

C++中string的拼接_c++ string字符串拼接_JasonLiu1919的博客-CSDN博客

成就一亿技术人!

字符拼接可以采用的方法:

1、多个字符串拼接时用+操作符

1)

代码:

如果不加红色部分的代码,则需要采用_sntprintf代替sntprintf.

```
#include <iostream>
#include <string>

<span style="background-color: rgb(255, 255, 255);"><span style="color:#FF0000;">#if _MSC_VER
#define snprintf _snprintf
#endif</span></span>
using namespace std;

string intToString(int v)
{
    char buf[32] = {0};
    snprintf(buf, sizeof(buf), "%u", v);

    string str = buf;
    return str;
}

int main()
{
    string data;
    int myid=7;

    string data1=intToString(myid) ;
    string data2;
    data = "{\\"status\\":200, \\"id\\":\\"\" +intToString(myid) + "\\"}";

    cout<<data.c_str()<<endl;

    return 0;
}
```



2) 引申使用str += "a", str =str+ "a" 效率差距:

str =str+ "a"加的运算产生的是一个新的对象,再把结果返回,而str += "a" 涉及到的应该是对象的引用,操作之后直接返回引用,避免了产生新的对象。因此,两者的性能有一定的差距。

```
int main()
{
    static int num = 1000000;
    time_t timeBegin, timeEnd;

    timeBegin = time(NULL);

    string str = "";
    for(int i =0; i<num; i++)
    {

        str =str + "a";

    }

    timeEnd = time(NULL);
    cout<<"str=str +a所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;
}
```

```

        timeBegin = time(NULL);
        string str1 = "";
for(int i =0; i<num; i++)
    {
        str1 += "a";
    }
        timeEnd = time(NULL);
        cout<<"str+=a所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;
return 0;
}

```

所耗费的时间差距如下图所示：差得真不是一丢丢。。。 （此处用的是debug版本）



2、使用append。

```

string s1 = "Hello ";
string s2 = "World! ";
string s3 = " China";
string s4;
s4.append(s1);
cout<<s4.c_str()<<endl;
s4.append(s2);
cout<<s4.c_str()<<endl;
s4.append(s3);
cout<<s4.c_str()<<endl;

```

将其与str+=a进行对比：

```

#include <iostream>
#include <string>
#include <time.h>

using namespace std;

int main()
{
    static int num = 100000000;
    time_t timeBegin, timeEnd;

        timeBegin = time(NULL);
        string str1 = "";
for(int i =0; i<num; i++)
    {
        str1 += "a";
    }
        timeEnd = time(NULL);
        cout<<"str+=a 所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

        timeBegin = time(NULL);
        string str2 = "";
for(int i =0; i<num; i++)
    {
        str2.append("a");
    }
        timeEnd = time(NULL);
        cout<<"str.append(a)所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;
}

```

```
return 0;
```

```
}
```

总体运行效率差不多：

```
str+=a 所耗费的时间: 38 ms
str.append(a)所耗费的时间: 37 ms
http://blog.csdn.net/
```

3. stringstream

结合这两种方法与上述方法进行对比：

```
#include <iostream>
#include <map>
#include <string>
#include <time.h>
#include <sstream>
using namespace std;

int main()
{
    static int num = 100000000;
    time_t timeBegin, timeEnd;

    timeBegin = time(NULL);
    string str1 = "";
    for(int i =0; i<num; i++)
    {
        str1 += "a";
    }
    timeEnd = time(NULL);
    cout<<"str+=a 所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

    timeBegin = time(NULL);
    string str2 = "";
    for(int i =0; i<num; i++)
    {
        str2.append("a");
    }
    timeEnd = time(NULL);
    cout<<"str.append(a)所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

    timeBegin = time(NULL);
    string str3 = "";
    stringstream ss;
    for(int i =0; i<num; i++)
    {
        ss<<"a";
    }
    str3=ss.str();
    timeEnd = time(NULL);
    cout<<"stringstream 方法所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

    return 0;
}
```

运行结果如下：可知stringstream方法是最快的！（这里的循环次数和上面是一样，对比运行时间也是可以看出）

```
str+=a 所耗费的时间: 38 ms
str.append(a)所耗费的时间: 37 ms
stringstream 方法所耗费的时间: 33 ms
```

<http://blog.csdn.net/>

4、sprintf进行字符的拼接

代码:

```
#include <iostream>
#include <map>
#include <string>
#include <time.h>
#include <sstream>
using namespace std;

static int num = 10000000;

int main()
{
    time_t timeBegin, timeEnd;

    timeBegin = time(NULL);
    string str1 = "";
    for(int i =0; i<num; i++)
    {
        str1 += "a";
    }
    timeEnd = time(NULL);
    cout<<"str+=a 所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

    timeBegin = time(NULL);
    string str2 = "";
    for(int i =0; i<num; i++)
    {
        str2.append("a");
    }
    timeEnd = time(NULL);
    cout<<"str.append(a)所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

    timeBegin = time(NULL);
    string str3 = "";
    stringstream ss;
    for(int i =0; i<num; i++)
    {
        ss<<"a";
    }
    str3=ss.str();
    timeEnd = time(NULL);
    cout<<"stringstream 方法所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

    timeBegin = time(NULL);
    string s4 = "";

    char* cp = new char [num];
    char *tt=cp;
    char *t1="a";
```

```
size_t strLength=sizeof(t1);
for(int i=0; i<num; i++)
{
    sprintf(cp,"%s",t1 );

    cp++;
}

s4 = cp;


timeEnd = time(NULL);

cout<<"sprintf 方法所耗费的时间: "<<timeEnd - timeBegin<<" ms"<<endl;

return 0;
}
```

运行结果如下:

从中可以知道, sprintf是目前这四者速度最快的。其次分别是stringstream、str.append和str+=a方法。



```
str+=a 所耗费的时间: 37 ms
str.append(a)所耗费的时间: 36 ms
stringstream 方法所耗费的时间: 32 ms
sprintf 方法所耗费的时间: 24 ms
http://blog.csdn.net/
```

注意, sprintf是不安全的, 该函数无法检查目的缓存区是否溢出, 现在一般采用snprintf对其进行替代使用。类似的函数还有gets, strcat和strcpy, 建议分别用fgets,strncat和strncpy进行替代使用。