Example

c++向文件写日志，当文件大小超过10k时，新建一个继续写

<io.h>部分函数说明（需要使用时可以查询）

IMG_256 首先需要了解句柄

<https://blog.csdn.net/perfectguyipeng/article/details/71189822>

<io.h>中常用函数

主要包括文件读写函数

文件打开方式函数

移动文件指针函数

<https://blog.csdn.net/babygjx/article/details/5740038>

IMG_257 缓冲区介绍：

<https://blog.csdn.net/caoshangpa/article/details/78920823>

代码：

#include <iostream>

#include <fstream>

#include <cstring>

#include <io.h>

Using namespace std;

Long int getLen(ifstream &fs);

Void main(){

String fpath = "tmp1.log";

String fpath1 = "tmp2.log";

Ofstream os("tmp1.log",ios:app);

Ifstream is("tmp1.log");

Char \* logContent = "this's a log file";

Long int restBytes = 10 \* 1024 - getLen(is);

While(restBytes > (int)strlen(logContent)){

Os << logContent << endl;

restBytes -= 1 + strlen(logContent);

}

Os.flush();//使缓冲区紧接着前文输出到文件

Os.close();

Is.close();

Os.open("tmp2.log",ios:app);

Is.open("tmp2.log");

While(restBytes > (int)strlen(logContent)){

Os << logContent << endl;

restBytes -= 1 + strlen(logContent);//strlen对应的是单位长度，如果是中文就是一个字strlen = 2

}

Os.flush();

Os.close();

Is.close();

}

Long int getLen(ifstream &fs){

Fs.seekg(0,ios::end);

Long int len = fs.tellg();

Return len;

}

IMG_258 seekg()/seekp()与tellg()/tellp()的用法详解

对输入流操作：seekg（）与tellg（）

对输出流操作：seekp（）与tellp（）

下面以输入流函数为例介绍用法：

seekg（）是对输入文件定位，它有两个参数：第一个参数是偏移量，第二个参数是基地址。

对于第一个参数，可以是正负数值，正的表示向后偏移，负的表示向前偏移。而第二个参数可以是：

ios：：beg：表示输入流的开始位置

ios：：cur：表示输入流的当前位置

ios：：end：表示输入流的结束位置

tellg（）函数不需要带参数，它返回当前定位指针的位置，也代表着输入流的大小。

假设文件test.txt为以下内容：

hello,my world

name:hehonghua

date:20090902

程序为：

#include <iostream>

#include <fstream>

#include <assert.h>

using namespace std;

int main()

{

    ifstream in("test.txt");

    assert(in);

    in.seekg(0,ios::end);       //基地址为文件结束处，偏移地址为0，于是指针定位在文件结束处

    streampos sp=in.tellg(); //sp为定位指针，因为它在文件结束处，所以也就是文件的大小

    cout<<"filesize:"<<endl<<sp<<endl;

    in.seekg(-sp/3,ios::end); //基地址为文件末，偏移地址为负，于是向前移动sp/3个字节

    streampos sp2=in.tellg();

    cout<<"from file topoint:"<<endl<<sp2<<endl;

   in.seekg(0,ios::beg);        //基地址为文件头，偏移量为0，于是定位在文件头

   cout<<in.rdbuf();            //从头读出文件内容

    in.seekg(sp2);

    cout<<in.rdbuf()<<endl; //从sp2开始读出文件内容

    return 0;

}

则结果输出：

file size:

45

from file to point:

30

hello,my world

name:hehonghua

date:20090902

date:20090902

四、C++文件流检测EOF

成员函数eof()用来检测是否到达文件尾，如果到达文件尾返回非0值，否则返回0。原型是inteof();

例：if(in.eof())ShowMessage("已经到达文件尾！");

五、C++文件流文件定位

和C的文件操作方式不同的是，C++ I/O系统管理两个与一个文件相联系的指针。一个是读指针，它说明输入操作在文件中的位置；另一个是写指针，它下次写操作的位置。每次执行输入或输出时，相应的指针自动变化。所以，C++的文件定位分为读位置和写位置的定位，对应的成员函数是 seekg()和 seekp()，seekg()是设置读位置，seekp是设置写位置。它们最通用的形式如下：

1.  istream &seekg(streamoff offset,seek\_dir origin);

2.  ostream &seekp(streamoff offset,seek\_dir origin);

streamoff定义于 iostream.h 中，定义有偏移量 offset 所能取得的最大值，seek\_dir 表示移动的基准位置，是一个有以下值的枚举：

ios::beg： 文件开头

ios::cur： 文件当前位置

ios::end： 文件结尾

这两个函数一般用于二进制文件，因为文本文件会因为系统对字符的解释而可能与预想的值不同。

例：

1.  file1.seekg(1234,ios::cur);//把文件的读指针从当前位置向后移1234个字节

2.  file2.seekp(1234,ios::beg);//把文件的写指针从文件开头向后移1234个字节

如果vc编程的话最好使用CFile类等更加方便于文件操作.