

实验三课后作业说明

作业一 实现以下函数

utils.asm 文件：

- **void clr_screen()**
 - 功能：清除屏幕
- **void key_detect(int echo, int block)**
 - 功能：按键检测
 - 参数说明：
 - echo —— 指定检测到某一按键是否在屏幕上显示检测到的字符；
 - block —— 指定调用该函数时是否阻塞，即是否等待任意按键再继续执行；
 - 返回值：
 - 检测得到的按键值，未检测到返回0；
 - 应用场景：
 - 循环调用draw()函数在屏幕上持续绘制图形，要求检测到按键'q'退出，未检测到按键'q'则持续绘制。
 - 若调用的key_detect为阻塞型函数，则调用该函数会进入等待任意按键的状态，无法在循环中持续调用draw()进行绘制。

```
while(1){
    draw();
    if( 'q' == key_detect(0, 0) ) //不阻塞，不回显
        exit(0);
}
```

- **void read_floppy(int SegStart, int SegNum)**
 - 功能：读软盘功能（读入内存的位置可以自定义，也可以作为参数）
 - 参数说明：
 - SegStart——从软盘的SegStart扇区开始将软盘读入内存；
 - SegNum——读取软盘的扇区数

main.c 文件：

- **void print_str_uppercase(char *str)**
 - 功能：
 - 若str为空指针（str == 0），则把“success”转换为大写，“success”已经在entry.asm文件中预定义好，然后调用print_char函数输出SUCCESS；
 - 若str不为空指针（str != 0），则将地址str所指向的字符串转为大写，调用print_char函数输出。

- 要求在entry.asm中调用print_str_uppercase这个函数，通过压栈向print_str_uppercase传递参数str。参数str可以为空（str == 0），输出结果为SUCCESS。
- void choose(int a, int b, int c, int d)
 - 功能：
 - 在entry.asm中调用该函数，打印输出a, b, c, d中的最大值和最小值

作业二 结合前面的内容完成下面任务（附加题，选做）

- 在软盘上建立一个表，记录用户程序的名称/软盘中的位置/字节数/内存中的加载地址；
- 设计一组控制台命令，来控制程序的执行和查询程序信息；
- 显示用户程序在软盘中位置、大小以及在内存中的加载地址等相关信息