ICS LABA REPORT

【设计思路】

①main.cpp：构建整体框架，通过对assemble类的实例化实现汇编器的功能。

②assemble.cpp

Ⅰ、函数作用分析

```
assemble(std::string input_filename，std::stringoutput_filename)
从输入文件中读取信息，将转化内容放于输出文件中


TranslateOprand(int current_address, std::string str, int opcode_length)
将需要跳转的距离转换成指定长度的offset

ConvertBin2Hex(std::string bin)
将二进制串转成16进制

NumberToAssemble(const std::string& number)
传入数值字符串，返回转换后的16位01补码值

NumberToAssemble(const int& number)
传入数值，返回16位01补码值

RecognizeNumberValue(std::string str)
传入数值字符串，返回十进制数值

Trim(std::string& s, const char* t = " \t\n\r\f\v")
引用字符串s，并将其左右两边的指定特殊字符段删去
```

Ⅱ、设计思路解析

```
一共扫描三次。
第一次扫描：将注释和有效代码分开

第二次扫描：
将标签和地址相关联，区分Pseudo和Operation指令，并确定每行代码对应的绝对存储位置

第三次扫描：依据前两次扫描信息，对照指令集转成01码
```

Ⅲ、重要变量解析

```
std::vector<std::string> file_content;
//存储有效代码

std::vector<std::string> origin_file;
//存储原始代码

std::vector<LineStatusType> file_tag;
//存储代码类型标志

std::vector<std::string> file_comment;
//存储代码注释

std::vector<int> file_address;
//存储代码对应地址
```

```
class label_map_tp;
//标签与地址映射集合

class assembler
//汇编器
```

【实验收获】

在阅读程序的过程中，本人最大的收获是代码框架的构建思路：通过清晰的层级关系将项目构建起来，既方便维护，又方便阅读。其次，关于类的使用方法也给了本人很多启发。使用类，可以清晰明确地理清各变量间的层级关系，同时更加直观反映出对象的属性，方便理解项目构建思路。更多地，本次实验还教会了我使用make一键编译，这大大简化了多文件编译的步骤。

总的来说，此次实验令我受益匪浅.