

# IRAF/PyRAF インストール講習会

磯貝 瑞希

国立天文台 天文データセンター

2017.12.19

-- ウェブ掲載版 --

# はじめに

本日の講習会の目的:

IRAF/PyRAFをシステム(Linux, CentOS 7)にインストールできるようになること。

講習内容:

## 1. IRAF (v2.16.1) のインストール

- IRAF, STSDAS/TABLES, x11iraf, SAOimage ds9

## 2. PyRAFのビルド・インストール

- (Python のビルドに必要な)パッケージのインストール
- Python (v3.5.4) + (PyRAFに必要な)モジュールのインストール
- PyRAF (v2.1.14) のインストール

# 参考: CentOS 7 の情報

## OSの情報:

- CentOS 7.4
- ソフトウェアの選択: Gnome Desktop + 開発ツール
- OSインストール後にインストールしたパッケージ:
  - emacs
- アカウント名: 本テキストでは一般ユーザのアカウント名を「**user**」とする
- OSインストール後の設定変更:
  - スクリーンセーバーの解除、
  - 日本語入力設定(「半角/全角」キーで切替可能)
  - フォルダ名 (ダウンロード、ドキュメント など) の英語化

# 1-A. IRAF 2.16.1 のインストール

インストール用ファイルの入手:

ウェブブラウザ(アプリケーション → お気に入り → Firefox Web ブラウザー)  
を起動し、本家サイト: <http://iraf.noao.edu>  
より、「Linux 64bit版」をダウンロード:

`ftp://iraf.noao.edu/iraf/v216/PCIX/iraf.linux.x86_64.tar.gz`

インストール

ターミナルを開き(右クリックしてメニューの一番下を選択)、rootになる

`$ su -`

インストール先ディレクトリを作成する (下記以外でも可)

`# mkdir -p /usr/local/iraf/v2161/iraf`

作成したディレクトリに移動する

```
# cd /usr/local/iraf/v2161/iraf
```

ダウンロードしたファイルを展開する:

```
# tar xvfz /home/user/Downloads/iraf.linux.x86_64.tar.gz
```

環境変数 \$iraf を設定する

```
# export iraf=$(pwd)
```

```
# echo $iraf
```

→ /usr/local/iraf/v2161/iraf と表示されればOK.

インストールの実施

```
# ./install -system
```

→ 実行後、以下については、青文字を入力。それ以外は「enter」でOK

```
New iraf root directory (/usr/local/iraf/v2161/iraf): [enter]
```

Default root image storage directory (/usr/local/iraf/imdirs): **/usr/local/iraf/v2161/imdirs**

Default root cache directory (/usr/local/iraf/cache): **/usr/local/iraf/v2161/cache**

Local unix commands directory (/usr/local/bin): **[enter]**

```
=====
===== Verifying System Settings =====
=====
```

Hostname = myhost.mydomain OS version = Linux 3.10.0-693.11.1.el7.x86\_64

Architecture = linux64 HSI arch = linux64

Old iraf root = /iraf/iraf New iraf root = /usr/local/iraf/v2161/iraf

Old imdir = /iraf/imdir New imdir = /usr/local/iraf/v2161/imdirs

Old cache = /iraf/cache New cache = /usr/local/iraf/v2161/cache

Local bin dir = /usr/local/bin

Proceed (yes): **[enter]**

→ インストールが最後まで進む。

## 出力情報の続き その1

```

=====
===== Begin Installation =====
=====

Checking for existing commands directory...          [ OK ]

                Editing Paths
                -----
Editing the user .login/.cshrc paths ...             ./install: 行 1006: ((
: 0=0 : 非変数に割り当てを行おうとしています (エラーのあるトークンは "=0 ")
*** Cannot
[ OK ]
Editing iraf/imdir/cache paths into system files ... [ OK ]

                Checking File Permissions
                -----
Checking iraf file permissions ...                   [ OK ]
Creating root imdir at /usr/local/iraf/v2161/imdirs ... [ OK ]
Creating root cache at /usr/local/iraf/v2161/cache ... [ OK ]
Reset /tmp sticky bit setting ...                   [ OK ]

                Creating File Links
                -----
Checking for /iraf symlink ...                       [ FAIL ]
Checking /usr/include directory ...                  [ OK ]
Creating <iraf.h> symlink ...                       [ OK ]
Creating iraf command links in local bin dir ...    [ OK ]
Marking system update time hlib$utime ...           [ OK ]

                Installing VOClient Code
                -----
Creating 'voclientd' symlink ...                    [ OK ]
Creating 'voclient.jar' symlink ...                  [ OK ]

```

```
                Creating Graphics Device Files
                -----
Checking /usr/local/lib directory ...           [ OK ]
[ ]reating /usr/local/lib/imtoolrc link ...     [ OK ]
Creating X11IRAF links ...                       [ OK ]

=====
Congratulations!  IRAF has been successfully installed on this system.
=====

    To begin using the system simply log in as any user and from the
    directory you wish to use as your iraf login directory type:

        % mkiraf      # create a login.cl file
        % cl          # start IRAF

    The 'iraf' user is already configured with a login.cl file so a simple
    'cl' command is sufficient to start the system.
    Additional user information can be found at the IRAF.NET web site:

        http://iraf.net

    Please contact http://iraf.net with any questions or problems.

=====
=====  Installation Completed With No Errors  =====
=====
```

→ 以上が表示されれば、install が完了。

## リンクの修正:

(補足: リンクの修正をしなくても一見問題なく動くようだが、全てのタスクで正常に動作する保証はないため、事前に修正しておくことを推奨する)

bin, unix/bin, unix/as, unix/hlib/iraf.h のリンクを修正する

```
# rm bin
# ln -s bin.linux64 bin
# cd unix
# rm bin
# ln -s bin.linux64 bin
# rm as
# ln -s as.linux64 as
# cd hlib
# rm iraf.h
# ln -s iraf64.h iraf.h
```

→ 以上で IRAF 本体インストール後の修正作業が完了。

## 動作確認

新しくターミナルを開き、mkirafを実行後、IRAFを起動する

```
$ mkdir iraf
```

```
$ cd iraf
```

```
$ mkiraf -i -t=xterm
```

```
$ cl
```

```
vocl> epar imstat
```

→ IRAFが正常に起動し、imstatのパラメータ編集モードになればOK.

## 1-B. 外部パッケージのインストール

外部パッケージ STSDAS/TABLES をインストールする

(補足: STSDAS/TABLES以外の外部パッケージのインストールも同様)

externディレクトリに移動

```
# cd /usr/local/iraf/v2161/iraf/extern
```

configure, make を実行

```
# ./configure
```

```
# make stsdas tables
```

リンクを修正

```
# cd stsdas ; rm iraf.h ; ln -s /usr/local/iraf/v2161/unix/hlib/iraf.h .
```

```
# rm bin ; ln -s bin.linux bin
```

```
# cd ../tables ; rm bin ; ln -s bin.linux bin
```

「;」 複数行のコマンドを1行に  
まとめて書くための区切り文字

補足: binのリンクを修正しない場合、IRAF では問題ないが PyRAF ではエラーとなる。

# 1-C. x11iraf のインストール

IRAF v2.16.1 のインストールファイル一式にx11irafも同梱されている。ユーザインストールでは、同梱されているx11iraf関連ファイルのリンクが自動生成され、利用できるようになるが、システムへのインストールでは自動生成されないため、手動でリンクを作成する。

(補足: ./install実行時の出力 Creating X11IRAF link... [OK] でも実際は何もしていない)

## 変数の定義

```
# dir=/usr/local/iraf/v2161/iraf/vendor/x11iraf
```

## 変数定義の確認

```
# cd $dir ; pwd
```

## リンク作成

```
# ln -s $dir/bin.linux/* /usr/local/bin/
```

```
# ln -s $dir/lib.linux/* /usr/local/lib/
```

```
# ln -s $dir/include/* /usr/local/include/
```

```
# ln -s $dir/app-defaults/* /usr/share/X11/app-defaults/
```

```
# ln -s $dir/man/* /usr/local/share/man/man1/
```

## 動作確認

新しくターミナルを開き、xgtermでIRAFを起動できるか確認する

login.cl を初期化し、ターミナルとしてxgterm を設定する

```
$ cd ~/iraf
```

```
$ mkiraf -i -t=xgterm
```

xgterm上で IRAF を起動する

```
$ xgterm -sb -e cl &
```

IRAF上で以下を実行し、グラフが表示できるか確認する

```
vocl> implot dev$pix
```

→ グラフが表示できればOK.

## 1-D. ds9 のインストール

FITS画像ビューア SAOimage ds9 をインストールする  
バイナリファイルの入手:

本家サイト: <http://ds9.si.edu/site/Home.html>

のダウンロードページより、CentOS7版

<http://ds9.si.edu/download/centos7/ds9.centos7.7.5.tar.gz>

を入手する。

インストール(インストール先で展開するだけ)

```
$ su -
```

```
# cd /usr/local/bin
```

```
# tar xvfz /home/user/Downloads/ds9.centos7.7.5.tar.gz
```

## 動作確認:

起動しているIRAFプロンプト上で以下を実行する

```
vocl> ! ds9 &
```

```
vocl> display dev$pix 1
```

→ 画像が表示できればOK

以上で IRAF と関連ソフトのインストールは全て完了。

## 2. PyRAF のインストール

PyRAFのインストールは、以下の通り、インストール前に必要な環境・ソフトのインストールを先行実施する。

2-A: Pythonのビルドに必要なパッケージのインストール

2-B: Python 3.5.4 のビルド・インストール

2-C: モジュールのインストール

2-D: PyRAF 2.1.14 のビルド・インストール

## 2-A. Pythonのビルドに必要なパッケージのインストール

tcl, tk, tix, libzip と開発用パッケージをインストールする  
(最低限必須なのは tcl, tk, readline とその開発用パッケージ)

```
$ su -
```

```
# yum -y install tcl tk tix tcl-devel tk-devel tix-devel
```

→ 16パッケージがインストールされる。

```
# yum -y install zlib-devel readline-devel bzip2-devel openssl-devel sqlite-devel
```

→ 12パッケージがインストールされる。

```
# yum -y install xz-devel libzip libzip-devel
```

→ 3パッケージがインストールされる。

## 2-B. Python 3.5.4 のビルド・インストール

ソースファイルの入手:

本家サイト: <https://www.python.org/>

より、Downloads -> Source Code を選択、Python 3.5.4 の「Gzipped source tarball」を選択する

<https://www.python.org/ftp/python/3.5.4/Python-3.5.4.tgz>

展開・ビルド:

```
$ cd ; tar xvfz Downloads/Python-3.5.4.tgz
```

```
$ cd Python-3.5.4
```

```
$ ./configure --prefix=/usr/local |& tee configure.log
```

```
$ make |& tee make.log
```

tee: 標準入力の内容を  
標準出力とファイル  
に出力する

テストの実施 (しばらくかかる。実行環境に強く依存)

```
$ make test |& tee make_test.log
```

→ 結果:

385 tests OK.

13 tests skipped:

```
test_dbm_gnu test_dbm_ndbm test_devpoll test_kqueue test_msilib  
test_ossaudiodev test_startfile test_tix test_tk test_ttk_guionly  
test_winreg test_winsound test_zipfile64
```

Tests result: SUCCESS

→ 以上が出力されればOK.

## インストール

システムバンドル版のPython 2.7 と共存する形でインストールを実施 (altinstall)

```
$ sudo make altinstall |& tee make_altinstall.log
```

← パスワードを聞かれた場合、ログインユーザアカウントのものを入力

## 動作確認

新しくターミナルを開いて、Python 3.5.4 を起動する

```
$ python3.5
```

← Python 3.5.4 は /usr/local/bin/ 以下に python3.5 という名前でインストールされている。

Python 3.5.4 が起動後、以下を実行する。

```
>>> import os
```

```
>>> os.system('date')
```

-> 日時が表示され、上下キーの操作で履歴も表示されればOK。

## 2-C. モジュールのインストール

PyRAFに必須のモジュール (numpy, d2to1, stsci.distutils, stsci.tools) に加えて、オプションのモジュール (nose, urwid, astropy, matplotlib) もインストールする。

インストール方法: pip を使用

```
$ su -
```

```
# pip3.5 install numpy
```

```
# pip3.5 install nose
```

```
# pip3.5 install urwid
```

```
# pip3.5 install astropy
```

→ 依存性の関係で 「pytest, py, attr, six, pluggy」 もインストールされる。

# **pip3.5 install matplotlib**

→ 依存性の関係で「cyclcr, pyparsing, python-dateutil, pytz」もインストールされる。

# **pip3.5 install d2to1 stsci.distutils stsci.tools**

インストール先:

→ /usr/local/lib/python3.5/site-packages/ 以下

確認:

\$ **pip3.5 list --format columns**

→ インストール済みのモジュールの一覧が表示される。  
この一覧にモジュール名があればOK.

pip3.5 install:

複数のモジュールをまとめて  
指定することも可能

## 2-D. PyRAFのインストール

PyRAF は ソースファイルをビルドしてインストールする  
(pip でインストールしたバイナリ版では、グラフウィンドウやパラメータウィンドウ  
などが表示されないという問題があるため)

ソースファイルの入手:

本家サイト: [http://www.stsci.edu/institute/software\\_hardware/pyraf](http://www.stsci.edu/institute/software_hardware/pyraf)

より、「stand-alone PyRAF download page」に移動、PyRAF v2.1.14 をダウンロード。

<http://stsdas.stsci.edu/download/pyraf/pyraf-2.1.14.tar.gz>

展開・ビルド:

```
$ cd ; tar xvfz Downloads/pyraf-2.1.14.tar.gz
```

```
$ cd pyraf-2.1.14/
```

```
$ python3.5 setup.py build |& tee build.log
```

チェック・インストール:

```
$ python3.5 setup.py check
```

```
$ sudo /usr/local/bin/python3.5 setup.py install --prefix=/usr/local |& ¥  
tee install.log
```

補足: ¥ 改行のエスケープ。

¥ の直後に[enter]を入力する

確認:

新しくターミナルを開き、pyraf を実行する

```
$ pyraf
```

→ IRAF 2.16.1 起動時のメッセージが表示され、その後に

PyRAF や Python のバージョン番号の表示 (2.1.14, 3.5.4) があればOK.

練習1: 追加の動作確認:

PyRAF上で ds9 を起動し、fits画像の表示や imexam で簡易測光や radial profileの表示、さらにタスク「imstat」のパラメータウインドウを表示する

## 練習1の回答例:

```
vocl> !ds9 &
```

```
vocl> display dev$pix 1
```

```
vocl> imexam
```

→ fits画像上の星にポインタを合わせ、「a」や「r」を押す

```
vocl> epar imstat
```

## 補足:

- Python 3系のPyRAFでは、タスク名のtab補完が機能しない。

(確認: PyRAF v2.1.14 on Python 3.4.7, 3.5.4, 3.6.4

PyRAF v2.1.13 on Python 3.5.4)

- Python 2.7系のPyRAFでは、上記問題が起きないことを確認済み。

(確認: PyRAF v2.1.14 on Python 2.7.14)