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

2018.06.05

-- ウェブ掲載版 --

講習の目的と内容:

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:

0. 仮想マシンの起動

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

3. AstroCondaを利用したインストール

参考: CentOS 7の情報

OSの情報:

- **CentOS 7.5**
- **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.lnux.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

行頭の「$」は一般ユーザでのコマンド実行を、「#」はrootでの実行を表す

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.lnux.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 -s
```

→ 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 = vm01.mydomain OS version = Linux 3.10.0-862.3.2.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  ]
```

出力情報の続き その2


```
                    Creating Graphics Device Files
                    ------------------------------
Checking /usr/local/lib directory ...                               [  OK  ]
Creating /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
```

make時に内部で「wget」コマンドを実行している。wget がない場合には「# yum -y install wget 」でインストールする

リンクを修正

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

→ グラフが表示されれば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 の ver. 7.5

http://ds9.si.edu/archive/centos7/ds9.centos7.7.5.tar.gz

を入手する。(Download – 左側メニューの Archive – centos7)

※ 最新版(7.6)はIRAFからの画像表示でエラー(Cannot open device)となるため。

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

テストの実施準備:

接続先閉鎖によりエラーとなるテスト(test_xmlrpc_net)を除外する

```
$ mkdir bkup; mv Lib/test/test_xmlrpc_net.py  bkup/
```

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:

```
384 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
# pip3.5 install matplotlib
```

→ 依存性の関係で 「cycler, kiwisolver, pyparsing, python-dateutil, pytz, six」
もインストールされる（注:本講習実施時では）。

pip3.5 install d2to1 stsci.distutils stsci.tools

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

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.

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　ではグラフウインドウやパラメータウインドウ
　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:
　本家サイトではソースを配布されなくなったため、PyPI (=Python Package Index)
　　https://pypi.org/
　より入手する。「pyraf」で検索、「pyraf 2.1.14」を選択、左側のメニューより
　「Download files」を選択、表示される「pyraf-2.1.14.tar.gz」を選択。

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)

3. AstroCondaを利用したインストール

現在のPyRAFの本家サイトでは、最新版のダウンロード・インストールについて、
AstroCondaの説明ページ:

http://astroconda.readthedocs.io/en/latest/

へ誘導しており、Condaを利用した方法「のみ」を紹介している。

Conda:

Python用のパッケージ管理・仮想環境構築ツール(オープンソース)

本家サイト: https://conda.io/docs/

クロスプラットフォームで、Windows, Mac, Linux版が提供されている。

AstroConda:

フリーなCondaチャンネルの一つ。PyRAFと同じく、STScI(=Space Telescope Science Institute) によって維持されている。HST などのデータを処理・解析するために必要なソフトやツールなどを提供している。

3. AstroCondaを利用したインストール

AstroCondaを利用したインストール手順は、以下の通り:

3-A: Condaのインストール

3-B: Condaの設定: AstroCondaチャンネルの登録

3-C: IRAF, PyRAF, ds9 のインストール

3-D: PATHの設定

この方法では、IRAF, x11iraf, STSDAS/TABLES, PyRAF, 関連Pythonモジュールも
まとめてインストールされる。
ただし、インストールされる IRAF は 2.16 32bit版 のため、32bit用の各種ライブラリ
が必要。以下、インストール手順(「¥」は改行のエスケープ):

```
$ su -
#  yum -y install glibc.i686  zlib.i686 ncurses-libs.i686 bzip2-libs.i686  ¥
        uuid.i686 libxcb.i686
```

3-A: Condaのインストール

Condaの導入方法:

2つのdistribution(Miniconda/Anaconda)があり、それぞれに Python 2.7系 (Miniconda2/Anaconda2) と Python 3.x系 (Miniconda3/Anaconda3) がある。

Miniconda: 必要最小限のConda 管理環境を提供

Anaconda: 完全なConda 管理環境 + 数百の有用なツールやライブラリをデフォルトで提供

→ 本講習では、軽量な Python 3系の Miniconda3 を使用する。

Miniconda3 の入手:

本家サイトのdownloadページ

URL: https://conda.io/miniconda.html

より、Python 3.6, Linux, 64bit を選択し、「Miniconda3-latest-Linux-x86_64.sh」を入手。

3-A: Condaのインストール

root で　ダウンロードしたファイルを bash で実行:

$ su -

 # bash /home/user/Downloads/Miniconda3-latest-Linux-x86_64.sh

→　Enterを押すとライセンス条項が表示される。最後まで進んで、「yes」を入力。

→インストール先を変更: [/usr/local/miniconda3」を入力。

(インストール先は変更可。変更した場合は、これ以降の設定箇所もそれに合わせて変更)

→ /root/.bashrcファイルの編集: yes

確認:

 # source ~/.bashrc

 # which conda

→「/usr/local/miniconda3/bin/conda」と表示されればOK.

3-B: Condaの設定
3-C: IRAF/PyRAF/ds9 のインストール

3-B: Condaの設定: AstroCondaチャンネルの登録

```
# conda config --add channels http://ssb.stsci.edu/astroconda
```

3-C: IRAF/PyRAF/ds9 のインストール (環境名は変更可。本講習では「iraf36」とする)

```
# conda create -n iraf36 python=3.6 iraf-all pyraf-all ds9
```

確認

インストールした環境の有効化(アクティベート)

```
# source activate iraf36
# which cl;  which pyraf; which xgterm; which ds9
```

→「/usr/local/miniconda3/envs/iraf36/bin/cl」などと表示されればOK.

インストールした環境の無効化(ディアクティベート)

```
# source deactivate
```

3-D: PATHの設定

インストールした環境を一般ユーザが利用できるように、PATHの設定を行う。

/etc/profile.d/以下に/usr/local/miniconda3/binのPATHへの追加を書いたファイルを置く (この設定方法の場合、一般ユーザによるPATHの設定は不要)

```
# echo 'export PATH=/usr/local/miniconda3/bin:${PATH}' ¥
      > /etc/profile.d/miniconda3.sh
```

確認:

```
# cat /etc/profile.d/miniconda3.sh
```

3-E: 一般ユーザ環境での動作確認

root環境の終了

```
# exit
```

環境設定の反映 & 一般ユーザ(user)環境での有効化

```
$ source ~/.bashrc ; source activate iraf36
```

動作確認1 (IRAF 2.16　の起動と画像表示、imexamの実行・グラフの表示)

```
$ mkdir -p ~/conda/iraf ; cd ~/conda/iraf
$ echo 'xgterm' | mkiraf
$ xgterm -sb -e cl &
```

→ IRAF 2.16　と表示されることを確認。

```
ecl> !ds9 &
```

ecl> displ dev$pix 1

ecl> imexam

→ 星にポインタを合わせて、a, r, s, e などを押す。

動作確認2 (PyRAFの起動と、起動後は動作確認1と同じこと)

$ pyraf

→ PyRAF 2.1.14, Python 3.6.5 と表示されることを確認。

- -> !ds9 &

- -> displ dev$pix 1

- -> imexam

→ 星にポインタを合わせて、a, r, s, e などを押す。

iraf36環境の無効化

$ source deactivate

3-F: 補足1

1: Python 2.7 環境のインストール方法:

今回 AstroCondaでインストールした PyRAF は Python 3.6 系　をベースにしているが、AstroCondaの本家サイトのFAQ

https://astroconda.readthedocs.io/en/latest/faq.html

では、Python 2.7系の使用を推奨している。

(「STSDASのPythonコードが　Python 2.7　とそれ以前のバージョンを特にターゲットにしているため」とのこと)

Python 2.7 環境をインストールする場合は、本テキストと以下の2点が異なる:

- **3-A: Miniconda2/Anaconda2 をインストール**
- **3-C: # conda create -n iraf27 python=2.7 iraf-all pyraf-all ds9**

3-F: 補足2

2: IRAF 32bit版がインストールされる理由:

本家サイトのFAQによれば、「多くのタスクで64bit版バイナリを用意するには、ソースコードの大幅な変更が必要だったため」とのこと。

3: 従来のインストール との違い:

	従来の方法	AstroConda	備考
IRAF	2.16.1 (32/64bit)	2.16 32bit	
STSDAS/TABLES	3.17	3.18.3	違い: aXe (※1)のバグ修正など
Pythonモジュール の導入・更新・削除	easy_install /pip	conda	

※1: aXe: スリットレス分光データの整約・解析・可視化・シミュレーション用ソフト