可以使用C语言中scanf、gets、fgets，C++中cin、getline函数读取字符串，当字符串字符数量非常大时，这些函数的效率究竟是如何的呢？本博客主要解决这一问题。  
****准备测试数据****  
我们先在桌面新建一个名为test的文件夹，作为本次实验的文件夹。我们希望向该文件夹下的input.txt文件（这个文件程序会自动生成）中写入5000个字符串，每个字符串为10000个0字符，字符串间用换行符分隔开，共计5000万个字符，以此作为输入数据， 程序如下：

#include<bits/stdc++.h>

using namespace std;

int main(){

ofstream out("C:/Users/Administrator/Desktop/test/input.txt");//文件路径可按你自己的想法替换

for(int i=0;i<5000;++i){

for(int j=0;j<10000;++j){

out<<"0";

}

out<<endl;

}

out.close();

puts("successful");//控制台输出successful时，表示文件数据写入完毕

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14

****读取数据****  
共对C语言中scanf、gets、fgets，C++中cin、getline这5个函数进行比较，我们可以让这5个函数分别读取1000万个0字符。程序如下：  
#include<bits/stdc++.h>  
using namespace std;  
int main(){  
char input[100005];  
string str;

clock\_t s= clock();//读取开始时间

for(int i=0;i<1000;++i)

scanf("%s",input);

clock\_t e= clock();//读取完毕时间

printf("scanf:%.0fms\n",(double)(e-s)/CLOCKS\_PER\_SEC\*1000);//输出整个读取过程花费时间，单位:毫秒

s= clock();//读取开始时间

for(int i=0;i<1000;++i)

gets(input);

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

e= clock();//读取完毕时间  
printf(“gets:%.0fms\n”,(double)(e-s)/CLOCKS\_PER\_SEC\*1000);//输出整个读取过程花费时间，单位:毫秒

s= clock();//读取开始时间

for(int i=0;i<1000;++i)

fgets(input,10005,stdin);

e= clock();//读取完毕时间

printf("fgets:%.0fms\n",(double)(e-s)/CLOCKS\_PER\_SEC\*1000);//输出整个读取过程花费时间

，单位:毫秒\*/

s= clock();//读取开始时间

for(int i=0;i<1000;++i)

cin>>str;

e= clock();//读取完毕时间

printf("cin:%.0fms\n",(double)(e-s)/CLOCKS\_PER\_SEC\*1000);/\*输出整个读取过程花费时间

，单位:毫秒\*/

s= clock();//读取开始时间

for(int i=0;i<1000;++i)

getline(cin,str);

e= clock();//读取完毕时间

printf("getline:%.0fms\n",(double)(e-s)/CLOCKS\_PER\_SEC\*1000);/\*输出整个读取过程花费时间

，单位:毫秒\*/

return 0;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22

编译运行后会生成一个exe可执行文件，找到这个exe文件并复制粘贴到test文件夹下。并重命名为test.exe  
****进行比较****  
在test文件夹下新建一个compare.txt文件，并向其中写入以下程序：

test < input.txt > output.txt

pause

* 1
* 2

保存并关闭文件后，将compare.txt的后缀名txt更改为bat。  
经过以上操作后，test文件夹下应该有3个文件：

input.txt：测试数据  
test.exe：测试程序  
compare.bat：执行脚本  
此时双击compare.bat脚本，会跳出一个dos窗口，等待一段时间，当dos窗口输出请按任意键继续. . .的字样时，关闭dos窗口，会发现test文件夹下新生成了一个output.txt文件，打开该文件，就可以看到各个函数读取数据所用时间。  
我执行时output.txt文件内容如下：

scanf:960ms

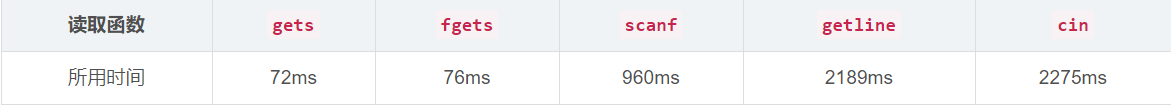
gets:72ms

fgets:76ms

cin:2275ms

getline:2189ms

* 1
* 2
* 3
* 4
* 5

****结论****  
按照所用时间从小到大进行排序，得到的结果如下：  
  
我们可以发现gets和fgets函数读取字符串的速度是最快的，读取时间约为scanf函数的1/13，约为getline和cin的1/30，由于gets函数由于不安全已被c11标准废弃，在以后读取字符串时，如果字符串很长，应尽可能使用fgets函数。  
另外，scanf函数的读取效率约为getline和cin的1/2，getline比cin稍快。

那么为什么会出现这样的情况呢？  
以fgets函数和scanf函数为例，由于fgets只负责读取字符串，而scanf函数还可以读取其他如int、double类型的数据，所以scanf函数就多了确定读取的数据是何种类型的步骤，因而花费了更多的时间。  
而getline和cin函数只能向string类型中读入数据，由于string类型是在堆上分配的，在读取数据的时候需要不断从堆上申请动态分配内存，不如char数组直接提前分配好了所有内存来得更为简便。此外在向string类型读入数据的时候，还需要调用string类型的构造函数，初始化包括begin()、end()、size()等类内数据成员和成员函数，这也花费了很多时间。所以string类型比char数组提供了许多使用更为简便的函数和成员，但也要为此承担时间上的代价。