



# 计算机网络实验报告

## 实验三：Chat实验

数据科学与计算机学院 17大数据与人工智能

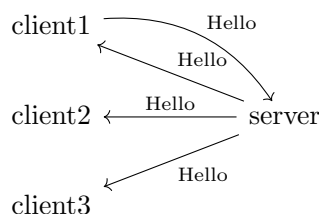
17341015 陈鸿峥

### 一、实验目的

掌握套接字的多线程编程方法，实现多人聊天。

### 二、实验介绍

利用客户/服务器(Client/Server或CS)模式实现一个多人聊天（群聊）程序，其功能是每个客户发送给服务器的消息都会传送给所有的客户端。



### 三、参考资料

- Introduction to Multi-Threaded Programming (Linux), <https://www.linuxjournal.com/article/3138>

### 四、实验环境

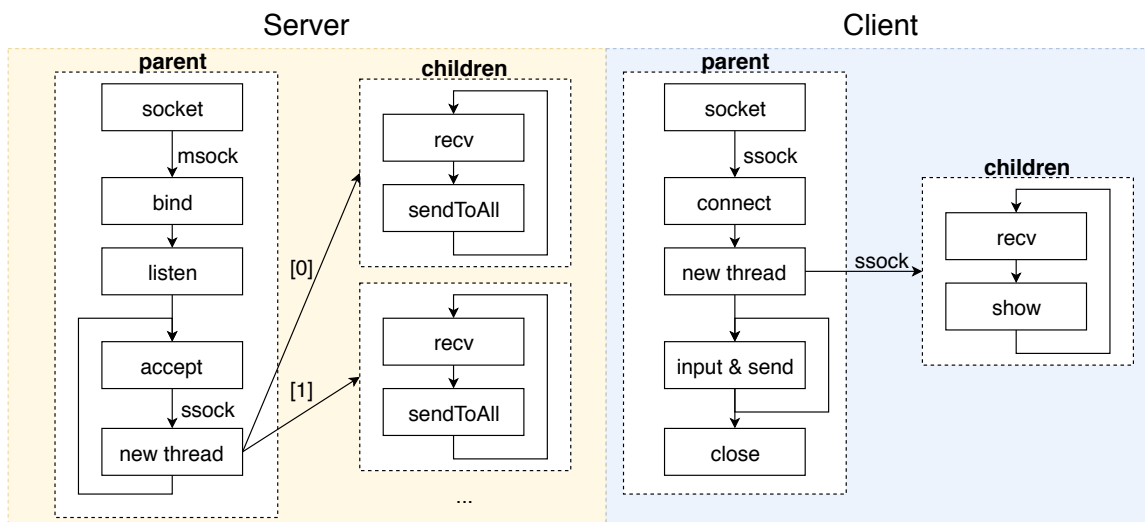
本机为Ubuntu 18.04 (LTS) + gcc 7.3.0

### 五、注意事项

- 依步骤完成以下实验内容，尽量多完成一些步骤
- 把出现的问题、解决方法和一些思考和体会写在实验体会部分
- 对典型的运行情况的客户和服务器控制台进行截屏并放在实验结果部分
- 截屏用按键(Ctrl+Alt+PrintScreen)单独截取控制台窗口。截屏应尽量windows绘图程序缩小尺寸(能看清就行)后粘入
- 端口号采用50500
- 字符串可以采用函数scanf或gets输入

### 六、实验内容

先阅读课件“套接字并发编程.PDF”，重点是读懂课件中“chat并发编程(服务器)”和“chat并发编程(客户端)”的流程图，然后完成下面步骤（截屏要同时显示服务器和至少两个客户端）。



## 1. 多人聊天程序

编写多人聊天程序，要求客户端和服务端都采用多线程方式进行编程。每个客户端都采用TCP协议连接服务器并保持连接。服务器同时与所有客户端建立和保持连接。每个客户端输入的消息都会通过服务器转发给所有客户。

Linux环境下线程调用需要引入头文件<pthread.h>，同时添加编译指令-pthread。创建线程为pthread\_create，退出线程为pthread\_exit，线程等待为pthread\_join。

服务器端程序：

客户端程序：

在本实验中我还编写了Makefile文件，方便自动编译。

```

1 CC = gcc
2 FLAGS = -pthread
3 ALL = server client
4
5 all: $(ALL)
6
7 % : %.c
8     $(CC) $(FLAGS) $< -o $@
9
10 .PHONY : clean
11
12 clean :
13     -rm -f *.o $(ALL)

```

运行截屏：

## 2. 添加信息后转发

服务器程序转发某个客户端发来的消息时都在消息前面加上该客户端的IP地址和端口号以及服务器的当前时间。要求服务器程序把转发的消息也显示出来。

服务器程序（修改部分）：

运行截屏：

## 3. “进入”信息发送

新客户刚连接时服务器端把enter消息（包含客户端IP地址和端口号）发送给所有客户端。

服务器程序（修改部分）：

运行截屏：

## 4. “退出”信息发送

客户端输入exit时退出客户端程序（正常退出），或者客户端直接关闭窗口退出（异常退出），服务器都会把该客户leave的消息广播给所有客户。

服务器程序（修改部分）：

运行截屏：

## 5. 与老师服务器连接

运行客户端程序测试与老师的服务器程序的连接（172.18.187.9:50500）。

运行截屏（客户端）：

## 6. 与同学互测

与同学的程序进行相互测试，一个人可以与多人测试，截屏选择其中一个。

同学的学号姓名（可以多人）：17341059黄杨峻、17341111刘学海

作为服务器运行截屏：

作为客户端运行截屏：

## 七、完成情况

是否完成以下步骤？（✓完成 ✗未做）

1. [✓]    2. [✓]    3. [✓]

4. [✓]    5. [✓]    6. [✓]

## 八、实验体会