FITSIO C++ 接口库 LIBFIO 使用说明

顾俊骅

v1.0, June 7, 2007
v1.1, May 8, 2012, Minor changes

1 功能描述

`libfio` 是 `cfitsio` 软件包在 `C++` 语言上的包装。通过 `libfio`，可以在 `C++` 语言中方便地操纵 `fits` 格式的文件，主要是 `fits` 图像文件。`libfio` 隐藏了 `cfitsio` 库中所要求的 `C` 语言指针语法，可以大大地提高效率，精简程序长度。

2 安装

`libfio` 依赖于 `cfitsio` 和 `blitz++` 两个库。

2.1 安装 `blitz++`

Debian Linux 官方源中有 `blitz++`，分为 `libblitz0-dev`, `libblitz0ldbl` 和 `libblitz-doc` 三个包，可以使用 `apt` 或 `aptitude` 直接安装：

```
# apt-get install libblitz0-dev libblitz0ldbl libblitz-doc
```

Ubuntu Linux 官方源中未包含 `blitz++` 软件包，但可以直接从 `launchpad.net` (https://launchpad.net/ubuntu/+source/blitz%2B%2B) 下载编译好的二进制包，注意根据安装的系统选择 `i386` (32 位) 或者 `amd64` (64 位)，然后使用 `dpkg` 安装，如：

```
# dpkg -i libblitz0-dev_0.9-12build1_amd64.deb \
    libblitz0ldbl_0.9-12build1_amd64.deb \
    libblitz-doc_0.9-12build1_all.deb
```

对于其他 Linux 发行版，如果在源里找不到，也没有预编译好的二进制包，则可手动编译安装。注意应该安装好相应的开发工具和依赖软件包。

2.2 设置 fitsio

更新：本软件包所需要的 fitsio 库和头文件现已整理存放于子目录 cfitsio，所以可以直接进行之后的编译安装。

此处整理出来的 fitsio 等文件是从 HEASOFT 6.12 提取。

3 编译安装 libfio

解压本软件包后，进入解压后的目录，首先以 “普通用户” 执行以下命令编译：

```
$ make clean
$ make
```

如果编译顺利通过，则可以切换到 “超级用户” 安装：

```
# make install
```

默认安装路径为 /usr/local/include 和 /usr/local/lib。同时在系统目录 /usr/lib 建立符号链接，否则编译后的程序可能因无法找到所需的库而无法正常运行。如果不满意，当然可以亲自修改 Makefile。

4 使用 libfio

4.1 头文件包含

首先应该在使用到 libfio 的程序源文件中包含对应的头文件 fio.h：

```
#include <fio.h>
```

通常情况下，可能还需要再添加一行：

```
using namespace blitz;
```

4.2 编译选项

如果将 libfio 库和相关的头文件安装于默认路径，即 /usr/local/lib 和 /usr/local/include，则在编译使用到 libfio 的程序源文件时，应该添加如下编译选项：

```
-I/usr/local/include -L/usr/local/lib -lfio -lcfitsio -lblitz
```

此外，建议添加选项 -Wall 来让编译器报告所有警告，还可以添加选项 -g 以用于之后可在 GDB 中调试。如果确认程序没有错误或不足，则可以使用选项 -O2 来打开编译器的一些优化功能。

4.3 程序实例

下面通过分析一个实际程序，介绍 `libfio` 库的使用方法。

```
// 包含 fio 头文件
#include <fio.h>
#include <iostream>
#include <cmath>

// 命名空间
using namespace std;
using namespace blitz;

// 主程序开始
int main(int argc, char *argv[])
// argc 和 argv 用于从命令行获取参数
// argc 是命令行参数的数目，包含程序名称
// argv[0], argv[1], ..., argv[argc-1] 是各个参数字符串
{
    if (3 != argc) {        // 判断输入参数个数是否符合要求
        fprintf(stderr, "Usage:\n");
        fprintf(stderr, "     %s in.fits out.fits\n", argv[0]);
        exit(-1);           // 强制程序结束并返回错误代码 -1
    }

    cfitsfile ff1;          // 声明一个用于操作 fits 文件的对象
    ff1.open(argv[1]);      // 打开一个已经存在的 fits 文件
    Array<double,2> img1;   // 声明一个用于存放图像的矩阵
    ff1 >> img1;            // 将数据从 fits 文件导入矩阵

    // 然后可以对这个矩阵中的数据进行各种处理
    // 比如说，这里计算输入图像的泊松误差
    Array<double,2> img2(img1.shape());
    int i, j;
    for (i=0; i<img2.extent(0); ++i) {
        for (j=0; j<img2.extent(1); ++j) {
            img2(i,j) = sqrt(img1(i,j));
        }
    }

    cfitsfile ff2;          // 声明另一个用于操作 fits 文件的对象
    ff2.create(argv[2]);    // 新建一个 fits 文件
                            // 如果文件存在，则先删除再新建
    ff2 << img2;            // 将数据从矩阵导出到 fits 文件

    return 0;               // 程序正常结束
}
```

5 ChangeLogs

2012/05/08, v1.1

- 更正 `fio.cc` 中两处相同的错误：写错了变量名，导致编译时被警告有未使用的变量。

- 整理本软件包所需的 `fitsio` 库和对应的头文件，与软件一同打包，并更新相关头文件和 `Makefile`，支持 32 位与 64 位系统。
- 修改了 `Makefile`，添加所需要的 `-fPIC` 选项，修改安装路径到 `/usr/local`，以及其他一些小改动。
- 更新了说明文档。

2007/06/07, v1.0

- 原始程序