【C++ IO机制】stream_buf 解析

目录

正文

回到顶部

stream_buf 解析

1. stream_buf原型：

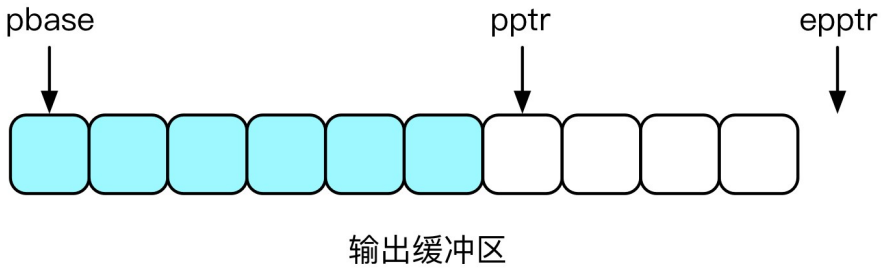
```
template <class CharT, class Traits = std::char_traits<CharT>>
class basic_streambuf;
```

【注意】

```
typedef basic_streambuf<char> streambuf;
```

用于输出的 streambuf

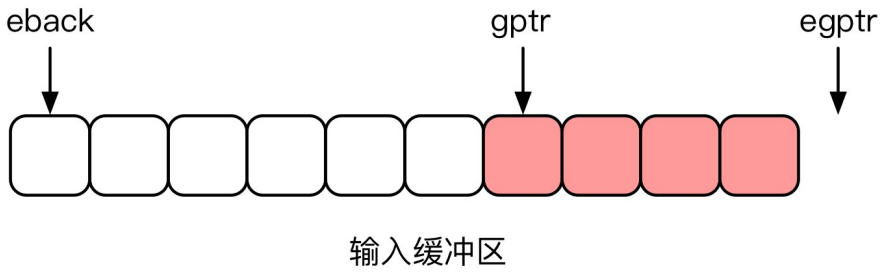
streambuf 使用三个指针来管理相应的输出缓冲区(缓冲区需要自行设置)，分别由接口 pbase，pptr和 epptr 返回。其中 pbase 是缓冲区的基指针，指向缓冲区的第一个字节，epptr 是缓冲区的尾指针，指向其最后一个字节的下一个字节(类似于 iter.end() 的作用)，而 pptr 指向缓冲区当前可用的位置，也就是pptr 之前都已经被数据所填充，如下图：



streambuf 定义的输出相关的函数主要有 sputc 和 sputn，前者输出一个字符到缓冲区，并且将指针 pptr 向后移动一个字符，后者调用函数 xsputn 连续输出多个字符，xsputn 默认的实现就是多次调用 sputc。由于缓冲区有限，当 pptr 指针向后移动满足 pptr() == epptr 时，说明缓冲区满了，这时将会调用函数 overflow 将数据写入到外部设备并清空缓冲区；清空缓冲区的方式则是调用 pbump 函数将指针 pptr 重置。我们可以通过如下的类来实现自定义的输出 buffer：

用于输入的 streambuf

同管理输出缓冲区一样，streambuf 也使用三个指针，eback()，gptr() 以及 egptr() 分别指示输入缓冲区的开始字节，当前可用字节以及缓冲区尾的下一字节，如下图所示：



streambuf 类同样定义了如下几个函数来支持对于输入缓冲区的读取和管理：

- sgetc: 从输入缓冲区中读取一个字符；
- sbumpc: 从输入缓冲区中读取一个字符，并将 gptr() 指针向后移动一个位置；
- sgetn: 从输入缓冲区中读取 n 个字符；
- sungetc: 将缓冲区的 gptr() 指针向前移动一个位置；
- sputbackc: 将一个读取到的字符重新放回到输入缓冲区中；

与输出缓冲区不同的是，输入缓冲区需要额外提供 putback 操作，也就是将字符放回到输入缓冲区内。

1. 示例

```
#include <iostream>
#include <fstream>

int main()
{
  std::ifstream ifs("test.txt");
  if (ifs.good())
  {
    std::streambuf *pbuf = ifs.rdbuf();
    char c;
    ifs >> c;
    std::streamsize size = pbuf->in_avail();
    std::cout << "first character in file: " << c << '\n';
    std::cout << size << " characters in buffer after it\n";
  }
  ifs.close();

  return 0;
}
```

说明：test.file 文件内容如下：

```
this is test file!
```

输出：

```
first character in file: t
18 characters in buffer after it
```

```
#include <iostream>
#include <sstream>

int main()
{
  std::stringstream stream("Hello, world");
  std::cout << "sgetc() returned '" << (char)stream.rdbuf()->sgetc() << "'\n";
  std::cout << "peek() returned '" << (char)stream.peek() << "'\n";
  std::cout << "get() returned '" << (char)stream.get() << "'\n";
}
```

输出：

```
sgetc() returned 'H'
peek() returned 'H'
get() returned 'H'
```

说明：

sgetc()	返回当前位置的字符。
sbumpc()	返回当前位置的字符，并将当前位置前进到下一个字符。
snextc()	将当前位置前进到下一个字符并返回下一个字符。

回到顶部

参考资料

1. std::streambuf从示例到应用
2. stream_buff 解析【cppreference.com】
3. C/C++编程：流缓冲类std::basic_streambuf

分类：【02】C++ 基础

标签：C++ IO

苏格拉底的落泪
粉丝 - 9 关注 - 31
+加关注

« 上一篇：【C++ IO机制】文件输入输出 ①
» 下一篇：【C++ IO机制】标准输入输出 ①

posted @ 2021-12-25 13:59 苏格拉底的落泪 阅读(482) 评论(0) 编辑 收藏 举报

评论排行榜

1. 【ceph运维】删除osd和host(2)
2. 【ceph集群】docker 安装 ceph 集群(1)
3. 【C++ Primer 第15章】定义派生类拷贝…

推荐排行榜

1. 【osd】OSD的内部队列(1)
2. install(1)
3. 【linux命令】查询命令安装路径(1)
4. epoll(二)(1)
5. 字节对齐（内存对齐）(1)

最新评论

1. Re:【ceph集群】docker 安装 ceph 集群
都用了三台虚拟机，还用docker干啥？不会ceph部署吗？
--亚正中
2. Re:【ceph | 运维】删除osd和host
@xu_jd 查看一下 /usr/lib/systemd/system 目录下是否存在 ceph-osd@.service 文件。这个文件 在 ceph 源码 systemd 目录下。我的操作系统为 c...
--苏格拉底的落泪
3. Re:【ceph | 运维】删除osd和host
为啥我的15.2 没有这样的命令啦？
[root@ceph1 ceph]# ceph osd tree ID CLASS WEIGHT TYPE NAME STATUS REWEIGHT PRI-AF...
--xu_jd
4. Re:【C++ Primer 第15章】定义派生类拷…
天才呀，看了此文，恍然大悟
--追梦的路上