# NOT course 2017

**Linux & IRAF**

# 1 Linux

## Basics

- **man** <command> – Shows a manual page for a given <command>
- **pwd** – Shows <path> of current working directory[1] (cwd)
- **ls** – Lists files in cwd
  - -a – Shows all (also hidden files)[2]
  - -l – Long listing format (incl. permissions)
- **cd** – Change working directory
  - cd <dir> – Go to <dir> in cwd
  - cd .. – Go back one directory
  - cd – Return to <home>[3]
- **mkdir** <dir> – Create a new <dir> in cwd
- **rmdir** <dir> – Remove an empty <dir> in cwd
- **rm** <file> – Remove a <file> in cwd
  - -r – Remove a non-empty <dir>
- **cp** <file> <dir> – Copies a <file> in cwd to <dir>
  - cp <file1> <file2> – Copies <file1> to <file2>
  - cp <path>/<file> . – Copies a <file> from <path> to cwd
  - -r – Copies directories recursively
- **mv** <file> <dir> – Moves a <file> to <dir>
  - mv <file1> <file2> – Move <file1> to <file2>
- **cat** <file> – Concatenate files and print on the standard output
  - For instance, cat file.txt prints whatever text is in the file.txt to the terminal
- **more/less** – Same as **cat** but different output and scrolling. Convenient for reading FITS headers
- **locate** <file> – Find <file> by name
- **tar** – Compress files
  - tar -cvf archive.tar <file> – Compress <file> to archive.tar
  - tar -tvf archive.tar – View archive.tar
  - tar -xvf archive.tar – Extract (unpack) archive.tar

[1] folder
[2] Options: ls -l, ls -l -a, ls -la
[3] <home>=/home/username/

### Bit more advanced

- **Tab**/double tab can be used to fill commands instead of writing, get used to it!
- **Multiple files** can usually be given to a command e.g.
  - rm <file1> <file2> <file3> – Removes files <file1>, <file2>, <file3>
  - tar -cf archive.tar <file1> <file2> <file3> – Compress <file1>, <file2>, <file3> to archive.tar
- **Usage of paths is flexible**, e.g.
  - mkdir <path>/<dir> – Creates a new directory <dir> to <path>
  - cd <path>/<dir> – Change cwd to <dir>
- **Wild card *** (examples)
  - rm * – Removes everything in the cwd
  - tar -cf <archive>.tar *.txt – Compresses every file ending with .txt into <archive>.tar
- **Creating file lists**
  - ls *.txt > <file> – Creates a (text file) <file> where every .txt file in the cwd is listed
  - Using >> appends instead of creating/overwriting a file
- **Search certain words in text files**
  - grep -n '<pattern>' *.txt – Print file names and lines matching <pattern> of every .txt file in cwd
- **Stuck?**
  - ctrl+c – Interrupts stuff in the active terminal: writing of a command (typically faster than backspace), scripts stuck in a loop etc. Usually enough. If not:
  - ps -A – Lists all active processes and shows their process ID numbers (#ID)
  - kill #ID – Terminate a process
  - kill -9 #ID – Force terminate a process
  - top – Displays processes, more advanced than ps -A
- **(some) Text editors**
  - gedit – Windows hotkeys, easy to use
  - emacs – For programmers
  - zile – Command line version of emacs
  - nano – Command line
  - vim – Command line, for programmers
- **Misc.**
  - Windowed programs can usually be closed with ctrl+w or ctrl+q, terminals with ctrl+d
  - chmod – Change permissions (allow or deny read/write/execute privileges)

# 2 IRAF

## Getting started

1. **Create login.cl** into cwd by typing mkiraf
   You should see:

   ```
   -- creating a new uparm directory
   Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
   Enter terminal type [default xterm]:
   ```

   Choose xgterm
2. **Edit login.cl** (e.g. with gedit) and find line

   ```
   #set stdimage = imt800
   ```

   Uncomment and set the image size to at least 2048 (NOT/ALFOSC CCD size)

   ```
   set stdimage = imt2048
   ```
3. **Launch ds9** (image display server) **and xgterm** (IRAF terminal)
   - ds9 &[4]
   - xgterm -sb[5] &
4. **Launch IRAF** (in xgterm) by typing cl[6]

## Packages

- Different tasks are located in packages
- packages are listed when launching the IRAF client
- Load a package by typing its name
- Leave a package by typing 'bye'
- Type '?' or '??' to see what is available
- Type 'help package/task' to see various help pages[7]

Table 1: Useful packages

| package | task(s) |
|---|---|
| noao-imred-ccdred | CCD image reduction |
| noao-twodspec-apextract | Spectrum extraction |
| noao-onedspec | Spectrum handling |
| guiapps-spt | Spectool |
| notcam | NOTCam reduction* |

*`http://www.not.iac.es/instruments/notcam/guide/observe.html\#reductions`

## Tasks

- **Image/data manipulation** (e.g. displaying images, data reduction/calibration etc.) is performed with certain tasks.
- **lpar** task – shows current parameter values of a task
- **epar** task – edit parameters of a task
  - :go – Run task with current parameter values
  - :q – Quit task and save current parameter values
  - :q! – Quit task but don't save changes to parameter values
  - :e – Edit additional subparameters if available
- **unlearn** task – reset default parameter values

[4] & means that a process will be run in the background i.e. the working terminal is still available for use (note: cannot be killed with ctrl+c)

[5] Add scroll bar

[6] This is actually ecl in course system

[7] Also available online. Easier to use (search etc.)

- **General tasks**
  - **display** – Displays images on ds9
  - **imexam** – Starts image examination mode on current image visible in ds9. Commands are given with certain keys on the cursor position on ds9:
    - **a** – E.g. point source FWHM, coordinates and quick photometry
    - **r** – Radial profile of a point source
    - **s** – Surface plot
    - **m** – Statistics of a rectangular region
    - **q** – Quit examination mode
  - **splot** – Plots 1D spectra. Some commands:
    - **a** – Zooms to area in between two cursor positions. Double 'a' resets zoom
    - **k** – Gaussian fit in between two cursor positions
      - **-** – Removes a fitted profile in between two cursor positions
      - **r** – Replot
    - **m** – Statistics in between two cursor positions
    - **q** – Quit
  - **imcopy, scopy** – Copy images, spectra
  - **imarith, sarith** – Perform arithmetic operations on images, spectra
  - **hedit** – Edit .fits headers
  - **Use '?' in any interactive task to display the help**
- **Running tasks**
  - **Using epar** (preferred): Every parameter visible, no need to remember stuff
  - **Run and give parameters** on the command line e.g.

    ```
    display file.fits 1 zscale+
    ```

    displays file.fits in 1st frame of ds9 with optional parameter zscale=yes. In this case the mandatory parameters (file name, frame number) are given first and then any optional parameter(s)
  - **Run** a command on the command line (not recommended). Only values of mandatory parameters are asked e.g. file name and frame number in the previous example
- **Lists**
  - File lists[8] can be given as parameters to tasks
  - Lists must be marked with an '@'. For instance,

    ```
    imarith @list + 100 arith_@list
    ```

    adds a constant value of 100 to every pixel in all images given in the 'list'. Modified images have a prefix 'arith_'[9] in this case

[8] Text files containing one file name per line

[9] As a precaution IRAF rather outputs additional files than overwrites images

## Misc.

- **Some linux commands** will work also in cl, some will not. All commands can be used by typing '!' before a command. E.g. !ds9 & opens ds9 in **cl**
- **Only one ds9 should be open** to avoid problems, multiple xgterms are okay
- **AVOID closing windows with 'X'** as this may break IRAF and a restart (of ds9, xgterm) may be needed. This is a feature...
- **Automatic loading** of packages: find the following in login.cl and add packages that are often needed

```
# List any packages you want loaded at login time
images          # general image operators
plot            # graphics tasks
dataio          # data conversions, import export
lists           # list processing
```

# 3 FITS headers

- **FITS:** Flexible Image Transport System, standard data format in astronomy (.fits)
- **Image metadata** e.g. object name, exposure time etc. are stored in the fits header
- **KEYWORDS** are used to identify different metadata, e.g. part of NOT header

```
DATE_OBS= '2017-10-04T23:08:09.915'
OBSERVAT= 'LaPalma '
TELESCOP= 'NOT     '
INSTRUME= 'ALFOSC_FASU'
DETNAME = 'E2V CCD 231-42 2kx2k'
```

- **more/less** <file> – Read the header of a <file>
- **gethead** *.fits KEYWORDS – Display KEYWORDS values of all .fits files in cwd
  For instance

```
gethead *fits OBJECT EXPTIME DATE-OBS
```

  prints the object name, exposure time and the time of the observation
- **Different telescopes** typically have (partially) different keywords!

# 4 NOT data

- **File naming:** for instance the first ALFOSC file in April 2016): ALzd010001.fits
  1. instrument: AL = ALFOSC (NC = NOTCam)
  2. running letter for the year: z (2016)
  3. running letter for the month: d (April)
  4. running number for the day: 01
  5. running file number for this night: 0001
- **Multi-extension FITS format (MEF):** NOT data are in MEF format i.e. there can be more than one image in a single .fits file, especially with NOTCam. This makes running IRAF tasks a bit more inconvenient as the extension must be explicitly given. E.g.

```
display ALzd010001.fits[1] 1
```

  shows the first extension (image). The header is located in 0th extension. For instance NOTCam files have several intermediate steps in extension 2,3,... depending on the exposure mode; final image is in 1st extension.
- **Separate first extensions** before data reduction. There are many ways to do this and probably every project supervisor has their own. One of the easiest (but not quickest) way is:
  1. ls *fits > list (input list)
  2. cp list list2 (output list)
  3. Edit[10] file names in list: .fits -> .fits[1]
  4. Edit file names in list2: .fits -> _1.fits
  5. In IRAF run: imcopy @list @list2 verbose+

  This creates files with only the header and 1st extension with an identifier '_1' in the file names

## Some NOT KEYWORDS

| KEYWORD | Explanation |
|---|---|
| **General** | |
| DATE-OBS | Date of observation |
| MJD | Modified Julian Date |
| UT | Universal Time |
| EXPTIME | Exposure time |
| OBJECT | Object name |
| TCSTGT | Object name in TCS |
| AIRMASS | Airmass at start of exposure |
| OBJRA | Right ascension |
| OBJDEC | Declination |
| IMAGETYP | Image type |
| IMAGECAT | Image category |
| **ALFOSC** | |
| GAIN | CCD gain |
| RDNOISE | CCD readout noise |
| ALFLTNM | ALFOSC filter name |
| FAFLTNM | FASU A filter name |
| FBFLTNM | FASU B filter name |
| ALGRNM | Grism name |
| ALAPRTNM | Aperture name (slit) |
| DETWIN1 | Detector window size |
| DETXBIN | Detector x binning |
| DETYBIN | Detector y binning |
| **NOTCam** | |
| EXPMODE | Exposure (dither) mode |
| NCCAMNM | Camera name |
| NCFLTNM1 | Filter 1 name |
| NCFLTNM2 | Filter 2 name |
| NCAPRNM | Aperture name (slit) |
| NCGRNM | Grism name |

v1: 30 October 2017, Jussi Harmanen

[10] Search & replace: ctrl+h in gedit