

C++ 文件结束符

问题描述：windows下

```
int main()

{

    string str;

    while(cin>>str);

    cout<<"\n"<<"end"<<endl;

}
```

控制台输入多个字符串后，输入文件结束符ctrl-Z，并未使while循环结束，该如何跳出while循环？

解决方法：

方法1：回车 - ctrl-z - 回车

方法2：改代码，设置自己的文件结束符

```
String a;
while((cin>> a)&&a!='X')
X可以为自己设置的文件结束符。
```

原因解析：

当从键盘输入一串字符并按回车后，这些字符会首先被送到输入缓冲区中存储。每当按下回车键后，cin.get()就会检测输入缓冲区中是否有可读了的数据。cin.get()还会对键盘上是否有流结束标识的Ctrl-z或者ctrl-d键按下做出检查，其检查的方式有两种，阻塞式以及非阻塞式。

windows系统一般采用阻塞式检查ctrl-z, Unix/Linux系统下一般采用非阻塞式的检查ctrl-d。

阻塞式检查方式是指只有在回车键按下之后才对此前是否有ctrl-z组合键按下进行检查。另外，如果输入缓冲区中有可读的数据则不会检测ctrl-z，因为有可读数据，说明还未到流的末尾。需要注意的时，ctrl-z产生的不是一个普通的ascii字符，因此ctrl-z不会跟其他从键盘输入的字符一样能够存放在输入缓冲区中。因此，若使ctrl-z产生作用，输入ctrl-z之前不应输入任何字符，即输入回车键把之前输入的字符全部送入缓冲区中后，紧接着应输入ctrl-z，然后输入回车键，告知系统检测流结束符ctrl-z。

非阻塞式是指按下ctrl-d之后立即响应的方式。如果在ctrl-d输入之前已经从键盘输入了字符，则ctrl-d相当于回车，即把这些字符送到了输入缓冲区供读取使用，此时ctrl+d不起流结束符的作用。若ctrl-d输入值钱没有任何键盘输入，则ctrl-d就是流结束的信号。

windows 下ctrl-z使用举例说明如下：

从键盘上输入abcd^z 再输入回车键后，系统处理如下：因为回车的作用，前面的abcd等字符被送到输入缓冲区中，而^z不会产生自负，因此不会存储到缓冲区中。这是,cin.get()检测到输入缓冲区中已经有数据存在，就不会检测有无^z存在，cin.get()从缓冲中读取相应的数据，如果读取完了，则输入缓冲区重新边为空，cin.get()等待新的输入，可见尽管有ctrl-z输入，但是之前有其他字符，系统并未监测到流结束符。

因此输入流结束符的条件就是：^z之前不能有任何字符输入（回车键除外），然后再用回车键通知系统做流结束符检测。