<http://blog.chinaunix.net/uid-22566367-id-381994.html>

在一个问题中遇到了一个小问题，自己试了一下，小结一下  
关于fgets的用法，在man手册中是这样解释的：  
    char \*fgets(char \*s, int size, FILE \*stream);  
       fgets()  reads  in  at most one less than size characters from stream and stores them  
       into the buffer pointed to by s.  Reading stops after an EOF or a newline.  If a new-  
       line  is read, it is stored into the buffer.  A '\0' is stored after the last charac-  
       ter in the buffer.  
这个的大致意思是这样的：当读到一个新行或者是一个EOF标志时，fgets()就会结束，并且这个函数会自动给末尾加'\0'。  
我有这样一个问题:这个fgets()到底是读了多少个字节？  
man里面所说的'\0'是如何加上的？  
这个size包括'\0'这个字符吗？  
我写了一个测试程序：

|  |
| --- |
| #include <stdio.h> #include <string.h> #define N 10 int main() {     char a[N] = {0};     int i=0;     while(fgets(a, N, stdin)) {         fprintf(stdout, "%d %s@%d\n", i, a, strlen(a));         i++;     }     return 0; } |

输入与输出结果：  
12345                          //输入  
0 12345  
@6  
123456789                      //输入  
1 123456789@9  
2   
@1  
说明：在上面的结果中，“//输入”是我自己加上的，是为了看起来方便， 并不是实际的运行与输入结果，它表示的是这个数据是要输入的。剩余行全都是输出的行。  
当我输入“123456”，并敲了一个回车之后，从结果中可 以看到，其读了"123456"和回车。所以长度为6，数组a的第7个单元是'\0'，第6个单元是10（就是回车），前5个单元是"12345"。  
到 这里我有一个疑问：这个'\0'是自动加上的，当输入的长度超过了指定的长度时，存储又是怎样的？  
于是我输入了"123456789"这个测试数 据，很明显，按照我的猜想，数组a的前9个单元存放“123456789”，第10个单元存放'\n'，如果再赋'\0'的话，数组就会越界，所以' \0'是不赋的。  
可是测试结果却和我的想法大不一样，分析结果之后，实际是这么存放的：数组a的肖9个单元存放“123456789”，第10个 单元存放'\0'，放弃了'\n'，而不是放弃'\0'。  
  
结论:  
    char \*fgets(char \*s, int size, FILE \*stream);  
fgets()读一次只读size个字节，而且其中把'\0'也包括在内。  
由此可见，认真阅读 外文文档是多么重要。