Open source for securing data with advanced Crypto-Steganography technology

ClubHACK 2009

5th, 6th & 7th December, 2009 | Pune , INDIA

Suhas Desai

About me

- **Security analyst**.
- Contributing writer - “**Linux Journal**”,” **LFY**”, “**Linux+”** magazines.
- Authored several research papers on RFID, Image processing and Linux security.
- Co-author - “**Security in Computing**” / Pearson Education [2010].
- Over **175** workshops across the globe to promote Linux and Open Source.
- Frequent speaker at prominent industry forums and conferences, has delivered noted sessions at **Universiti Sains Malaysia**; **OSSPAC'09** (Open source Singapore Pacific Asia Conference), Singapore and at **INTEROP** 2009, Mumbai.

Agenda

- Crypto-steganography Overview
- Open Source role
- Python to achieve Security

Promotion of Open Source technologies to secure data!

Agenda

- Crypto-steganography Overview
- Open Source role
- Python to achieve Security

Introduction

Steganography and Cryptography are two important technologies used to secure data. It has gained major attention since Second World War. In Second World War it has been widely used to hide and send sensitive information of military operations.

Cryptography is art of writing secret code. Cryptography is combination of ‘Crypto’ and ‘graphy’ words. Crypto means ‘secret’ and graphy means ‘art of’.

Steganography is the art of hiding information in images. In Steganography, confidential data is hidden in images to protect it from unauthorized users. Steganography means “covered writing” in Greek.

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, Steganography focuses on keeping the existence of message secret .The strength of Steganography can thus be amplified by combining it with Cryptography.

Agenda

- Crypto-steganography Overview
- Open Source role
- Python to achieve Security

Open Source Role

- Source code available

- Easy to customize , code reuse and redistributable.

- Cost Savings

Agenda

- Crypto-steganography Overview
- Open Source role
- Python to achieve Security

Why Python?

- Python plays vital role in information security.

Essentials :

- Python Image Library – Image processing Library from Python.
- EzPyCrypto - EzPyCrypto is used for cryptography.
- Stepic - Stepic is used for Steganography

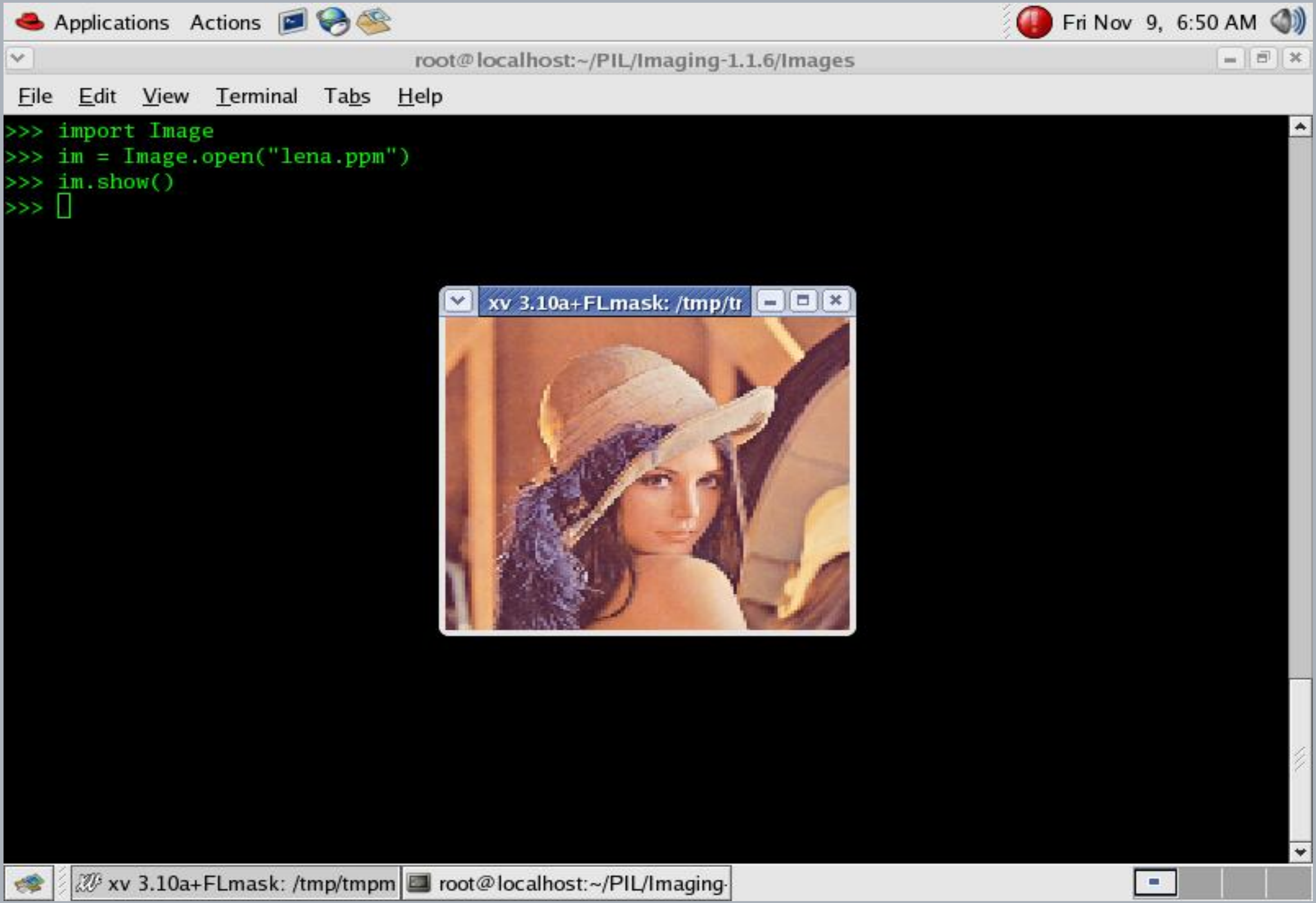
Python Image Library (PIL)

- Python interpreter is having various capabilities to perform image processing applications very efficiently.

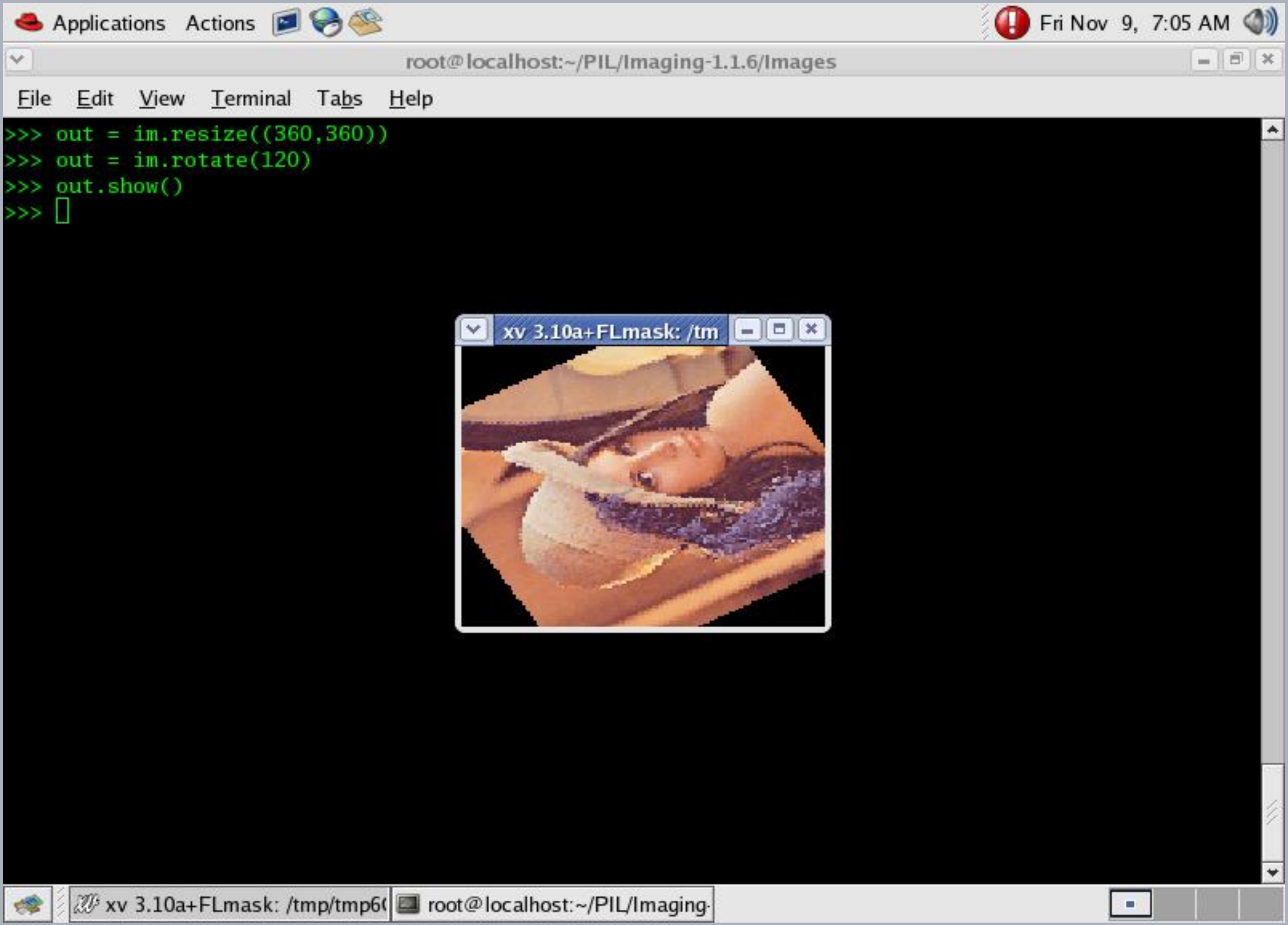
- Python, xv and the PIL package are essential packages to perform image processing in python. xv is an interactive image display for the X window system.

- Steps for Installation of python imaging library (PIL):

```
#gunzip Imaging-1.1.6.tar.gz
#tar xvf Imaging-1.1.6.tar
#cd Imaging-1.1.6
#python setup.py install
```

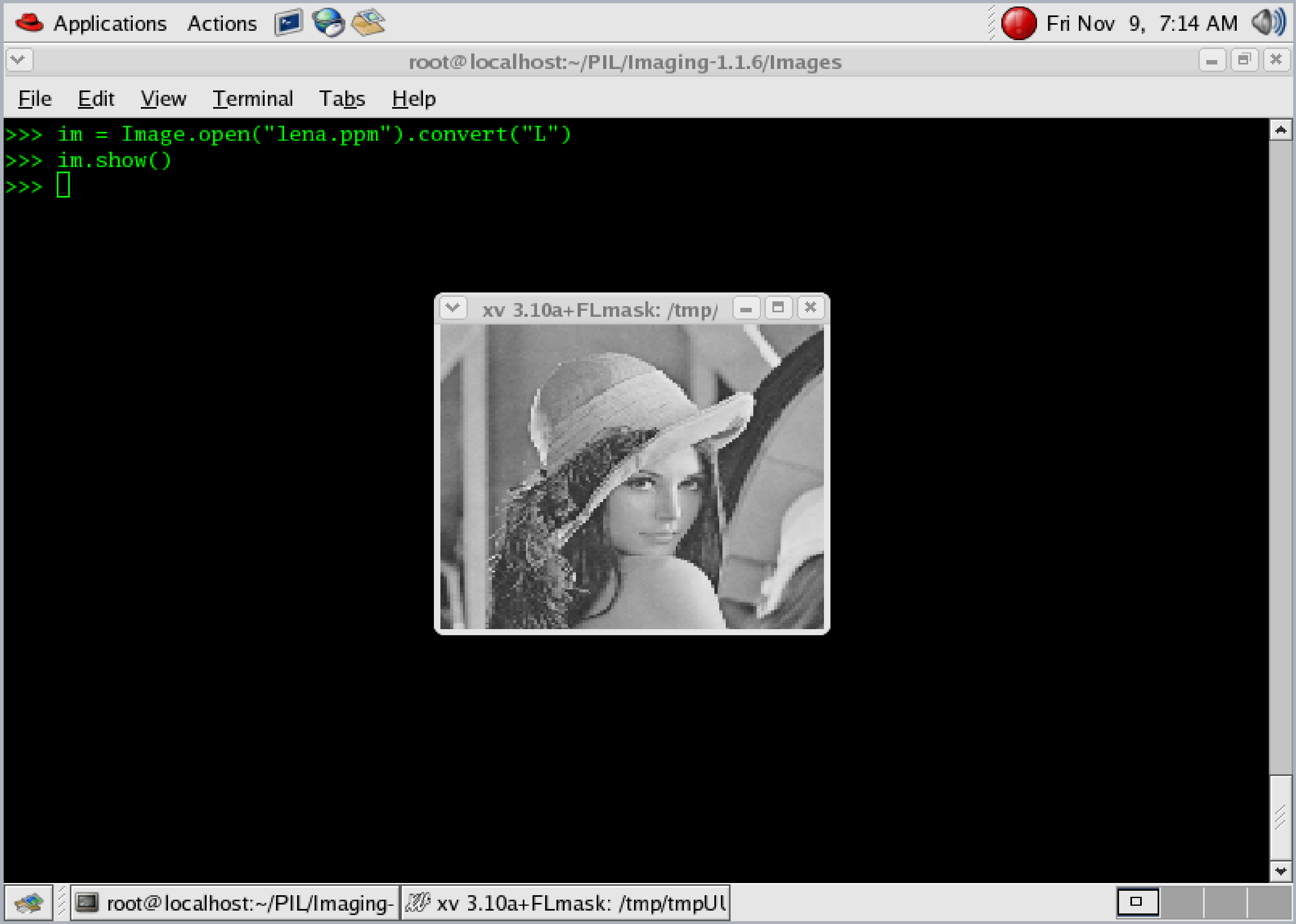

Applications
Actions
Fri Nov 9, 6:50 AM
root@localhost:~/PIL/Imaging-1.1.6/Images
File
Edit
View
Terminal
Tabs
Help
xv 3.10a+FLmask: /tmp/tı
xv 3.10a+FLmask: /tmp/tmpm
root@localhost:~/PIL/Imaging·

```
>>> import Image
>>> im = Image.open("lena.ppm")
>>> im.show()
>>> 
```



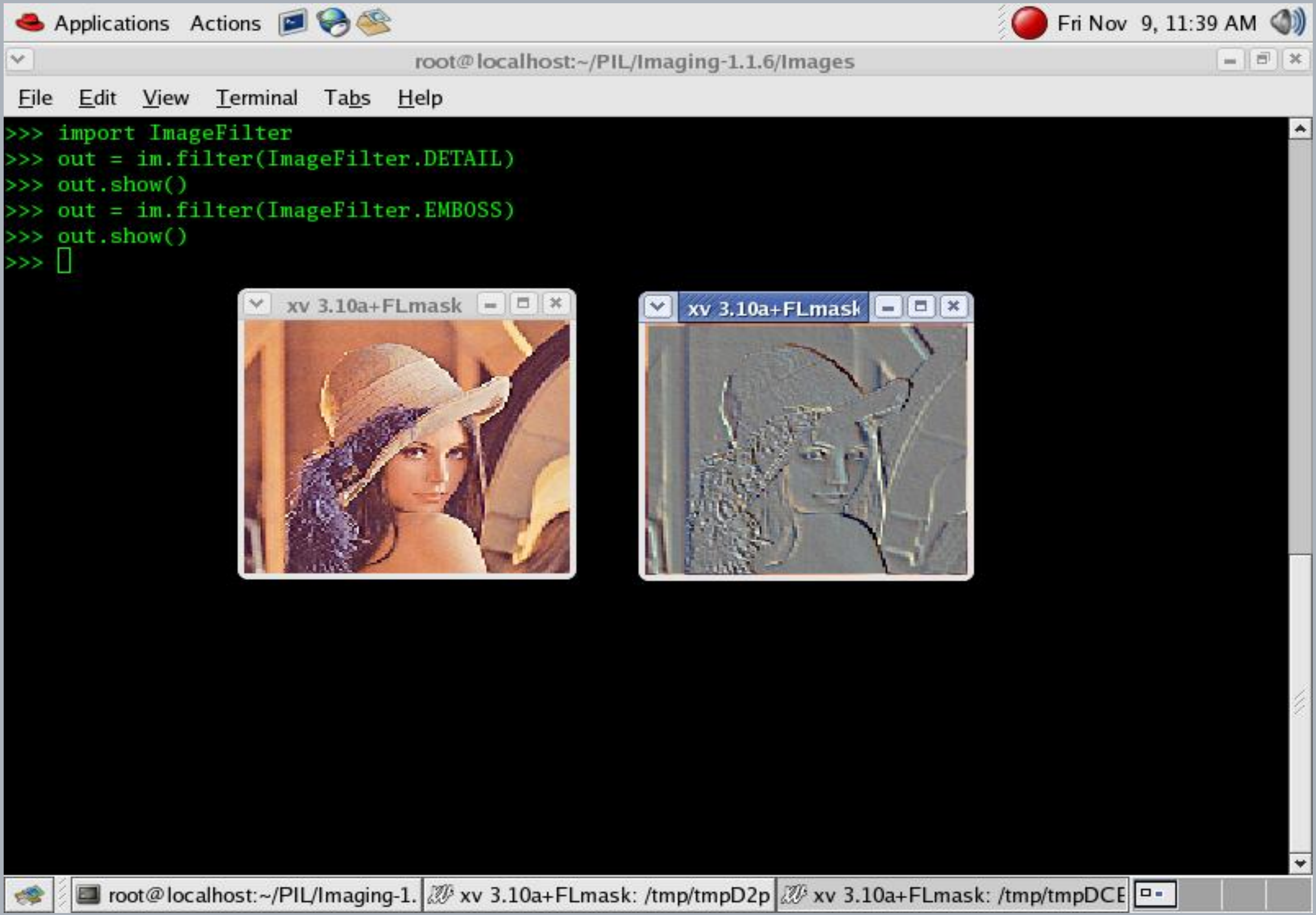
Applications
Actions
Fri Nov 9, 7:05 AM
root@localhost:~/PIL/Imaging-1.1.6/Images
File
Edit
View
Terminal
Tabs
Help
xv 3.10a+FLmask: /tm
xv 3.10a+FLmask: /tmp/tmp6(
root@localhost:~/PIL/Imaging-

```
>>> out = im.resize((360,360))
>>> out = im.rotate(120)
>>> out.show()
>>> 
```

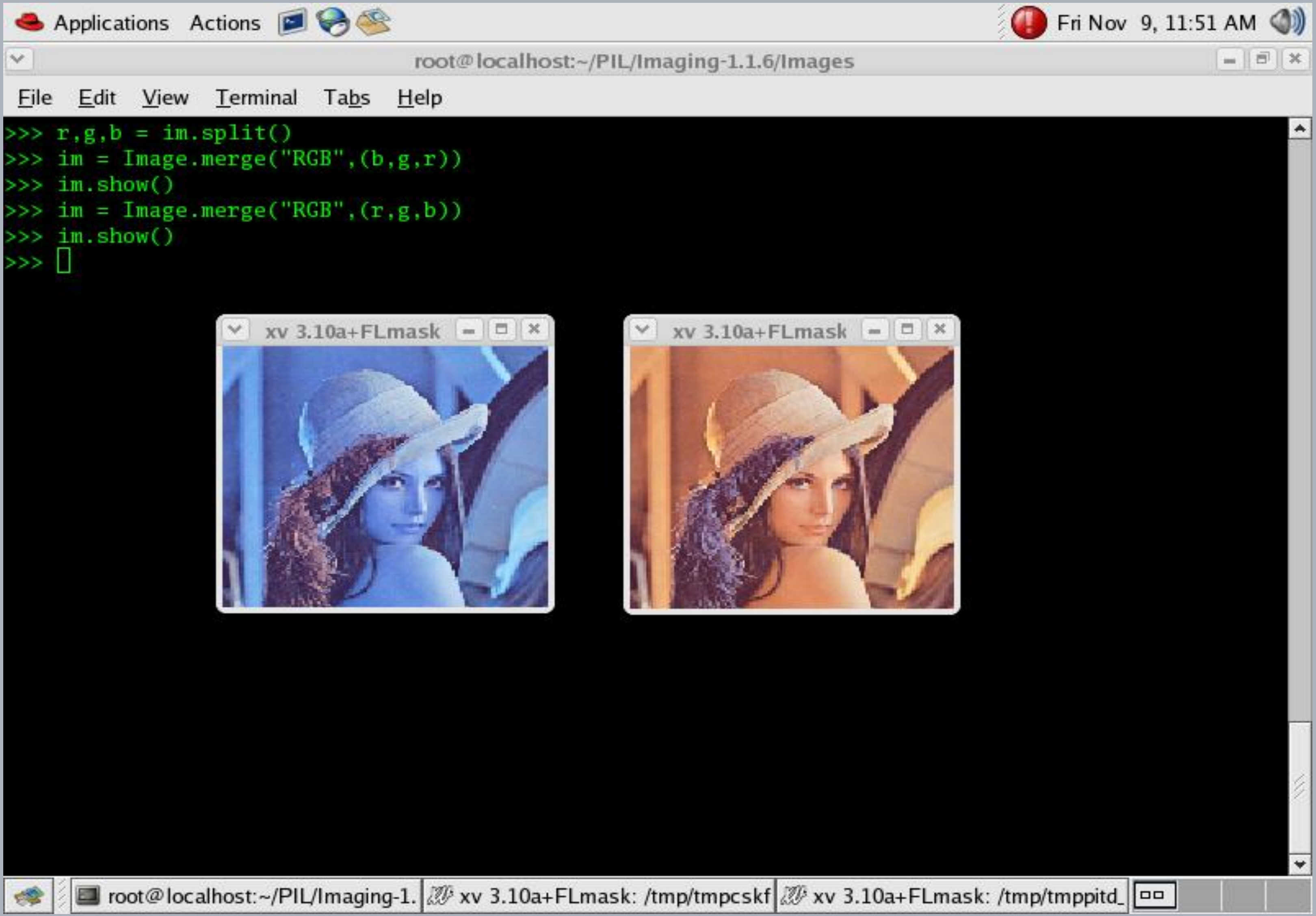


Applications
Actions
Fri Nov 9, 7:14 AM
root@localhost:~/PIL/Imaging-1.1.6/Images
File
Edit
View
Terminal
Tabs
Help
xv 3.10a+FLmask: /tmp/
root@localhost:~/PIL/Imaging-
xv 3.10a+FLmask: /tmp/tmpUl

```
>>> im = Image.open("lena.ppm").convert("L")
>>> im.show()
>>> 
```


Applications Actions
Fri Nov 9, 11:39 AM
root@localhost:~/PIL/Imaging-1.1.6/Images
File Edit View Terminal Tabs Help
xv 3.10a+FLmask
xv 3.10a+FLmask
root@localhost:~/PIL/Imaging-1.
xv 3.10a+FLmask: /tmp/tmpD2p
xv 3.10a+FLmask: /tmp/tmpDCE

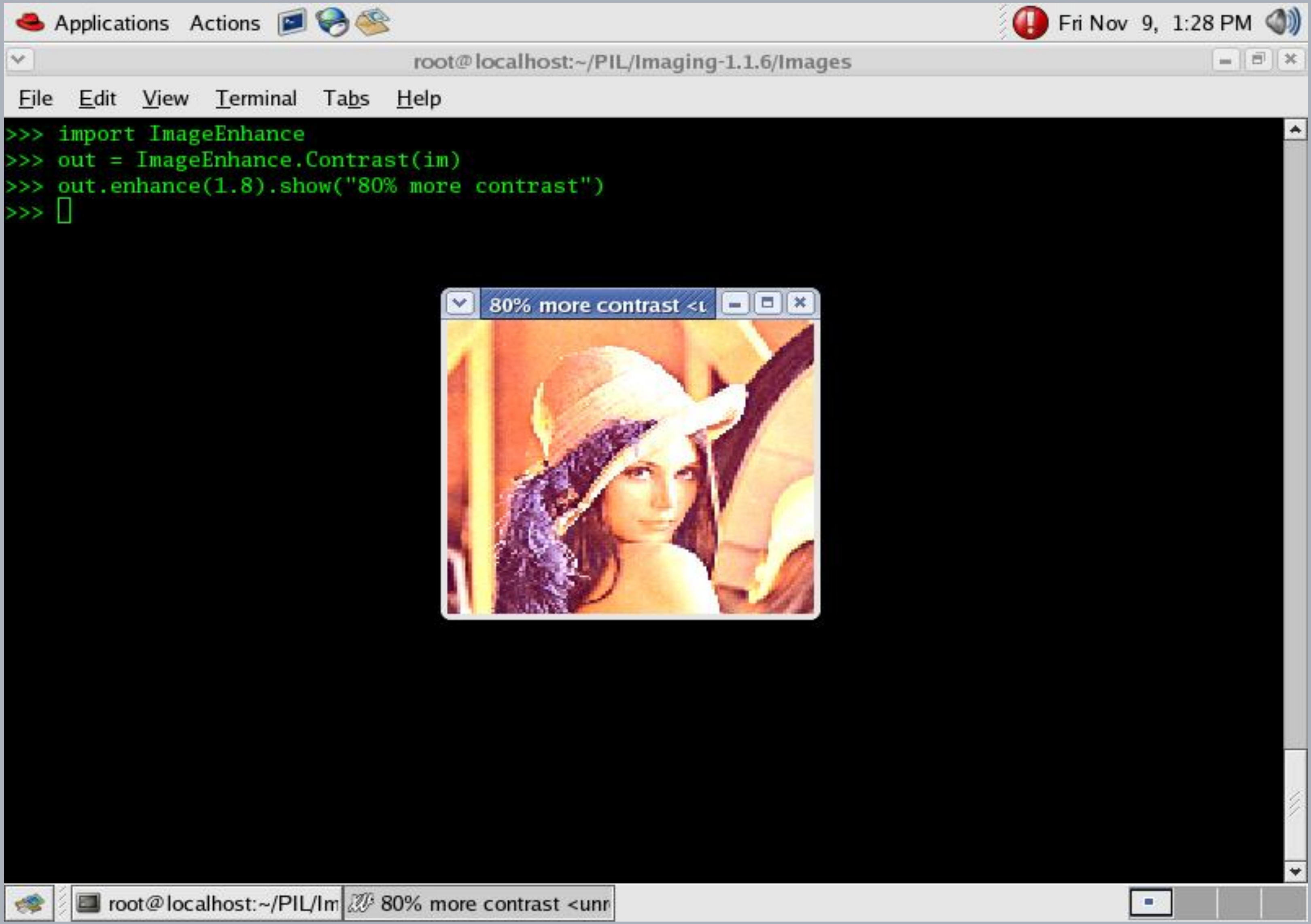
```
>>> import ImageFilter
>>> out = im.filter(ImageFilter.DETAIL)
>>> out.show()
>>> out = im.filter(ImageFilter.EMBOSS)
>>> out.show()
>>> 
```



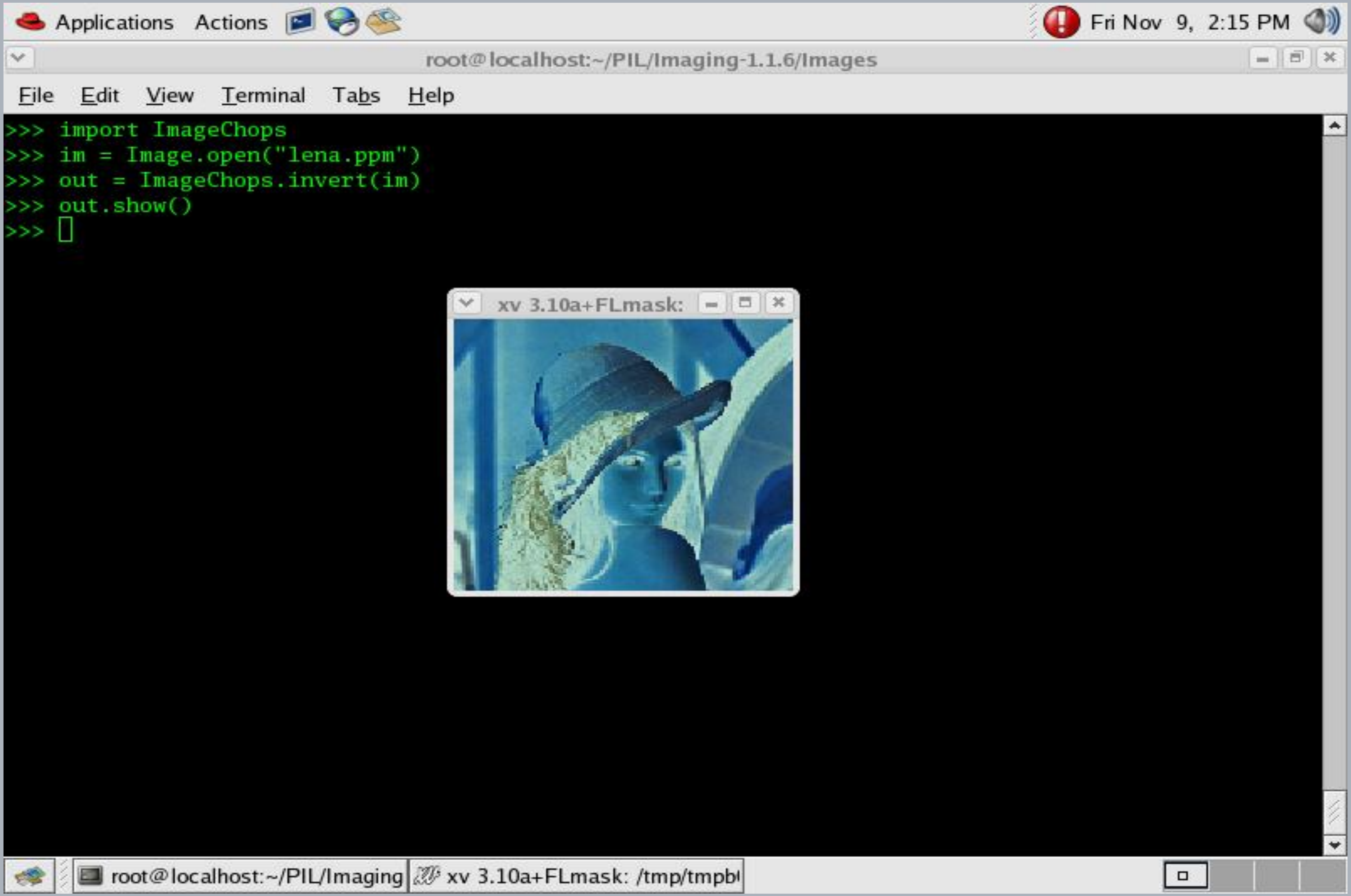
Applications
Actions
Fri Nov 9, 11:51 AM
root@localhost:~/PIL/Imaging-1.1.6/Images
File
Edit
View
Terminal
Tabs
Help

```
>>> r,g,b = im.split()
>>> im = Image.merge("RGB",(b,g,r))
>>> im.show()
>>> im = Image.merge("RGB",(r,g,b))
>>> im.show()
>>> 
```

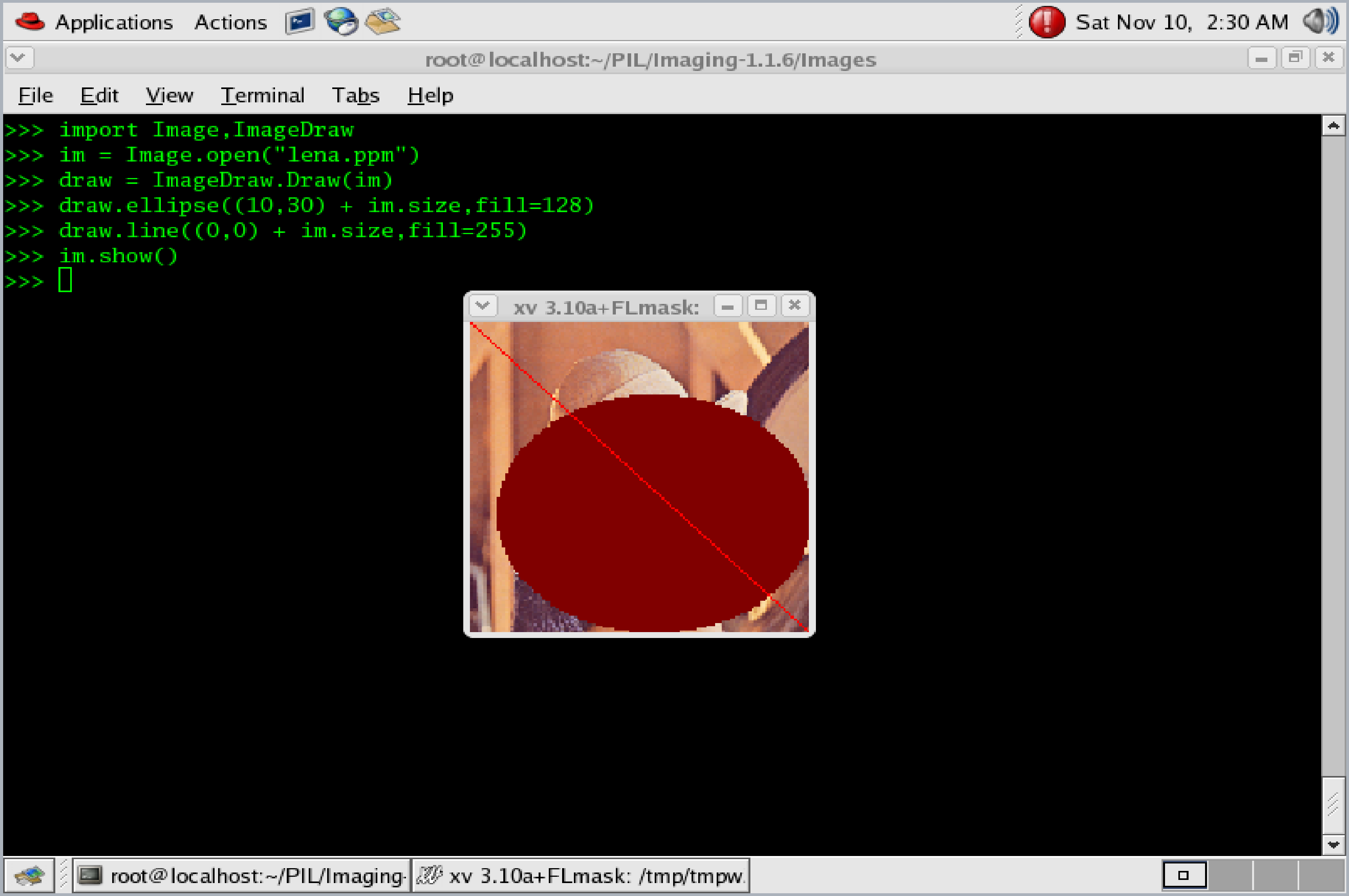
xv 3.10a+FLmask
xv 3.10a+FLmask
root@localhost:~/PIL/Imaging-1.
xv 3.10a+FLmask: /tmp/tmpcskf
xv 3.10a+FLmask: /tmp/tmppitd_



```
>>> import ImageEnhance
>>> out = ImageEnhance.Contrast(im)
>>> out.enhance(1.8).show("80% more contrast")
>>> 
```

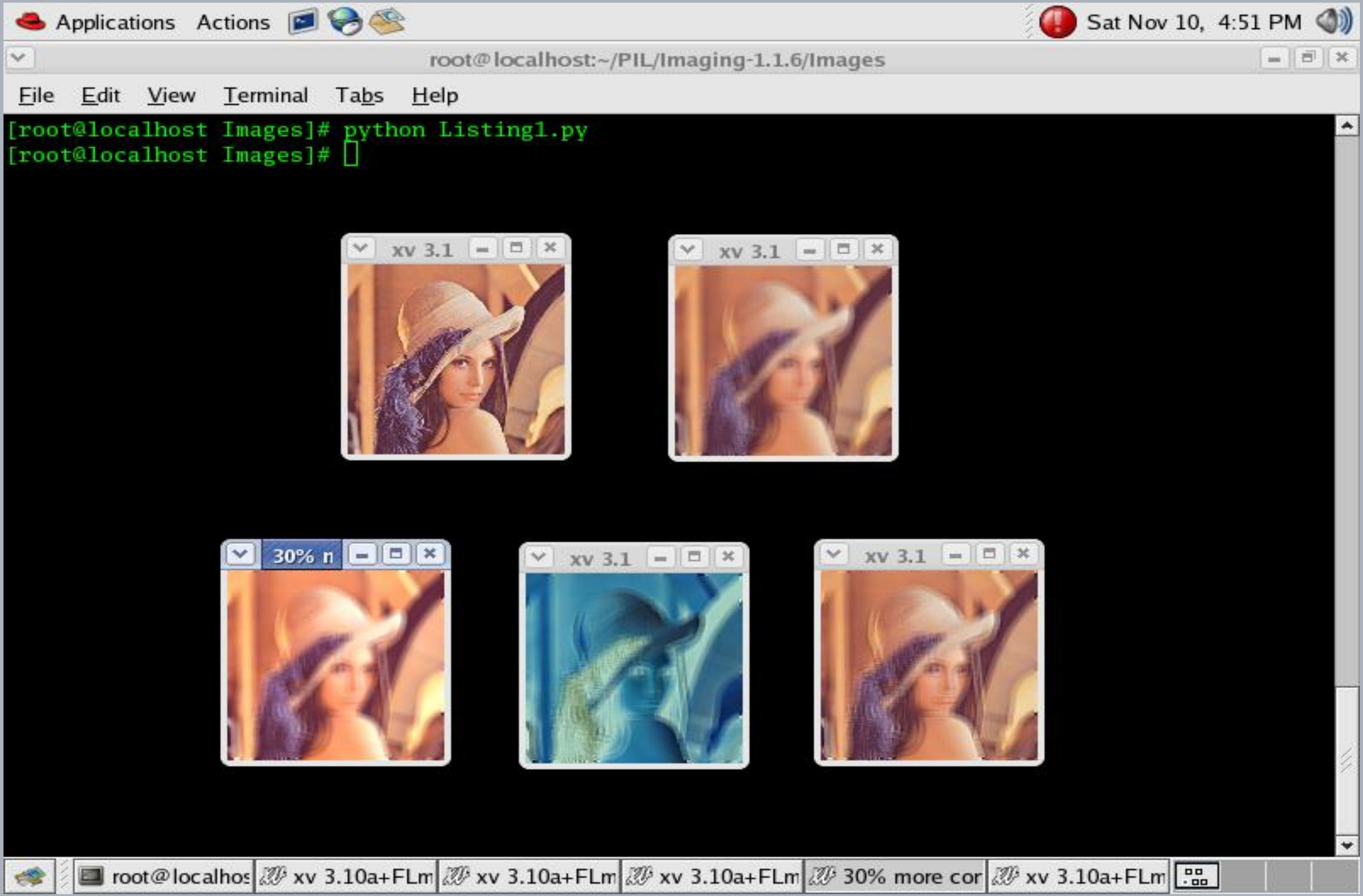

Applications
Actions
Fri Nov 9, 2:15 PM
root@localhost:~/PIL/Imaging-1.1.6/Images
File
Edit
View
Terminal
Tabs
Help
xv 3.10a+FLmask:
root@localhost:~/PIL/Imaging
xv 3.10a+FLmask: /tmp/tmpb

```
>>> import ImageChops
>>> im = Image.open("lena.ppm")
>>> out = ImageChops.invert(im)
>>> out.show()
>>> 
```



Applications
Actions
Sat Nov 10, 2:30 AM
root@localhost:~/PIL/Imaging-1.1.6/Images
File
Edit
View
Terminal
Tabs
Help
xv 3.10a+FLmask:
root@localhost:~/PIL/Imaging-
xv 3.10a+FLmask: /tmp/tmpw

```
>>> import Image,ImageDraw
>>> im = Image.open("lena.ppm")
>>> draw = ImageDraw.Draw(im)
>>> draw.ellipse((10,30) + im.size,fill=128)
>>> draw.line((0,0) + im.size,fill=255)
>>> im.show()
>>> 
```



```
[root@localhost Images]# python Listing1.py
[root@localhost Images]# 
```

What is ezPyCrypto class?

EzPyCrypto Overview

EzPyCrypto is very simple API for military-grade cryptography in Python. It encrypts and decrypts arbitrary-sized pieces of data like strings or files. EzPyCrypto class performs public and private key cryptography. You can use any size public key. Programming with EzPyCrypto class is relatively simple. You can import or export public and private keys also.

EzPyCrypto Setup

```
#tar xvf ezPyCrypto-0.1.1.tar
#cd ezPyCrypto-0.1.1
#python setup install
```

Methods of ezPyCrypto Class

- EzPyCrypto.key (number)-This method generates the key based on passed number.

- EncString (string)-This method encrypts the data or string which is passed to the method. It uses the key passed to above method. This method is called with key generated by ezPyCrypto.key() method.

- DecString (string)-This method decrypts the data. This method is called with key generated by ezPyCrypto.key() method. The string which we want to decrypt is passed to this method.

- EncStringToAscii (string)-This method encrypts the data using the key passed to ezPyCrypto class and stores it in ASCII format. This method is called with key generated by ezPyCrypto.key() method. The string which we want to encrypt is passed to this method.

- DecStringFromAscii (string)-This method decrypts the ASCII format data. This method is called with key generated by ezPyCrypto.key() method. The string which you may want to decrypt is passed to this method.

- There are also many more methods of ezPyCrypto class like signString (), verifyString (), makeNewKeys (), importkey (), exportkey() etc.

Stepic

Setup of Stepic and related packages:

To perform Steganography operation you need to use Stepic class of python. Following packages are essential to perform Steganography operations with Stepic class.

1. Imaging-1.1.6.tar.gz
2. stepic-0.3.tar.tar
3. ezPyCrypto-0.1.1.tar
4. libpng10-1.0.42-1.fc11.i386.rpm
5. xv-3.10a-13.i386.rpm

- Limitation of imaging packages are you cannot directly use show () method of python image class. This problem can overcome with xv utility to use show () method of python image class. For that it is required to install xv utility. Libpng10-1.0.42-1.fc11.i386.rpm and xv-3.10a-13.i386.rpm are the essential packages to get XV utility.

Steganography in Images using Stepic class of Python

Stepic is a new Python module and command line tool. It hides arbitrary data within images. Stepic is having a very simple behavior. Methods available in stepic class are easy to implement steganography. Stepic is having a disadvantage. It slightly modifies the colors of pixels in the image to hide the data. These modifications are imperceptible to humans. These minor modifications we can detect through programs.

Stepic encodes or hides text inside image and also decodes/extracts hidden text from the image. It allows storing the text or image data within an existing image without original image being affected. Stepic has very simple and easy implementation in python. But stepic doesn’t perform any encryption or compression of data while hiding it inside image. For that it is required to use ezPyCrypto tool with stepic.

Here we will see how to use this stepic for image steganography.

Steps for installation of stepic -

1. Install python imaging library (PIL)

2. Steps for installation of stepic 3.0

```
#tar xvf stepic-0.3.tar.tar
#cd stepic-0.3
#python setup.py install
```

Methods of stepic class

- Encode (string) - This method hides data inside image. This method is called with steganographer object. We can call this method directly by using stepic class.

- stepic.steganographer ( ) – It creates image object which is ready for undergoing steganography.

- decode ()-It extracts data from images. This method is also called with steganographer object

- Open (), save (), show () methods from image class of python are compatible with stepic class. After installation of stepic now you can develop your steganography application.

Encode or hide data inside an image

Import Image and stepic classes -

```
>>> import Image
>>> import stepic
```

Open an image in which you want to hide the data -

```
>>> im=Image.open ("lena.jpg")
```

Create steganographer object

```
>>> s = stepic.Steganographer(im)
```

You may get deprecation warning during steganographer call method at first time.

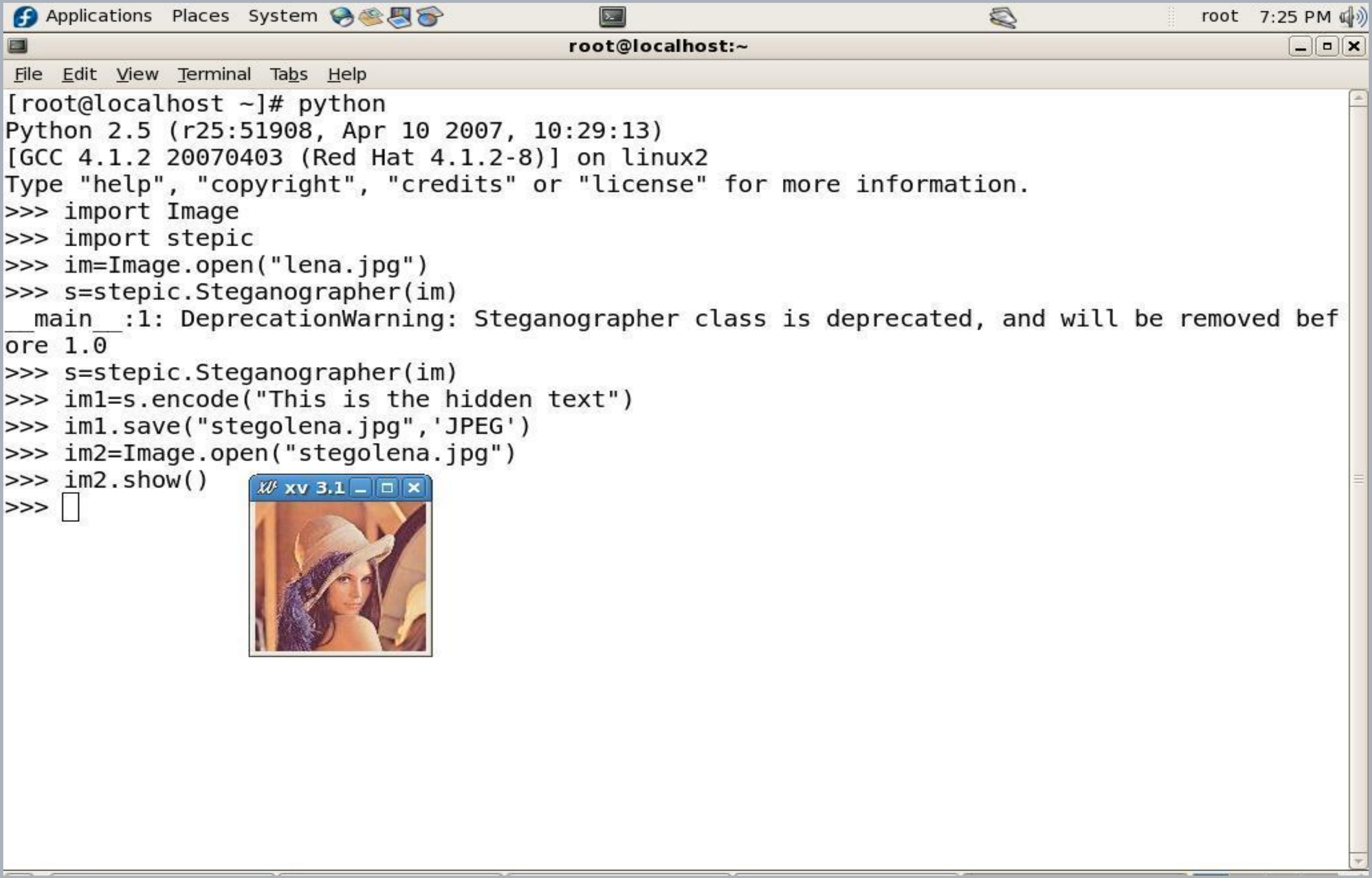
Use steganographer object to encode the data in some another object -

```
>>> im1=s.encode("This is the hidden text")
```

Save the data inside the image -

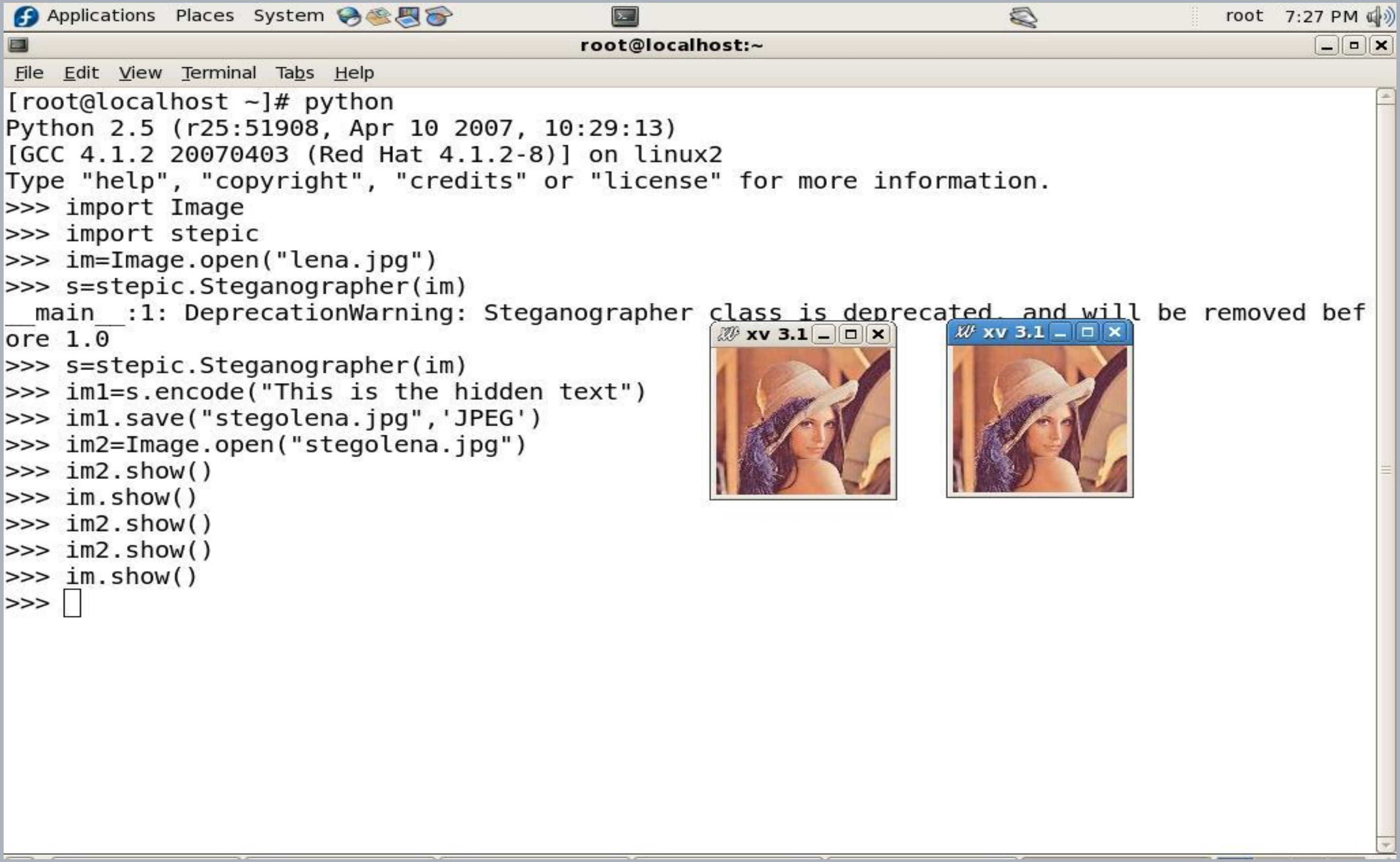
```
>>> im1.save ("stegolena.jpg",'JPFG')
```

Data Hiding



```
[root@localhost ~]# python
Python 2.5 (r25:51908, Apr 10 2007, 10:29:13)
[GCC 4.1.2 20070403 (Red Hat 4.1.2-8)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import Image
>>> import stepic
>>> im=Image.open("lena.jpg")
>>> s=stepic.Steganographer(im)
__main__:1: DeprecationWarning: Steganographer class is deprecated, and will be removed bef
ore 1.0
>>> s=stepic.Steganographer(im)
>>> im1=s.encode("This is the hidden text")
>>> im1.save("stegolena.jpg",'JPEG')
>>> im2=Image.open("stegolena.jpg")
>>> im2.show()
>>> 
```

Comparison: Original Image and Hidden Data Image



```
[root@localhost ~]# python
Python 2.5 (r25:51908, Apr 10 2007, 10:29:13)
[GCC 4.1.2 20070403 (Red Hat 4.1.2-8)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import Image
>>> import stepic
>>> im=Image.open("lena.jpg")
>>> s=stepic.Steganographer(im)
__main__:1: DeprecationWarning: Steganographer class is deprecated, and will be removed bef
ore 1.0
>>> s=stepic.Steganographer(im)
>>> im1=s.encode("This is the hidden text")
>>> im1.save("stegolena.jpg",'JPEG')
>>> im2=Image.open("stegolena.jpg")
>>> im2.show()
>>> im.show()
>>> im2.show()
>>> im2.show()
>>> im.show()
>>> 
```

Here, Instead of every time creating Steaganographer class instance, you can use stepic.encode() method directly for hiding the data.

```
>>> import Image
>>> import stepic
>>> im=Image.open("lena.jpg")
>>> im2=stepic.encode(im, ‘This is the hidden text’)
>>> im2.save('stegolena.jpg','JPEG')
```

Decoding or extracting hidden data from an Image

1.Use decode () function for decoding or extracting data from image.

```
>>> im1=Image.open(“stegolena.jpg”)
>>> s=stepic.decode(im1)
>>> data=s.decode()
```

2.Print the data

```
>>> print data
This is the hidden text
```

Combine ezPyCrypto with stepic class to hide encrypted data inside images

Instead of hiding plain data inside images, if you encrypt that data with some key and hide it inside image then that is more secure.

As stepic doesn’t support encryption or compression of data while hiding it inside images, you can use ezPyCrypto tool of python along with stepic class for hiding encrypted data inside images to obtain more security.

EzPyCrypto is the more powerful tool that we can combine with stepic. Other cryptography algorithm classes like md5, bz2 cannot work with stepic.

Stepic class hides data in ASCII format. So after encrypting data you have to convert it to ASCII format. You can use encStringToAscii () and encStringFromAscii() methods with stepic class to convert this data in ASCII format.

Encode or Hide Encrypted message inside image

1. Import Image, stepic and ezPyCrypto class in python

```
>>> import Image
>>> import stepic
>>> import ezPyCrypto
```

2. Now open an image in which you want to hide data

```
>>> im=Image.open("lena.jpg")
```

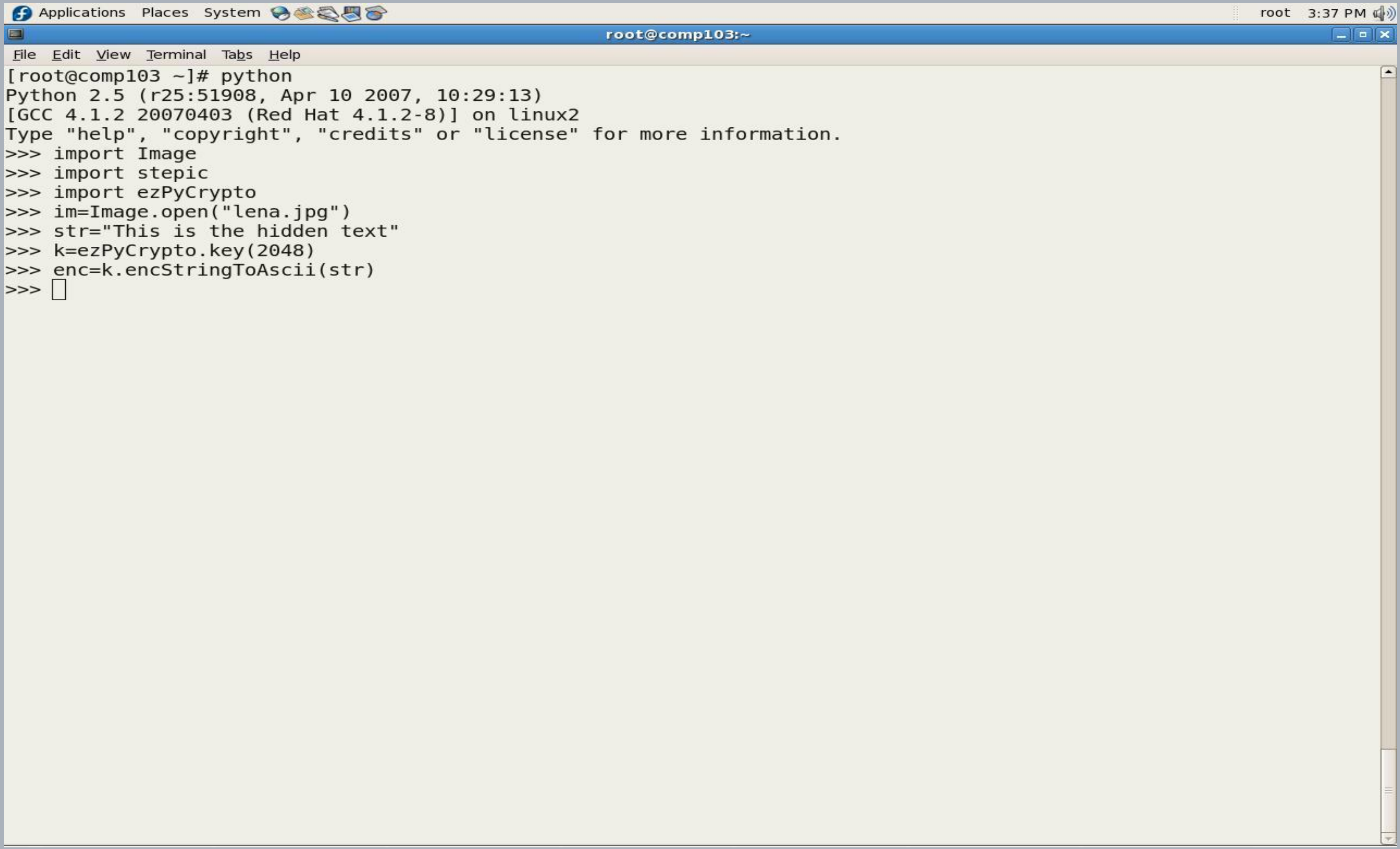
3. Create a key that you want to use for cryptography

```
>>>k=ezPyCrypto.key(2048)
```

4. Using key message is encrypted to ASCII format. This is shown in Figure 3 where actual data is encrypted.

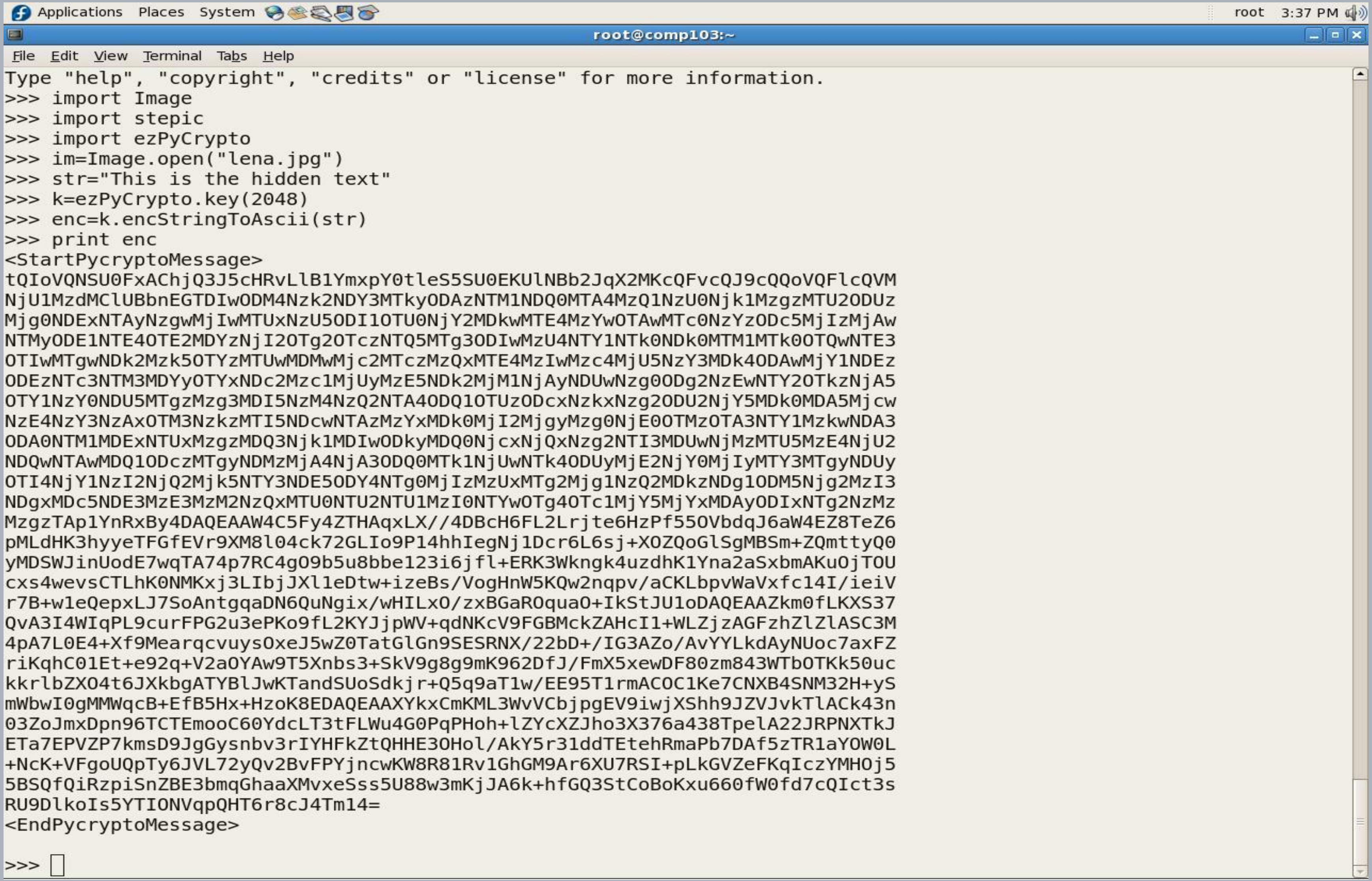
```
>>>enc=k.encStringToAscii(“This is the hidden text”)
```

Encryption using ezPyCrypto



```
[root@comp103 ~]# python
Python 2.5 (r25:51908, Apr 10 2007, 10:29:13)
[GCC 4.1.2 20070403 (Red Hat 4.1.2-8)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import Image
>>> import stepic
>>> import ezPyCrypto
>>> im=Image.open("lena.jpg")
>>> str="This is the hidden text"
>>> k=ezPyCrypto.key(2048)
>>> enc=k.encStringToAscii(str)
>>> 
```

Display encrypted data



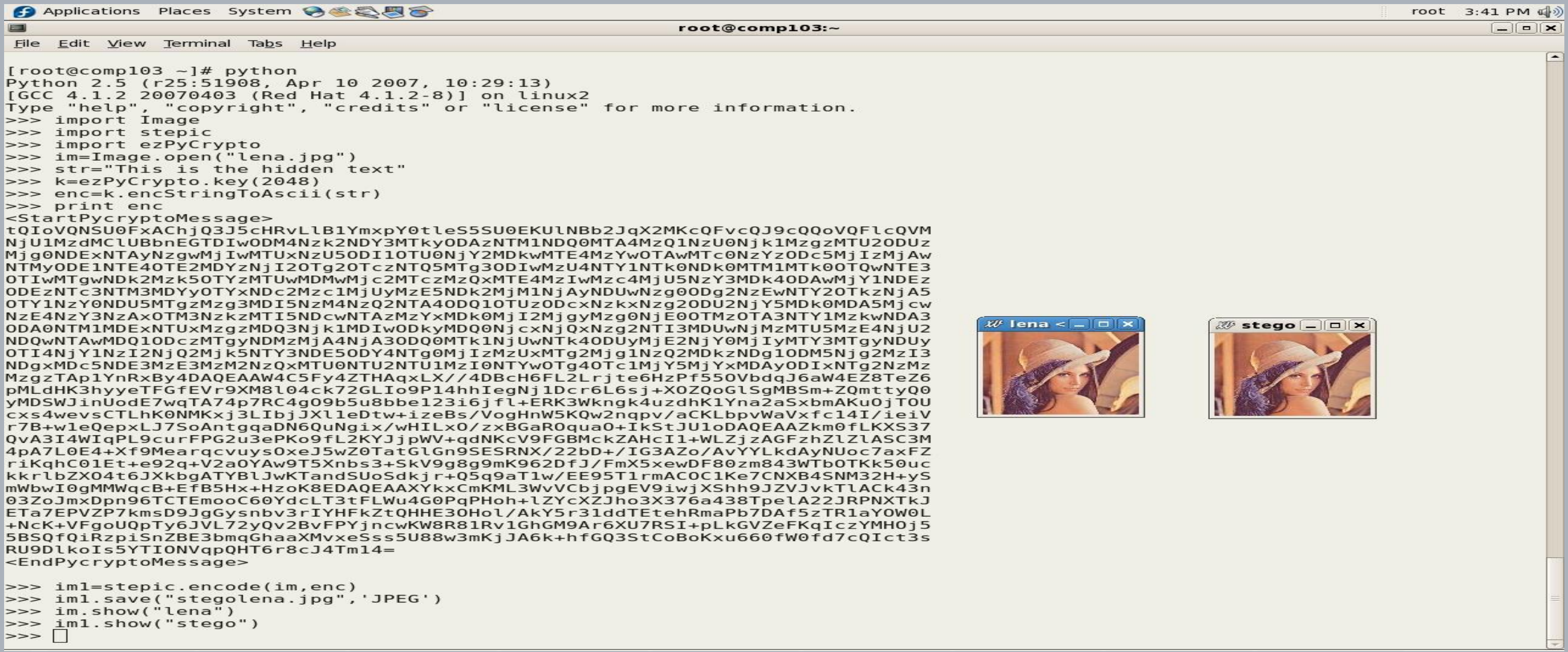
```
Type "help", "copyright", "credits" or "license" for more information.
>>> import Image
>>> import stepic
>>> import ezPyCrypto
>>> im=Image.open("lena.jpg")
>>> str="This is the hidden text"
>>> k=ezPyCrypto.key(2048)
>>> enc=k.encStringToAscii(str)
>>> print enc
<StartPycryptoMessage>
tQIoVQNSU0FxAChjQ3J5cHRvLlB1YmxpY0tleS5SU0EKUlNBb2JqX2MKcQFvcQJ9cQQoVQFlcQVM
NjU1MzdMClUBbnEGTDIwODM4Nzk2NDY3MTkyODAzNTM1NDQ0MTA4MzQ1NzU0Njk1MzgzMTU2ODUz
Mjg0NDExNTAyNzgwMjIwMTUxNzU5ODI1OTU0NjY2MDkwMTE4MzYwOTAwMTc0NzYzODc5MjIzMjAw
NTMyODE1NTE4OTE2MDYzNjI2OTg2OTczNTQ5MTg3ODIwMzU4NTY1NTk0NDk0MTM1MTk0OTQwNTE3
OTIwMTgwNDk2Mzk5OTYzMTUwMDMwMjc2MTczMzQxMTE4MzIwMzc4MjU5NzY3MDk4ODAwMjY1NDEz
ODEzNTc3NTM3MDYyOTYxNDc2Mzc1MjUyMzE5NDk2MjM1NjAyNDUwNzg0ODg2NzEwNTY2OTkzNjA5
OTY1NzY0NDU5MTgzMzg3MDI5NzM4NzQ2NTA4ODQ1OTUzODcxNzkxNzg2ODU2NjY5MDk0MDA5Mjcw
NzE4NzY3NzAxOTM3NzkzMTI5NDcwNTAzMzYxMDk0MjI2MjgyMzg0NjE0OTMzOTA3NTY1MzkwNDA3
ODA0NTM1MDExNTUxMzgzMDQ3Njk1MDIwODkyMDQ0NjcxNjQxNzg2NTI3MDUwNjMzMTU5MzE4NjU2
NDQwNTAwMDQ1ODczMTgyNDMzMjA4NjA3ODQ0MTk1NjUwNTk4ODUyMjE2NjY0MjIyMTY3MTgyNDUy
OTI4NjY1NzI2NjQ2Mjk5NTY3NDE5ODY4NTg0MjIzMzUxMTg2Mjg1NzQ2MDkzNDg1ODM5Njg2MzI3
NDgxMDc5NDE3MzE3MzM2NzQxMTU0NTU2NTU1MzI0NTYwOTg4OTc1MjY5MjYxMDAyODIxNTg2NzMz
MzgzTAp1YnRxBy4DAQEAAW4C5Fy4ZTHAqxLX//4DBcH6FL2Lrjte6HzPf55OVbdqJ6aW4EZ8TeZ6
pMLdHK3hyyeTFGfEVr9XM8l04ck72GLIo9P14hhIegNj1Dcr6L6sj+XOZQoGlSgMBSm+ZQmttyQ0
yMDSWJinUodE7wqTA74p7RC4gO9b5u8bbe123i6jfl+ERK3Wkngk4uzdhK1Yna2aSxbmAKuOjTOU
cxs4wevsCTLhK0NMKxj3LIbjJXl1eDtw+izeBs/VogHnW5KQw2nqpv/aCKLbpvWaVxfc14I/ieiV
r7B+w1eQepxLJ7SoAntgqaDN6QuNgix/wHILxO/zxBGaROquaO+IkStJU1oDAQEAAZkm0fLKXS37
QvA3I4WIqPL9curFPG2u3ePKo9fL2KYJjpWV+qdNKcV9FGBMckZAHcI1+WLZjzAGFzhZlZlASC3M
4pA7L0E4+Xf9MearqcvuysOxeJ5wZ0TatGlGn9SESRNX/22bD+/IG3AZo/AvYYLkdAyNUoc7axFZ
riKqhC01Et+e92q+V2aOYAw9T5Xnbs3+SkV9g8g9mK962DfJ/FmX5xewDF80zm843WTbOTKk50uc
kkrlbZXO4t6JXkbgATYBlJwKTandSUoSdkjr+Q5q9aT1w/EE95T1rmACOC1Ke7CNXB4SNM32H+yS
mWbwI0gMMWqcB+EfB5Hx+HzoK8EDAQEAAXYkxCmKML3WvVCbjpgEV9iwjXShh9JZVJvkTlACk43n
03ZoJmxDpn96TCTEmooC60YdcLT3tFLWu4G0PqPHoh+lZYcXZJho3X376a438TpelA22JRPNXTkJ
ETa7EPVZP7kmsD9JgGysnbv3rIYHFkZtQHHE3OHol/AkY5r31ddTEtehRmaPb7DAf5zTR1aYOW0L
+NcK+VFgoUQpTy6JVL72yQv2BvFPYjncwKW8R81Rv1GhGM9Ar6XU7RSI+pLkGVZeFKqIczYMHOj5
5BSQfQiRzpiSnZBE3bmqGhaaXMvxeSss5U88w3mKjJA6k+hfGQ3StCoBoKxu660fW0fd7cQIct3s
RU9DlkoIs5YTIONVqpQHT6r8cJ4Tm14=
<EndPycryptoMessage>

>>> 
```

Hide encrypted message inside image -

>>>im1=stepic.encode(enc)

Save this image with new name and show this new image -

>>>im1.save(“stegolena.jpg”,’JPEG’)

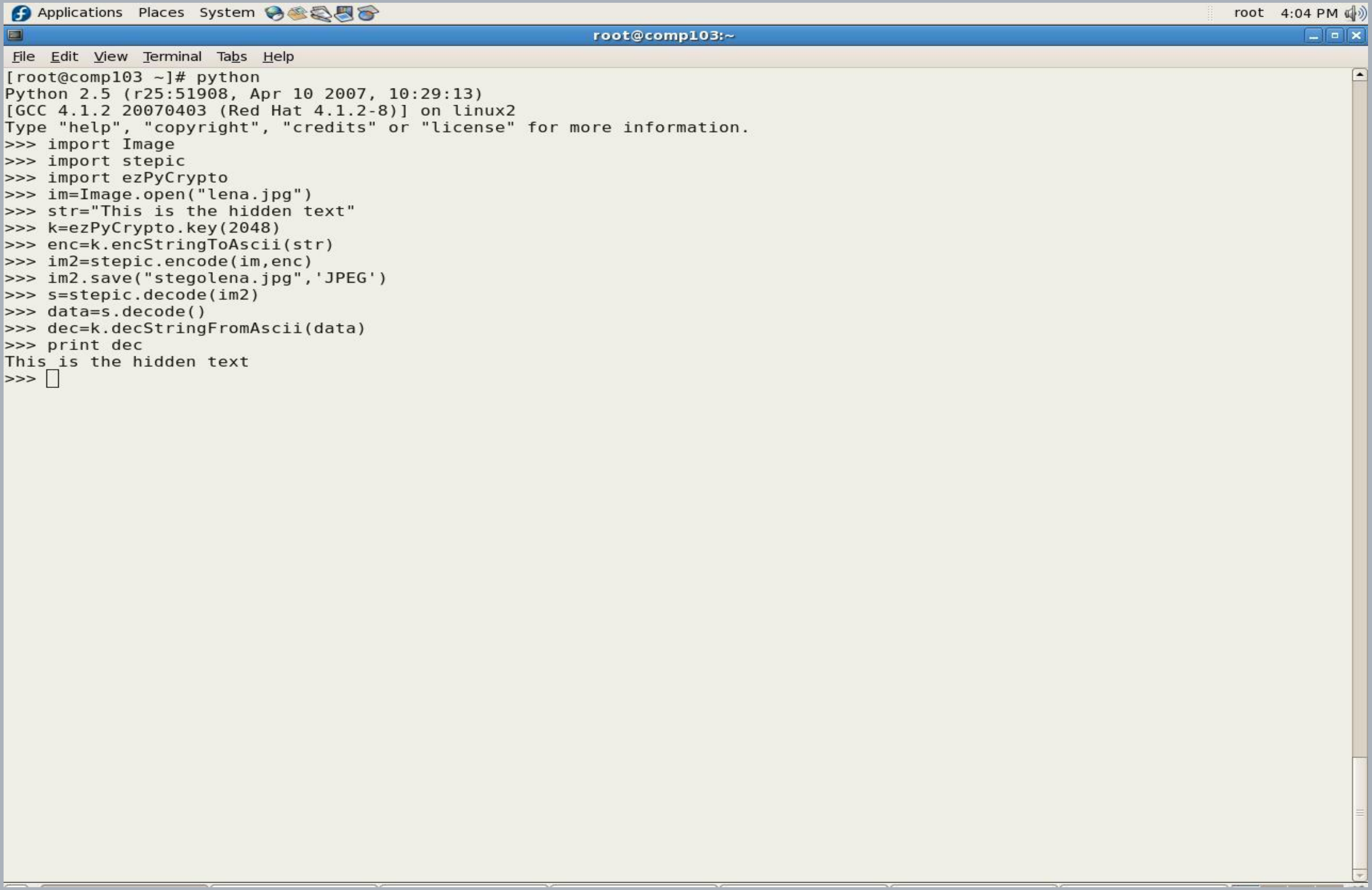


```
[root@comp103 ~]# python
Python 2.5 (r25:51908, Apr 10 2007, 10:29:13)
[GCC 4.1.2 20070403 (Red Hat 4.1.2-8)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import Image
>>> import stepic
>>> import ezPyCrypto
>>> im=Image.open("lena.jpg")
>>> str="This is the hidden text"
>>> k=ezPyCrypto.key(2048)
>>> enc=k.encStringToAscii(str)
>>> print enc
<StartPycryptoMessage>
tQIoVQNSU0FxAChjQ3J5cHRvLlB1YmxpY0tleS5SU0EKUlNBb2JqX2MKcQFvcQJ9cQQoVQFlcQVM
NjU1MzdMClUBbnEGTDIwODM4Nzk2NDY3MTkyODAzNTM1NDQ0MTA4MzQ1NzU0Njk1MzgzMTU2ODUz
Mjg0NDExNTAyNzgwMjIwMTUxNzU5ODI1OTU0NjY2MDkwMTE4MzYwOTAwMTc0NzYzODc5MjIzMjAw
NTMyODE1NTE4OTE2MDYzNjI2OTg2OTczNTQ5MTg3ODIwMzU4NTY1NTk0NDk0MTM1MTk0OTQwNTE3
OTIwMTgwNDk2Mzk5OTYzMTUwMDMwMjc2MTczMzQxMTE4MzIwMzc4MjU5NzY3MDk4ODAwMjY1NDEz
ODEzNTc3NTM3MDYyOTYxNDc2Mzc1MjUyMzE5NDk2MjM1NjAyNDUwNzg0ODg2NzEwNTY2OTkzNjA5
OTY1NzY0NDU5MTgzMzg3MDI5NzM4NzQ2NTA4ODQ1OTUzODcxNzkxNzg2ODU2NjY5MDk0MDA5Mjcw
NzE4NzY3NzAxOTM3NzkzMTI5NDcwNTAzMzYxMDk0MjI2MjgyMzg0NjE0OTMzOTA3NTY1MzkwNDA3
ODA0NTM1MDExNTUxMzgzMDQ3Njk1MDIwODkyMDQ0NjcxNjQxNzg2NTI3MDUwNjMzMTU5MzE4NjU2
NDQwNTAwMDQ1ODczMTgyNDMzMjA4NjA3ODQ0MTk1NjUwNTk4ODUyMjE2NjY0MjIyMTY3MTgyNDUy
OTI4NjY1NzI2NjQ2Mjk5NTY3NDE5ODY4NTg0MjIzMzUxMTg2Mjg1NzQ2MDkzNDg1ODM5Njg2MzI3
NDgxMDc5NDE3MzE3MzM2NzQxMTU0NTU2NTU1MzI0NTYwOTg4OTc1MjY5MjYxMDAyODIxNTg2NzMz
MzgzTAp1YnRxBy4DAQEAAW4C5Fy4ZTHAqxLX//4DBcH6FL2Lrjte6HzPf55OVbdqJ6aW4EZ8TeZ6
pMLdHK3hyyeTFGfEVr9XM8l04ck72GLIo9P14hhIegNj1Dcr6L6sj+XOZQoGlSgMBSm+ZQmttyQ0
yMDSWJinUodE7wqTA74p7RC4gO9b5u8bbe123i6jfl+ERK3Wkngk4uzdhK1Yna2aSxbmAKuOjTOU
cxs4wevsCTLhK0NMKxj3LIbjJXl1eDtw+izeBs/VogHnW5KQw2nqpv/aCKLbpvWaVxfc14I/ieiV
r7B+w1eQepxLJ7SoAntgqaDN6QuNgix/wHILxO/zxBGaROquaO+IkStJU1oDAQEAAZkm0fLKXS37
QvA3I4WIqPL9curFPG2u3ePKo9fL2KYJjpWV+qdNKcV9FGBMckZAHcI1+WLZjzAGFzhZlZlASC3M
4pA7L0E4+Xf9MearqcvuysOxeJ5wZ0TatGlGn9SESRNX/22bD+/IG3AZo/AvYYLkdAyNUoc7axFZ
riKqhC01Et+e92q+V2aOYAw9T5Xnbs3+SkV9g8g9mK962DfJ/FmX5xewDF80zm843WTbOTKk50uc
kkrlbZXO4t6JXkbgATYBlJwKTandSUoSdkjr+Q5q9aT1w/EE95T1rmACOC1Ke7CNXB4SNM32H+yS
mWbwI0gMMWqcB+EfB5Hx+HzoK8EDAQEAAXYkxCmKML3WvVCbjpgEV9iwjXShh9JZVJvkTlACk43n
03ZoJmxDpn96TCTEmooC60YdcLT3tFLWu4G0PqPHoh+lZYcXZJho3X376a438TpelA22JRPNXTkJ
ETa7EPVZP7kmsD9JgGysnbv3rIYHFkZtQHHE3OHol/AkY5r31ddTEtehRmaPb7DAf5zTR1aYOW0L
+NcK+VFgoUQpTy6JVL72yQv2BvFPYjncwKW8R81Rv1GhGM9Ar6XU7RSI+pLkGVZeFKqIczYMHOj5
5BSQfQiRzpiSnZBE3bmqGhaaXMvxeSss5U88w3mKjJA6k+hfGQ3StCoBoKxu660fW0fd7cQIct3s
RU9DlkoIs5YTIONVqpQHT6r8cJ4Tm14=
<EndPycryptoMessage>

>>> im1=stepic.encode(im,enc)
>>> im1.save("stegolena.jpg",'JPEG')
>>> im.show("lena")
>>> im1.show("stego")
>>> □
```

Decode or Extract Encrypted message from image and then decrypt it to original text.

```
//Decode/extract/decrypt encrypted messages Program.

>>> import Image
>>> import stepic
>>> import ezPyCrypto
>>> im=Image.open("stegolena.jpg")
>>> data=stepic.decode(im)   //Extract data from image
>>>k=ezPyCrypto(2048) //Create key (key should be same as key used at the time of encryption)
>>> dec=k.decStringFromAscii(data) //Decrypt the message using the key
>>>print dec //Print the message
This is the hidden text
```

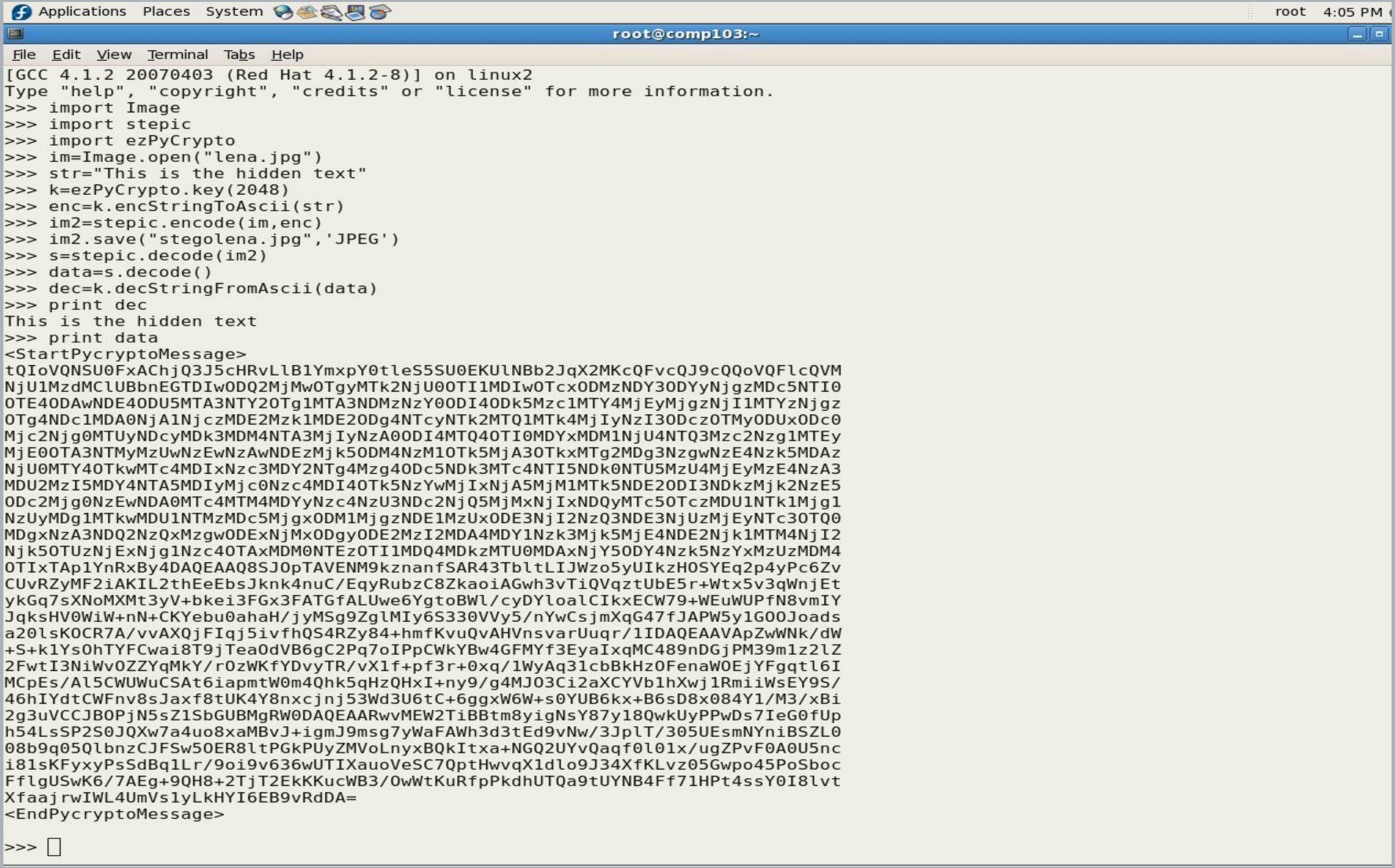
Decrypted data “This is the hidden text”.



```
[root@comp103 ~]# python
Python 2.5 (r25:51908, Apr 10 2007, 10:29:13)
[GCC 4.1.2 20070403 (Red Hat 4.1.2-8)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import Image
>>> import stepic
>>> import ezPyCrypto
>>> im=Image.open("lena.jpg")
>>> str="This is the hidden text"
>>> k=ezPyCrypto.key(2048)
>>> enc=k.encStringToAscii(str)
>>> im2=stepic.encode(im,enc)
>>> im2.save("stegolena.jpg",'JPEG')
>>> s=stepic.decode(im2)
>>> data=s.decode()
>>> dec=k.decStringFromAscii(data)
>>> print dec
This is the hidden text
>>> 
```

Now display data which is extracted from the image then it will displayed in encoded form as shown in Figure . To crack this encoded text you need to have a key that is used at the time of encryption of the message.

Display Extracted data without decryption



```
[GCC 4.1.2 20070403 (Red Hat 4.1.2-8)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import Image
>>> import stepic
>>> import ezPyCrypto
>>> im=Image.open("lena.jpg")
>>> str="This is the hidden text"
>>> k=ezPyCrypto.key(2048)
>>> enc=k.encStringToAscii(str)
>>> im2=stepic.encode(im,enc)
>>> im2.save("stegolena.jpg",'JPEG')
>>> s=stepic.decode(im2)
>>> data=s.decode()
>>> dec=k.decStringFromAscii(data)
>>> print dec
This is the hidden text
>>> print data
<StartPycryptoMessage>
tQIoVQNSU0FxAChjQ3J5cHRvLlB1YmxpY0tleS5SU0EKUlNBb2JqX2MKcQFvcQJ9cQQoVQFlcQVM
NjU1MzdMClUBbnEGTDIwODQ2MjMwOTgyMTk2NjU0OTI1MDIwOTcxODMzNDY3ODYyNjgzMDc5NTI0
OTE4ODAwNDE4ODU5MTA3NTY2OTg1MTA3NDMzNzY0ODI4ODk5Mzc1MTY4MjEyMjgzNjI1MTYzNjgz
OTg4NDc1MDA0NjA1NjczMDE2Mzk1MDE2ODg4NTcyNTk2MTQ1MTk4MjIyNzI3ODczOTMyODUxODc0
Mjc2Njg0MTUyNDcyMDk3MDM4NTA3MjIyNzA0ODI4MTQ4OTI0MDYxMDM1NjU4NTQ3Mzc2Nzg1MTEy
MjE0OTA3NTMyMzUwNzEwNzAwNDEzMjk5ODM4NzM1OTk5MjA3OTkxMTg2MDg3NzgwNzE4Nzk5MDAz
NjU0MTY4OTkwMTc4MDIxNzc3MDY2NTg4Mzg4ODc5NDk3MTc4NTI5NDk0NTU5MzU4MjEyMzE4NzA3
MDU2MzI5MDY4NTA5MDIyMjc0Nzc4MDI4OTk5NzYwMjIxNjA5MjM1MTk5NDE2ODI3NDkzMjk2NzE5
ODc2Mjg0NzEwNDA0MTc4MTM4MDYyNzc4NzU3NDc2NjQ5MjMxNjIxNDQyMTc5OTczMDU1NTk1Mjg1
NzUyMDg1MTkwMDU1NTMzMDc5MjgxODM1MjgzNDE1MzUxODE3NjI2NzQ3NDE3NjUzMjEyNTc3OTQ0
MDgxNzA3NDQ2NzQxMzgwODExNjMxODgyODE2MzI2MDA4MDY1Nzk3Mjk5MjE4NDE2Njk1MTM4NjI2
Njk5OTUzNjExNjg1Nzc4OTAxMDM0NTEzOTI1MDQ4MDkzMTU0MDAxNjY5ODY4Nzk5NzYxMzUzMDM4
OTIxTAp1YnRxBy4DAQEAAQ8SJOpTAVENM9kznanfSAR43TbltLIJWzo5yUIkzHOSYEq2p4yPc6Zv
CUvRZyMF2iAKIL2thEeEbsJknk4nuC/EqyRubzC8ZkaoiAGwh3vTiQVqztUbE5r+Wtx5v3qWnjEt
ykGq7sXNoMXMt3yV+bkei3FGx3FATGfALUwe6YgtoBWl/cyDYloalCIkxECW79+WEuWUPfN8vmIY
JqksHV0WiW+nN+CKYebu0ahaH/jyMSg9ZglMIy6S330VVy5/nYwCsjmXqG47fJAPW5y1GOOJoads
a20lsKOCR7A/vvAXQjFIqj5ivfhQS4RZy84+hmfKvuQvAHVnsvarUuqr/1IDAQEAAVApZwWNk/dW
+S+k1YsOhTYFCwai8T9jTeaOdVB6gC2Pq7oIPpCWkYBw4GFMYf3EyaIxqMC489nDGjPM39m1z2lZ
2FwtI3NiWvOZZYqMkY/rOzWKfYDvyTR/vX1f+pf3r+0xq/1WyAq31cbBkHzOFenaWOEjYFgqtl6I
MCpEs/Al5CWUWuCSAt6iapmtW0m4Qhk5qHzQHxI+ny9/g4MJO3Ci2aXCYVb1hXwj1RmiiWsEY9S/
46hIYdtCWFnv8sJaxf8tUK4Y8nxcjnj53Wd3U6tC+6ggxW6W+s0YUB6kx+B6sD8x084Y1/M3/xBi
2g3uVCCJBOPjN5sZ1SbGUBMgRW0DAQEAARwvMEW2TiBBtm8yigNsY87y18QwkUyPPwDs7IeG0fUp
h54LsSP2S0JQXw7a4uo8xaMBvJ+igmJ9msg7yWaFAWh3d3tEd9vNw/3JplT/305UEsmNYniBSZL0
08b9q05QlbnzCJFSw5OER8ltPGkPUyZMVoLnyxBQkItxa+NGQ2UYvQaqf0l01x/ugZPvF0A0U5nc
i81sKFyxyPsSdBq1Lr/9oi9v636wUTIXauoVeSC7QptHwvqX1dlo9J34XfKLvz05Gwpo45PoSboc
FflgUSwK6/7AEg+9QH8+2TjT2EkKKucWB3/OwWtKuRfpPkdhUTQa9tUYNB4Ff71HPt4ssY0I8lvt
XfaajrwIWL4UmVs1yLkHYI6EB9vRdDA=
<EndPycryptoMessage>

>>> 
```

Here secret key cryptography has been demonstrated. Same key is used at receiver and sender side. Similarly you can use public key cryptography using ezPyCrypto tool that is one common public key for encryption but different keys for decryption. EzPyCrypto is also useful to add digital signatures inside the image.

Summary

Stepic provides additional security by hiding the message inside images. You can encrypt the messages using ezPyCrypto tool and hide it inside image. This will be additional layer of security for the confidential data.

Enjoy the power of Stepic with ezPyCrypto!

References

Jain Ankit,” Steganography: A solution for data hiding”.

Robert Krenn,” Steganography Implementation & Detection”

Christian Cachin,” Digital Steganography”

James Madison,” An Overview of Steganography”

Neil F. Johnson,Sushil Jajodia, ”Exploring Steganography:Seeing the Unseen”

Max Weiss, “Principles of Steganography”

T. Morkel , J.H.P. Eloff , M.S. Olivier,” An Overview of Image Steganography”

Andreas Westfeld and Andreas Pfitzmann,” Attacks on Steganographic Systems”

www.python.org

http://domnit.org/stepic/doc/

http://www.freenet.org.nz/ezPyCrypto/

Thanks!

desai.suhas@gmail.com