实验10
状态寄存器的相关应用

实验内容

- 问题1：对上一次实验的改进

使得执行完自己编写的add128函数后退回到主程序时，状态寄存器中的信息表示了128位加法的状态。

ZF CF SF OV PF

例如，如果128位加法为0，则 ZF = 1

问题1：实现128位整数加法–回顾

编写2个c++文件: main.cpp add128.h

```
#include <iostream>
#include<iomanip>
#include "add128.h"
using namespace std;
#define MYCOUT(x) cout << setbase(16) << setw(8) << setfill('0') << (x) << ' ';
void printint128(int128 x){
        MYCOUT(x.l3);
        MYCOUT(x.l2);
        MYCOUT(x.l1);
        MYCOUT(x.l0);

        return;
}
int main(){
        int128 a = {1, 0, 0, 0};
        int128 b = {2, 0, 0, 0};
        int128 c;

        add128(a, b, c);

        printint128(c);

        return 0;
}
```

```
/* add128.h */
struct int128
{
        int l0; //最低32位
        int l1;
        int l2;
        int l3; //最高32位
};

void add128(int128 a, int128 b, int128& c);
```

编写汇编文件，实现add128函数的功能

问题1：实现128位整数加法 –回顾

- 实验步骤

步骤1：分析main.cpp中的结构体传参的过程

步骤2：自己手动编写一个add128.s文件，实现两个128位的整数相加的功能。

步骤3：然后将自己编写的add128.s 生成目标文件，并和原来的main.o进行连接生成可执行文件。

问题1：实现128位整数加法 –改进

- 改进要求：

使得执行完自己编写的 add128 函数后退回到主程序时，状态寄存器中的信息表示了128位加法的状态。

ZF CF SF OV PF

例如，如果128位加法为0，则 ZF = 1

问题1：实现128位整数加法 –改进

讨论：

- 如果是无符号数，如何知道是加法是否有进位？
- 如果是有符号数，如何确定最终结果的符号位和是否溢出？
- 如何确定结果是否等于0？
- 如何在完成计算后，让状态寄存器的ZF，CF，OF SF, PF表示对应的结果？

问题2

- 编写一个子程序 letter，将以0（不是字符‘0’）结尾的字符串中小写字母转换为大写字母。

名称：letterc
功能：将以 0 结尾的字符串中的小写字母转变成大写字母
参数：ds:si 指向字符串首地址

问题2

```
assume  cs:codesg

datasg segment
     db "Beginner's All-purpose Symbolic Instruction Code.",0
datasg ends
codesg segment

  begin: mov ax,datasg
         mov ds,ax
         mov si,0
         call letterc

         mov ax,4c00h
         int 21h

letterc:     :
             :

codesg ends
end begin
```