

OS 实验 1 实验报告

一、实验题目：

UNIX 环境下客户/服务器网络编程—模拟机选彩票的分布式 C/S 程序

二、实验目标：

1. 熟悉客户/服务器编程模式。
2. 熟悉 socket 网络编程，包括建立、绑定、连接、监听、关闭等。
3. 熟悉 TCP、UDP 协议的通信过程。
4. 熟悉文件处理，包括打开、读取、写入等。
5. 巩固 LINUX/UNIX 环境下的 system call。

三、实验内容：

采用 C/S 模式，模拟机选彩票。由客户端发起请求，服务器首先提示用户输入用户名，用户在客户端输入用户名，服务器收到用户名之后随机生成彩票的数字序列，发送给用户。然后服务器端将用户名及相应的彩票数字序列存储到日志文件中。客户端收到彩票数字序列后，将其输出到屏幕上，并存到本地日志文件中。

四、实验原理与设计：

1. 实验原理

(1) C/S 模式

Client 和 Server 常常分别处在不同的计算机上，Client 程序的任务是提供用户界面和接口，将用户的要求提交给 Server 程序，再将 Server 程序返回的结果以特定的形式显示给用户；Server 程序的任务是接收 Client 程序提出的服务请求，进行相应的处理，再将结果返回给 Client 程序。

(2) socket 网络编程

socket 通常称为“套接字”，用于描述 IP 地址和端口，是一个通信链的句柄。应用程序通常通过“套接字”向网络发出请求或者应答网络请求。每一个 Socket 用一个半相关描述：（协议，本地地址，本地端口）。一个完整的 Socket 有一个本地唯一的 Socket 号，由操作系统分配。最重要的是，Socket 是面向客户/服务器模型而设计的，针对客户端和服务端程序提供不同的 Socket 系统调用。客户随机申请一个 Socket（相当于一个想打电话的人可以在任何一台入网电话上拨号呼叫），系统为之分配一个 Socket 号；服务器拥有全局公认的 Socket，任何客户都可以向它发出连接请求和信息请求（相当于一个被呼叫的电话拥有一个呼叫方知道的电话号码）。

根据连接启动的方式以及本地套接字要连接的目标，套接字之间的连接过程可以分为三个步骤：服务器监听，客户端请求，连接确认。

1) 服务器监听：是服务器端套接字并不定位具体的客户端套接字，而是处于等待连接的状态，实时监控网络状态。

2) 客户端请求：是指由客户端的套接字提出连接请求，要连接的目标是服务器端的套接字。为此，客户端的套接字必须首先描述它要连接的服务器的套接字，指出服务器端套接字的地址和端口号，然后就向服务器端套接字提出连接请求。

3) 连接确认：是指当服务器端套接字监听到或者说接收到客户端套接字的连接请求，它就响应客户端套接字的请求，建立一个新的线程，把服务器端套接字的描述发给客户端，一旦客户端确认了此描述，连接就建立好了。而服务器端套接字继续处于监听状态，继续接收其他客户端套接字的连接请求。

(3) TCP 传输协议

TCP（Transmission Control Protocol 传输控制协议）是一种面向连接（连接导向）的、

可靠的、基于 IP 的传输层协议。

TCP 是因特网中的传输层协议，使用三次握手协议建立连接。当主动方发出 SYN 连接请求后，等待对方回答 SYN+ACK，并最终对对方的 SYN 执行 ACK 确认。这种建立连接的方法可以防止产生错误的连接，TCP 使用的流量控制协议是可变大小的滑动窗口协议。

TCP 三次握手的过程如下：1) 客户端发送 SYN (SEQ=x) 报文给服务器端，进入 SYN_SEND 状态。2) 服务器端收到 SYN 报文，回应一个 SYN (SEQ=y)ACK(ACK=x+1) 报文，进入 SYN_RECV 状态。3) 客户端收到服务器端的 SYN 报文，回应一个 ACK(ACK=y+1) 报文，进入 Established 状态。三次握手完成，TCP 客户端和服务端成功地建立连接，可以开始传输数据了。

(4) 文件操作

文件指针在 C 语言中用一个指针变量指向一个文件，这个指针称为文件指针。通过文件指针就可对它所指的文件进行各种操作。定义说明文件指针的一般形式为：FILE* 指针变量标识符，其中 FILE 应为大写，它实际上是由系统定义的一个结构，该结构中含有文件名、文件状态和文件当前位置等信息。文件在进行读写操作之前要先打开，使用完毕要关闭。所谓打开文件，实际上是建立文件的各种有关信息，并使文件指针指向该文件，以便进行其它操作。关闭文件则断开指针与文件之间的联系，也就禁止再对该文件进行操作。

(5) LINUX/UNIX 环境下的 system call

系统调用是由操作系统实现提供的程序接口或应用编程接口 (API)，是应用程序同系统之间的接口。操作系统的主要功能是为管理硬件资源和为应用程序开发人员提供良好的环境来使应用程序具有更好的兼容性，为了达到这个目的，内核提供一系列具备预定功能的多内核函数，通过一组称为系统调用的接口呈现给用户。系统调用把应用程序的请求传给内核，调用相应的内核函数完成所需的处理，将处理结果返回给应用程序。

2. 实验设计

在本实验中，首先，服务器调用 socket 函数得到一个描述符，调用 bind 函数将地址绑定到套接字，调用 listen 函数开始监听端口，调用 accept 获得连接请求并建立连接。客户端调用 socket 函数得到一个描述符，调用 connect 函数建立连接。然后客户端与服务器双方用 send 和 recv 函数进行发送和接收消息。

通信结束后，双方分别用 FILE 类的 fopen、fwrite 等函数对文件进行操作，将通信信息写回文件中。

最后，关闭 socket 和 file 流。

五、伪码算法：

<pre>服务器： { 创建套接字，绑定地址，开始监听； while(1) { 获得连接请求，建立连接； 请用户输入用户名； 接收用户在客户端输入的用户名； 生成随机彩票数字序列； 将彩票内容发送给用户； 将用户名及彩票信息写入服务器日志文件； } }</pre>	<pre>客户端： { 创建套接字； 连接服务器； 接收服务器输入用户名的提示； 输入用户名； 将用户名发送给服务器； 接收服务器发送的彩票信息； 将彩票信息写入客户端日志文件； }</pre>
--	---

六、程序源码：

服务器：

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>

#define PORT 6666
#define BACKLOG 10
#define MAX_SIZE 100
#define FILE_NAME "lottery.txt"
```

//将选出的随机数转成字符串

```
void IntToString(int n,char s[])
{
    int i,j;
    char s1[MAX_SIZE]="";
    for(j=0;n!=0;j++)
    {
        s1[j]=(n%10)+'0';
        n=n/10;
    }
    for(i=0,j--;j>=0;i++,j--)
    {
        s[i]=s1[j];
    }
    s[i]='\0';
}
```

//机选彩票：六个红色（范围 1-33），一个蓝色（范围 1-16）

```
void lottery(char lott[])
{
    int temp,i,j;
    char t[MAX_SIZE]="";
    strcat(lott,"red:");
    // srand(time(NULL));
    //printf("seed=%d\n",time(NULL));
    for(j=0;j<6;j++)
    {
        temp=rand()%33+1;
        IntToString(temp,t);
```

```

        strcat(lott, " ");
        strcat(lott, t);
        strcpy(t, "");
    }
    strcat(lott, " blue:");
    temp=rand()%16+1;
    IntToString(temp, t);
    strcat(lott, " ");
    strcat(lott, t);
}

int main()
{
    int sock_fd, new_fd;
    struct sockaddr_in server_addr, client_addr;
    int sin_size;
    int bytes;
    printf("Welcome to lottery world!\n");

    if((sock_fd=socket(AF_INET, SOCK_STREAM, 0))==-1)
    {
        perror("socket failed");
        exit(1);
    }
    //printf("socket ok\n");

    bzero(&server_addr, sizeof(struct sockaddr_in));
    server_addr.sin_family=AF_INET;
    server_addr.sin_addr.s_addr=htonl(INADDR_ANY);
    server_addr.sin_port=htons(PORT);

    //防止 bind 出现 Address already in use 解决方法
    int reuse=1;
    if((setsockopt(sock_fd, SOL_SOCKET, SO_REUSEADDR, &reuse, sizeof(reuse)))<0)
    {
        perror("setsockopt error");
        exit(1);
    }

    if(bind(sock_fd, (struct sockaddr *)&server_addr, sizeof(struct sockaddr)))
    {
        perror("bind failed");
        exit(1);
    }
}

```

```

//printf("bind ok\n");

if(listen(sock_fd,BACKLOG)==-1)
{
    perror("listen failed");
    exit(1);
}
//printf("listen ok\n");

while(1)
{
    char buffer[MAX_SIZE]="";//存放用户名和彩票信息
    sin_size=sizeof(struct sockaddr_in);
    new_fd=accept(sock_fd,(struct sockaddr *)&client_addr,&sin_size);
    if(new_fd==-1)
    {
        perror("accept failed");
        exit(1);
    }
    //printf("accept ok\n");

    //请用户输入用户名
    if(send(new_fd,"Please input your name:",MAX_SIZE,0)==-1)
    {
        perror("write failed");
        exit(1);
    }
    //printf("write ok\n");

    char name[MAX_SIZE]={0};//用户名
    if((bytes=recv(new_fd,name,MAX_SIZE,0))==-1)
    {
        perror("read failed");
        exit(1);
    }
    //printf("read ok\n");
    name[bytes]='\0';
    printf("name=%s\n",name);
    strcat(buffer,name);
    strcat(buffer,":");

    char lott[MAX_SIZE]="";//彩票
    srand(time(NULL));
    lottery(lott);//随机生成彩票

```

```

        if(send(new_fd,lott,MAX_SIZE,0)==-1)
        {
            perror("write failed");
            exit(1);
        }
        //printf("write ok\n");
        strcat(buffer,lott);
        strcat(buffer,"\n");

        //将用户名及彩票信息写入文件
        FILE *fp;
        if((fp=fopen(FILE_NAME,"a+"))==NULL)
        {
            perror("file open failed");
            exit(1);
        }
        // fseek(fp,0L,SEEK_END);
        fwrite(buffer,strlen(buffer),1,fp);
        printf("lottery:%s\n",buffer);
        fclose(fp);

        close(new_fd);
    }
    close(sock_fd);
    return 0;
}

```

客户端:

```

#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>

#define PORT 6666
#define MAX_SIZE 100

int main(int argc,char *argv[])
{
    if(argc!=2)
    {
        perror("usage:client serverIP");
        exit(1);
    }
}

```

```

int sock_fd;
int bytes;
int sin_size;
char buffer[MAX_SIZE]={0};
struct sockaddr_in server_addr,client_addr;
printf("I am lucky.\n");

if((sock_fd=socket(AF_INET,SOCK_STREAM,0))==-1)
{
    perror("socket failed");
    exit(1);
}
//printf("socket ok\n");

bzero(&server_addr,sizeof(struct sockaddr_in));
server_addr.sin_family=AF_INET;
server_addr.sin_addr.s_addr=inet_addr(argv[1]);
server_addr.sin_port=htons(PORT);

if(connect(sock_fd,(struct sockaddr *)&server_addr,sizeof(struct sockaddr)))
{
    perror("connect failed");
    exit(1);
}
//printf("connect ok\n");

if((bytes=recv(sock_fd,buffer,MAX_SIZE,0))==-1)
{
    perror("read failed");
    exit(1);
}
//printf("read ok\n");
buffer[bytes]='\0';
printf("%s\n",buffer);

char name[20];
scanf("%s",name);//输入用户名
if(send(sock_fd,name,100,0)==-1)
{
    perror("write failed");
    exit(1);
}
//printf("write ok\n");

```

```

if((bytes=recv(sock_fd,buffer,MAX_SIZE,0))==-1)
{
    perror("read failed");
    exit(1);
}
//printf("read ok\n");
buffer[bytes]='\0';
printf("lottery:%s\n",buffer);//打印彩票
strcat(buffer,"\n");

//将自己的彩票信息写入文件
char fileName[MAX_SIZE]="";
strcat(fileName,name);
strcat(fileName, "_lottery.txt");
FILE *fp;
if((fp=fopen(fileName,"a+"))==NULL)
{
    perror("file open failed");
    exit(1);
}
fwrite(buffer,strlen(buffer),1,fp);
//printf("buffer=%s\n",buffer);
fclose(fp);

close(sock_fd);
return 0;
}

```

七、执行结果截图：

1. 启动服务器（IP 为 192.168.190.129），开始监听客户端请求：

```

argenbarbie@ubuntu:~/Desktop/lottery$ ./server
Welcome to lottery world!

```

2. 启动客户端，向服务器发起连接请求，连接成功后，收到服务器请输入用户名的提示：

```

argenbarbie@ubuntu:~/Desktop/lottery$ ./client 192.168.190.129
I am lucky.
Please input your name:

```

3. 在客户端输入用户名：

```

Please input your name:
murong

```

4. 服务器收到客户端输入的用户名后，在服务器端打印出用户名，然后，随机生成彩票数字序列，并在服务器端打印出彩票信息：

```

name=murong
lottery:murong:red: 28 32 21 23 25 1 blue: 14

```


5. 客户端收到服务器端发送的彩票信息，将其在客户端输出：

```
lottery:red: 28 32 21 23 25 1 blue: 14
```

6. 服务器端日志文件记录用户名及其彩票信息，文件内容如下：

```
lottery.txt x  
murong:red: 28 32 21 23 25 1 blue: 14  
ime:red: 13 18 17 17 29 13 blue: 7
```

7. 客户端日志文件记录本地用户的彩票信息，文件内容如下：

```
murong_lottery.txt x  
red: 28 32 21 23 25 1 blue: 14
```

```
ime_lottery.txt x  
red: 13 18 17 17 29 13 blue: 7
```

八、总结与体会：

本实验主要是在理解 C/S 网络编程的基础上，基于 TCP 协议实现模拟彩票生成器的功能，实现了客户端与服务器之间的交互。并利用了文件处理等系统调用函数，将所学内容前后联系起来，并应用于实际案例中，收获很大。