

酒店分布式温控计费系统

基本要求：某快捷廉价酒店响应节能环保理念，推行自助计费式中央温控系统，使得入住的客户可以根据自身需求设定温度和风速的调节，同时房间内的控制面板显示所需支付的金额。客户退房时酒店须出具空调使用的账单及详单。空调运行期间，酒店的空调管理员能够监控各房间空调的使用状态；需要的情况下可以生成酒店及房间的空调使用统计报表。

基本需求分析：

- 1. 入住的客户可以根据自身需求设定温度和风速的调节，（客户端功能）
- 2. 房间内的控制面板显示所需支付的金额。（客户端UI，实时信息）
- 3. 客户退房时酒店须出具空调使用的账单及详单。（服务器端功能，统计信息）
- 4. 空调运行期间，酒店的空调管理员能够监控各房间空调的使用状态；需要的情况下可以生成酒店及房间的空调使用统计报表。（服务器端功能）

课程目标：在给定成本、进度的前提下，开发出满足用户需求且具有可修改性、有效性、可靠性、可理解性、可重用性、可适应性、可移植性、可追踪性、可互操作性的软件产品。基于软件工程的目标，我们分析题目的实际需求，结合组员自身能力以及特点，下面给出开发的系统开发的技术路线：

客户端和服务端通信：

是否能深刻地理解网络连接的过程将影响着我们这一项目服务器——客户端的通信问题。并且，对于现在这一课程题目而言，他的本地功能比较有限，更多的是端到端的通信，如何保证在服务器端能够实时地检测客户端（房间）的信息，以及并发的问题将是我们解决的重中之重。基于以上观点，我们本周深入了解了网络编程的相关概念，结合实际要求提出了适用于我们组的网络通信架构。

如今大多的网络编程都是基于socket套接字，利用TCP/IP协议进行连接的建立与释放，从而完成信息的传输。OSI模型的重要性不言而喻，OSI模型包括了物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

OSI模型	各层主要协议
应用层	HTTP、FTP等
表示层	
会话层	
传输层：定义传输数据的协议、端口号等	TCP/UDP，我们将基于TCP的有连接方式进行网络通信，以维持账单的准确性。
网络层：进行逻辑地址寻址、路由	IP协议，ICMP等
数据链路层：建立逻辑连接、进行硬件设备寻址、差错检验、成帧等	
物理层：建立、维护、断开物理连接	

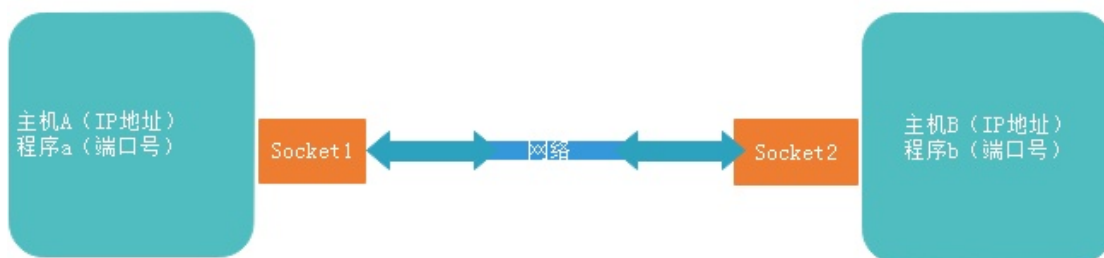
面向连接的套接字：实现这种连接类型的主要协议是TCP。因为这些套接字的网络版本使用因特网协议（IP）来搜索网络中的主机，所以整个系统通常结合这两种协议（TCP和IP）来进行

- TCP协议

是建立在IP协议上的，TCP协议负责在两台计算机之间建立可靠连接，保证数据包按顺序到达。TCP协议会通过握手建立连接，然后，对每个IP包进行编号，确保对方按顺序收到，如果包丢掉了，就自动重发。一个TCP报文除了包含要传输的数据外，还包含源IP地址和目的IP地址，源端口和目标端口。端口有什么用？在两台计算机通信时，只有IP地址是不够的，因为同一台计算机上跑着多个网络程序。一个TCP报文来了以后，到底是交给浏览器还是QQ，就需要端口来区分。每个网络程序都向操作系统申请唯一的端口号，这样，两个进程在两台计算机之间建立网络连接就需要各自的IP地址和端口号。

许多常用的更高级的协议都是建立在TCP协议基础上的，比如用于浏览器的HTTP协议、发送邮件的SMTP协议等。

- socket



简单地说，Socket是一种实现TCP协议的接口，他让我们显式地使用TCP协议，但是需要注意地是，TCP协议是一种传输层协议，并且这一协议是一种流协议（stream protocol），这就以为着他是以字节流的形式传递给接收者的，没有固定的“报文”或“报文边界”的概念，从这方面来说，读取TCP数据就像从串口数据中读取数据一样，无法预先知道一次指定的读调用中会返回多少字节，这对本问题的处理不太友好，因为对于本问题而言，服务器端和客户端之间的交互信息可以封装为一个使用状态数据包（因为账单问题比较少突发状况）

使用状态数据包
温度
风速
客户ID（房间号）
实时金额
开始时间

- HTTP协议

基于流的协议不太好用了，更高层的协议HTTP解决了这一问题（应用层）。使用HTTP可能会有如下优点：

- 文本型协议，方便使用者理解。
- 可以借用HTTP的工具，比如curl等进行测试
- 可以借用HTTP的部署设施；例如，使用proxy等完成公网、私网代理，使用nginx等进行负载均衡。
- 为客户端提供标准的HTTP API

当然，我们还得面对HTTP所带来的缺点：

- 是否相比Socket带来了其他的价值？有的时候，socket编程更为直接，引入一个HTTP带来了框架层面的负担。
- payload？ HTTP有更大的开销

但是主要问题在于，HTTP无法实现服务器主动向客户端推送信息，这一点可能会导致前台（服务器端）无法控制房间的空调。当然，现在仍在需求分析阶段，还未考虑具体的实现问题，先把这个问题保留。

运行方式：

在WWW中，“客户”与“服务器”是一个相对的概念，只存在于一个特定的连接期间，即在某个连接中的客户在另一个连接中可能作为服务器。基于HTTP协议的客户/服务器模式的信息交换过程，它分四个过程：建立连接、发送请求信息、发送响应信息、关闭连接

HTTP协议是基于请求/响应范式的。一个客户机与服务器建立连接后，发送一个请求给服务器，请求方式的格式为，统一资源标识符、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和可能的内容。服务器接到请求后，给予相应的响应信息，其格式为一个状态行包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。

其实简单说就是任何服务器除了包括HTML文件以外，还有一个HTTP驻留程序，用于响应用户请求。你的浏览器是HTTP客户，向服务器发送请求，当浏览器中输入了一个开始文件或点击了一个超级链接时，浏览器就向服务器发送了HTTP请求，此请求被送往由IP地址指定的URL。驻留程序接收到请求，在进行必要的操作后回送所要求的文件。

报文格式：

HTTP报文由从客户机到服务器的请求和从服务器到客户机的响应构成。

请求报文格式如下：

请求行 - 通用信息头 - 请求头 - 实体头 - 报文主体

请求行以方法字段开始，后面分别是 URL 字段和 HTTP 协议版本字段，并以 CRLF 结尾。SP 是分隔符。除了在最后的 CRLF 序列中 CF 和 LF 是必需的之外，其他都可以不要。有关通用信息头，请求头和实体头方面的具体内容可以参照相关文件。

应答报文格式如下：

状态行 - 通用信息头 - 响应头 - 实体头 - 报文主体

状态码元由3位数字组成，表示请求是否被理解或被满足。原因分析是对原文的状态码作简短的描述，状态码用来支持自动操作，而原因分析用来供用户使用。客户机无需用来检查或显示语法。有关通用信息头，响应头和实体头方面的具体内容可以参照相关文件。

计费系统

前面讨论了可能涉及到的通信协议问题，在这一部分，我们将讨论关于计费系统的设计。

一个可靠的计费系统，实际上就是给定一个计费函数，通过设定温度、风速并结合室内温度最后得出一个结果。

$$f(t, s) = cost \quad (1)$$

经过我们组充分考虑，并结合现在空调的工作原理，我们得出变频空调的功耗的简化模型：

$$W_t = \begin{cases} w1 & T1 < t \\ w2 & T1 = t \\ w3 & T1 > t \end{cases} \quad (2)$$

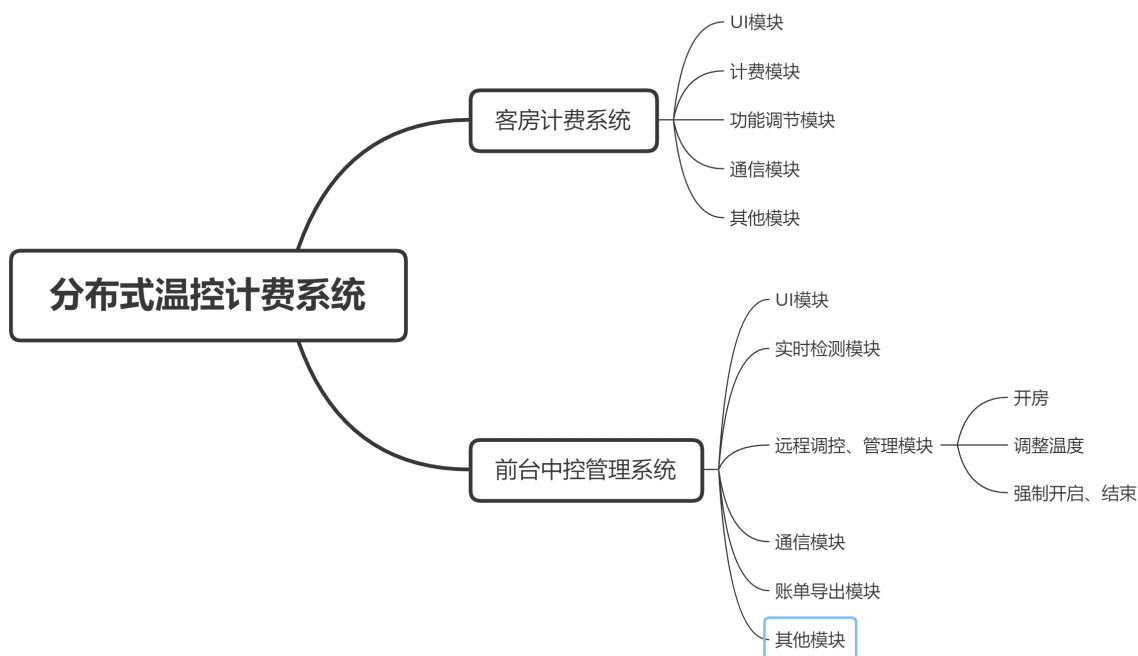
$$W_f = \begin{cases} w1 & f = f1 \\ w2 & f = f2 \\ w3 & f = f3 \end{cases} \quad (3)$$

$$\begin{aligned} W &= \alpha W_t + \beta W_f \\ cost &= \sigma W \end{aligned} \quad (4)$$

其中 $w1 > w2 > w3$,

当然了，这一计费函数随时都能进行改动。一般而言，如果室内温度高于设定温度，那么空调将最大功率运行（制冷功率），同时，如果风速调大，对应功率也会更大。当室内温度接近设定温度时，变频空调无极变频，将以一种恒定的较低功率运行以维持室内温度保持恒定。

系统结构



账单设计

账单设计内容将结合实际需求进行变动，现在设计如下：

房间号	室内温度（摄氏度）	设定温度（摄氏度）	风速	起始时间（单位：分钟）	终止时间（单位：分钟）	花费（元）
106	32	26	high	2020.3.12 12:30	2020.3.12 16:20	1.25