



C 语言手写 QQ-AI 版

1. 项目介绍

使用 C 语言实现腾讯 QQ 界面风格的聊天软件，实现 AI 对话。

2. 项目目标

1. 掌握 C 语言开发客户端软件的基本方法
2. 掌握 C 语言实现用户界面渲染、用户界面响应的底层机制
3. 掌握网络编程的基本概念
4. 掌握多线程开发的基本用法

适合初学者的“提升项目”

3. 项目演示

开始表演...

4. 讲师介绍

讲师介绍 Rock老师



» 高薪就业 包就业服务 IT外包 技术变现



资深架构师、项目经理
12年一线研发、8年技术培训
已培养3000余名高级程序员.



负责研发了多个广州军区南海无线通信项目、卫星通信监控平台、联通IDU系统.



擅长以最简单的方式深入阐述复杂的概念.



二十余所高校项目实训经验.



5. 学习条件

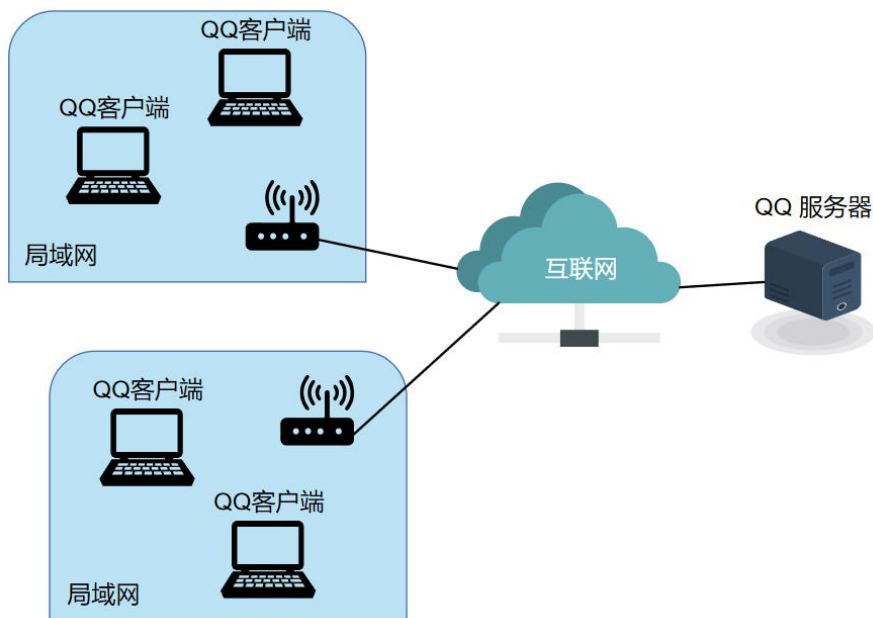
已经掌握 C 语言的基础用法:

变量、字符串, 条件判断 if, 循环语句 for/while 等基础内容

如果还没有 C 语言基础, 建议先学习【零基础-5 天入门 C 语言】

https://ke.qq.com/webcourse/index.html#cid=377567&term_id=100449934&taid=3062105523995359

6. 项目架构



7. 项目准备

1. 安装好 VS2019 或者 VS2010

VS2010 下载地址:

https://ke.qq.com/webcourse/index.html?cw_id=339501&ac_type=3#cid=377567&term_id=100449934

2019 下载地址:



2. 安装好 easyx

Easyx-适用于 VS2019 版本:

https://ke.qq.com/webcourse/index.html?cw_id=339507&ac_type=3#cid=377567&term_id=100449934&taid=11919272981152479&type=1024

Easyx-适用 VS2019 以下的版本:

https://ke.qq.com/webcourse/index.html?cw_id=339508&ac_type=3#cid=377567&term_id=100449934&taid=11919277276119775&type=1024

3. 领取项目素材

1. 图片素材



res.rar

2. 库文件



tx_qq.h



tx_qq.lib

1. 客户端项目实施

1. 创建空项目

2. 导入素材

导入素材 res 目录到项目目录下。

3. 添加第 3 方库文件

tx_qq.lib



tx_qq.h

```
#include "tx_qq.h"

#pragma comment(lib, "tx_qq.lib")

int main(void) {
    printf("hello world\n");
    return 0;
}
```

大佬必经之路，先学会使用库，再开发自己的库。

VIP 学员凌云，入职的第一个项目，是智能电视 APP，项目中有一千多个文件，几十个库文件。【大专学历，入职 C++ 软件工程师，开发智能电视 APP，月薪 15K】

4. 输入对方 IP 地址

```
char ip_addr[32];

int main(void) {
    printf("请输入对方的 IP 地址: ");
    scanf_s("%s", ip_addr, sizeof(ip_addr));

    return 0;
}
```

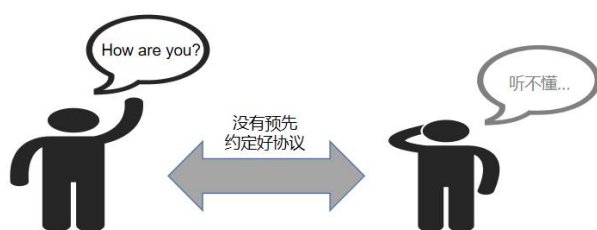
IP 地址是什么？



5. 网络通信之前的初始化

5.1 网络开发快速入门

什么是网络协议

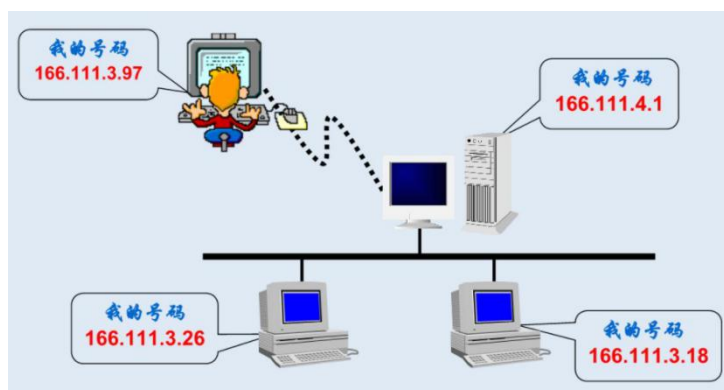


最常用的网络协议 TCP 与 UDP





什么是 IP 地址



什么是端口

端口号，就是“在同一个地方，问需要什么服务”





298 服务



198 服务



最常用的端口号：

80 端口 - WEB 服务

www.baidu.com 就是 www.baidu.com:80

什么是套接字

【IP + 端口号】组合成一个“插座”，这个“插座”就是套接字
使用“套接字”，发送数据/接收数据

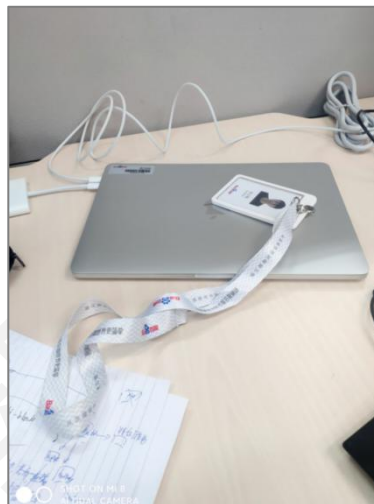
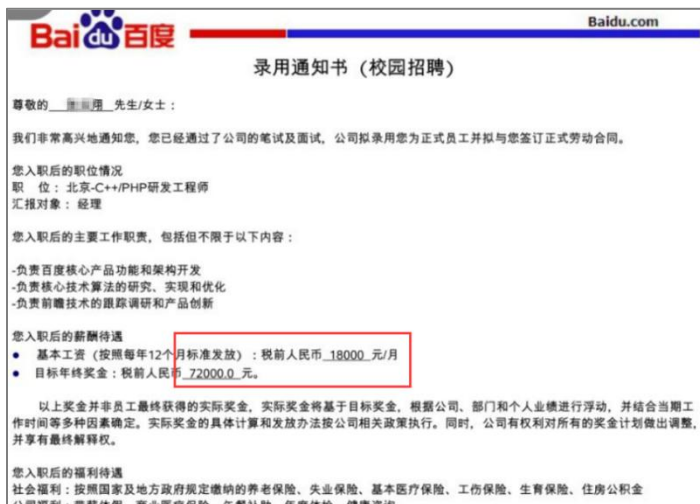




大佬必经之路，网络编程。不懂网络开发的程序员不是真正的程序员。

VIP 学员小翔，入职 C++ 服务器开发工程师，研发搜索引擎架构。服务器端的搜索引擎，是网络编程中的服务器端的“数据发动机”。

【小翔，入职百度，二本学历（河南工程学院）， 年薪 28.8 万。】



5.2 网络通信前的初始化

- 1) 指定网络协议
- 2) 设置网络地址
- 3) 指定通信的端口
- 4) 指定对端的 IP 地址
- 5) 创建套接字

```
SOCKET serverSocket;  
sockaddr_in sockAddr;  
#define PORT 2021  
  
#include <winsock.h>  
#pragma comment(lib, "ws2_32.lib")
```

调用初始化函数

```
int main(void) {  
    printf("请输入对方的 IP 地址: ");  
    scanf_s("%s", ip_addr, sizeof(ip_addr));
```




```
if (!TCPInit(&serverSocket, &sockAddr, ip_addr, PORT)) {  
    printf("网络初始化失败!\n");  
    return -1;  
}  
  
return 0;  
}
```

6. 连接奇牛客户端

```
void connect() {  
    // 连接服务器端（发起网络连接请求）  
    connect(serverSocket, (SOCKADDR*)&sockAddr, sizeof(sockAddr));  
    printf("已经接入!\n");  
}
```

调用 connect

```
int main(void) {  
    printf("请输入对方的 IP 地址: ");  
    scanf_s("%s", ip_addr, sizeof(ip_addr));  
  
    if (!init()) {  
        printf("网络初始化失败!\n");  
        return -1;  
    }  
  
    printf("正在连接奇牛客服...\n");  
    connect();  
  
    return 0;  
}
```

大佬必经之路，学会封装。善于封装，是程序员的基本修养。

VIP 学员辰星，入职第一个项目，重构公司的一个 MFC 上位机项目，改用 Qt 框架重新实现。入职的第二周，哭了，原始项目代码太烂了。最后放弃重构。在 Rock 的建议下，自己重写核心功能，封装成新项目的底层接口，最后提前完成项目，成为公司的核心开发人员。

【辰星，大专学历（深圳职业技术学院），转正薪资 15000】



7. 初始化图形 QQ

```
HWND hwnd; // 聊天窗口句柄
int screenWidth;
int screenHeight;
int msgHeight; // 新气泡的 y 坐标

IMAGE imgBg; //聊天窗口背景
IMAGE imageArrows[2]; //气泡的箭头
IMAGE imageHeads[2]; //人物的头像

// 三个按钮
Button btnClose;
Button btnTitle;
Button btnSend;

void initUI() {
    // 1. 创建聊天窗口
    initgraph(WINDOW_WIDTH, WINDOW_HEIGHT, EW_SHOWCONSOLE); // 创建绘图窗口
    setbkmode(TRANSPARENT);

    // 2. 移动窗口位置
    screenWidth = GetSystemMetrics(SM_CXSCREEN);
    screenHeight = GetSystemMetrics(SM_CYSCREEN);
    hwnd = GetHwnd(); //获取当前窗口句柄
    SetWindowLong( //设置窗口属性
        hwnd,
        GWL_STYLE, //设定一个新的窗口风格。
        //GetWindowLong 获取指定串口的属性
        GetWindowLong(hwnd, GWL_STYLE) - WS_CAPTION); //WS_CAPTION 带标题栏的窗口风格
    MoveWindow(hwnd, screenWidth / 8, 100, WINDOW_WIDTH, WINDOW_HEIGHT, false);

    // 3. 绘制背景图片
    loadimage(&imgBg, "res/bg.png");
    putimage(0, 0, &imgBg);

    // 4. 加载气泡尾巴和头像
    loadimage(&imageArrows[0], "res/left_arrow.jpg", 6, 8, true);
    loadimage(&imageArrows[1], "res/right_arrow.jpg", 6, 8, true);
    loadimage(&imageHeads[1], "res/niu.jpg", 44, 51, true);
    loadimage(&imageHeads[0], "res/rock.jpg", 55, 51, true);

    // 5. 初始化 3 个按钮
```



```
// 5.1 初始化关闭按钮
initButton(&btnClose, "res/close_normal.jpg", "res/close_press.jpg", 32, 33, 0);
btnClose.x = WINDOW_WIDTH - 32;
btnClose.y = 0;

// 5.2 初始化标题按钮
initButton(&btnTitle, "res/title.jpg", "res/title.jpg", 460, 39, 0);
btnTitle.x = 0;
btnTitle.y = 0;

// 5.3 初始化发送按钮
initButton(&btnSend, "res/send_normal.jpg", "res/send_press.jpg", 88, 28, 0);
btnSend.x = 337;
btnSend.y = 784;

// 6. 初始化气泡位置
msgHeight = 120;

// 7. 设置编辑区域的文本颜色
setcolor(BLACK);

}
```

位置说明：



调用 initUI

```
int main(void) {
```



```
printf("请输入对方的 IP 地址: ");
scanf_s("%s", ip_addr, sizeof(ip_addr));

if (!init()) {
    printf("网络初始化失败!\n");
    return -1;
}

printf("正在连接奇牛客服...\n");
connect();

initUI();

return 0;
}
```

大佬必经之路, 基础功能不够, 就创建这个基础功能。

这段代码, 使用了 Button 组件, 实际上 easyx 没有 Button 组件, 快速入门, 开发自己的高级组件!

VIP 学员正在缓冲, 在家做 IT 外包, 为了提高外包开发效率, 定义了大量外包开发中的常用组件, 最近接手了一个麻将棋牌游戏外包, 开发周期一个月, 收入 $20000 \times 0.8 = 16000$ 元。开发完这个棋牌外包后, 把项目进行优化, 开始承接专业棋牌游戏定制开发。

【正在缓冲, 高中学历, 外包开发月入 20000+】



12"

对的, 我一开始想的也是, 我现在不太想上班儿了, 就想自己在家做点儿东西, 借点儿包点儿包, 然后再慢慢深入学习。



10"

这个项目外包的费用是2万, 然后给介绍我的那人儿是20%, 剩下是我的。

8. 创建多线程，并发处理信息编辑和信息接收

一心多用

语音

编辑

讨论

上传视频

+

★ 收藏

👍 67

🗨 14

一心多用就是指同时干好几件事情，比如，一边做作业一边看电视，或者一边上网一边和家人交谈，对此专家研究发现，同时做几件事不仅不会提高效率，而且对大脑有害。“一心多用”会影响正常的脑力活动，引发多种疾病。

中文名	一心多用	来源	俄罗斯《报纸报》网站
外文名	Multitasking	相关因素	同额叶前部皮层提高工作效率有关
		影响	影响健康，工作效率，致病

目录

1 研究案例

2 影响

• 身心健康

• 导致事倍功半

• 使人暴躁好斗

• 导致疾病

3 相关因素

4 科学研究

5 研究意义

程序，通过“多线程”来实现“一心多用”

```
int main(void) {
    printf("请输入对方的 IP 地址: ");
    scanf_s("%s", ip_addr, sizeof(ip_addr));

    if (!init()) {
        printf("网络初始化失败!\n");
        return -1;
    }

    printf("正在连接奇牛客服...\n");
    connect();

    initUI();

    DWORD threadID = 0;
    HANDLE handleSecond = CreateThread(NULL, 0, ThreadFuncRcv, 0, 0, &threadID);
    HANDLE handleEdit = CreateThread(NULL, 0, msgEditHandle, NULL, 0, &threadID);

    return 0;
}
```



定义线程函数

```
DWORD WINAPI ThreadFuncRcv(LPVOID param) {  
    return NULL;  
}  
  
DWORD WINAPI msgEditHandle(LPVOID param) {  
    return NULL;  
}
```

大佬必经之路，多线程是程序开挂的必备工具。

VIP 学员清风，入职 C++ 软件工程师，从事军工仿真开发，在项目中大量使用多线程，把 CPU 的性能最大程度地挖掘出来。

【清风，本科学历，Qt 军工仿真开发，月薪 20000】



9. 实现编辑功能

```
DWORD WINAPI msgEditHandle(LPVOID param) {  
    textBox(10, 663, 420, 110, LINE_HEIGHT, WHITE, BLACK, msgEdit, &msgLen);  
    return NULL;  
}
```




此时还不能实现 UI 互动，其他功能还没有实现。

10. 实现界面的消息处理

```
void mainUI() {
    while (1) {
        MOUSEMSG m = GetMouseMsg();
        FlushMouseMsgBuffer(); //不能少，后缀快速拖动顶部的标题按钮，将导致鼠标消息太多，
        出现混乱！
        switch (m.uMsg) {
            case WM_MOUSEMOVE:
                if (checkButtonSelect(&btnTitle, &m)) {

                }
                else if (checkButtonSelect(&btnSend, &m)) {
                    btnSend.pressed = true;
                    drawButton(&btnSend);
                }
                else if (checkButtonSelect(&btnClose, &m)) {
                    btnClose.pressed = true;
                    drawButton(&btnClose);
                }
                else {
                    // 检查鼠标是否从按钮内移动到按钮之外
                    if (btnClose.pressed == true) { // 鼠标从关闭按钮移出
                        btnClose.pressed = false;
                        drawButton(&btnClose);
                    }
                    if (btnSend.pressed == true) { // 鼠标从发送按钮移出
                        btnSend.pressed = false;
                        drawButton(&btnSend);
                    }
                }
                break;
            case WM_LBUTTONDOWN:
                if (checkButtonSelect(&btnTitle, &m)) {

                }
                else if (checkButtonSelect(&btnClose, &m)) {
                    btnClose.pressed = true;
                    drawButton(&btnClose);
                }
                else if (checkButtonSelect(&btnSend, &m)) {
                    btnSend.pressed = true;
                }
            }
        }
    }
}
```



```
        drawButton(&btnSend);
    }
    break;
case WM_LBUTTONDOWN:
    if (checkButtonSelect(&btnClose, &m)) {
        //btnClose.pressed = false;
        //drawButton(&btnClose);
        closegraph();
        exit(0);
    }
    else if (checkButtonSelect(&btnSend, &m)) {

    }
    else if (checkButtonSelect(&btnTitle, &m)) {

    }
    break;
}
}
```

调用 mainUI

```
int main(void) {
    printf("请输入对方的 IP 地址: ");
    scanf_s("%s", ip_addr, sizeof(ip_addr));

    if (!init()) {
        printf("网络初始化失败!\n");
        return -1;
    }

    printf("正在连接奇牛客服...\n");
    connect();

    initUI();

    DWORD threadID = 0;
    HANDLE handleSecond = CreateThread(NULL, 0, ThreadFuncRcv, 0, 0, &threadID);
    HANDLE handleEdit = CreateThread(NULL, 0, msgEditHandle, NULL, 0, &threadID);

    mainUI();

    system("pause");
}
```



```
return 0;
}
```

如果服务端已经开启，可以检查以下效果。

1. 编辑功能：中文编辑，英文编辑，混合编辑，换行，退格删除
2. 发送按钮和关闭按钮的滑动变色效果
3. 关闭按钮的关闭功能

大佬必经之路，学会制造自己的轮子。大佬开发之道，避免重复制造轮子。

VIP 学员海将河推走，入职客户端开发工程师，手写了网络开发各个底层组件。

【海将河推走，二本学历， 入职字节跳动，月薪 25000，年终奖 75000】



11. 实现窗口的拖动功能

和客户端的窗口拖动相同。

因为已经把默认的标题栏隐藏，所以需要额外实现窗口的拖动功能。

```
void mainUI() {
    bool titleDrag = false; //表示“标题栏”是否被单击拖动
    int titleLastX; //窗口的上一次位置（X 坐标位置）
    int titleLastY; //窗口的上一次位置（X 坐标位置）

    while (1) {
        MOUSEMSG m = GetMouseMsg();
        FlushMouseMsgBuffer(); //不能少，后续快速拖动顶部的标题按钮，讲导致鼠标消息太多，
        出现混乱！
        switch (m.uMsg) {
```



```
case WM_MOUSEMOVE:
    // 鼠标滑过标题栏
    if (checkButtonSelect(&btnTitle, &m)) {
        if (btnTitle.pressed == true) {
            if (titleDrag == false) { // 此时标题栏已经被点击按下，正准备拖动
                titleLastX = m.x; // 记录初始坐标
                titleLastY = m.y;
                titleDrag = true;
            }
            else { // 此时标题栏已经被点击按下，正在拖动
                // 计算拖动偏移量
                int offX = m.x - titleLastX;
                int offY = m.y - titleLastY;
                moveWindow(hwnd, offX, offY); // 根据拖动偏移量，移动窗口
            }
        }
    }
    else if (checkButtonSelect(&btnSend, &m)) {
        btnSend.pressed = true;
        drawButton(&btnSend);
    }
    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    }
    else {
        // 检查鼠标是否从按钮内移动到按钮之外
        if (btnClose.pressed == true) { // 鼠标从关闭按钮移出
            btnClose.pressed = false;
            drawButton(&btnClose);
        }
        if (btnSend.pressed == true) { // 鼠标从发送按钮移出
            btnSend.pressed = false;
            drawButton(&btnSend);
        }
    }
    break;
case WM_LBUTTONDOWN:
    if (checkButtonSelect(&btnTitle, &m)) {
        btnTitle.pressed = true; // 单击按下标题栏
    }
    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    }
}
```



```
        else if (checkButtonSelect(&btnSend, &m)) {
            btnSend.pressed = true;
            drawButton(&btnSend);
        }
        break;
    case WM_LBUTTONDOWN:
        if (checkButtonSelect(&btnClose, &m)) {
            //btnClose.pressed = false;
            //drawButton(&btnClose);
            closegraph();
            exit(0);
        }
        else if (checkButtonSelect(&btnSend, &m)) {
        }
        else if (checkButtonSelect(&btnTitle, &m)) {
            // 松开标题栏按钮（左键抬起）
            btnTitle.pressed = false;
            titleDrag = false;
        }
        break;
    }
}
```

测试效果。

12. 发送信息

左键抬起时发送。

```
msg_t msgAll[5];
int msgCount = 0;
int currentMsgIndex = -1; //当前最近的一条信息的索引

void mainUI() {
    bool titleDrag = false; //表示“标题栏”是否被单击拖动
    int titleLastX; //窗口的上一次位置（X 坐标位置）
    int titleLastY; //窗口的上一次位置（X 坐标位置）

    while (1) {
        MOUSEMSG m = GetMouseMsg();
        FlushMouseMsgBuffer(); //不能少，后缀快速拖动顶部的标题按钮，讲导致鼠标消息太多，
        出现混乱！
    }
}
```



```
switch (m.uMsg) {
case WM_MOUSEMOVE:
    // 鼠标滑过标题栏
    if (checkButtonSelect(&btnTitle, &m)) {
        if (btnTitle.pressed == true) {
            if (titleDrag == false) { // 此时标题栏已经被点击按下，正准备拖动
                titleLastX = m.x; // 记录初始坐标
                titleLastY = m.y;
                titleDrag = true;
            }
            else { // 此时标题栏已经被点击按下，正在拖动
                // 计算拖动偏移量
                int offX = m.x - titleLastX;
                int offY = m.y - titleLastY;
                moveWindow(hwnd, offX, offY); // 根据拖动偏移量，移动窗口
            }
        }
    }
    else if (checkButtonSelect(&btnSend, &m)) {
        btnSend.pressed = true;
        drawButton(&btnSend);
    }
    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    }
    else {
        // 检查鼠标是否从按钮内移动到按钮之外
        if (btnClose.pressed == true) { // 鼠标从关闭按钮移出
            btnClose.pressed = false;
            drawButton(&btnClose);
        }
        if (btnSend.pressed == true) { // 鼠标从发送按钮移出
            btnSend.pressed = false;
            drawButton(&btnSend);
        }
    }
    break;
case WM_LBUTTONDOWN:
    if (checkButtonSelect(&btnTitle, &m)) {
        btnTitle.pressed = true; // 单击按下标题栏
    }
    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    }
}
```




```
}
else if (checkButtonSelect(&btnSend, &m)) {
    btnSend.pressed = true;
    drawButton(&btnSend);
}
break;
case WM_LBUTTONDOWN:
    if (checkButtonSelect(&btnClose, &m)) {
        //btnClose.pressed = false;
        //drawButton(&btnClose);
        closegraph();
        exit(0);
    }
    else if (checkButtonSelect(&btnSend, &m)) {
        btnSend.pressed = false;
        drawButton(&btnSend);

        int ret = send(serverSocket, msgEdit, msgLen, 0);
        printf("已经发送%d 个字符\n", ret);

        currentMsgIndex = (currentMsgIndex + 1) % (sizeof(msgAll) /
sizeof(msgAll[0]));
        msgEdit[msgLen] = 0;
        strcpy(msgAll[currentMsgIndex].msg, msgEdit);
        memset(msgEdit, 0, sizeof(msgEdit));
        msgLen = 0;

        msgAll[currentMsgIndex].type = SEND;
        msgCount++;

        //drawMsg();
        drawMsg(msgAll, currentMsgIndex, &msgHeight, imageArrows, imageHeads);
    }
    else if (checkButtonSelect(&btnTitle, &m)) {
        // 松开标题栏按钮（左键抬起）
        btnTitle.pressed = false;
        titleDrag = false;
    }
    break;
}
}
```



测试效果。

13. 接收信息

```
DWORD WINAPI ThreadFuncRcv(LPVOID param) {
    char buff[4096];
    while (1) {
        int ret = recv(serverSocket, buff, sizeof(buff), 0);
        if (ret <= 0) return false;

        buff[ret] = 0; //添加字符串结束符
        printf("收到: %s\n", buff);

        currentMsgIndex = (currentMsgIndex + 1) % (sizeof(msgAll) / sizeof(msgAll[0]));

        strcpy(msgAll[currentMsgIndex].msg, buff);
        msgAll[currentMsgIndex].type = RECEIVE;
        msgCount++;
        //drawMsg();
        drawMsg(msgAll, currentMsgIndex, &msgHeight, imageArrows, imageHeads);
    }

    return NULL;
}
```

添加音效。

14. 客户端项目开发总结

1. 使用多线程，实现主界面的消息处理、消息接收、信息编辑，三条线并发执行。
2. 使用图形库 `easyx`，实现界面组件的开发和使用。
3. 表情功能、文件传输、视频电话、音频电话、账号登录等待完善。

VIP 学员项目分享

<https://www.bilibili.com/video/BV1ua4y1n73P>

C 语言基础掌握了，还要学习什么？

1. 选择目标



- A 高薪就业
- B 自由程序员
- C 逆向开发

- 2. 确定路线
- 3. 加油干!

2. 服务端实现

1. 创建空项目

2. 导入素材

导入素材 res 目录到项目目录下。

3. 添加第 3 方库文件

tx_qq.lib
tx_qq.h

```
#include "tx_qq.h"

#pragma comment(lib, "tx_qq.lib")

int main(void) {
    printf("hello world\n");
    return 0;
}
```

4. 网络通信之前的初始化

初始化:

- 1) 初始化网络协议
- 2) 创建套接字
- 3) 设置网络地址
- 4) 绑定套接字



```
#include <winsock.h>
#pragma comment(lib, "ws2_32.lib")

#define PORT 2021
SOCKET serverSocket;

bool init()
{
    WSADATA wsaData;
    int err = WSStartup(MAKEWORD(1, 1), &wsaData);
    if (err != 0) {
        return false;
    }

    serverSocket = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);

    sockaddr_in sockAddr;
    sockAddr.sin_family = PF_INET;
    sockAddr.sin_addr.S_un.S_addr = 0;
    sockAddr.sin_port = htons(PORT);

    // 绑定套接字
    bind(serverSocket, (SOCKADDR*)&sockAddr, sizeof(sockAddr));
    // 创建监听队列
    listen(serverSocket, 1);

    return NULL;
}
```

调用 init

```
int main()
{
    init();
    .....
    system("pause");
    return 0;
}
```

5. 等待客户端发起连接

```
SOCKET clientSock;
```



```
void waitAccept() {
    SOCKADDR client; // 监控端的网络地址（相当于客户端的网络地址）
    int nSize = sizeof(client);
    printf("等待客户端发起连接...\n");
    clientSock = accept(serverSocket, &client, &nSize);
    printf("客户端已经接入! \n");
}
```

调用 waitAccept

```
int main(void) {
    init();
    waitAccept();

    return 0;
}
```

6. 初始化图形 QQ

```
HWND hwnd; // 聊天窗口句柄
int screenWidth;
int screenHeight;
int msgHeight; // 新气泡的 y 坐标

IMAGE imgBg; //聊天窗口背景
IMAGE imageArrows[2]; //气泡的箭头
IMAGE imageHeads[2]; //人物的头像

// 三个按钮
Button btnClose;
Button btnTitle;
Button btnSend;

void initUI() {
    // 1. 创建聊天窗口
    initgraph(WINDOW_WIDTH, WINDOW_HEIGHT, EW_SHOWCONSOLE); // 创建绘图窗口
    setbkmode(TRANSPARENT);

    // 2. 移动窗口位置
    screenWidth = GetSystemMetrics(SM_CXSCREEN);
    screenHeight = GetSystemMetrics(SM_CYSCREEN);
    hwnd = GetHwnd(); //获取当前窗口句柄
    SetWindowLong( //设置窗口属性说
```



```
hWnd,  
GWL_STYLE,    //设定一个新的窗口风格。  
//GetWindowLong 获取指定窗口的属性  
GetWindowLong(hWnd, GWL_STYLE) - WS_CAPTION); //WS_CAPTION 带标题栏的窗口风格  
MoveWindow(hWnd, screenWidth * 0.7, 100, WINDOW_WIDTH, WINDOW_HEIGHT, false);  
  
// 3. 绘制背景图片  
loadimage(&imgBg, "res/bg.png");  
putimage(0, 0, &imgBg);  
  
// 4. 加载气泡尾巴和头像  
loadimage(&imageArrows[0], "res/left_arrow.jpg", 6, 8, true);  
loadimage(&imageArrows[1], "res/right_arrow.jpg", 6, 8, true);  
loadimage(&imageHeads[0], "res/niu.jpg", 44, 51, true);  
loadimage(&imageHeads[1], "res/rock.jpg", 55, 51, true);  
  
// 5. 初始化 3 个按钮  
// 5.1 初始化关闭按钮  
initButton(&btnClose, "res/close_normal.jpg", "res/close_press.jpg", 32, 33, 0);  
btnClose.x = WINDOW_WIDTH - 32;  
btnClose.y = 0;  
  
// 5.2 初始化标题按钮  
initButton(&btnTitle, "res/title.jpg", "res/title.jpg", 460, 39, 0);  
btnTitle.x = 0;  
btnTitle.y = 0;  
  
// 5.3 初始化发送按钮  
initButton(&btnSend, "res/send_normal.jpg", "res/send_press.jpg", 88, 28, 0);  
btnSend.x = 337;  
btnSend.y = 784;  
  
// 6. 初始化气泡位置  
msgHeight = 120;  
  
// 7. 设置编辑区域的文本颜色  
setcolor(BLACK);  
}
```

位置说明：



调用 `initUI`

```
int main(void) {  
    init();  
    waitAccept();  
    initUI();  
  
    return 0;  
}
```

7. 创建多线程，并发处理信息编辑和信息接收

```
int main()  
{  
    init();  
  
    waitAccept();  
  
    initUI();  
  
    DWORD dwThreadID = 0;  
    HANDLE handleSecond = CreateThread(NULL, 0, ThreadFuncRcv, 0, 0, &dwThreadID);  
    HANDLE handleEdit = CreateThread(NULL, 0, msgEditHandle, NULL, 0, &dwThreadID);
```



```
mainUI();

system("pause");

return 0;

}
```

线程函数:

```
DWORD WINAPI ThreadFuncRcv(LPVOID param) {
    return NULL;
}

DWORD WINAPI msgEditHandle(LPVOID param) {
    return NULL;
}
```

8. 实现编辑功能

```
char msgEdit[1024];
int msgLen;

DWORD WINAPI msgEditHandle(LPVOID param) {
    textBox(10, 663, 420, 110, LINE_HEIGHT, WHITE, BLACK, msgEdit, &msgLen);
    return NULL;
}
```

此时还不能实现 UI 互动，其他功能还没有实现。

9. 实现界面的消息处理

```
void mainUI() {
    while (1) {
        MOUSEMSG m = GetMouseMsg();
        FlushMouseMsgBuffer(); //不能少，后缀快速拖动顶部的标题按钮，将导致鼠标消息太多，
        出现混乱！
        switch (m.uMsg) {
            case WM_MOUSEMOVE:
                if (checkButtonSelect(&btnTitle, &m)) {

                } else if (checkButtonSelect(&btnSend, &m)) {
                    btnSend.pressed = true;
                }
            }
        }
    }
}
```



```
        drawButton(&btnSend);
    }
    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    } else {
        // 检查鼠标是否从按钮内移动到按钮之外
        if (btnClose.pressed == true) { // 鼠标从关闭按钮移出
            btnClose.pressed = false;
            drawButton(&btnClose);
        }
        if (btnSend.pressed == true) { // 鼠标从发送按钮移出
            btnSend.pressed = false;
            drawButton(&btnSend);
        }
    }
    break;
case WM_LBUTTONDOWN:
    if (checkButtonSelect(&btnTitle, &m)) {
    }
    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    }
    else if (checkButtonSelect(&btnSend, &m)) {
        btnSend.pressed = true;
        drawButton(&btnSend);
    }
    break;
case WM_LBUTTONUP:
    if (checkButtonSelect(&btnClose, &m)) {
        //btnClose.pressed = false;
        //drawButton(&btnClose);
        closegraph();
        exit(0);
    }
    else if (checkButtonSelect(&btnSend, &m)) {
    }
    else if (checkButtonSelect(&btnTitle, &m)) {
    }
}
```



```
        break;
    }
}
}
```

调用 mainUI

```
int main(void) {
    init();
    waitAccept();
    initUI();

    DWORD dwThreadID = 0;
    HANDLE handleSecond = CreateThread(NULL, 0, ThreadFuncRcv, 0, 0, &dwThreadID);
    HANDLE handleEdit = CreateThread(NULL, 0, msgEditHandle, NULL, 0, &dwThreadID);

    mainUI();

    return 0;
}
```

10. 实现窗口的拖动功能

因为已经把默认的标题栏隐藏，所以需要额外实现窗口的拖动功能。

```
void mainUI() {
    bool titleDrag = false; //表示“标题栏”是否被单击拖动
    int titleLastX; //窗口的上一次位置（X 坐标位置）
    int titleLastY; //窗口的上一次位置（X 坐标位置）

    while (1) {
        MOUSEMSG m = GetMouseMsg();
        FlushMouseMsgBuffer(); //不能少，后缀快速拖动顶部的标题按钮，讲导致鼠标消息太多，
        出现混乱！
        switch (m.uMsg) {
            case WM_MOUSEMOVE:
                // 鼠标滑过标题栏
                if (checkButtonSelect(&btnTitle, &m)) {
                    if (btnTitle.pressed == true) {
                        if (titleDrag == false) { // 此时标题栏已经被点击按下，正准备拖动
                            titleLastX = m.x; // 记录初始坐标
                            titleLastY = m.y;
                            titleDrag = true;
                        }
                    }
                }
            }
        }
    }
}
```



```
    }
    else { // 此时标题栏已经被点击按下，正在拖动
        // 计算拖动偏移量
        int offX = m.x - titleLastX;
        int offY = m.y - titleLastY;
        moveWindow(hwnd, offX, offY); // 根据拖动偏移量，移动窗口
    }
}

} else if (checkButtonSelect(&btnSend, &m)) {
    btnSend.pressed = true;
    drawButton(&btnSend);
}

else if (checkButtonSelect(&btnClose, &m)) {
    btnClose.pressed = true;
    drawButton(&btnClose);
} else {
    // 检查鼠标是否从按钮内移动到按钮之外
    if (btnClose.pressed == true) { // 鼠标从关闭按钮移出
        btnClose.pressed = false;
        drawButton(&btnClose);
    }

    if (btnSend.pressed == true) { // 鼠标从发送按钮移出
        btnSend.pressed = false;
        drawButton(&btnSend);
    }
}

break;
case WM_LBUTTONDOWN:
    if (checkButtonSelect(&btnTitle, &m)) {
        btnTitle.pressed = true; // 单击按下标题栏
    }

    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    }

    else if (checkButtonSelect(&btnSend, &m)) {
        btnSend.pressed = true;
        drawButton(&btnSend);
    }

    break;
case WM_LBUTTONUP:
    if (checkButtonSelect(&btnClose, &m)) {
        //btnClose.pressed = false;
    }
}
```



```
        //drawButton(&btnClose);
        closegraph();
        exit(0);
    }
    else if (checkButtonSelect(&btnSend, &m)) {

    }
    else if (checkButtonSelect(&btnTitle, &m)) {
        // 松开标题栏按钮（左键抬起）
        btnTitle.pressed = false;
        titleDrag = false;
    }
    break;
}
}
```

11. 发送消息

```
msg_t msgAll[5];
int msgCount = 0;
int currentMsgIndex = -1; //当前最近的一条信息的索引

void mainUI() {
    bool titleDrag = false; //表示“标题栏”是否被单击拖动
    int titleLastX; //窗口的上一次位置（X 坐标位置）
    int titleLastY; //窗口的上一次位置（X 坐标位置）

    while (1) {
        MOUSEMSG m = GetMouseMsg();
        FlushMouseMsgBuffer(); //不能少，后缀快速拖动顶部的标题按钮，将导致鼠标消息太多，
        出现混乱！
        switch (m.uMsg) {
            case WM_MOUSEMOVE:
                if (checkButtonSelect(&btnTitle, &m)) {
                    if (btnTitle.pressed == true) {
                        if (titleDrag == false) { // 此时标题栏已经被点击按下，正准备拖动
                            titleLastX = m.x; // 记录初始坐标
                            titleLastY = m.y;
                            titleDrag = true;
                        }
                    }
                    else { // 此时标题栏已经被点击按下，正在拖动
```




```
// 计算拖动偏移量
int offX = m.x - titleLastX;
int offY = m.y - titleLastY;
moveWindow(hwnd, offX, offY); // 根据拖动偏移量，移动窗口
}
}
}
else if (checkButtonSelect(&btnSend, &m)) {
    btnSend.pressed = true;
    drawButton(&btnSend);
}
else if (checkButtonSelect(&btnClose, &m)) {
    btnClose.pressed = true;
    drawButton(&btnClose);
}
else {
    // 检查鼠标是否从按钮内移动到按钮之外
    if (btnClose.pressed == true) { // 鼠标从关闭按钮移出
        btnClose.pressed = false;
        drawButton(&btnClose);
    }
    if (btnSend.pressed == true) { // 鼠标从发送按钮移出
        btnSend.pressed = false;
        drawButton(&btnSend);
    }
}
break;
case WM_LBUTTONDOWN:
    if (checkButtonSelect(&btnTitle, &m)) {
        btnTitle.pressed = true; // 单击按下标题栏
    }
    else if (checkButtonSelect(&btnClose, &m)) {
        btnClose.pressed = true;
        drawButton(&btnClose);
    }
    else if (checkButtonSelect(&btnSend, &m)) {
        btnSend.pressed = true;
        drawButton(&btnSend);
    }
    break;
case WM_LBUTTONUP:
    if (checkButtonSelect(&btnClose, &m)) {
        closegraph();
    }
```



```
        exit(0);
    }
    else if (checkButtonSelect(&btnSend, &m)) {
        btnSend.pressed = false;
        drawButton(&btnSend);

        //int ret = send(serverSocket, msgEdit, msgLen, 0);
        int ret = send(clientSock, msgEdit, msgLen, 0);
        printf("已经发送%d 个字符\n", ret);

        currentMsgIndex = (currentMsgIndex + 1) % (sizeof(msgAll) /
sizeof(msgAll[0]));
        msgEdit[msgLen] = 0;
        strcpy(msgAll[currentMsgIndex].msg, msgEdit);
        memset(msgEdit, 0, sizeof(msgEdit));
        msgLen = 0;

        msgAll[currentMsgIndex].type = SEND;
        msgCount++;

        //drawMsg();
        drawMsg(msgAll, currentMsgIndex, &msgHeight, imageArrows, imageHeads);
    }
    else if (checkButtonSelect(&btnTitle, &m)) {
        // 松开标题栏按钮（左键抬起）
        btnTitle.pressed = false;
        titleDrag = false;
    }
    break;
}
}
```

12. 接收信息

```
DWORD WINAPI ThreadFuncRcv(LPVOID param) {
    while (1) {
        char buff[1024];
        int ret = recv(clientSock, buff, sizeof(buff), 0);
        if (ret <= 0) {
            printf("对方已下线\n");
        }
    }
}
```



```
        closesocket(clientSock);
        //TerminateThread();
        waitAccept();
    }
    else {
        buff[ret] = 0;
        printf("[收到] %s\n", buff);

        currentMsgIndex = (currentMsgIndex + 1) % (sizeof(msgAll) /
sizeof(msgAll[0]));

        strcpy(msgAll[currentMsgIndex].msg, buff);
        msgAll[currentMsgIndex].type = RECEIVE;
        msgCount++;
        //drawMsg();
        drawMsg(msgAll, currentMsgIndex, &msgHeight, imageArrows, imageHeads);
    }

}

return NULL;
}
```

3. 项目提升

完整版 QQ 项目演示

<https://www.bilibili.com/video/BV1ua4y1n73P>

多线程的并发和同步

编辑线程，接受消息线程，主线程，没有对共享资源做互斥，理论上存在技术风险。
解决方案，对多线程的共享资源实现互斥访问（比如：easyx 的颜色控制）。

万分之一的 BUG 概率（我还没有遇到）



服务端的高并发实现

百万级并发，即是电信级企业项目。

其它功能的实现

视频聊天

语音聊天

文件传输

远程控制

.....

学员案例

学员小翔

零基础起步，二本，学完课程，入职百度，C++后台开发工程师，年薪 28.8 万

Baidu.com

录用通知书（校园招聘）

尊敬的 先生/女士：

我们非常高兴地通知您，您已经通过了公司的笔试及面试，公司拟录用您为正式员工并拟与您签订正式劳动合同。

您入职后的职位情况
职 位：北京-C++/PHP研发工程师
汇报对象：经理

您入职后的主要工作职责，包括但不限于以下内容：

- 负责百度核心产品功能和架构开发
- 负责核心技术算法的研究、实现和优化
- 负责前瞻技术的跟踪调研和产品创新

您入职后的薪酬待遇

- 基本工资（按照每年12个月标准发放） 税前人民币 18000.00元/月
- 目标年终奖金：税前人民币 72000.00元

以上奖金并非员工最终获得的实际奖金，实际奖金将基于目标奖金，根据公司、部门和个人业绩进行浮动，并结合当期工作时间等多种因素确定。实际奖金的具体计算和发放办法按公司相关政策执行。同时，公司有权对所有的奖金计划做出调整，并享有最终解释权。

您入职后的福利待遇

社会福利：按照国家及地方政府规定缴纳的养老保险、失业保险、基本医疗保险、工伤保险、生育保险、住房公积金

公司福利：带薪休假、商业医疗保险、午餐补助、年度体检、健康咨询。

所支付的工资中已包含国家法定的独生子女奖励费等福利待遇；同时，公司有权根据实际情况调整公司福利，对此，公司保

学员 Marin

自学过 Java，后来改学 C/C++，函授专科学习

入职音视频服务器工程师，15K，已跳槽，现在月薪 18K



录用通知书	
尊敬的 [姓名] 先生/女士： 您好！我们诚挚邀请您加入广州 [公司名称] 电子科技有限公司。有关您的聘用条件如下：	
一、协议期限	试用期：3 个月，如试用期表现良好，可申请提前转正。 合同期：3 年，到期可续签。
二、报到日期	2020 年 5 月 18 日 (offer 报道日期与实际报道日期时间间隔不可超过 5 个工作日，如有特殊情况需提前 2 个工作日以邮件形式发送给人事说明原因与递交申请，得到许可，则可延期入职，否则，offer 可作废)
三、报到地点	广州市 番禺区 广州市 [公司名称]
四、报到所需资料	1. 身份证原件； 2. 工商银行储蓄卡账户信息 (用于发放工资，如没有也可在入职后次月 10 日前办理)； 3. 毕业证原件； 4. 上家公司所开具的离职证明 5. 体检证明 (公司认可距入职日期三个月以内的体检报告，体检报告费用 在转正后报销，如试用期内离职，则公司退还体检报告原件)
五、工作岗位及汇报对象	您的工作岗位：C/C++ 软件开发工程师 工作汇报对象：部门负责人
六、工作时间	1. 大小周休息，如有特殊部门有特定要求，则需配合部门安排。 2. 各部门根据项目工作需求执行，在公司制度之下以所在部门实际通知为准。
七、薪酬奖金	基本薪酬：转正后整体年薪 18 万 全年总基本工资占年薪 80%+全年总月考核奖金占年薪 15%+年终奖占年薪 5%，具体以实际工作表现绩效考核结果以及公司薪酬制度为准) 试用期薪资：转正后薪资的 80%； 转正后薪资：公司会根据与公司业绩挂钩的工资奖金分配制度核发每月工资 备注：16-20 号会转入到个人银行账户。

学员海哥

零基础起步，二本，学完课程，入职字节跳动，客户端开发工程师，年薪 37.5 万

发件人：字节跳动 ByteDance >

详情

字节跳动2022校园招聘录用意向书

2021-07-15 23:28:59



亲爱的 [姓名] 同学 (身份证号：36012120000801)：

恭喜你顺利通过字节跳动2022校园招聘的考核，我们非常高兴地告知，你即将成为字节跳动团队的最新成员啦！

更多案例

<https://www.yuque.com/qiniuxueyuan/jiuye/ny150b>