

C++ 课程设计 – 综合设计

1. 编写一个航空预约系统。该系统在命令行运行和用户进行交互，系统包括以下功能：

- a) 添加航班信息
- b) 删除航班信息
- c) 显示所有航班信息
- d) 按照航班号，显示单个航班信息
- e) 显示旅客信息
- f) 预约航班
- g) 取消航班

航班信息包括：航班编号，出发地，到达地，出发时间，到达时间，票价，飞机类型，座位数和目前已经预约的座位数。

乘客信息包括：姓名，护照号，国籍，电话。

选取相应的数据结构来存放航班，旅客等信息。系统能够将数据保存至文件，并从文件读取数据到系统里。推荐利用 C++ 标准库里自带的数据结构方法来实现，类的设计没有硬性规定，注意程序中要做鲁棒性检查（如日期是否合法，到达日期必须比出发时间晚等）。

认证模块：

1. 编写工作人员信息登陆模块，用来记录航空预约系统使用工作人员的信息并实现认证登陆的功能。这里每次登陆的时候用户会被提示输入用户名 (ID) 和密码 (pw)，密码可以为 8-10 个字符长。后台服务端会以 ID, salt, $H(pw + salt)$ 的形式储存用户的资料。当用户输入 ID 和密码后，系统会找到对应 ID 的 salt 和 $H(pw + salt)$ 值。这里的 salt 值是在用户首次注册的时候系统随机生成的一个 128 比特的字符串；H 是一个密码杂凑函数，这个函数接收一个任意长的输入，输出一个固定长的输出。这里 H 我们统一使用 SHA256 算法，输出的长度为 256 比特。系统会基于用户提供的 pw 和自己本地存储的 salt 值计算出 $H(pw + salt)$ ，看看这个值是否与本地存储的 H 值一致，如果一致则用户验证通过，否则拒绝用户登陆。SHA256 不用自己编写，可以调用库实现即可。
2. 改进上述的登陆模块，变为一次一密类型从而可以进一步增强安全性。假设用户的用户名和密码为 ID 和 pw，设 $i=1000$ ，用户注册的时候发送 ID 和 $H^{(i+1)}(pw)$ 给服务器。当用户登陆系统的时候用户发送 $H^{(i)}(pw)$ 给服务器，并更新客户端参数 $(pw, i) \rightarrow (pw, i-$

1)。服务端验证 $H(H^{(i)}(pw)) = H^{(i+1)}(pw)$, 如果相等则认证通过并更新服务端参数 $(ID, H^{(i+1)}(pw)) \rightarrow (ID, H^{(i)}(pw))$.