



深蓝学院
shenlanxueyuan.com

第八章作业 - 情报督察 作业分享



主讲人 水乐



纲要

➤ 第一部分：需求分析

➤ 第二部分：思路分享

需求分析

- 基本要求：利用字母码本对文件加密；
- 拓展1，拓展2：数字码本(0-255)对任意文件进行加密；

拓展要求 > 基本要求

- Part1 接收参数处理;
- Part2 码本文件处理;
- Part3 读取待加密/解密文件;
- Part4 加密;
- Part5 解密;

思路分享 - 处理码本文件

● 1. 生成随机码本文件(数字码本)

```
1. #include <algorithm>
2. //生成0-255的数字码本
3. for(int i = 0; i <= 255; i++)
4.     num_data.push_back(i);
5. //将数组随机重新排列
6. std::random_shuffle(num_data.begin(), num_data.end())
```

● 2. 读取处理码本内容

```
1. //以二进制读方式打开文件;
2. std::ifstream sfin(sfname, std::ios::in |
    std::ios::binary);
```

```
1. //读取文件中全部数据
2. while (sfin.read(&readdata, sizeof(readdata)))
3.     readdatastr.push_back(readdata);
```

```
src > ≡ file_readtest.txt
1      | a c d r
2      | b g ex

src > ≡ file_readtest.txt
1      | 111 123 2
2      | 32 23 4
```

采用循环方法:

1. 去掉空格, 换行符;
2. 检查非法字符;
3. 判断码本类型;
4. 保存码本数据到encryptdata中;

```
1. std::vector<uint8_t> encryptdata;
2. encryptdata="qwertyuiopasdfghjklzxcvbnm";
```

思路分享 - 码本文件接收

● 1. 读取加密文件(数字码本+任意类型)

用二进制方式读取，每次读取8bit数据，采用vector<uint8_t>容器来存储读取的数据

```
1. //待加密文件数据
2. std::vector<uint8_t> ifdata;
3. uint8_t readdata;
4. //读取文件中所有数据
5. while (ifin.read(&readdata, sizeof(readdata)))
6.     ifdata.push_back(readdata);
```

● 2. 判断代价密文件内容与码本类型是否匹配

思路分享 - 加密

●1. 字母码本

●2. 数字码本

```
// 待加密文件内容
1. std::vector<uint8_t> ifdata = "hello";
// 码本内容
2. std::vector<uint8_t> sfdata =
    "qwertyuiopasdfghjklzxcvbnm";
//加密后内容
3. std::vector<uint8_t> encrypteddata;
    encrypteddata[0]=sfdata['h' - 'a'];

//encrypteddata[0] = sfdata[104-97] = sfdata[7] 'i';
```

```
1. std::vector<uint8_t> ifdata; //待加密文件内容
2. std::vector<uint8_t> sfdata; //码本内容
3. std::vector<uint8_t> encrypteddata; //加密后内容
4. int sftype; //码本类型
5. //遍历待加密文件
6. for(auto sptr = ifdata.begin(), eptr = ifdata.end(); sptr !=
    eptr; sptr++){
7.     if (*sptr != ' '){//空格符号不进行加密
8.         if (sftype == 0){//字母码本
9.             if(*sptr>= 'a' && *sptr<='z')
10.                 encrypteddata.push_back(sfdata[*sptr - 'a']);
11.             else
12.                 encrypteddata.push_back(sfdata[*sptr - 'A']);
13.             }
14.             else{//数字码本
15.                 encrypteddata.push_back(sfdata[*sptr]);
16.             }
17.         else{
18.             encrypteddata.push_back(*sptr);
19.         }
20. }
```

思路分享 - 解密

解密就是加密的反过程：遍历待解密文件的每个元素；从码本文件中找到相同的元素；码本文件元素的下标经过转换即为解密内容

```
1. if(*sptr >= 'a' && *sptr <= 'z')
2.     for(int i = 0; i < sfdata.size(); ++i){//在码本文件中找相同元素
3.         if (*sptr == sfdata[i]){
4.             cha = i + 'a';
5.             dncrypteddata.push_back(cha);
6.             break;
7.         }
8.     }
9.     else
10.         for(int i = 0; i < sfdata.size(); ++i){
11.             if (*sptr == sfdata[i]){
12.                 dncrypteddata.push_back('A' + i);
13.                 break;
14.             }
15.         }
}
```

```
1. std::vector<uint8_t> ifdata;//待解密内容
2. std::vector<uint8_t> sfdata;//码本内容
3. int sftype;//码本类型
4.
5. for(int i = 0; i < sfdata.size(); ++i){
6.     if (*sptr == sfdata[i]){
7.         dncrypteddata.push_back(i);
8.         break;
9.     }
10. }
```


感谢各位聆听 !
Thanks for Listening

