

项目名称：聊天软件企鹅拿大顶

项目编号：

文档编号：

聊天软件企鹅拿大顶

项目要求书

版本：Ver1.0

作者：01 组

组员介绍

组号：1

组长：应天成

成员：陈阳，高志，张璨，钟飞文，应天成，谢亚男

分工：

应天成：客户端，服务器后台程序

张璨：客户端

陈阳：客户端 UI，前台程序

高志：客户端 UI，前台程序

谢亚男：文件传输系统

钟飞文：测试，书写文档

北京理工大学 07111306 班

2015 年 9 月 11 日星期五

目录

目录

- 1 项目概要..... 4
 - 1.1 项目名称..... 4
 - 1.2 项目目标..... 4
 - 1.3 软件概要..... 4
 - 1.4 功能描述..... 4
 - 1.5 开发环境..... 5
 - 1.6 关键技术..... 5
- 2.软件详细需求..... 6
 - 2.2.1 发送信息..... 7
 - 2.2.2 接受信息..... 7
 - 2.3.3 收发文件..... 7
 - 2.2.4 用户注册..... 7
- 3.系统整体设计..... 7

1 项目概要

1.1 项目名称

聊天软件--企鹅拿大顶

1.2 项目目标

1. 体会从需求理解出发，到软件整体设计，详细设计，开发，测试，发布的整体流程。熟悉软件开发整体过程
2. 了解 linux 操作系统下的应用程序开发方法
3. 掌握 linux 系统下 socket 的使用
4. 掌握 linux 系统下开发工具 vi 编辑器、gtk、gdb、gcc 的使用
5. 培养快速学习新的知识，并在项目中使用的能力
6. 培养团队合作项目的相关能力(如分工等)

1.3 软件概要

开发一个运行于 LINUX 平台的聊天软件，可以实现电脑间的信息（包含以文件形式表示的信息）收发送。

1. 聊天
2. 发送文件

1.4 功能描述

包含以下的功能：

- 1、接受连接到服务器上其他电脑的数据（甚至可以是以各种文件的形式呈现的数据）；
- 2、发送数据（甚至可以是以各种文件的形式呈现的数据）给局域网上其他电脑；
- 3、图形界面显示收发送信息（同时附有收发送时间）；
- 4、发送及相关功能按键
- 5、用户需要注册账号，提供需要密码才能登录的服务；
- 6、显示在线用户，同时支持群聊和私聊
- 7、有文件传送过来时，界面切换为绿色主题，提醒用户接收文件。

1.5 开发环境

OS: Ubuntu 7.10

Compiler: GCC+GDB

GTK+

1.6 关键技术

GTK;

Socket;

C 语言编程

多线程

文件读取

2.软件详细需求

2.1 界面

登录界面

Login Window

Username:

Password:

Login

Regester

注册界面

Regester Window

User:

Password:

Confirm Passwd:

Confirm

聊天界面

zc

record:

Server Connected!

message:

friends:

SEND

file

2.2 数据处理

这部分就界面相关联功能进行描述：

点击 **register**, 弹出注册界面, 根据提示输入账号密码, 点击弹出界面中的“**register**”, 即可实现注册;

启动客户端后, 输入已注册的账号密码, 点击 **login**, 即可登录到聊天界面;

聊天界面上有一滑动选项, 显示 “**group**”和在线用户; 选择 “**group**”, 可进行群聊; 选择相应在线用户, 可进行私聊。在文本框输入信息, 点击 “**send**”, 即可发送信息; 点“**file**”可实现发送文件; 在有人发送文件给你时, 会有提示, 并且出现“**recvfile**”按钮; 点击该按钮, 选择相应文件夹, 命名后即可收取。聊天界面上方为聊天记录。聊天记录由收到信息时间和相应信息构成。

2.2.1 发送信息

将指定信息发送给指定在线用户, 通过界面的文本框进行信息输入, 点击 “**send**”按钮进行发送。

2.2.2 接受信息

接受相关用户发送过来的信息, 对方发送过来的信息自动显示在界面接受文本框中(附带发送信息的时间)。

2.3.3 收发文件

点击“**file**”可实现发送文件; 在有人发送文件给你时, 会有文字提示, 整个界面会变换为绿色主题, 并且出现 “**recvfile**”按钮; 点击该按钮, 选择相应文件夹, 命名后即可收取。

2.2.4 用户注册

用户在注册后才能够享受相应服务。注册过程较为简单: 点击 “**register**”, 进入注册界面, 输入自定义账号密码, 点 “**confirm**”, 如提示 “**register success!**”则为注册成功。

3.系统整体设计

3.1 系统结构图

3.2 模块要求

各个模块简单说明

GUI 层:

- GUI MainFrame: 利用 gtk+实现
- Sendmsg 按钮: 点击后将数据发送, 利用 gtk+实现
- 退出按钮: 点击退出程序, 利用 gtk+实现
- 发送数据框: 发送数据填写处, 利用 gtk+实现
- 接受数据框: 显示接受数据, 利用 gtk+实现
- Ip 地址设置框: 设置要发送数据的电脑 ip, 利用 gtk+实现

数据处理:

1.客户端函数:

(1). void receive_file_zc(char* save_path,int socket_fd)

功能: 接受文件发送方发送过来的文件

参数:

save_path: 所收到的文件的保存路径

socket_fd: 发送方的 socket 标示符

返回值: 空

(2). void send_file_zc(char* path,int socket_fd)

功能: 向他人发送文件

参数:

path: 所发送的文件的路径

socket_fd: 接收方的 socket 标示符

返回值: 空

(3). void input_destination_user_zc()

功能: 输入目标用户

参数: 无

返回值: 无

(4). char* time_display_zc();

功能: 以字符串的形式输出系统时间

参数: 无

返回值: 包含系统时间的字符串的起始地址

(5). void send_to_zc(char* des, char* buff)

功能: 将信息发送给指定用户

参数:

des: 目标用户

buff: 所发送的信息

返回值: 无

(6) . void g_chat_send(GtkWidget *widget,gpointer *data)

功能：将 entry 中的内容发送，调用 send 函数，并且在本地 text 上显示

参数：widget 控件 data 数据

返回值：无

(7). void g_text_show(gchar *message)

功能：将 message 在 text 中显示

参数：message 想打印的字符串

返回值：无

(8) void g_chat_file(GtkWidget *button,gpointer userdata)

功能：单击聊天窗口中的 file 打开选择文件窗口

参数：widget 控件 data 数据

返回值：无

(9) void g_file_select (GtkWidget *button ,GtkFileSelection *fs)

功能：单击选择文件窗口的确定按钮，会得到文件的路径。在其中调用 send 函数。

参数：widget 控件 data 数据

返回值：无

(10) .void g_list_add(gchar name[])

功能：将 name 中的字符串添加到下拉列表中

参数：name 想要添加的字符串

返回值：无

(11) . void g_list_delete(gchar name[])

功能：将 name 中的字符串从下拉列表中删除

参数：name 想要删除的字符串

返回值：无

(12) . void g_create_chat_window()

功能：创建聊天窗口

参数：无

返回值：无

(13) on_login_button_clicked()

功能：登录验证，判断用户名和密码是否正确

参数：无

返回值：无

(14) on_reg_window_X()

功能：关闭注册窗口后，取消登录窗口的隐藏状态，恢复显示状态

参数：无

返回值：无

(15) reg_check()

功能：注册检查，判断用户名是否已存在，密码和再次输入的密码是否匹配

参数：无

返回值：无

(16) on_reg_button_clicked()

功能：初始化注册界面

参数：无

返回值：无

(17) create_login_window ()

功能：初始化登录界面

参数：无

返回值：FALSE

(18) void pthread_recv_file(char *file_IP)

功能：用于接收文件的线程

参数：文件发送端的 IP

返回值：无

(19) void pthread_send_file(char *filename)

功能：用于发送文件的线程

参数：文件的路径+文件名 （相对路径绝对路径均可）

返回值：无

(20) void send_pack_owen(char* order , char* data , char* result)

功能：将原始数据打成数据包，从而实现不同的数据包实现的功能不同

参数：

order: 指令（一个大写字母，比如 “I”）

data: 需要发送的数据

result: 打包后的字符串，即打好的包

返回值： 无

(21) char receive_pack_owen(char *raw_data , char *pro_data)

功能：对收到的数据包进行解包

参数：

raw_data: 收到的没有处理过的数据包

pro_data: 经过解包后的数据包

返回值： char: 包头所包含的指令，即打包时的第一位，比如 ‘I’

(22) void input_ip_port_owen(char *IP , int PORT)

功能：输入要连接的 IP 和端口

参数： IP ， 端口

返回值： 无

(23) void get_socket_owen()

功能：进行 socket 连接

参数：无

返回值：无

(24) void connect_socket_owen()

功能：用于 socket 的连接

参数：无

返回值：无

(25) void input_name_owen()

功能：将本机的用户名发送到服务器

参数：无

返回值：无

(26) void send_text_owen(char *send_buf)

功能：向服务器发送数据

参数：

send_buf: 要向服务器发送的数据

返回值：无

(27) void pthread_function(int sockfd)

功能：用于接收服务器发过来的数据，并且对信息进行解包

参数：无

返回值：无

2.服务器函数

(1).void send_zc(char* des,char* buf,char* source)

功能：将源用户通过服务器转发的信息发送给目标用户

参数：

des: 目标用户

buf: 要发送的内容

source: 发送信息的用户

返回值：无

(2) void init_count_owen()

功能：用于初始化存储当前注册用户数量的函数，当文件不存在时，会自动创建文件

参数：无

返回值：无

(3) void count_plus_owen()

功能：当有新用户注册的时候，将文件中的用户数量读取出来，自增 1，再存回去

参数：无

返回值：无

(4) int judge_reg_owen(char *username , char *password , char *confirm_password)

功能：对注册时的用户名，密码，确认密码进行校验

参数：

Username: 用户名

Password: 密码

Confirm_password: 确认密码

返回值：

是否注册成功：0-success 1-passwdnotsame 2-userhasbeenexist 3-overlength

(5) int reg_owen(char *username , char *password , char *confirm_password)

功能： 验证注册是否成功

参数： 同（4）

返回值： 同（4）

注： 本函数调用了（4）函数，并包装了一些新的功能。

(6) void read_from_file_owen(int struct_number , struct user_info *user)

功能： 从文件中读取文本数据

参数： struct_number: 要读取用户的数量

User_info: 读取出的用户信息

返回值： 无

(7) int login_owen(char *username , char *password)

功能： 验证登陆函数

参数：

Username: 用户名

Password: 密码

返回值： 是否登陆成功

4--emptyusernameorpassword , 0--success , 3--password wrong , 2--username not exist

(8) void pthread_function(int client_fd)

功能： 服务器的关键线程，用于对数据的转发，注册，登陆等所有主要功能

参数： 当前接入客户端的文件描述符

返回值： 无

- 线程 1: 利用 socket 通信发送数据
- 线程 2: 利用 socket 通信接收数据

使用说明书

启动客户端；点击 **register**，弹出注册界面，根据提示输入账号密码，点击弹出界面中的“**register**”，即可实现注册；

启动客户端后，输入已注册的账号密码，点击 **login**，即可登录到聊天界面；

聊天界面上有一滑动选项，显示“**group**”和在线用户；选择“**group**”，可进行群聊；选择相应在线用户，可进行私聊。在文本框输入信息，点击“**send**”，即可发送信息；聊天界面上方为聊天记录。聊天记录由收到信息时间和相应信息构成。