<https://blog.csdn.net/u012349696/article/details/50757110>

一、fflush函数：更新缓存区

头文件：#include <stdio.h>

函数定义：int fflush(FILE \*stream);

函数说明：fflush()会强迫将缓冲区内的数据写回参数stream 指定的文件中. 如果参数stream 为NULL,fflush()会将所有打开的文件数据更新.

二、fflush(stdin):刷新缓冲区把缓冲里面的东西丢掉

fflush(stdout):刷新缓冲区把缓冲里面的东西输出到设备上去

三、简单来说 例如熟悉的scanf 、和printf 都是先把数据放入缓存里面，等缓存’满了‘再把他们丢出来

like this：

for(int i=0;i<10;i++)

{

printf("%d",i);

//fflush(stdiout);

}

结果：当没有fflush的时候，结果是0123456789 有fflush的时候 虽然结果也0123456789，不同的是

他们是一个一个蹦出来的

四、对于stdout fflush的作用看的也许不是很明显

看下stdin

int a,c;  
scanf("%d",&a);  
getchar();  
输入:  
12（回车）  
那么 a=12 ,c= '\n'   
而：  
int a,c;  
scanf("%d",&a);  
fflush(stdin);  
getchar();  
输入:  
12（回车）  
那么a=12, c暂时未得到输入值，还需要再输入c，因为getchar也是缓冲输入，'\n'本还在缓冲区，但是被清空了。  
另外fflush不能作用于重定向输入流。

五、关于缓冲区残余

scanf()函数接收输入数据时，遇以下情况结束一个数据的输入：（不是结束该scanf函数，scanf函数仅在每一个数据域均有数据，并按回车后结    束）。   
  ①   遇空格、“回车”、“跳格”键。   
  ②   遇宽度结束。   
  ③   遇非法输入。   
键盘缓冲区就可能有残余信息问题。

scanf()函数应该只是扫描stdin流，这个残存信息是在stdin中   
解决就要在scanf()函数之后加个fflush(stdin)

六、写文件的时候 不会马上进行物理磁盘的文件读写，而是先写入缓存，当缓存中内容达到一定程度后再写

  #include <stdio.h>  
  #include <stdlib.h>  
  int main(void){  
     FILE \*fp;  
     if((fp=fopen("test", "rb"))==NULL) {  
        printf("Cannot open file.\n");  
        exit(1);  
     }  
      char ch = 'C';  
      int i;  
      for(i=0; i<5; i++) {  
        fwrite(ch, sizeof(ch), 1, fp);  
        fflush(fp);  
      }  
      fclose(fp);  
      return 0;  
  }  
注意：如果在写完文件后调用函数fclose关闭该文件，同样可以达到将缓冲区的内容写到文件中的目的，但是那样系统开销较大。