THE CEG LUG TUX

The Open Source Magazine of CEG

Volume 1, Number 1 | Guindy, Chennai 600025 | Saturday, August 15, 2009

Adios! Amigos!! Welcome to the first edition of the online magazine of the Linux Users Group of College of Engineering, Guindy, Anna University. What actually started as a hobbyist touch to tamper and improvise the world of Linux in 2008, CEGLUG has blossomed into a professional venture vying against all odds to achieve perfection. This first edition is an earnest attempt to channelize talent against a deadline. It is in the hands of readers like you that these contributions realize their complete potential.

Knowledge is Open. It can by no means be put in a closed box to be put away for eternity. Rather, every time the box is opened, the quantum of Knowledge inside it doubles – obviously, due to the efforts of the person who utilizes it. We have just opened THE box for you and it is upto you to double its contents. So, use the Knowledge, double it, share it, give it back to the community you owe, your source to. The community of Open Source and the world of Linux.

Let me outline what this newsletter holds for you.

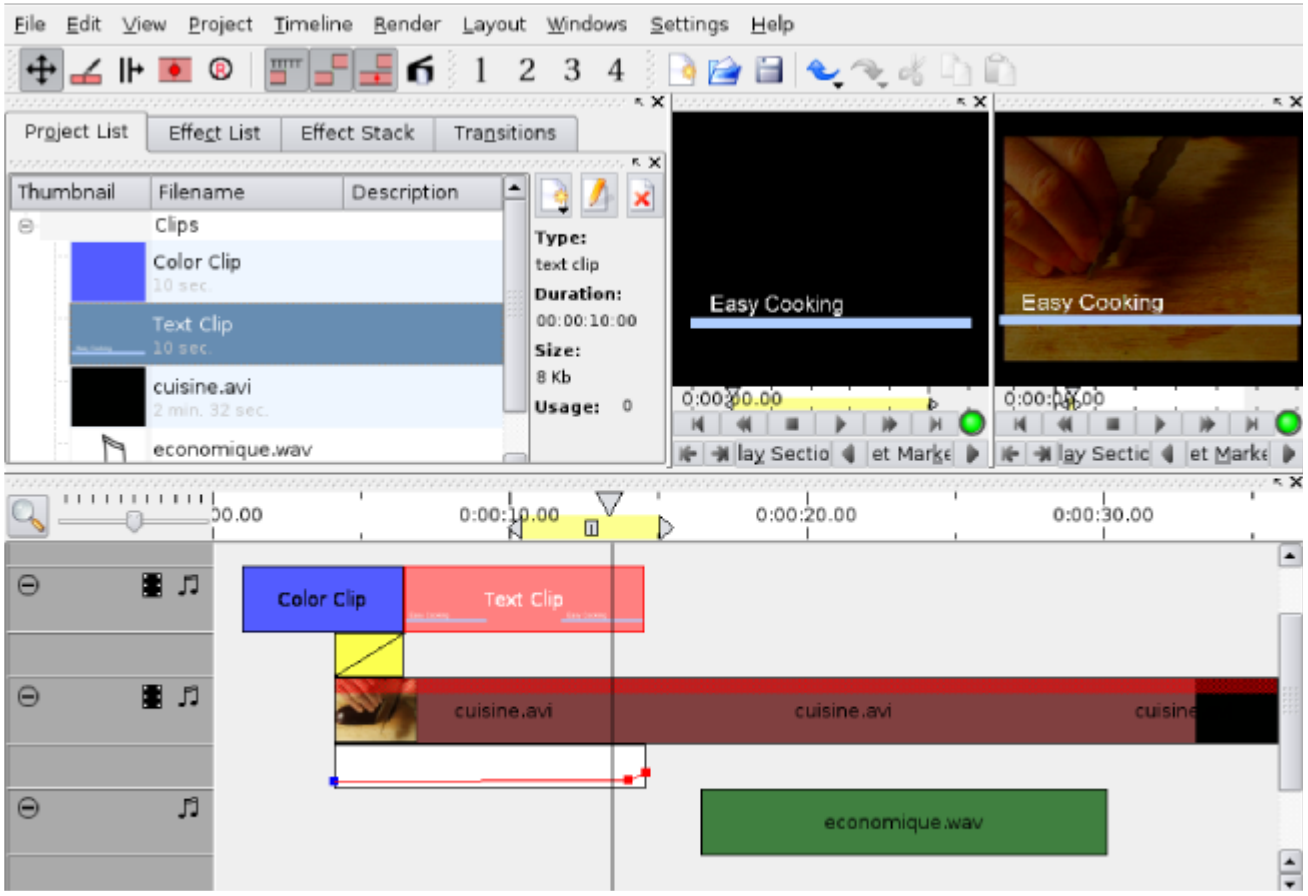
Reviews of the latest Linux OS releases; Open Source software; Tips to Tweak the Kernel; Fundas & a lot more. This first edition holds the first article in the "Python programming" series. The power of python, unleashed in simple words. Edit videos in Linux? We hand you "Kdenlive" – a review. Ever thought about a single authentication system for all your Logins? OpenId shows you the way. Kernel? Oh! Operating systems' base services. An article on kernel compilation. Take a little time off. Think open and complete this crossword. It is all in for you.

And what you can do to this newsletter.
Contribute in any form – Articles, Cartoons, Reviews – anything connected with open source and linux. You are in.

Mail to,

editor@ceglug.org.

Kdenlive, an elegant video editor



Pooja V. Rao, V Sem, CSE

Kdenlive is a video editing Graphical User Interface available in Linux, similar to Movie Maker in Windows. Though the early versions are prone to crashes on startup and there are no proper correction tools, the later releases handled the stability issues well. Much has to be emphasized about its splendid user interface. The only flaw in the later releases seems to be the not so user-friendly rendering of titles.

The basic features include the different transitions that could be added on to the video clips that are dragged and dropped into the timeline. To mention a few, they include the creative effects like embossing, glow and mirror. In addition, the multiple video and audio tracks could be organized in layers. There are also provisions to record from a web camera. The videos can be then exported in formats like mpeg2.

The Linux video editor would definitely be more sophisticated if there could be a preview window to view the transitions before including them in our videos. Moreover, the option of applying a setting to the entire video track could be introduced, thus moving to the zenith of the multi-track versatility and intuitiveness of Kdenlive.

Lets learn about the snakes in computer world

Venkatnathan.V, VII Sem, CSE

Programming with Python: Use the snake, to solve a problem - the easy way!

When I was stuck on what programming language is best suited for a project, my lecturer instructed me, “Why don’t you use Python?”, I went “Huh?” and then “What? Python?”

Most of us are familiar with the word C, C++, Java, etc. the common parlance of a Computer Engineer but, “Python”? Many are hardly aware of this very powerful programming language, without which, for instance, Google can’t run its “Bots” to crawl the Web.

So, here we are trying to understand what is Python and what is it capable of? Lets get into some details of what is python.

Python is an interpreted, interactive, object-oriented programming language.

Interpreted?
We all know what type of programming language C is. Lets go from C to Python, by stating the difference between them. The things you do to write a C program and execute it are:
Firstly, you write the code, called source, then you subject this code to a process called Compiling, using the cc or gcc command. What results is an executable, which is by default a.out. You then try to execute the executable with ./a.out command in the terminal.

Here, I like to point out some significant observations: 1) Can you execute without compiling your source code? Or 2) Can you execute a statement without completing the whole source code?

This is where the power of an interpreted language wins! In Python, and with any other interpreted language, a statement is interpreted (and not compiled?) and executed as and when it is reached. That is compilation is over a complete source code, whereas interpretation is over each and every statement.

That clears us with the first one, two more to go but, let me tell you a small anecdote on the origin of the name Python. Father of Python, Guido van Rossum, was thinking over a name for his new programming language. He wanted it to be short, unique and to some extent mysterious. He was also going through a sketch comedy series by BBC called “Monty Python’s Flying Circus”. So he simply called it Python. When I heard this I went agape, “Is that it?” Funny reason behind such a powerful and widely used language, don’t you think?

Lets move on to the next characteristic

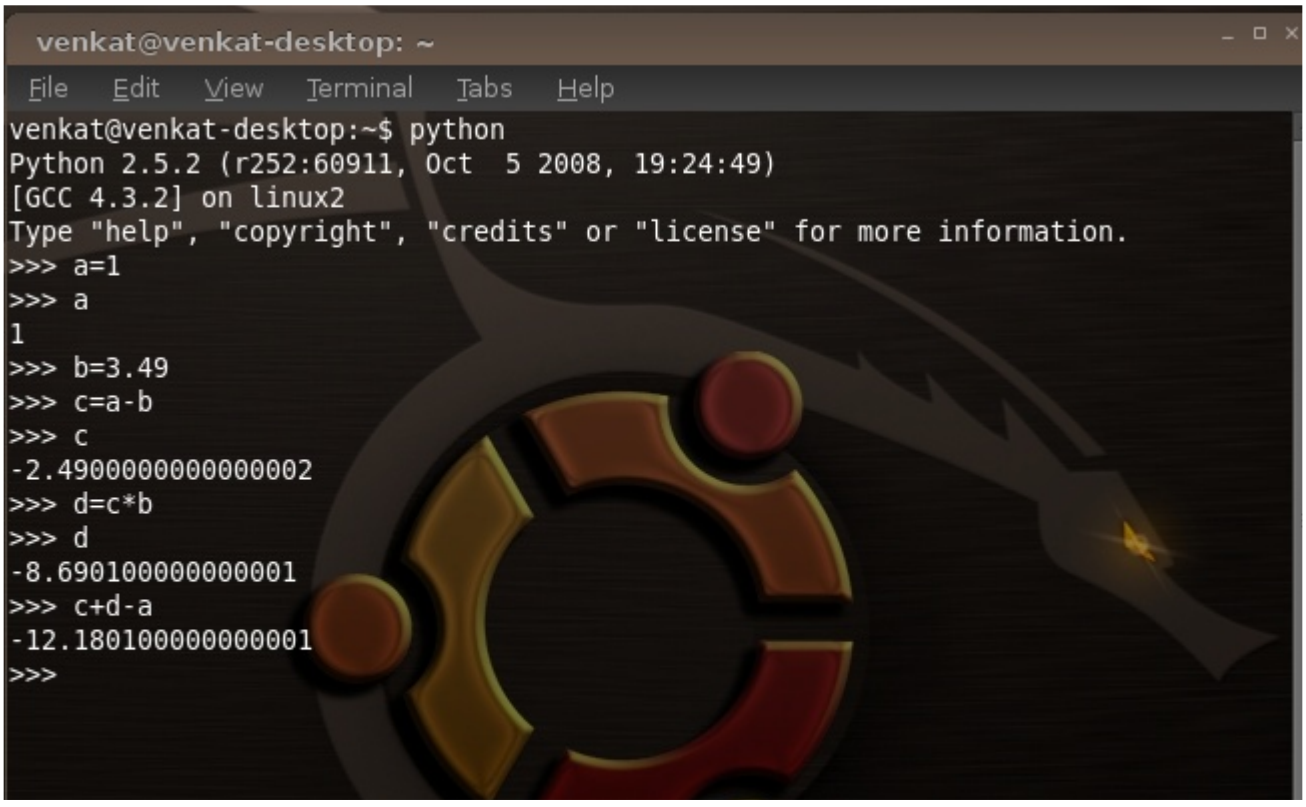
of Python.
Interactive?
This characteristic similar to any interpreted programming language. You will be able to appreciate by understanding the following screen-shot. (see below)

Here, You can see how one can use python as a calculator. You didn't declare with variables as integer or float, you simply used them (loosely typed language). For instance, I didn't declare a as an Integer, or b as float. This is the interactive attribute of Python.

Then the third one,
Object Oriented Language?
I don't want to bore you with the OO definition. It is simply that, Python has support for classes, objects, etc.

What did we learn today?
1. What is an Interpreted language?
2. Why is Python called so?
3. Who wrote Python?
4. The various features of Python?

We will probably end this first series of Programming with Python with some facts about it.
As I already told you Google and many such biggies can't live without Python. You might have asked what a programming language got to do with a company related to Web services? Answer is, Python is used widely as a Web Scripting language. You can write Python scripts to generate dynamic web pages. Many more facts and tutorials on Python will follow this article. So see you with the snake in the next series. Bye!



Identify yourself the digital way!

Sornakumar. S, V Sem, CSE

Internet has undoubtedly been the ultimate tool of the century every human being is powered with. Service providers have been striving hard to make every bit of data they provide, highly promising. With the increased number of users, the service providers increase and so do their specializations. For instance, when you utter the word search, the next thing which comes to your mind would be "Google". Wanna know what's cookin' hot?, get along with "Yahoo!" Meet your old pals, go nowhere, but "Facebook". Can't wait to express yourself? You are the much awaited customer at the "Blogger".

I've tried my hands over all these. Every service seem to impress me every little day I use 'em. But I've always had the trouble of remembering so many usernames and passwords. I always wondered if I could find some nice software which could remember all my usernames and passwords, and could also fill up my boring registration forms. One day, when I was browsing through some online magazines I found a heading which attracted my eyes. It read, "Have you ever imagined of one ID which would work on thousands of sites? Get to know OpenID,.." It precisely meant a solution to my problem! I decided to explore. I googled out OpenID and it showed me the way to solve my problem. Here is what it is.

OpenID is an open, decentralized standard for authentication. The user can use the same digital identity to login into different websites. It is highly different from the traditional username-password login structure. Instead it uses a URL or XRI based authentication. In establishing an OpenID authentication we have 3 main players.

User: The public who needs an OpenID. Identity Provider: The service provider, typically is a website which provides the URL to identify a user. Relying Party: The site (or) service which needs the user to be authenticated.

Users who need an OpenId should register to these Identity providers with a unique name and a password. Once they do so, they are given a URL which uniquely identifies them. Let us consider "kumar"(username) registers to "provider.com". Then his OpenID can be "provider.com/kumar". Whenever "kumar" needs to login to a website, he uses this URL. Suppose he is posting a

comment at the blogger, he is prompted first to enter his OpenID. When he does so, he is redirected to the Identity Provider's site and he's asked to enter his password. He can also choose whether to allow or deny the current site in which he's logging into, to access information about him or not.

It is a natural curiosity for a computer science student to find out what goes behind-the-scenes. I decided to find out how OpenID was more secure than usual authentication. Here are some of the results of my experiment. When the user enters his authentication URL in the Relying-party's website, the relying party establishes a "shared secret" with the identity provider. A shared secret, in cryptography terms mean, a passphrase exchanged between two people who communicate. This shared secret is stored by the relying party after authentication. Once this is established the user is redirected to the identity provider's website, where the user is asked to enter his password and his preferences for the particular site (whether or not the refering party can be allowed or denied to access his information). In this method of authentication there is no exchange of password between the Identity Provider and the Relying party. Thus the only person which has access to your password is the Identity Provider thus enhancing security. This is the one way of providing authentication. There are

Continued on Page 8

THE TUX TEAM

CHAIR PERSON
Dr.Ranjani Parthasarathy

CHIEF EDITOR
Ms.Bama

STUDENT EDITOR
Mahavir Gautham.R

SPECIAL THANKS TO
Sanath Kumar. R
Venkatakrishnan. G
Venkatnathan. V

This newsletter "THE TUX" is brought out by CEGLUG, an open source community under CEG's ACM's Student Chapter.

For a more details about CEGLUG, please visit,
www.ceglug.org

Incase of any queries, comments or contributions for further newsletters, please write to,
editor@ceglug.org

Your Pocket Guide for Building Linux Kernel

Mahavir Gautham.R, V Sem, CSE

In this small article I'm going to illustrate on how to build a custom linux kernel from its source packages using Ubuntu 9.04 (Jaunty Jackaloop) Kernel (Linux-2.6.13.1 Kernel) as example. Open the terminal of ubuntu and proceed to do the following in it,

1) First login as super user into the system using the command.

su

Now, it asks for the root password for the system.

If it is correct, now the ~ prompt changes to # prompt. Also, if this step is skipped, then the user has to prefix 'sudo' before all the commands used henceforth during the process of compilation of the kernel.

2) Now, we need to update the package database of your system for better results.

apt-get update

3) We need to install certain tools that will enable us to compile the kernel. For this we use the command,

apt-get install kernel-package libncurses5-dev fakeroot wget bzip2

4) We are atlast ready for downloading and compiling the kernel now. We proceed to downloading the kernel package(Linux-2.6.13.1.tar.bz2) now from www.kernel.org using,

cd /usr/src

wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.13.1.tar.bz2

By executing the two commands we download and store the kernel source package in /usr/src directory.

5) The source package we downloaded is a compressed one and we uncompress it using,

tar xjf linux-2.6.13.1.tar.bz2

Now, we rename the unpacked directory as 'linux' for ease of use and move into that directory using the commands,

ln –s linux-2.6.13.1 linux

cd /usr/src/linux

6) If you want to add any new driver support to your kernel or any support to a cutting edge technology like virtualization techniques, that is not integrated with your kernel download the packages for all those patches and apply it as follows (I call the downloaded patches as 'patch.bz2'),

First test for the integrity of your patches using the command,

bzip2 –dc /usr/src/patch.bz2 | patch -p1 –dry-run

If the execution of the above command does not show any errors then proceed to apply the patches to the kernel using the command,

bzip2 –dc /usr/src/patch.bz2 | patch –p1

7) Now we get on to building the actual kernel from the source.

First, we get on to cleaning the previous make leftovers if any and then go on to build,

make –kpkg clean

fakeroot make –kpkg --initrd --append-to-version=-custom kernel_image kernel_headers

Note: You can substitute any string other than 'custom' in the above command that helps you identify the .deb packages and the kernel, but it should be prefixed with – sign.

Also this process starts the building of kernel which may take a few hours depending on the processor speed and kernel configuration.

8) Once the kernel is built we concentrate on installing .deb packages built using,

dpkg –i linux-image-2.6.13.1-custom_2.6.13.1-custom-10.00.Custom_i386.deb

dpkg –i linux-headers-2.6.13.1-custom_2.6.13.1-custom-10.00.Custom_i386.deb

9) Now if all the steps have been done successfully, the new custom kernel will be there in menu.lst which can we found in, vi /boot/grub/menu.lst

10) Now, restart your computer and boot into your new kernel and enjoy!

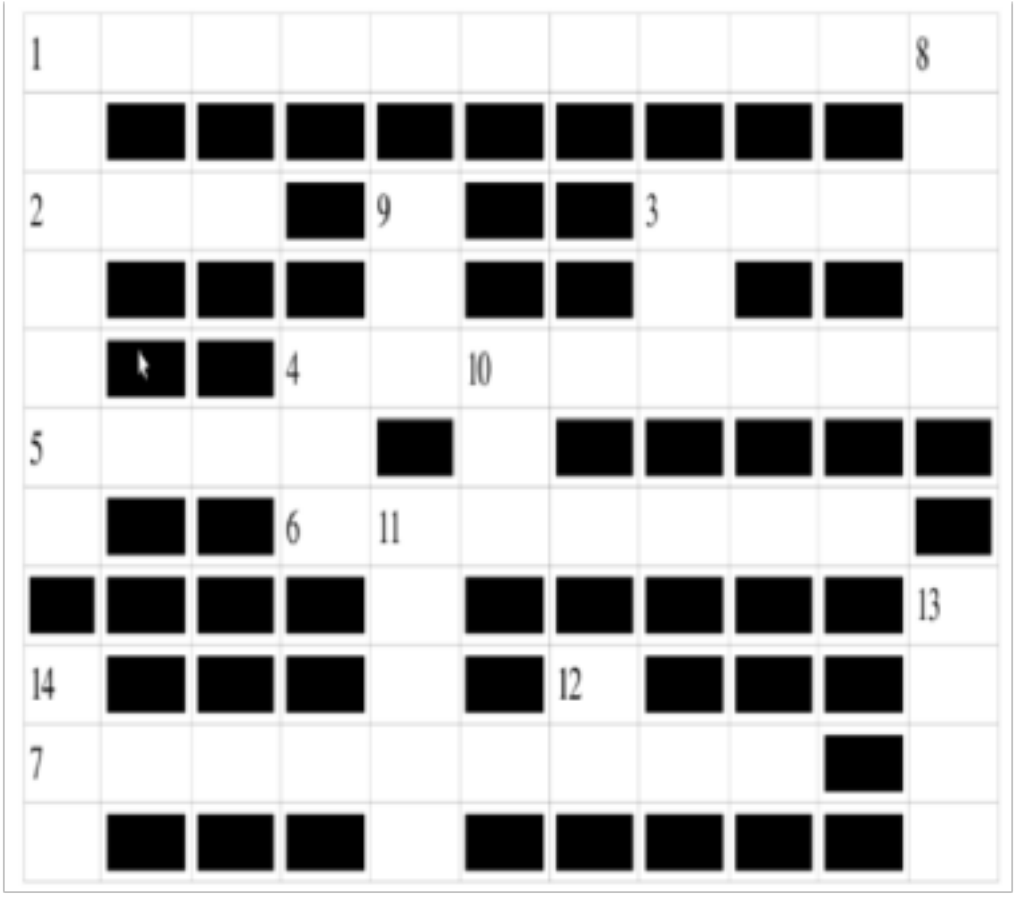
Continued from Page 6

Identify yourself the Digital Way!

other methods used in OpenID authentication which can identify users by direct communication between their browsers and Identity Providers, thus having no user intervention.

If you would like to know how popular OpenID is, then these names can help you out. Google, Microsoft, Yahoo!, AOL, MySpace, Facebook, SourceForge, Orange, VeriSign and 50,000+ sites have accepted on OpenID authentication. Imagine you having one ID and accessing 50,000+ sites. Man.., that is a real huge number! It doesnt end there. It keeps ticking everyday and even Governments are getting their hands onto it. No more Registration forms, and no more mnemonics to remember your IDs and passwords. All you need is an OpenID.

Open Crosswords



Venkatakrishnan. G, VII Sem, CSE

Across

1. tar (11)
2. file system (3)
3. sound(4)
4. Linus _____(8)
5. umunt- ngu-untu n-abant- (4)
6. GUI (7)
7. LZW, LZO, LZS (6,3)

Down

1. GNU C++ (7)
3. Language (3)
4. Mascot (3)
8. Text Editor (5)
9. AWK, one (3)
10. RPC, OO (3)
11. Emulator with Games (5)
12. Compression with armament (2)
13. Printing (4)
14. Most Popular Media Player (3)