

15 级种子班通宵测试任务书

目录

15 级种子班通宵测试任务书.....	1
环境要求.....	1
通宵测试能力考察点：.....	1
作品提交方式：.....	2
任务内容.....	2
任务步骤.....	2
加分内容 1：.....	5
实现方法要求：.....	5
加分内容 2：.....	5
实现方法要求：.....	5
加分内容 3：.....	5
实现方法要求：.....	5

环境要求

1. 运行环境：Windows/Linux
2. 编程语言：C/C++

通宵测试能力考察点：

1. c/c++ 基本掌握程度
2. 基本编程解决应用问题的能力
3. 网络检索问题、迁移应用的能力
4. 网络编程相关知识
5. 多线程开发,拓展线程同步与互斥)

(以上可提前公布给考生)

作品提交方式：

对于任务的每一步骤，都需要单独保存一份工程文件，在答辩的时候分步进行展示。最终答辩之前，将所有工程文件打包，命名：姓名_学号.zip(tar.gz)，以邮件附件形式（邮件主题和文件名保持一致）发送到邮箱
seedclass15@163.com

任务内容

根据任务步骤，逐步实现命令行多线程控制和 tcp 通信模块，并整合到给定的贪吃蛇代码，实现双人即时贪吃蛇对战

任务步骤

1. 用 C 语言实现一段代码，能够重复读取键盘输入 “WSAD,UJHK” 等八个按键，并即时输出其 ASCII 码到控制台。

如依次键入 “WSAD, UJHK” 时控制台对应会输出
“8783656885727475”

2. 学习多线程开发，在我们运行的程序中，可能我们会遇到堵塞的情况，比如通过调用 while((getchar())!='\n')来获取命令行输入，这时候我们的程序是堵塞在等待用户输入的，但如果我们希望在等待的同时，程序还能处理其它事情，比如定时向控制台输出一些信息，这时候我们就需要用到多线程开发了，现要求通过上网学习多线程的操作 API，并：

实现异步按键控制程序：主线程即时捕获键盘输入，并输出到命令行，子线程每隔一秒输出 “SeedClass\n”，如下图所示：

```
8368SeedClass
SeedClass
SeedClass
SeedClass
87SeedClass
8365SeedClass
68SeedClass
8574SeedClass
72SeedClass
SeedClass
```

（SEEDCLASS 前面的数字，是另一线程监听按键并输出其 ASCII 码）

3. 通过 TCP socket 实现服务器、客户端的正常通信，TCP 是一个网络通信协议，对于各种编程语言，都提供有相应的操作 API，通过这个协议，我们可以在网络连通的情况下，实现多台电脑的相互连接通信，现要求通过上网学习，实现一个客户端和服务端，其中服务器监听本地 ip : 127.0.0.1，客户端连接这个 ip,通过一台电脑来模拟多台电脑通信：

1. 客户端读取键盘输入，向服务器请求，向服务器发送任意内容
2. 服务器解析客户端内容,在控制台进行打印

下面分别是服务器、客户端控制台输出示意图：

服务器：

```
Server Socket Port:15001
start accepting..
new connection come
hello
who are you
哈哈
```

前三句是服务器的日志输出，后三句是从客户端接收的信息

客户端：

```
hello
data send successfully
who are you
data send successfully
哈哈
data send successfully
```

HELLO, WHO ARE YOU, 哈哈都是用户输入，DATA SEND SUCCESSFULLY 是程序发送成功后的日志输出。

4. 实现客户端、服务器的相互通信
1. 在任务 3 的基础上，客户端向服务器发信息，然后等待服务器回复
 2. 服务器接收后解析，加上前缀“ server:” 再发送给客户端
 3. 客户端接收到后打印到控制台。

服务器客户端输出分别如下所示：

	haha
start accepting..	server:haha
new connection come	qwe
haha	server:qwe
qwe	asd
asd	server:asd

5. 整合第 2 步多线程操作到第 4 步中，实现客户端每次发送信息后，不必等待服务器回复，可以立即监听键盘并输出。实现效果要求：客户端服务器分别创建一条子线程：

1. 客户端主线程用于监听键盘输入，并发给服务器，子线程用于接收服务器发送的内容，并输出到控制台。
2. 服务器主线程用于接收客户端消息，加上前缀“server:”再发送给客户端，并打印到控制台，子线程每隔一秒向控制台输出“SeedClass”

服务端，客户端输出分别如下所示：

```
Server Socket Port:15001
start accepting..
SeedClass
SeedClass
new connection come
SeedClass
SeedClass
qweasd
SeedClass
sadhj
SeedClass
SeedClass
qwehjkasd
SeedClass
```

```
qweasd
server:qweasd
sadhj
server:sadhj
qwehjkasd
server:qwehjkasd
```

6. 熟悉贪吃蛇客户端：
1. 代码提供了两个全局变量 key1,key2，通过修改这两个变量，控制两条贪吃蛇的行为,如 key1 上下左右对应 WSAD,key2 上下左右对应 UJHK,通过其 **ASCII 码**控制对应关系，如用 key1=ASCII(W)=87 表示向上移动
 2. key1，key2 的修改可通过函数 void key_input() 控制。
 3. 对于贪吃蛇代码，可以不用过多理解，理论上只要修改文件最后的 key_input() 函数和主逻辑循环函数即可完成我们的所有要求。
7. 将 TCP 通信部分代码(客户端、服务器)整合进贪吃蛇程序中. 实现一个贪吃蛇服务器和 socket 客户端,客户端通过监控键盘输入，传输给服务端，远程操控服务端贪吃蛇移动，具体实现步骤要求：
1. 客户端监听键盘输入，将 wsad 按键指令发送给服务器
 2. 熟悉 queue.c 代码，通过阅读 queue.h 代码注释、运行 queue.c 注释的 main 函数，了解如何一个先进先出字符动作队列。
 3. 服务器接收按键指令，并存到一个队列尾部，同时另一个线程从队列头部读取按键指令，控制贪吃蛇 1 行为，实现客户端远程控制服务端的贪吃蛇。

加分内容 1:

解决需求：基于任务第 7 步的基础上，实现贪吃蛇双人即时对战（碰到自己、墙、对方都输，游戏逻辑已实现，最终实现效果）

实现方法要求：

1. 仿照贪吃蛇服务端实现方式，整合 socket 客户端和贪吃蛇实现一个贪吃蛇客户端
2. 客户端进程有两条贪吃蛇，通过监控按键（WSAD）控制贪吃蛇 1，通过接收服务器（UJHK）控制贪吃蛇 2
3. 服务器进程也有两条贪吃蛇，通过通过监控按键（UJHK）控制贪吃蛇 2，通过接收服务器（WSAD）控制贪吃蛇 1
4. 在单台电脑演示时，两个进程会同时对按键进行检测，所以要求客户端代码只能监控 WSAD,服务端代码只能监控 UJHK，两者都要求通过队列存取指令进行操控。

加分内容 2:

解决需求：对于游戏中的全局变量 queue,在代码中有两个地方进行修改，且在两个不同的线程，为确保数据的一致性，需要在操作变量时进行加锁，操作完再解锁，确保在任意时刻只有一个线程对该变量进行修改。

实现方法要求：

熟悉 WINDOWS/LINUX 多线程开发 API, 添加锁相关逻辑，完成需求。

加分内容 3:

解决需求：解决 TCP 即时通信粘包问题：客户端连续向服务器发送 100 个数字，服务器获取后解析到一个长度为 100 的数组，再累加后返回结果

1. 服务器收到的一次信息，可能包含客户端发送的多条信息，需要缓存接收服务端信息，再按传输格式进行拆分
2. 如客户端发送 1~100 个数字（每次发完上一个就发下一个，不等服务器返回），服务端可能一次收到 12345, 需要拆分开成 1, 2, 3, 4, 5

实现方法要求：

1. 基于分隔符解决（如用\n进行分隔）

2. 基于长度分隔，在每条消息前加上消息长度，每次接收消息后先获取消息长度，再根据消息长度切割信息。
3. 基于上面两种解决方案，二选一整合进贪吃蛇客户端中