计算机网络第二次实验报告

PB20111686 黄瑞轩

1. 实验内容

高级聊天程序：一个服务器，多个客户端，服务器负责消息中转，客户端之间可以互相聊天（广播）。

1. 实验环境

实验平台：Linux 实验语言：C++

1. 实验过程

**程序流程图**：

nc命令发出请求

接收请求，建立连接

IO复用

**主要函数如下**：

void epoll\_add(int epollfd, int fd); // 在EPOLL中注册新事件的函数

void client\_destroy(int epfd, Client\* user); // 用户退出时执行的函数

void client\_add(int epfd, int fd); // 用户加入时执行的函数

void send\_all(Client\* client, std::string s); // 将client发来的消息s广播

ssize\_t receive(Client\* client, void\* buf, size\_t n); // 接收消息至buf中

int main(int argc, char\*\* argv); // 主函数，初始化socket服务后循环收-发

**主要变量如下**：

Client client\_list[MAX\_CLIENT\_NUMBER]; // 目前的用户列表

std::map<int, int> client\_map; // 根据fd找到其对应的用户编号index

int connected\_clients\_num; // 当前已连接用户数

**程序操作过程**：

我编写的程序属于一个服务应用，在命令行上执行时，应提供端口名作为参数，如：

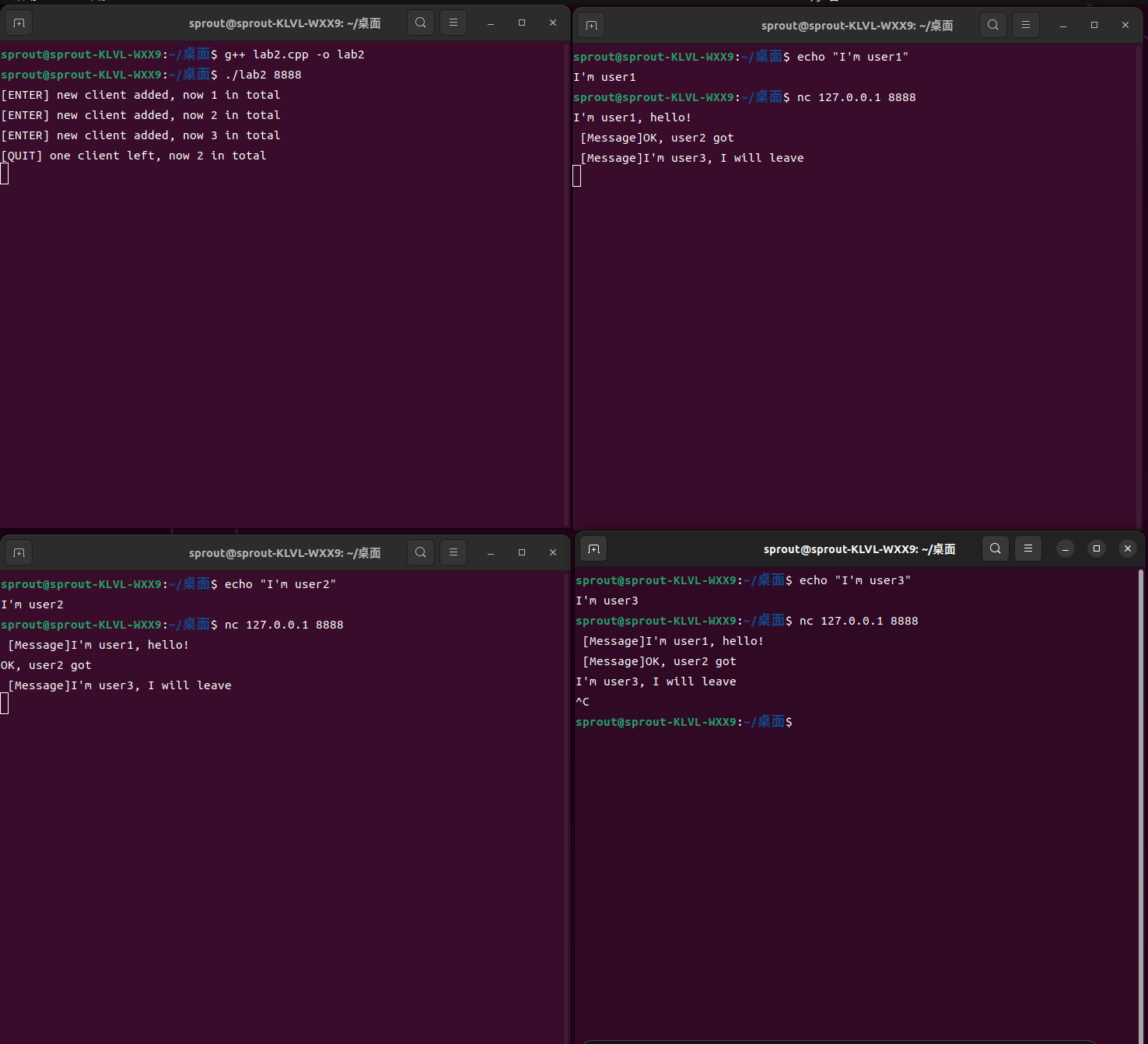
$ ./lab2 8888

这会在8888端口创建服务，各用户使用netcat包中的nc命令连接此端口即可，如：

$ nc 127.0.0.1 8888

此时服务应用建立连接，用户端发出的信息将被服务应用转发给每一个连接，用户端也会不断接收服务应用发来的信息。这里连接的建立采用的是EPOLL的IO复用技术，使用非阻塞SOCKET连接以提升recv的效率。

1. 实验结果截图



注：左上方是服务端，其余为用户端。

1. 提交文件说明

lab2.h // 头文件

lab2.cpp // 源文件

lab2 // 可执行文件

lab2.docx // docx 格式的报告

lab2.pdf // pdf 格式的报告

编译命令：

$ g++ lab2.cpp -o lab2