

Digital Photography with GNOME

Hubert Figuiere <hub@figuiere.net>

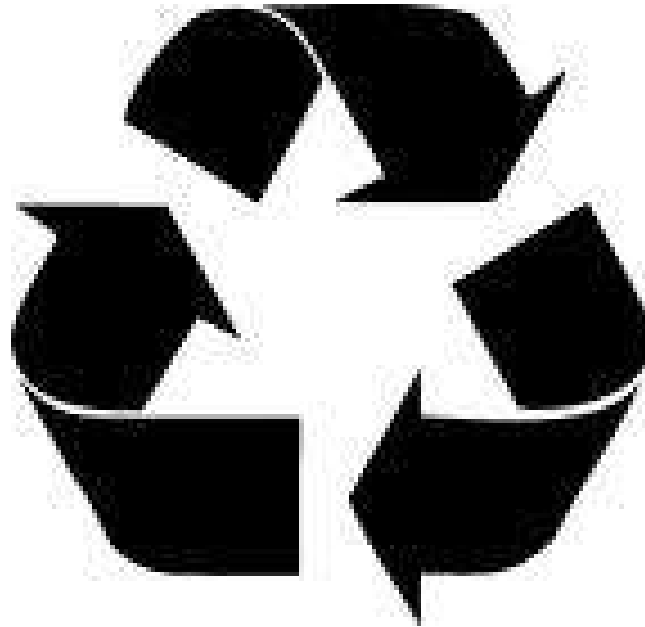
GUADEC 6

Stuttgart, Germany, May 31st 2005

Digital Photography with GNOME

- Getting pictures out of the camera
- Archiving
- Viewing and sorting
- Decoding
- Editing

This presentation is 100% recyclable



Why ?

- Because lot of people want to use their digital camera
- Lot of geeks still end up using Windows or MacOS
 - Adobe Photoshop CS
 - RAW processing: ACR, C1, or manufacturer's
- Because we all want freedom
- Because I wanted to talk about that

Views...

- Personal view because
 - Everybody has its own way
 - To provide an example
 - To share
- This is not the one true way of doing things

Part 1

Getting pictures from the camera

Getting pictures from the Camera



- With a card reader
 - USB or PCMCIA
- With the USB cable

With a card reader

- Simple
- Efficient
- Fast



With the USB cable

- No need for an accessory
- Can be tricky
- Limited to the camera capabilities
 - Some don't allow file deletion
- May be slow



How to connect ?

- Using Mass Storage
- Using libgphoto2



Copyright 2005 <http://www.libgphoto2.org>

Mass Storage



- as simple as the card reader
- mounts like a disk
 - most distro automount things
- browsed like any file system
 - use your favourite file manager

Which cameras are Mass Storage ?

- Olympus (since 2001)
- Nikon (since 2001)
- Minolta
 - might require Linux kernel patch
- Casio
 - might require Linux kernel patch
- Sony
- Fuji

Example: Olympus C-370

- Connect USB cable to camera
- Select “PC” option on the camera
- And voilà...

Example: Olympus C-370

```
$ dmesg
```

```
....
```

```
ohci_hcd 0001:01:19.0: wakeup
```

```
usb 2-1: new full speed USB device using address 9
```

```
scsi2 : SCSI emulation for USB Mass Storage devices
```

```
Vendor: OLYMPUS    Model: X450/D535Z/C370Z  Rev: 1.00
```

```
Type:    Direct-Access                      ANSI SCSI revision:
```

```
02
```

```
SCSI device sda: 256000 512-byte hdwr sectors (131 MB)
```

```
sda: assuming Write Enabled
```

```
sda: assuming drive cache: write through
```

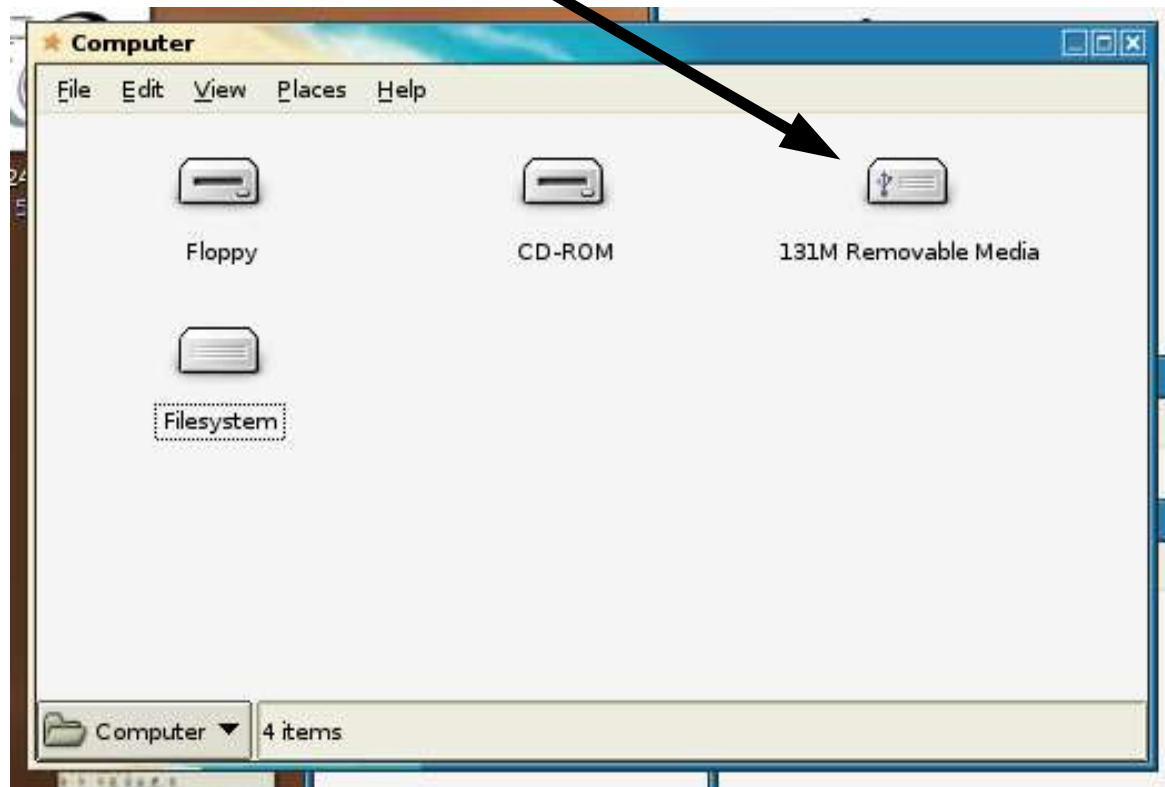
```
/dev/scsi/host2/bus0/target0/lun0: p1
```

```
Attached scsi removable disk sda at scsi2, channel 0, id 0, lun 0
```

```
USB Mass Storage device found at 9
```

Olympus C-370 on the desktop

- Appears as a new disk in Nautilus



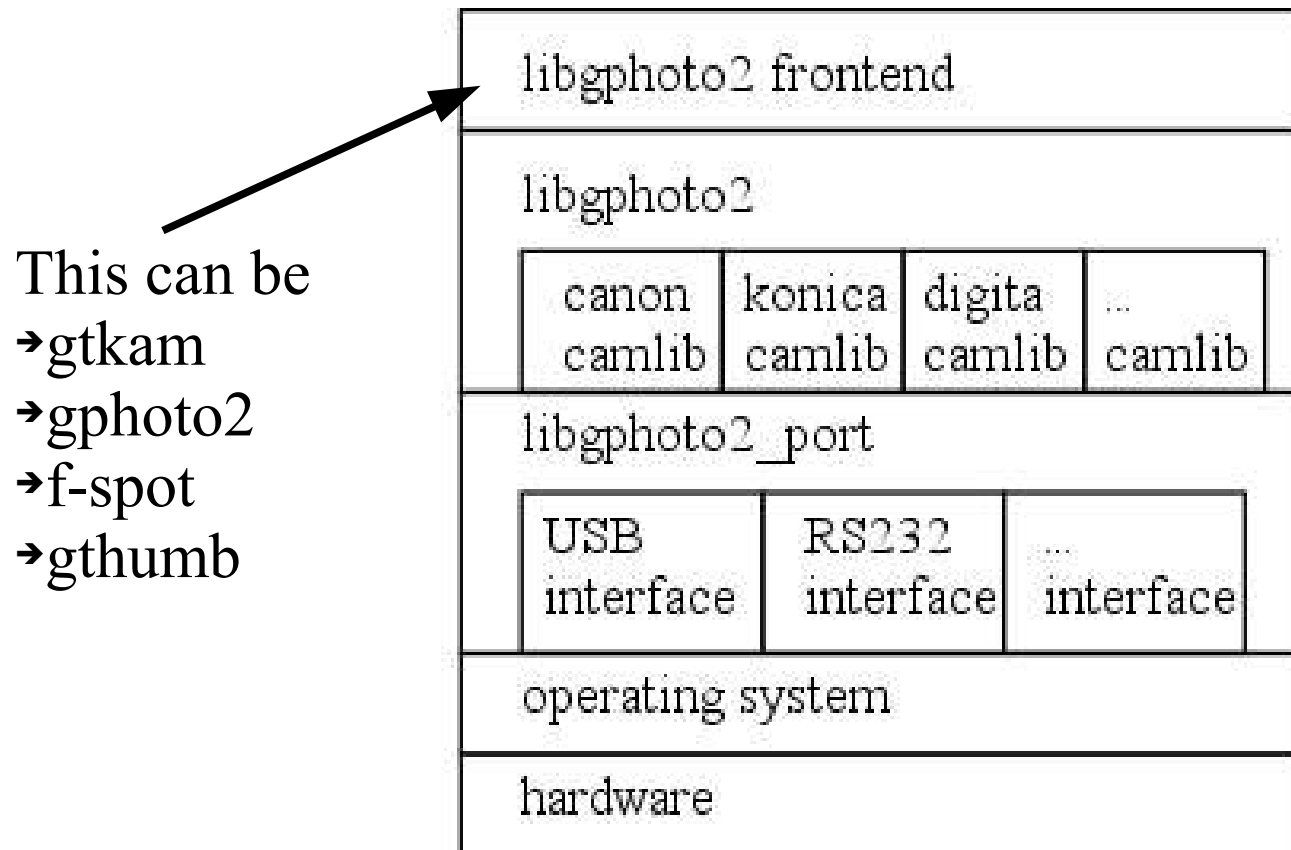
libphoto2



- THE library to access digital cameras (that are not Mass Storage)
- written without help and mostly no documentation from manufacturers
- just a library, several front-ends



libgphoto2 architecture



Which Cameras ?

- Those that use proprietary protocols
 - Canon
 - cheap cameras
 - old Nikon, Fuji, Olympus, Panasonic, Sanyo
- Those that use standard PTP
 - Canon (newer)
 - Kodak
- Those that you want to control
 - unsure which: You Mileage May Vary

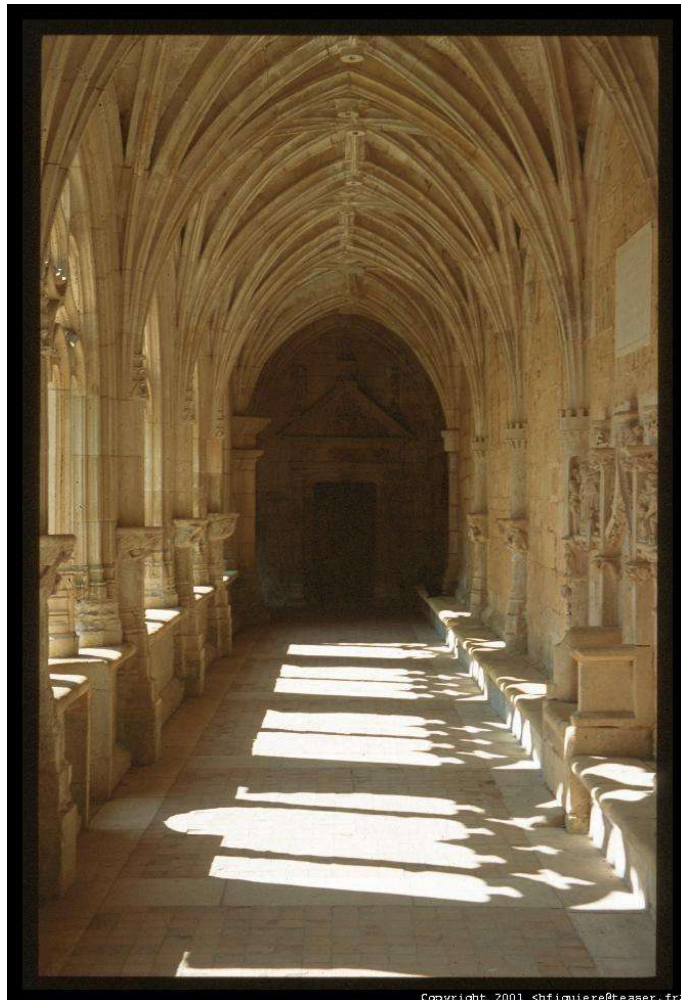
How to help ?

- Provide cameras to developers
- Badger your camera manufacturer by requesting him Open Source support for the camera
 - Canon only support MacOS and Windows in their proprietary SDK
 - Nikon require a NDA to access SDK
 - PTP standard now require to pay USD\$2500 as a I3A membership fee

How to help ? (cont.)

- Report bugs and provide debug information to gphoto developers

Frontends

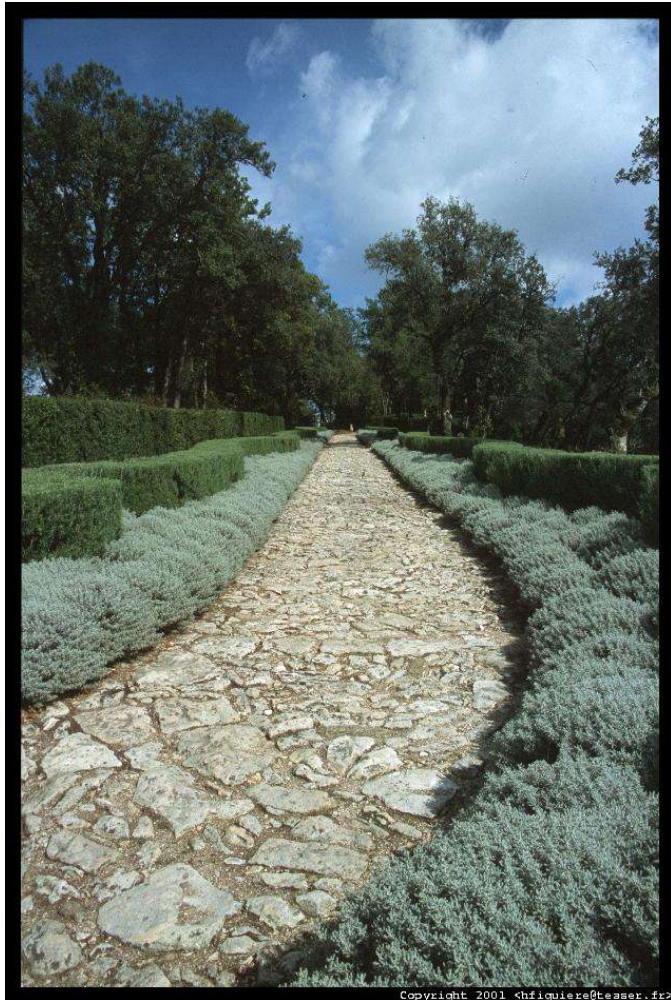


- gtkam
 - the gphoto project GUI
 - only for that use
- gThumb
- F-Spot

Part 2

Archiving

Archiving



- First thing I do because I'm paranoiac
- Thought for the lifetime

How do I archive

- 1 folder per day: 20050528
- Original camera files
- Burnt directly on a CD-R with Nautilus
- CD has a volume label with the date I create the CD and sequence: pcd2005052801
- Each CD has a burning date on the CD
- Good quality CD
- Stored in a sleeve inside a metallic box

Part 3

Viewing and sorting

Viewing and sorting

- What are Metadatas
- Which tools ?



Metadatas

- EXIF
 - Shooting data
 - Technical data
 - MakerNotes (proprietary)
- IPTC
 - Standard tagging for medias
 - No app that supports it
 - but libiptcdata exists
 - ...and a patch for gThumb to use it
 - f-spot will eventually use it too

“MakerNotes”

- Proprietary information inside EXIF data
- Specific to a manufacturer and even to a specific model
- Found in RAW files
- Contain data like lens type, camera program mode, etc.
- Decoded by most EXIF utilities, including libexif, ExifTools (Perl)

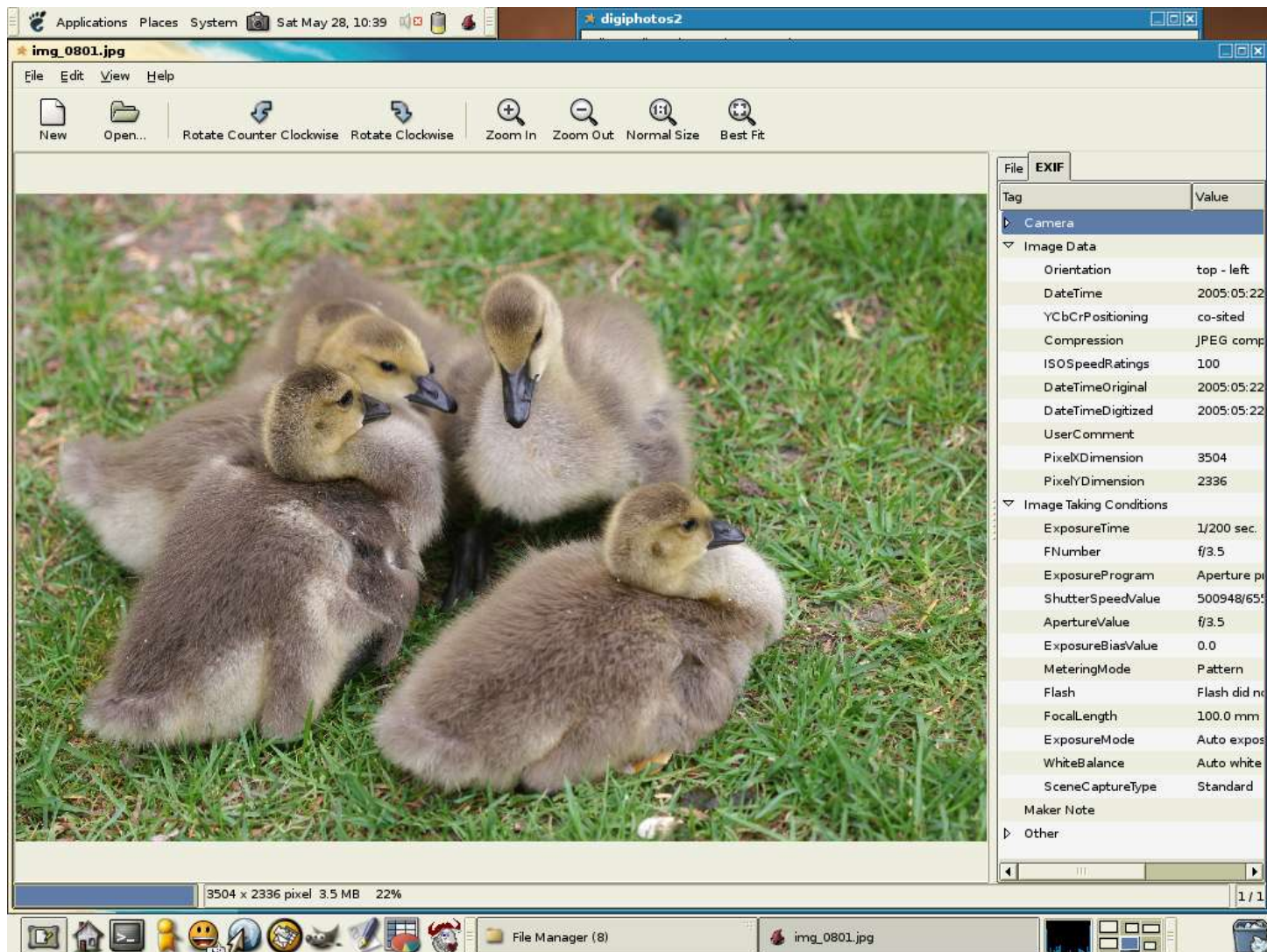
Viewing & sorting tools

- EOG
 - image viewer
- Nautilus
 - file manager
- gThumb
 - image browser
- F-Spot
 - photos collection manager

Eye Of Gnome

- GNOME standard image viewer
- Support EXIF
- Just a viewer

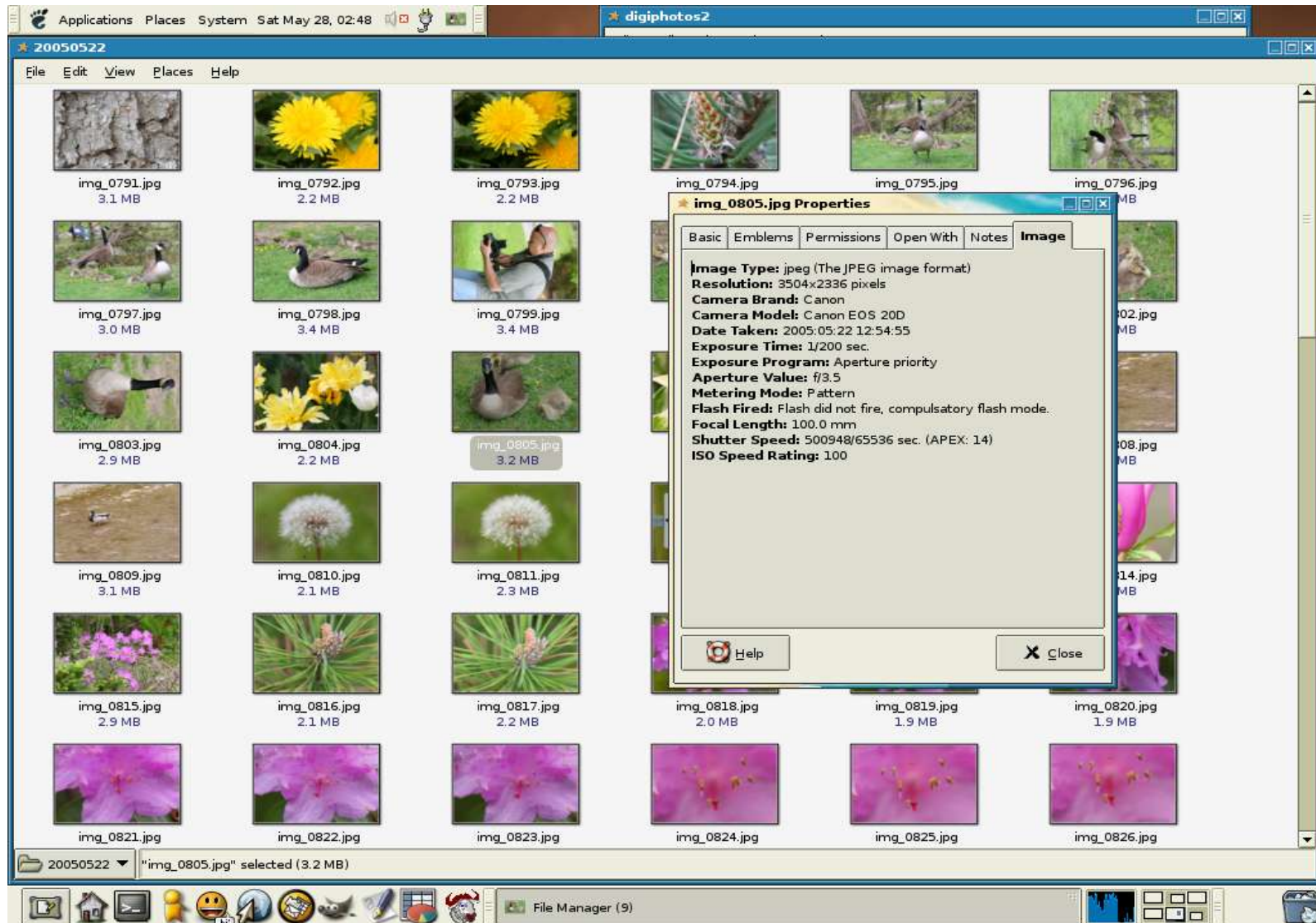
EOG: Screenshot



Nautilus

- Browsing files
- With Thumbnails
- Require and external application to view:
EOG by default

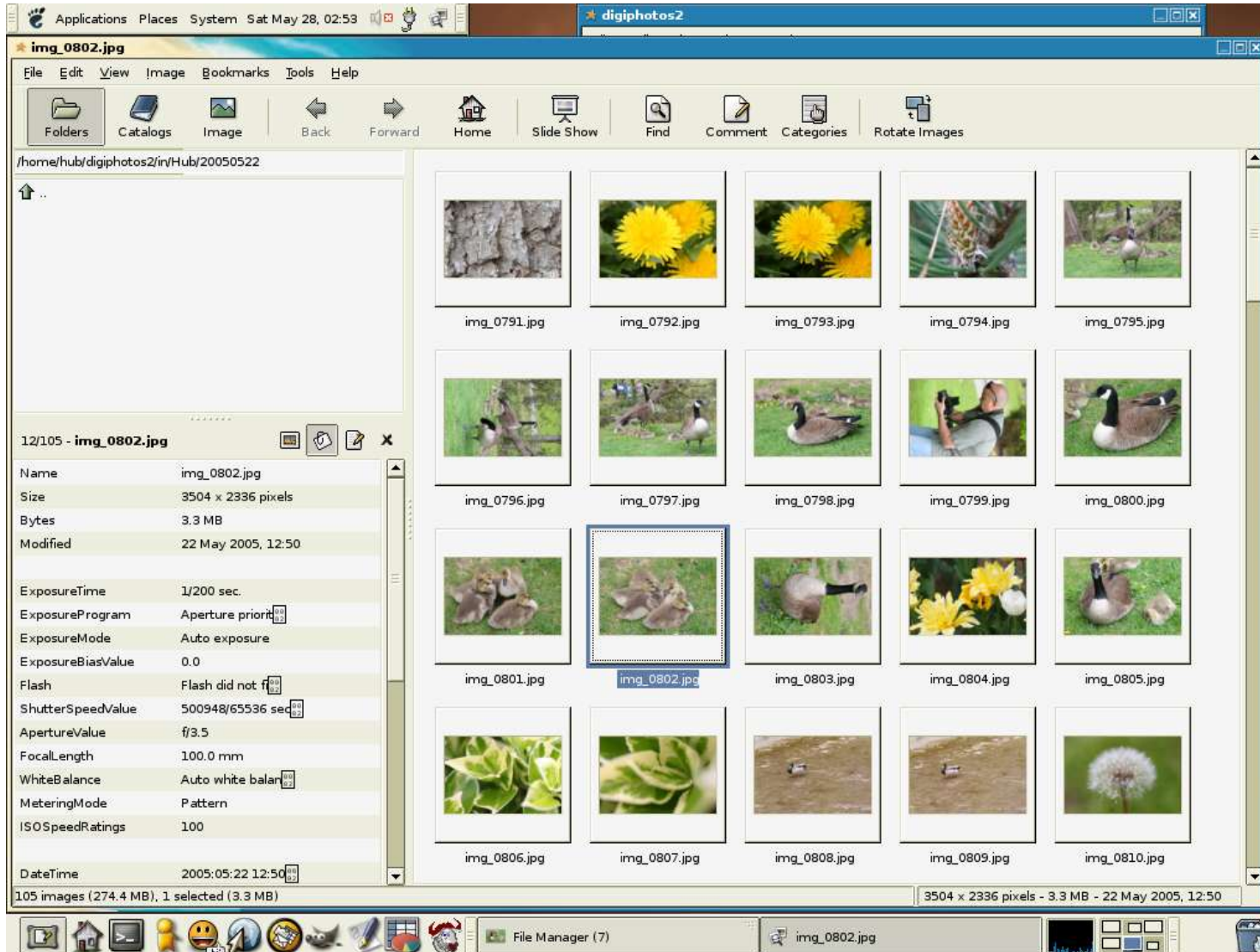
Nautilus: Screenshot



gThumb

- Thumbnailing
- Viewing
- EXIF support
- Slideshows
- Import from camera
- Commenting
- Albums

gThumb: screenshot



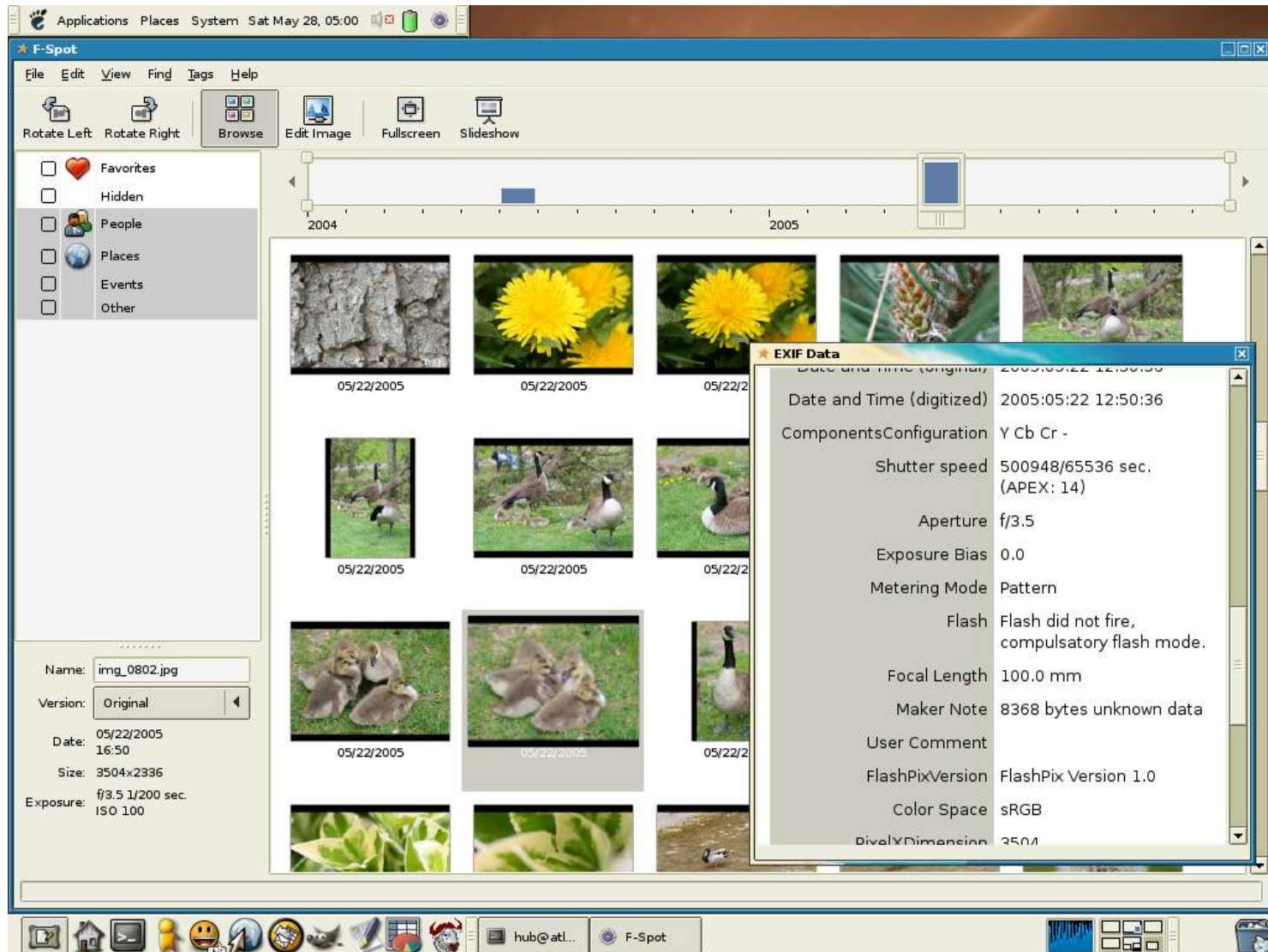
f-Spot

- Inspired by iPhoto
- Sorting
- Tagging
- EXIF support



f-spot

f-Spot: Screenshot



gThumb vs. f-spot

- Folder based
 - Catalog
 - Slower
 - Written in C: easier to install
 - A lot more editing tools
- Time based
 - Tagging
 - Faster
 - Written in C# and use Mono unstable
 - Red-eye removal

My preference?

- f-spot because
 - subjective speed
 - tagging
- What I miss
 - hierarchical tagging
 - IPTC support
 - MakerNotes decoding
 - grouping by folder or album (see DigiKam for KDE or Apple iPhoto)

Part 4

Decoding

Decoding pictures



- JPEG, no problem
- RAW files: the headache
- Metadatas: EXIF
MakerNotes

JPEG

- JPEG is an open format
- Works everywhere
- No issues but quality loss



RAW files

- Proprietary
 - Specific to a camera
 - No documentation
- The best for quality and post processing
- Not usable directly



RAW Files

- dcrw
- gimp-dcrw
- UFRaw
- Bibble Pro (proprietary)



Copyright 2005 <chifquiere@lecase.fr>

dcraw

- Command line
- The “reference”
used by everyone
 - Adobe (decoding)
 - iPhoto
- Only outputs 8-bits
and 16-bits linear
PPM and 16-bits
PSD (Photoshop) !



Copyright 2005 <hfiguiere@teaser.fr>

dcraw

Examples

Output 16-bits linear per channel PPM:

```
$ dcraw img_0123.cr2 img_0123.ppm
```

Output 8-bits per channel JPEG file:

```
$ dcraw -c imp_0123.cr2 | cjpeg img_0123.jpg
```

dcraw problems

- no meta data
- no color management
- not user friendly
- hard to use by other programs: they end up copying the code and not being up to date

gimp-dcraw

- simple gimp plugin for dcraw conversion
- still no meta data
- still no color management

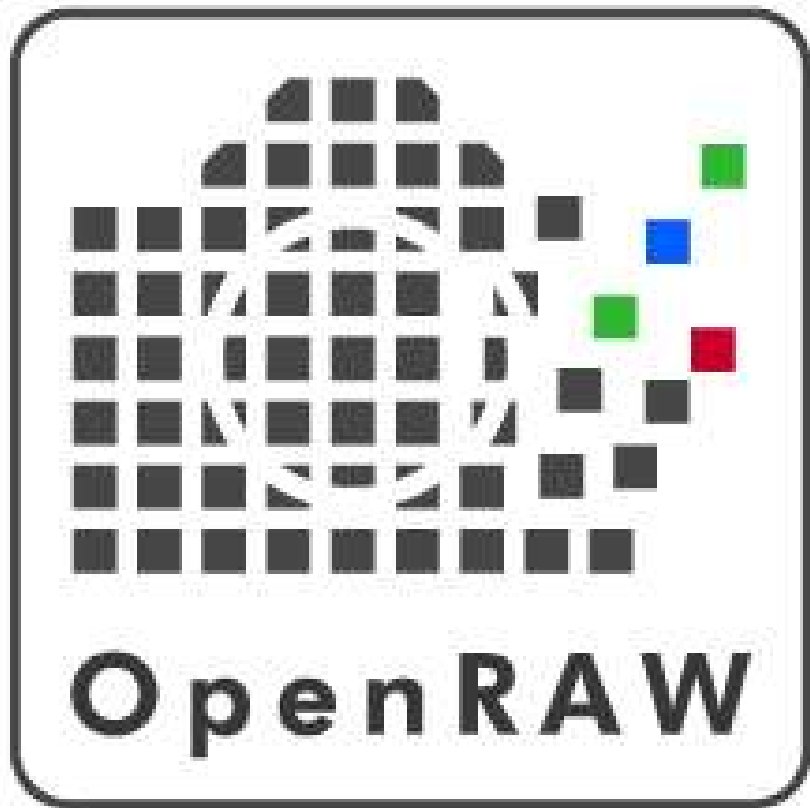
UFRaw

- much better UI
- color management with Little CMS
- still no meta-data
- standalone and GIMP plugin

Bibble Pro

- Proprietary Software for MacOS and Windows
- Also works on Linux i386
- Use dcraw
- Implements CMS
- Cost money but demo available

OpenRAW



- **Photographers** realized the importance of open file format
- Triggered by Nikon white balance “encryption” affair
- Open Source will benefit from it

Why do we need ?

- Reverse engineering takes a lot of time
- Results are not optimum
- Perenity of our pictures
 - there is no warranty they will support this format in 70 years with their proprietary software
 - there is no warranty that we can run today software in 70 years
 - we still can view pictures that are over 100 years old

Why don't manufacturer want ?

- Supposedly to protect their I.P. and camera design from competition
 - but since it has been reverse engineered, there is no point
- Because they don't have an open culture
- Because they want to control the market
 - Nikon even denied Adobe access to the file format as Nikon RAW decoding software cost additionnal money

Act Now!

- Request documentation to manufacturers
- <http://www.openraw.org/actnow/>

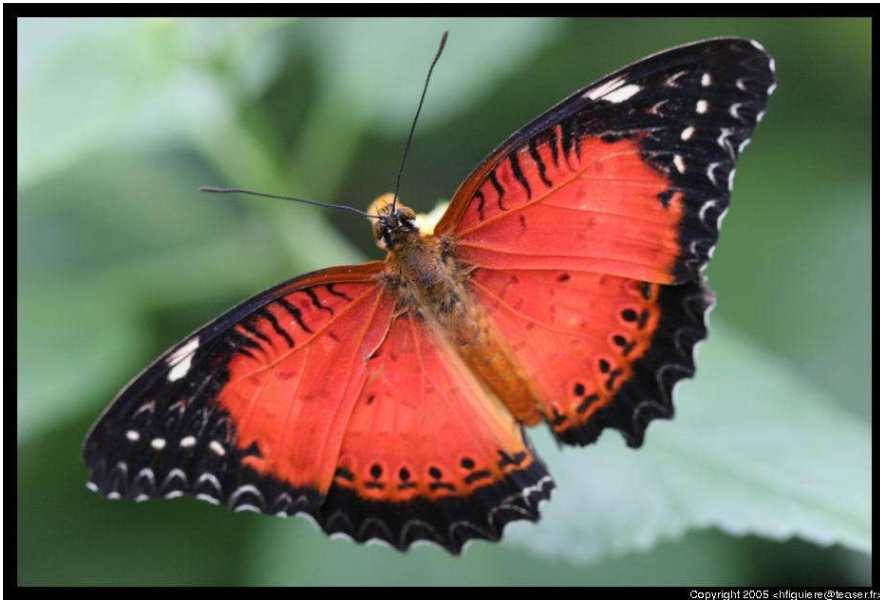


Part 5

Editing

Editing

- gimp



The Gimp



- Quite Complete and extensive photo editing
- Drawbacks
 - no CMS (currently in the work)
 - no 16-bits channels (require for optimum RAW processing)

Conclusion

- Solid foundation
- Great hackers
- Good apps
- But still lot of work to do
 - better file format support
 - better plug and play
 - better device support