成功后服务器为客户创建一个新的子进程与客户进行回显通信，服务器可以同时与多个客户进行回显通信。服务器如果收到客户方的 exit 文本，则只退出服务器所创建的对应客户的进程。服务器收到键盘输入的 exit 才会退出主进程。

（2）客户方：

建立一个用 TCP 连接的客户，以命令行的方式启动，命令行要求有服务器地址，服务器端口号。连接建立成功后可以输入文本与服务器进行回显通信。客户方通过输入 exit 文本结束语服务器的通信并退出程序。

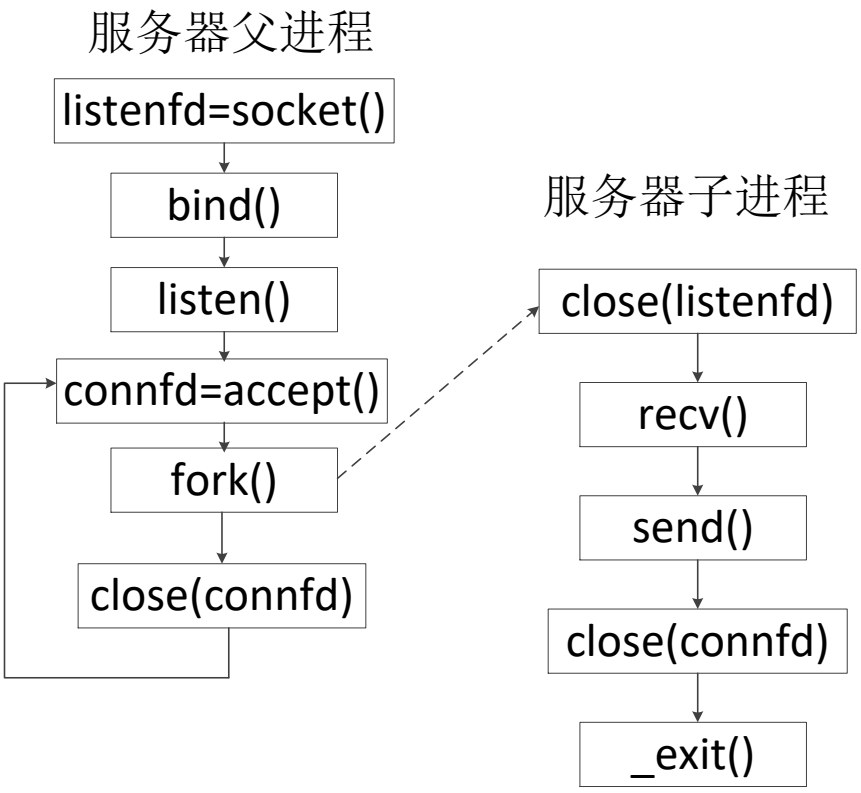
【实验环境】

Linux 操作系统、带有 C 语言编译插件的 Eclipse 开发工具。

【实验设备】

具有以太网通信接口的 PC 机一台。

【实验原理】



并发服务器流程描述

【实验核心函数说明】

创建进程函数：

格式：

pid_t fork(void);

功能说明：创建一个子进程。

返回值：在父进程中 pid_t 返回的是子进程的进程号；在子进程中 pid_t 返回 0。因此可以根据 pid_t 的值判断当前进程是父进程还是子进程。

终止进程函数：

格式：

void exit (int status);

功能说明：终止一个进程。

返回值：空。

参数：status 可以使用 EXIT_SUCCESS 和 EXIT_FAILURE。

【实验报告】

1．撰写设计过程并展示源代码，代码需要进行注释。
2．展示运行结果并论述。

实验 5 Linux 下基于串口的数据通信

实验所属系列：《嵌入式网络编程》课内实验　　实验对象：本科

相关课程及专业：嵌入式软件　　实验时数（学分）：4 学时

实验类别　课内上机

实验开发教师：计算机网络课程组

【实验目的】

掌握 Linux 下串口通信的基本流程、串口属性设置和数据传输的相关功能实现，实现两个计算机之间通过串口进行通信。

【实验内容】

两台计算机之间使用串口进行数据传输。

（1）、发送方：

初始化串口并打开串口，通过键盘输入一个字符串，调用串口发送功能发送字符串到接收方并等待接收方的响应。如果发送方键盘输入了 exit，则关闭串口并退出程序。

（2）接收方：

初始化串口并打开串口，调用串口接收功能等待发送方的数据，收到数据后返回相应给发送方。如果接收方键盘输入了 exit，则关闭串口并退出程序。

【实验环境】

Linux 操作系统、带有 C 语言编译插件的 Eclipse 开发工具。

【实验设备】

具有串口的 PC 机两台，一条串口电缆线。

【实验原理】

Linux 串口操作是以文件的形式实现的，串口编程包括如下步骤：

1）打开串口

2）串口初始化

3）读串口或写串口

4）关闭串口

【实验核心函数说明】

```
int fd; /*以读写方式打开串口*/
fd = open( "/dev/ttyS0", O_RDWR);
if (-1 == fd){
/*  不能打开串口一*/
perror("  提示错误！");
}
```

最基本的设置串口包括波特率设置、校验位和停止位设置。

串口设置的结构体

```
Struct termios{
tcflag_t c_iflag;    /* input mode flags */
tcflag_t c_oflag; /* output mode flags */
tcflag_t c_cflag; /* control mode flags*/
tcflag_t c_lflag;    /* local mode flags */
cc_t c_line;              /* line discipline */
cc_t c_cc[NCCS]; /* control characters */
speed_t c_ispeed;    /*input speed*/
speed_t c_ospeed; /*output speed*/
};
```

设置串口参数分为三个步骤：

1）获取当前串口的配置信息

```
tcgetattr(int fd, struct termios *termios_p)
```

2）设置结构体中的参数值

```
cfsetispeed(&options, speed);
options.c_cflag |= CSTOPB;
```

3）激活配置信息

tcsetattr(int fd,int optional_actions,const struct termios *termios_p)

读写串口与文件操作是一致的。

从串口读数据：

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

向串口写数据：

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

关闭串口就是关闭文件，其命令为：

int close(int fd)

【实验报告】

1．撰写设计过程并展示源代码，代码需要进行注释。

2．展示运行结果并论述。

实验 6 基于 GPRS 的文本通信

实验所属系列： 《嵌入式网络编程》课内实验　　实验对象： 本科

相关课程及专业：嵌入式软件　　实验时数（学分）：4 学时

实验类别　课内上机

实验开发教师： 计算机网络课程组

【实验目的】

掌握常用的 AT 命令，实现通过 GPRS 模块的字符串传输。

【实验内容】

使用串口调试助手发送 AT 命令道 GPRS 模块，实现短信的收发功能。

【实验环境】

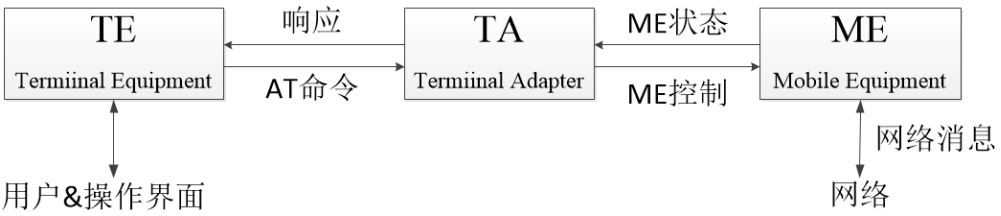
Windows 操作系统、STC-ISP 软件、友善串口调试助手、单片机程序 1.hex。

【实验设备】

具有串口的 PC 机一台，带有 GPRS 模块的实验箱一个。

【实验原理】

基于 GPRS 协议的数据通信控制主要使用 AT 命令实现，其连接和通信模型如下：



AT 命令控制移动设备结构图

实验前准备工作

1，　将手机SIM卡（大卡）插在开发板上的SIM卡插槽内

2，　将开发板上的天线安装好

3， 在PC端安装USB转串口驱动

4， 通过USB转串口线连接PC和开发板

5， 在PC端安装实验所用程序并为开发板烧写实验所用程序：

①打开STC-ISP软件

②选择单片机型号为STC12LE5A60S2

③点击“打开程序文件”，选择实验所用程序（1.hex）

④点击下载程序，为单片机重新上电

6，选择“友善串口助手”，选择正确的串口，然后点击“打开串口”

【实验核心命令说明】

测试 em310 模块时，首先以文本模式输入 em310<CR>，反馈 OK 后即可输入一系列 AT 命令进行测试。（比如直接输入 AT<CR>，会反馈 OK）。

若选择 AT 命令模式，用户需要了解 EM310 模块基本的 AT 命令，具体可详见 AT 命令手册。需要注意的是所有 AT 命令行必须以“AT”或“at”为前缀，以<CR>结尾。下面列出一些常用的 AT 命令并作出一些基本说明。注：<CR>即为回车符\r，十六进制代码为 0x0D。以下“发”代表用户的输入和发送，“收”代表 EM310 模块的反馈。

使用ATE命令设置回显信息。

查询制造商名称

AT+CGMI发

HUAWEI收

收

OK收

- 拨打电话

ATD<手机号>;发（将手机号改为自己想拨打的号码，<>不需要输入，;必须输入）

OK收（表示串口响应了该命令，不表示已经接通）

OK收（表示电话接通）

或者NO CARRIER收（表示未接通）

ATH发(挂断电话)

- 发送短信

首先设置文本格式参数：

AT+CSMP=17,169,0,0发

OK收

发送短信的流程较为复杂，首先我们应当选择短信发送的格式，text模式可以发送英文短信，pdu模式可以发送中文和英文短信：

AT+CMGF=<mode>发（mode为0表示PDU模式，为1表示text模式，<>不需要输入）

OK收

若选用text模式，接下来命令为：

AT+CMGS="15108306757"发（输入对方手机号）

>收（表示可以发送）

Hello world！发（输入短信内容，以CTRL-Z结束。注：CTRL-Z为标准中断符，十六进制代码0x1A）

+CMGS:35收

OK 收

接收与读取短信

①设置提醒方式：

AT+CNMI=2,1发（将短信提醒方式设置将短信存储到ME或SIM卡后，再给出新短信指示）

OK收

+CMTI: ”SM”,41收（当有新短信时模块就会给出这样的提示，SM表示储存位置为SIM卡，41表示当前储存的短信数目）

②查询短信：

AT+CMGF=1发（与发送短信相同，先设置短信格式，1为文本模式，0为PDU，这里以文本模式为例）

OK收

AT+CMGL="ALL"发（列举当前存储单元中所有短信）

+CMGL: 1,”REC

UNREAD”,”15108306757”,”14/05/27,18:47:30+32”,161,4收（在每一条短信的第一行列举短信的信息）

Test收（第二行显示出短信的内容）

OK收（所有短信显示完毕会显示OK）

③读出短信

AT+CPMS="SM","SM","SM"发（设置短信存储单元为SIM卡）

+CPMS:1,10,1,10,1,10收

收

OK收

AT+CMGR=1发（读取SIM卡中第一条短信）

+CMGR:"REC READ","15108306757",,"14/05/27,18:47:30+32",161,36,0,0,"

8613800280547",145,4收（显示第一条短信的信息）

Test收（显示第一条短信的内容）

收

OK 收

【实验报告】

1．撰写设计过程并展示源代码，代码需要进行注释。

2．展示运行结果并论述。