

1) 输入输出重定向 `stdin/stdout`  
`freopen`, `freopen` 可重定向 `cin`, `cout`

`cerr` 对应于标准错误输出流, 用于向屏幕输出出错信息.

2) 流操纵符, `#include <iomanip>`

- 整数流的基数: 流操纵符 `dec`, `oct`, `hex`, `setbase`.
  - 浮点数的精度: `precision`, `setprecision`
  - 设置域宽 (`setw`, `width`)
  - 用户自定义的流操纵符.
- 成员函数 → 流操纵符.

`cout << hex/dec/oct << ... <<`

3) 文件读写

① 文件打开方式: `ios::out` 输出到文件, 删除原有内容

`ios::app` 输出到文件, 保留原有内容, 总是在尾部添加.

`ios::binary` 以二进制文件格式打开.

`ios::ate` 打开后立即寻位到文件尾

路径的绝对路径和相对路径:

△ 绝对路径:

△ 相对路径:

`\\tmp\\mdir\\... .txt`

当前盘符的根目录下 for tmp \ ---  
 tmp \ \mdir \ \ --- .txt  
 当前文件夹的子文件夹 tmp \ ---  
 \ \ tmp \ \mdir \ \ --- .txt  
 当前文件夹的父文件夹下 for tmp 子文件夹是 ---

文件读写指针:

ios::beg 设置指针到流的开始

ios::end - - - - 流末尾

ios::cur 流位置指示器的当前位置

字符文件读写:

因为文件流也是流, 所以流的成员函数和流操作算子也同样适用于文件流. (可直接使用 operator >>, operator << 对文件进行输入、输出操作).

二进制文件读写:

二进制读文件:

istream & read(char \*s, long n);

写文件:

ostream & write(const char \*s, long n);