

**OBJECTIVE:**

To expose students to FOSS environment and introduce them to use open source packages

1. **Kernel configuration, compilation and installation :** Download / access the latest kernel source code from kernel.org, compile the kernel and install it in the local system. Try to view the source code of the kernel

2. **Virtualisation environment** (e.g., xen, kqemu or lguest) to test an applications, new kernels and isolate applications. It could also be used to expose students to other alternate OS's like \*BSD

3. **Compiling from source :** learn about the various build systems used like the auto\* family, cmake, ant etc. instead of just running the commands. This could involve the full process like fetching from a cvs and also include autoconf, automake etc.,

4. **Introduction to packet management system :** Given a set of RPM or DEB, how to build and maintain, serve packages over http or ftp. and also how do you configure client systems to access the package repository.

**5. Installing various software packages**

Either the package is yet to be installed or an older version is existing. The student can practice installing the latest version. Of course, this might need internet access.

Install samba and share files to windows

Install Common Unix Printing System(CUPS)

6. **Write userspace drivers using fuse** -- easier to debug and less dangerous to the system (Writing full-fledged drivers is difficult at student level)

7. **GUI programming : a sample programme** – using Gambas since the students have VB knowledge. However, one should try using GTK or QT

8. **Version Control System setup and usage** using RCS, CVS, SVN

9. **Text processing with Perl:** simple programs, connecting with database e.g., MySQL

10. **Running PHP** : simple applications like login forms after setting up a LAMP stack

11. **Running Python** : some simple exercise – e.g. Connecting with MySQL database

12. **Set up the complete network interface** using ifconfig command like setting gateway, DNS, IP tables, etc.,

**Resources :**

**An environment like FOSS Lab Server (developed by NRCFOSS containing the various packages)**

OR

Equivalent system with Linux distro supplemented with relevant packages

**Note:**

**Once the list of experiments are finalised, NRCFOSS can generate full lab manuals complete with exercises, necessary downloads, etc. These could be made available on NRCFOSS web portal.**

**TOTAL: 45 PERIODS**

**LIST OF EQUIPMENTS:****Hardware:****Minimum Requirements:**

- 700 Mhz X86 Processor
- 384 MB of system memory (RAM)
- 40 GB of disk space
- Graphics card capable of 1024\*768 resolution
- Sound Card
- Network or Internet Connection

**Software:**

Latest distribution of Linux.

## **Excercise No 1 : General Introduction & Linux OS Installation**

### **Aim :**

To introduce the fundamentals of Free/Open Source Software and to install Linux Operating System -Fedora Ver. 12 from FOSS Lab Server.

### **Description of the exercise :**

#### **a) Theory session for two hours covering the following topics**

- a) Introduction to FOSS
- b) FOSS Licenses
- c) Linux Distributions
- d) Packages – RPM based & Debian based – 2 or 3 samples chosen in tune with the syllabus
- e) Kernel – Kernel structure

### **Reference :**

FOSS Book available for download at

-<http://www.nrcfoss.au-kbc.org.in/down/3/> (from internet) or  
under 'Docs' in 'FOSS\_BOOKS' of your FLS(from your LAN)

#### **b) Hands on session for one hour on Linux installation (Fedora) from FLS**

### **About Fedora ...**

Fedora is a fast, stable, and powerful operating system for everyday use built by a worldwide community of friends. It's completely free to use, study, and share. More details about Fedora is available under 'Docs' in 'Fedora\_Document' of your FLS.

A practical exercise for students to learn basic linux installation of a rpm based operating system from Foss Lab Server

### **Pr-requisites**

1. Partition should already be done.
2. PXE boot option enabled in BIOS setup
3. Sufficient Free space available for installation of linux (Min 10 GB)
4. Session supervised by a trained staff member

### **Procedure with steps :**

Select Fedora by typing “fedora 12” near “**boot :-**”

```
Trying to load: pxelinux.cfg/C0A802A9
Trying to load: pxelinux.cfg/C0A802A
Trying to load: pxelinux.cfg/C0A802
Trying to load: pxelinux.cfg/C0A80
Trying to load: pxelinux.cfg/C0A8
Trying to load: pxelinux.cfg/C0A
Trying to load: pxelinux.cfg/C
Trying to load: pxelinux.cfg/default
- BOOT MENU -
=====

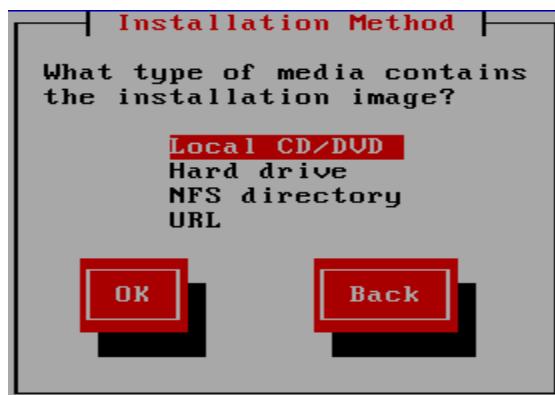
fedora 14
fedora 12
centos 5.5
opensuse 14
mandriva 2011
centos 5.6(test)

boot: -_
```

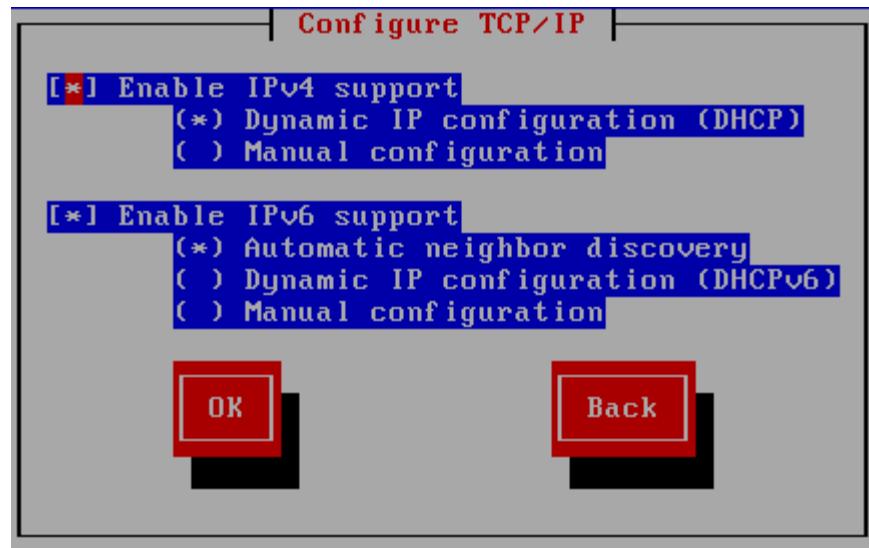
Choose a Language and then use TAB key to press OK and Press ENTER key.



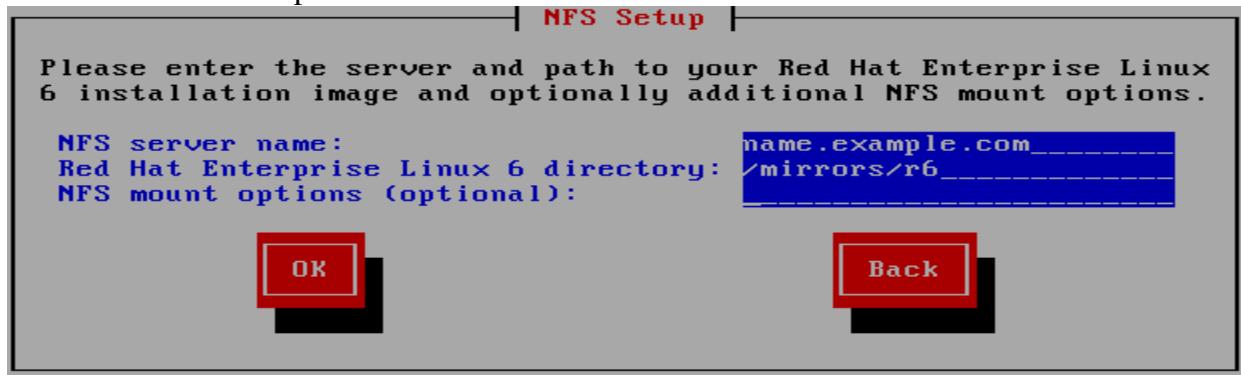
Use the arrow keys on your keyboard to select an installation method. Here select the NFS directory and use TAB key to select the OK and press ENTER.



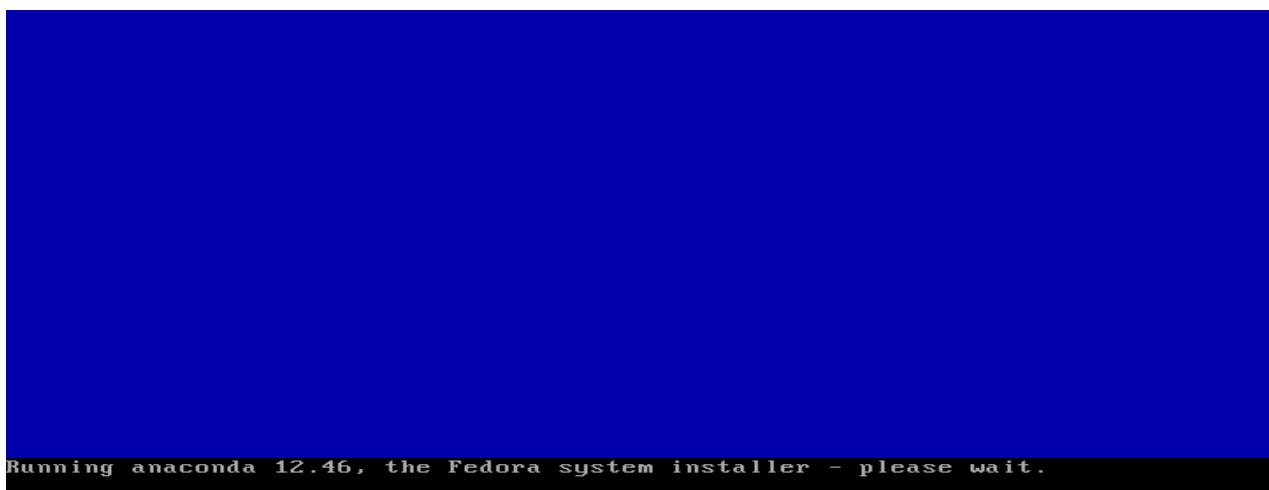
Press OK



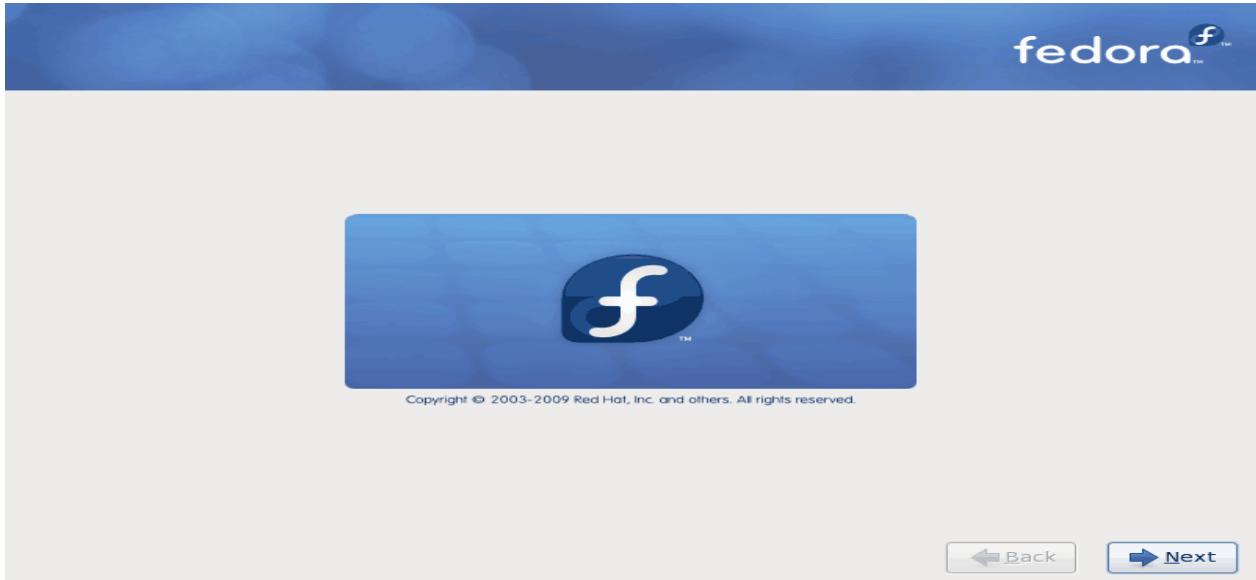
Enter the IP Address of the machine which is FLS in your LAN, in the place of “name.example.com” against NFS Server Name. The give the export directory as “/root/nfs/fed/fed” in the place of “/mirrors/r6”.



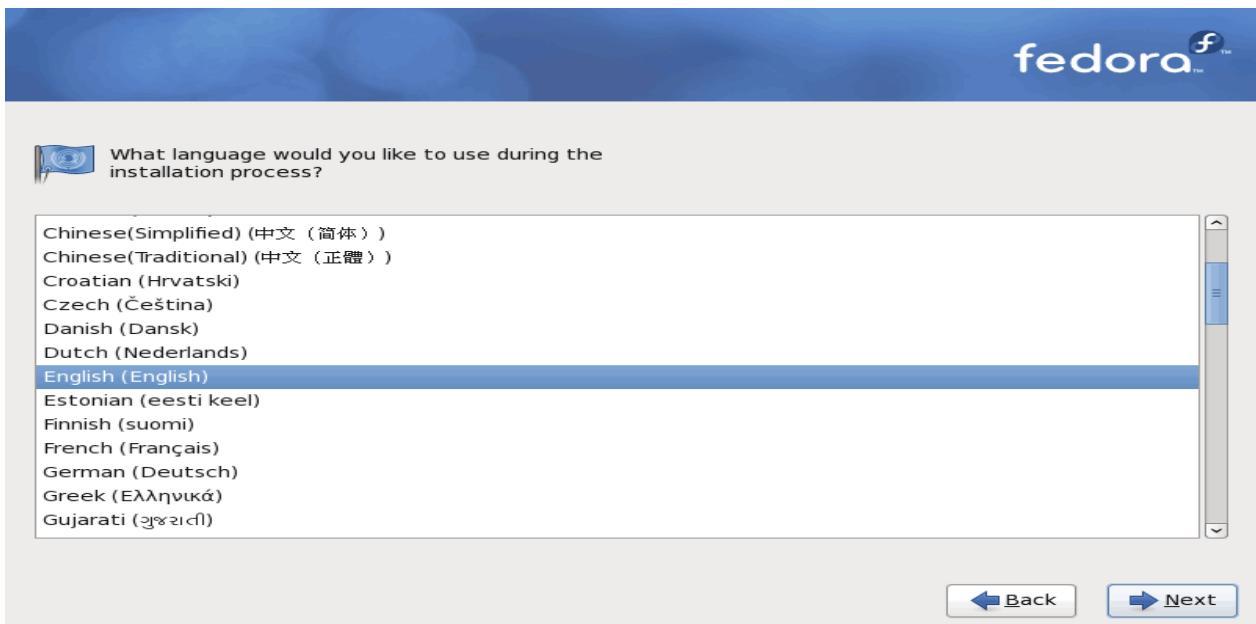
Then the installation proceed with the initialization of Anaconda the Fedora system installer. Just wait for Anaconda to initialize and give you the Fedora 12 graphical system installer.



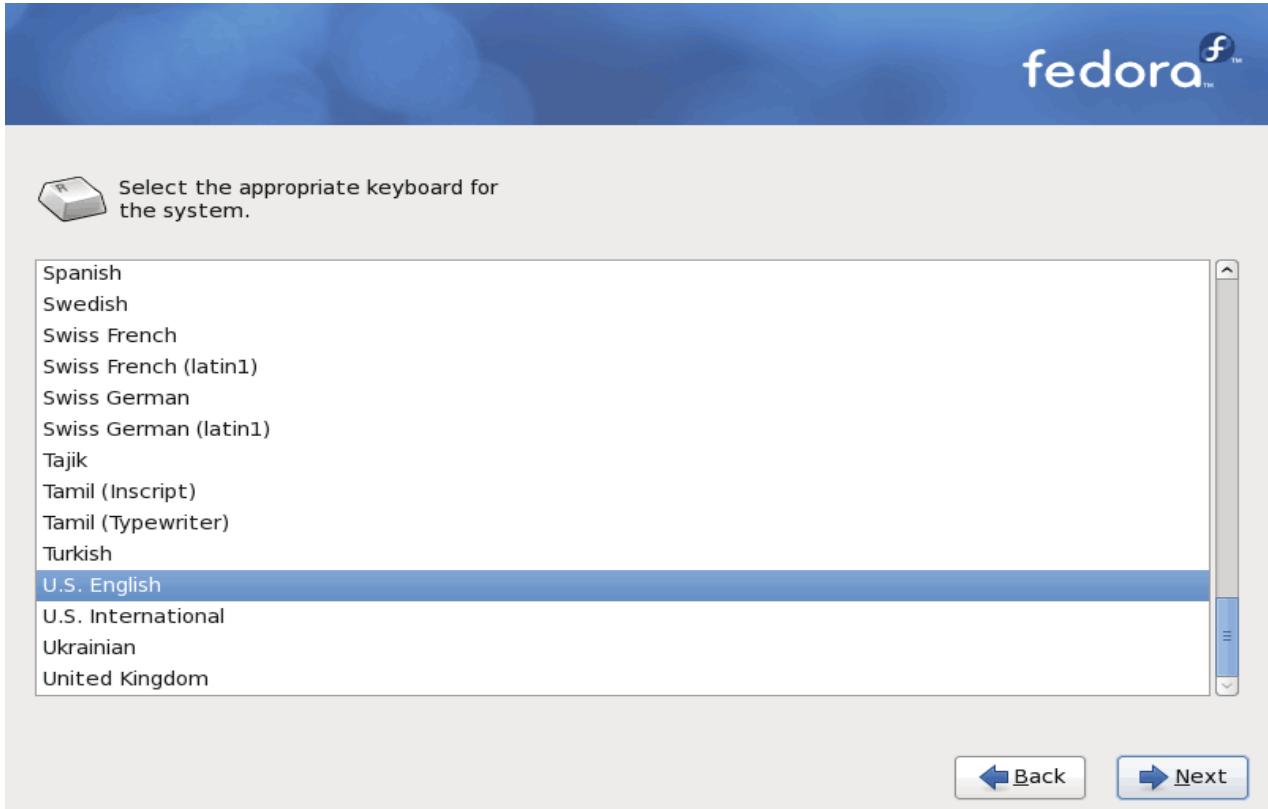
This is the start of Fedora 12 graphical installation setup . Click 'Next' button to proceed with the installation process



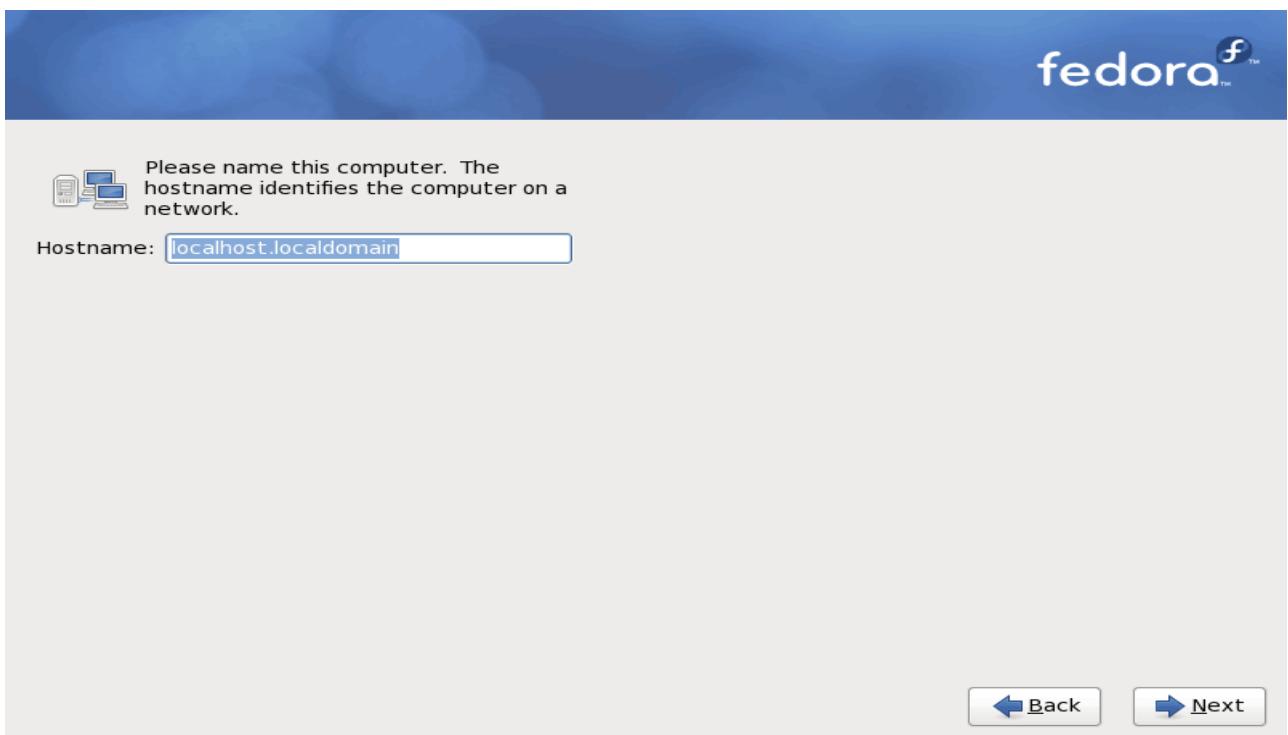
Choose the language that you want to use during the installation process of Fedora 12



On this installation window screen, select appropriate keyboard layout setting for your Fedora 12 system.

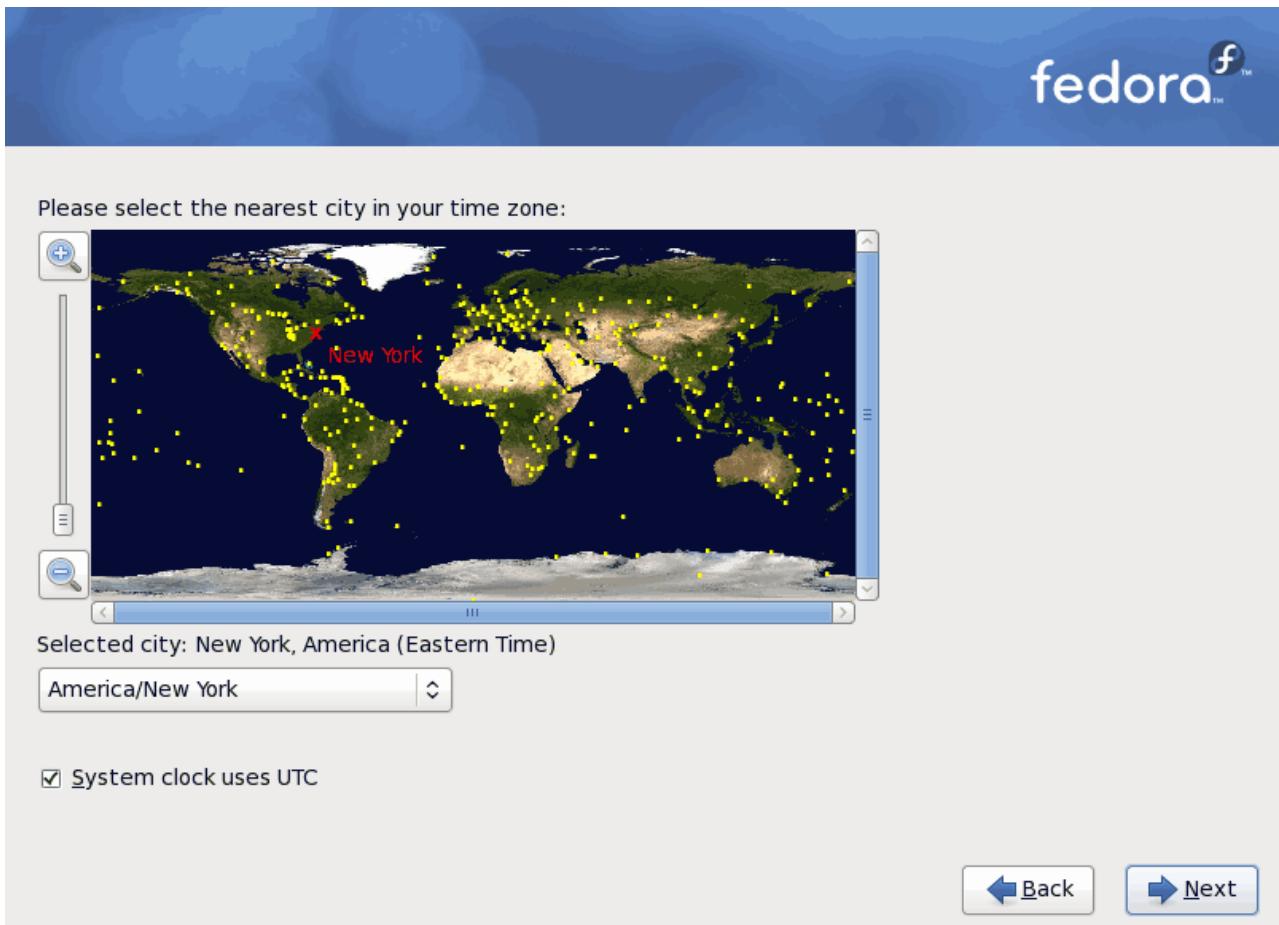


Give the hostname for your Fedora 12 system which is to be used to identify your Fedora 12 system on the network.

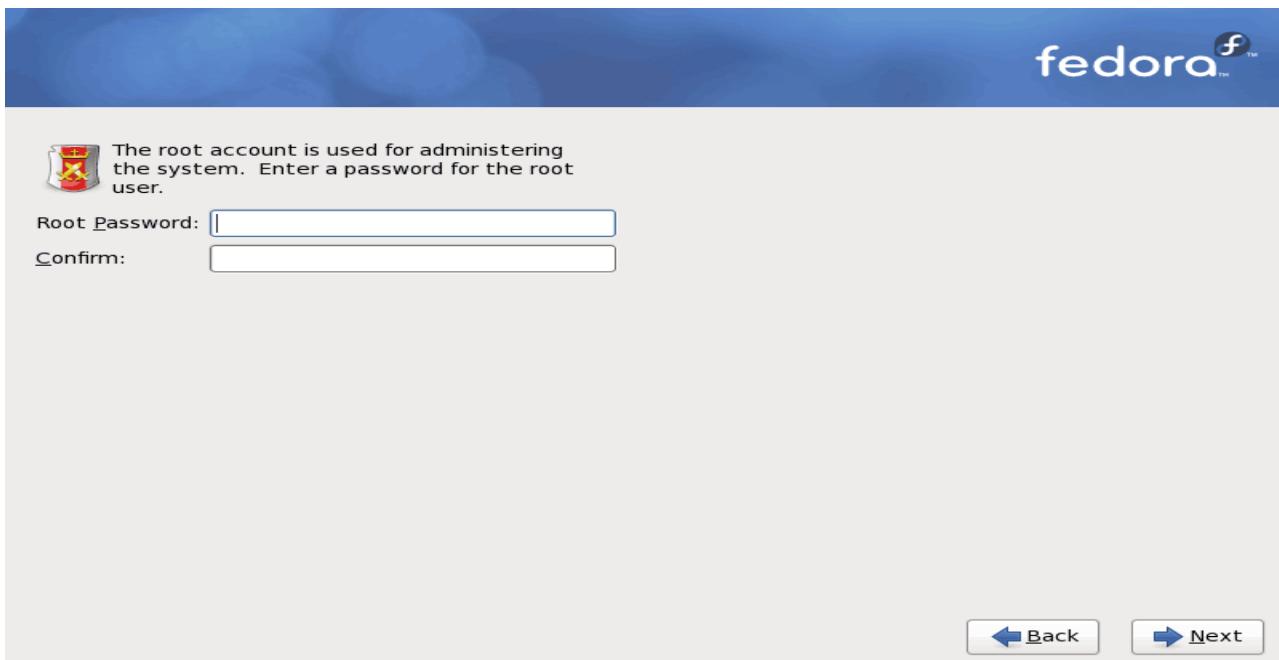


click 'Next' button to proceed

To configure Fedora 12 time zone, just click on nearest city available on the world map on the box provided on this screen to set correct time zone for your Fedora 12 system, or just click on the drop down box and choose the correct time zone by selecting the nearest city available.

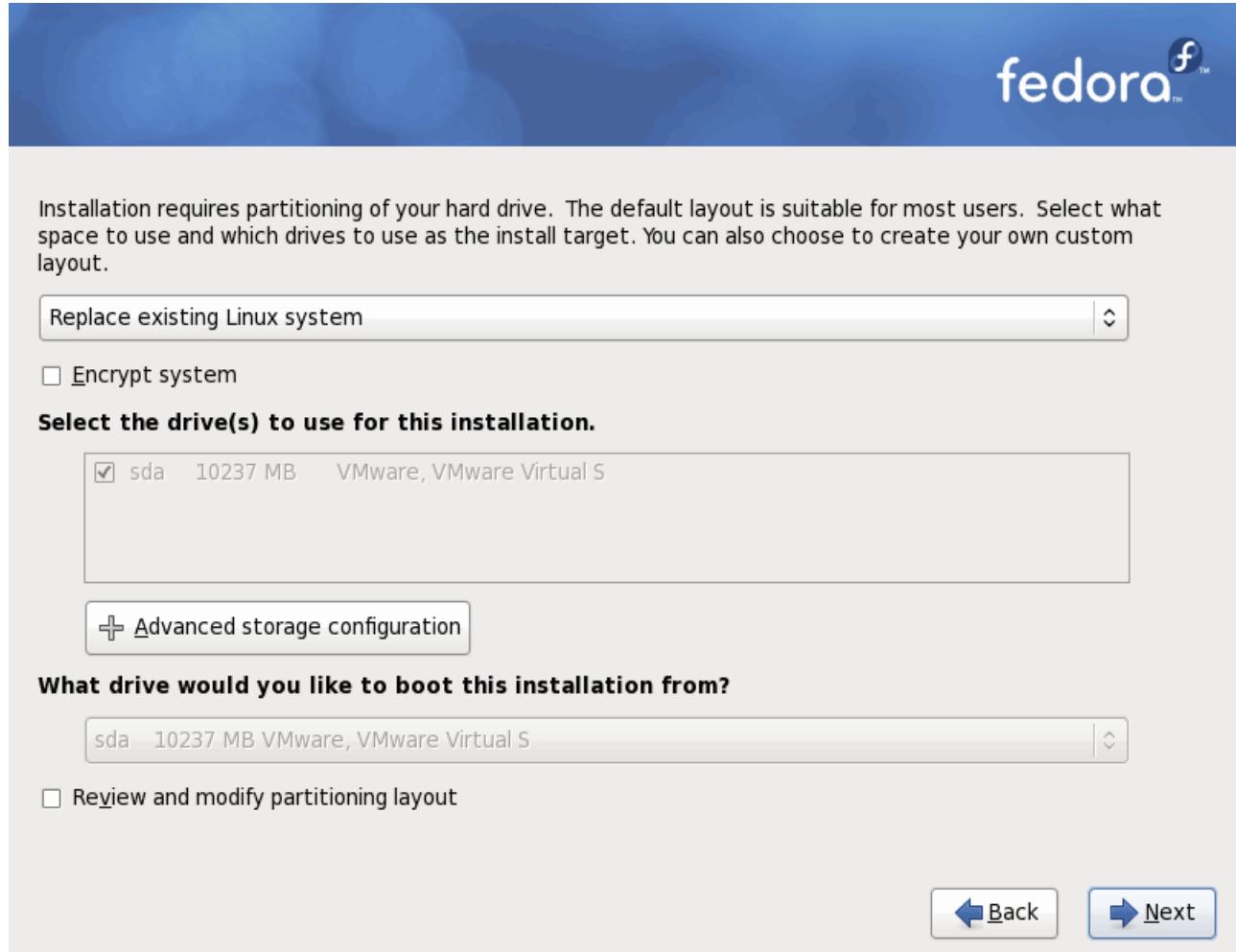


Click 'Next' button to proceed

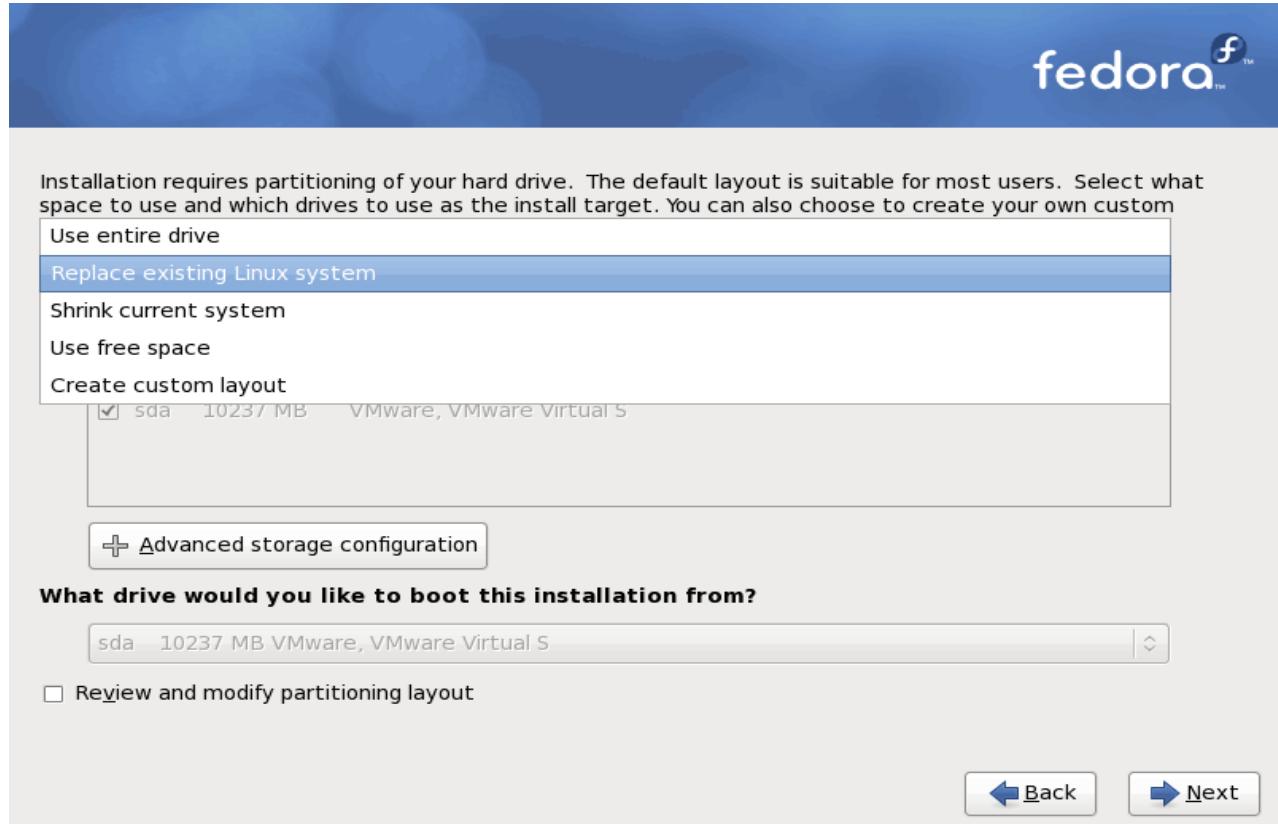


Key in root password for your Fedora 12 system and click on 'Next' button to proceed

Setup Fedora 12 partition. On this screen you need to choose the best partition option where to put Fedora 12 on your system hard disk.

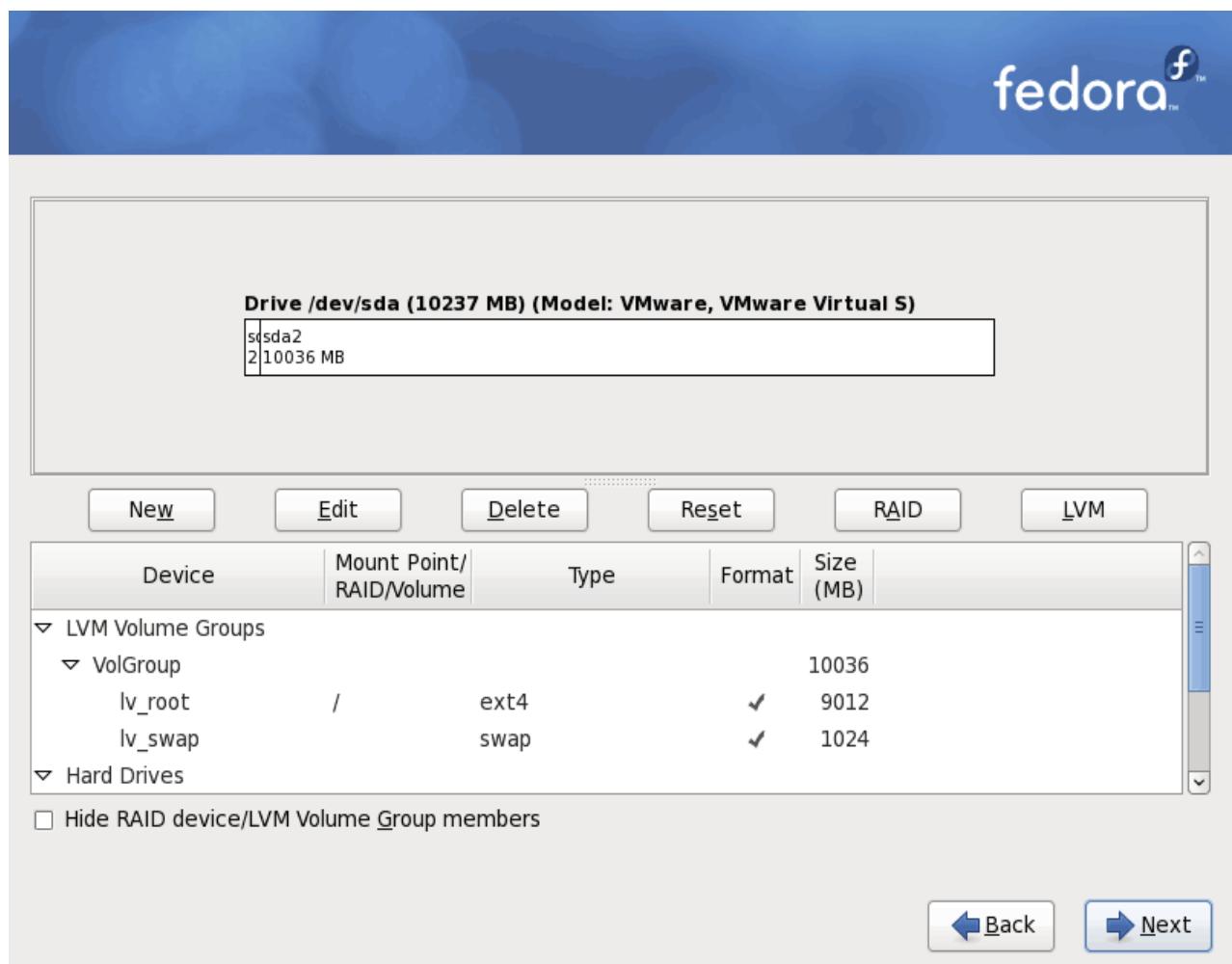


Installation requires partitioning of your hard drive. You can also choose to create your own custom layout.

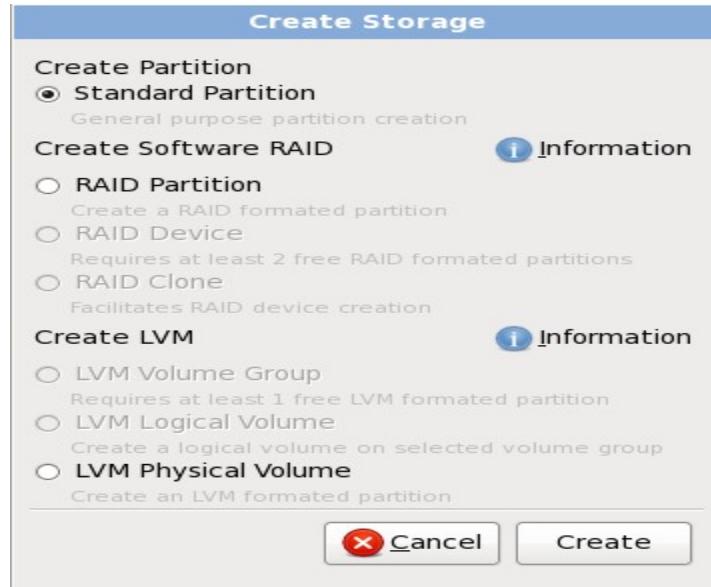


Click on the drop down menu and choose 'create custom layout'

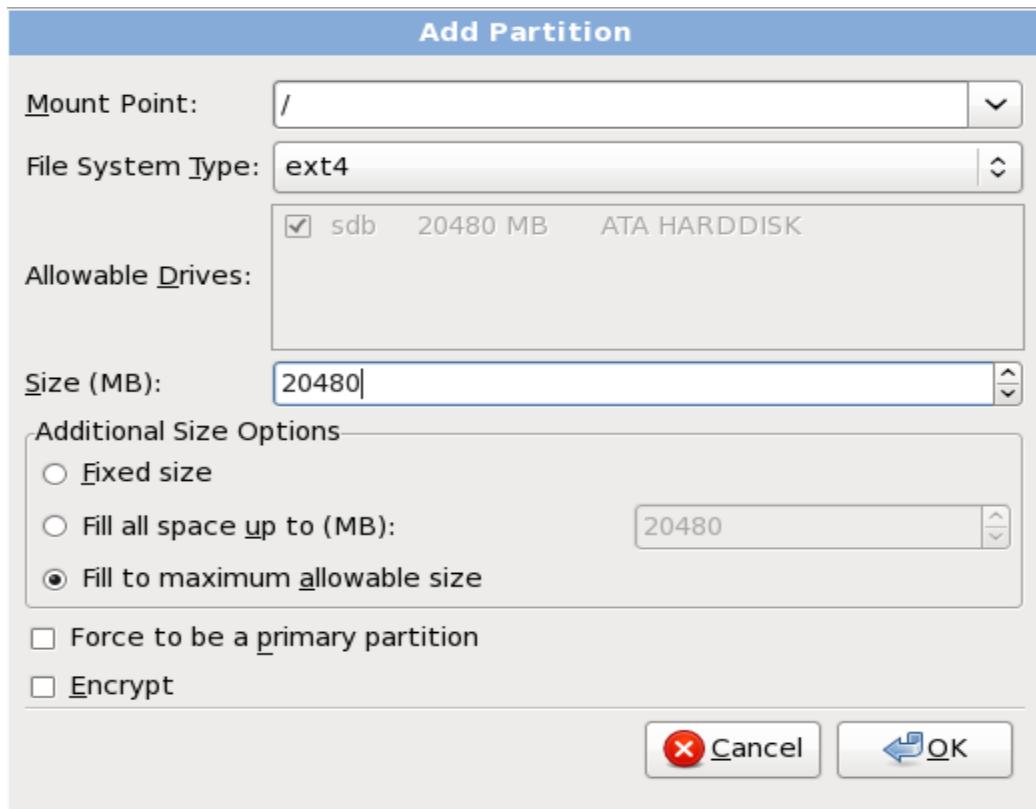
Create a root partition by clicking on the **New** Button and then create a swap partition.



Select Standard Partition to create a partition



Mount point must be '/' for a root partition and file system type Ext 4 . Size defines the amount of hard disk space for the particular partition.

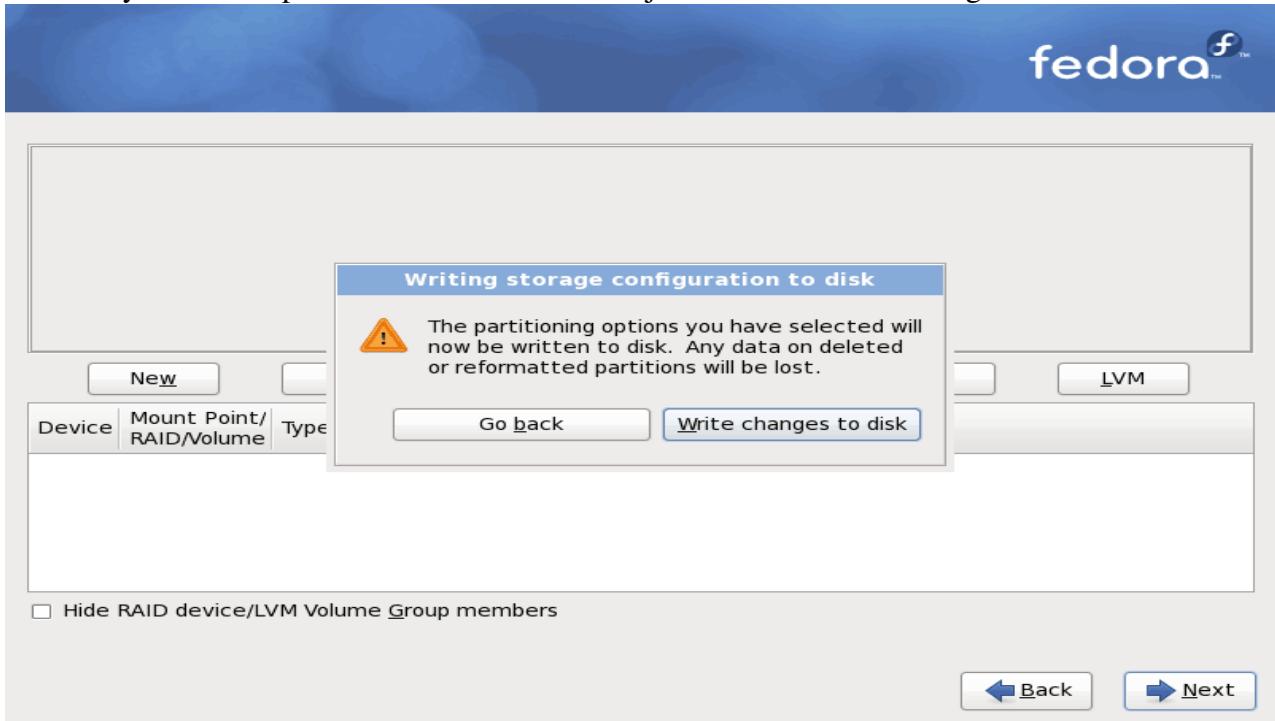


Similarly Create Swap partition also. Swap size can be double the size of RAM.

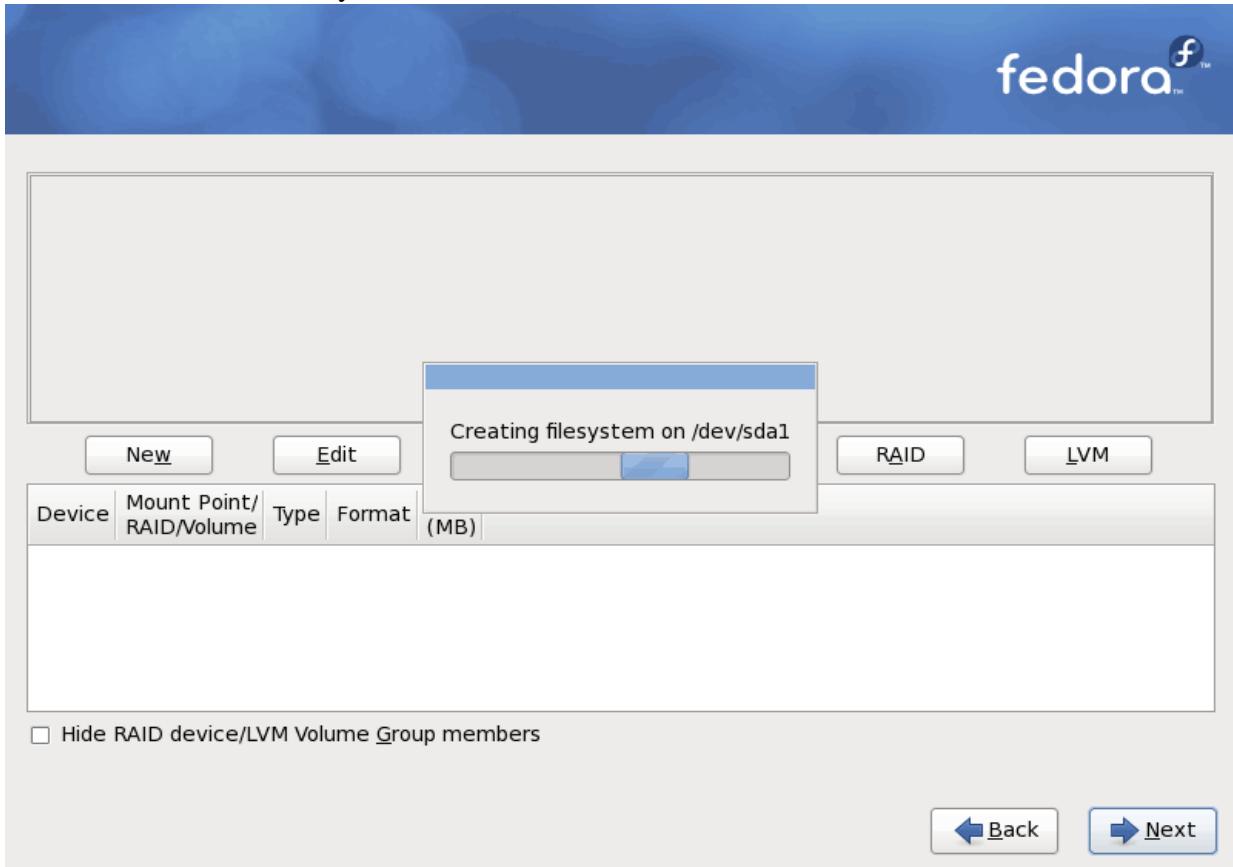
Click next to continue

Then the Fedora 12 prompt you with "Writing storage configuration to disk". What this box say is that the Fedora 12 system installation going to write partition and formatting your system hard disk as you configure on previous screen. and if you continue with write change to disk, all the data on

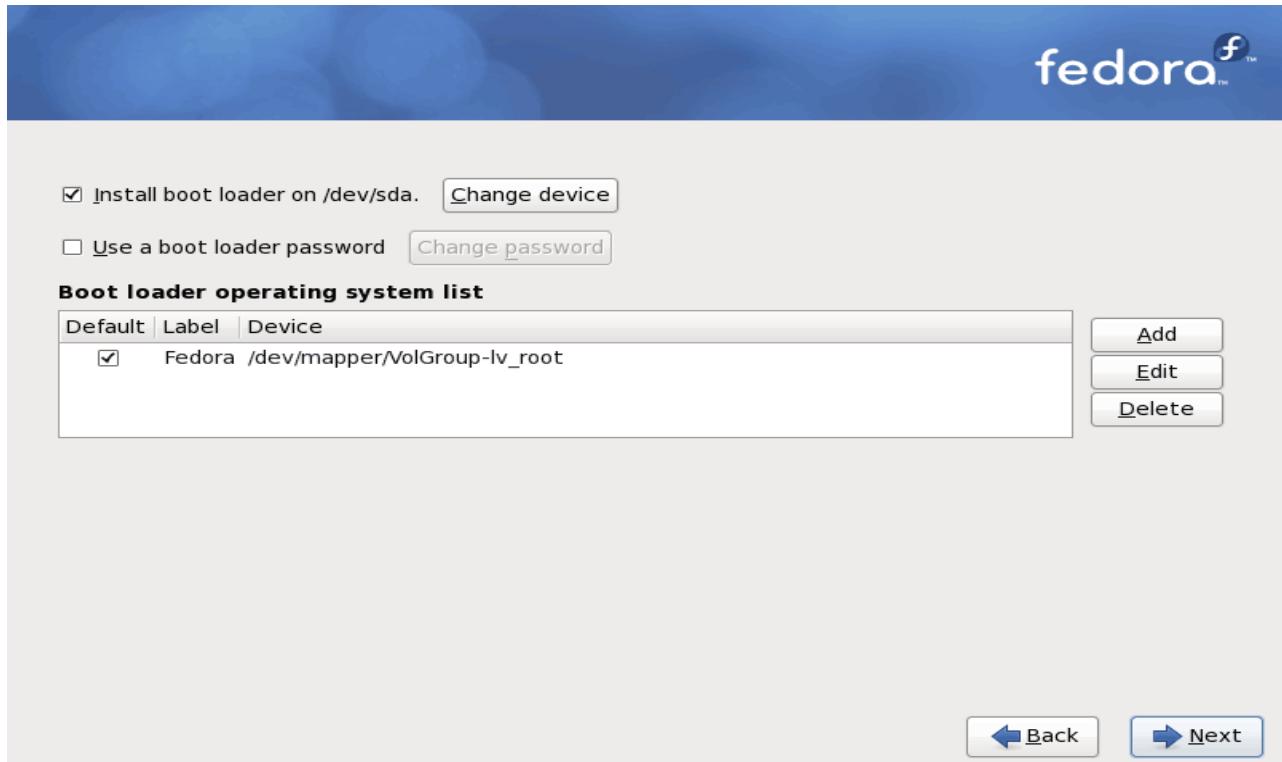
the hard disk will be lost. Click on "Go back" button to reconfigure partition layout or if you really sure that you want to proceed to install Fedora 12 just click on "Write change to disk".



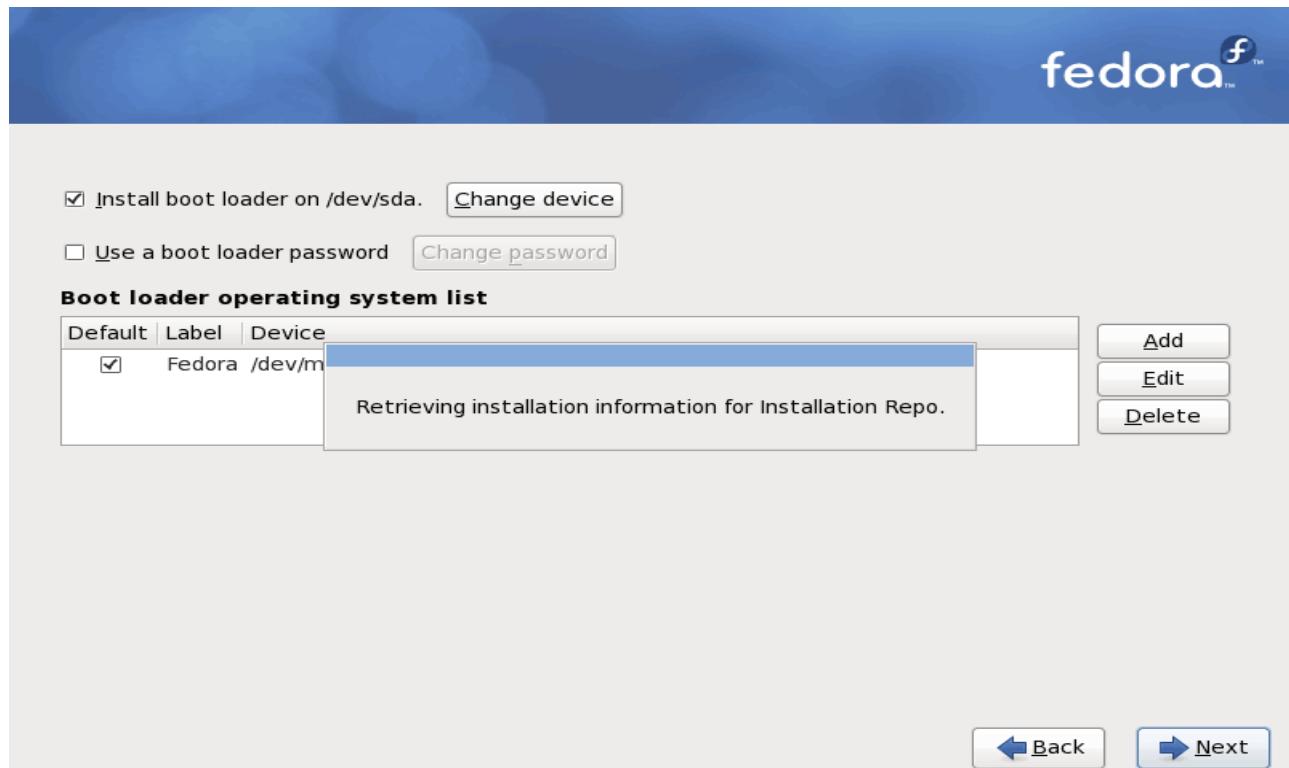
The screen show the Fedora 12 system installer creating filesystem on /dev/sda1 the first portion on the first hard disk on the system.



Then Fedora12 system installer display the screen where you want to place the Fedora 12 boot

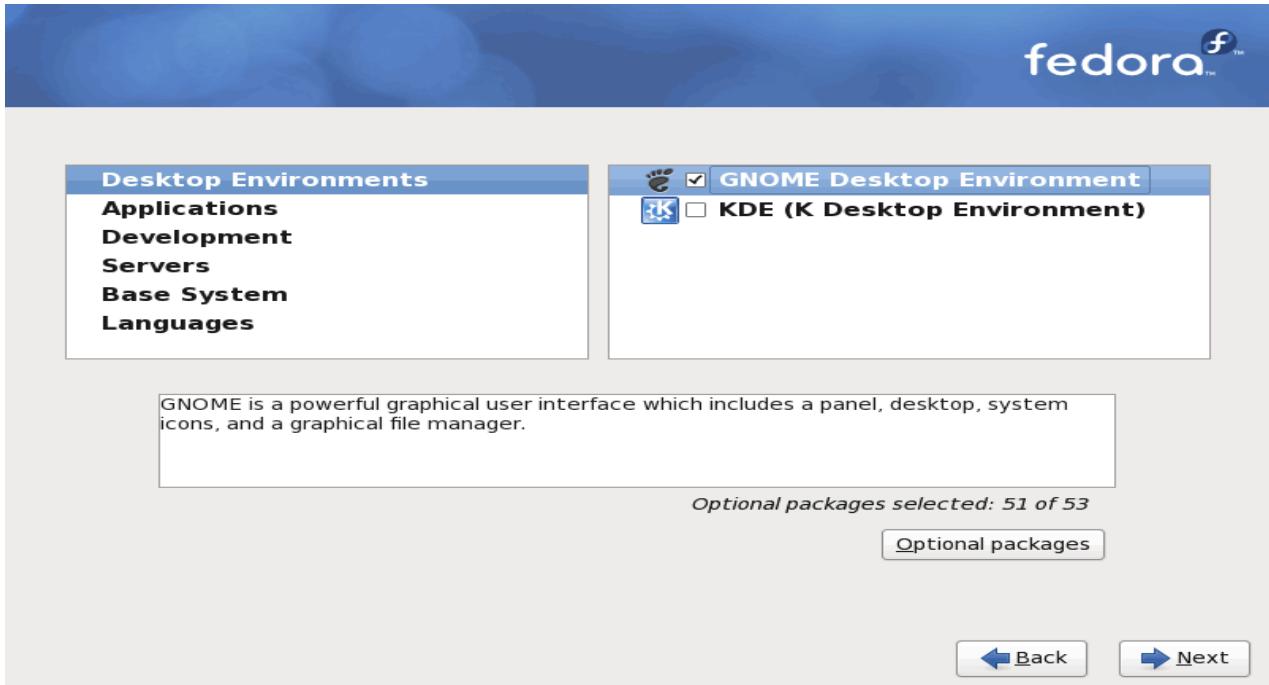


loader. The default option usually place on the first system hard disk, you can specify or change to other location by click on "Change device" button.

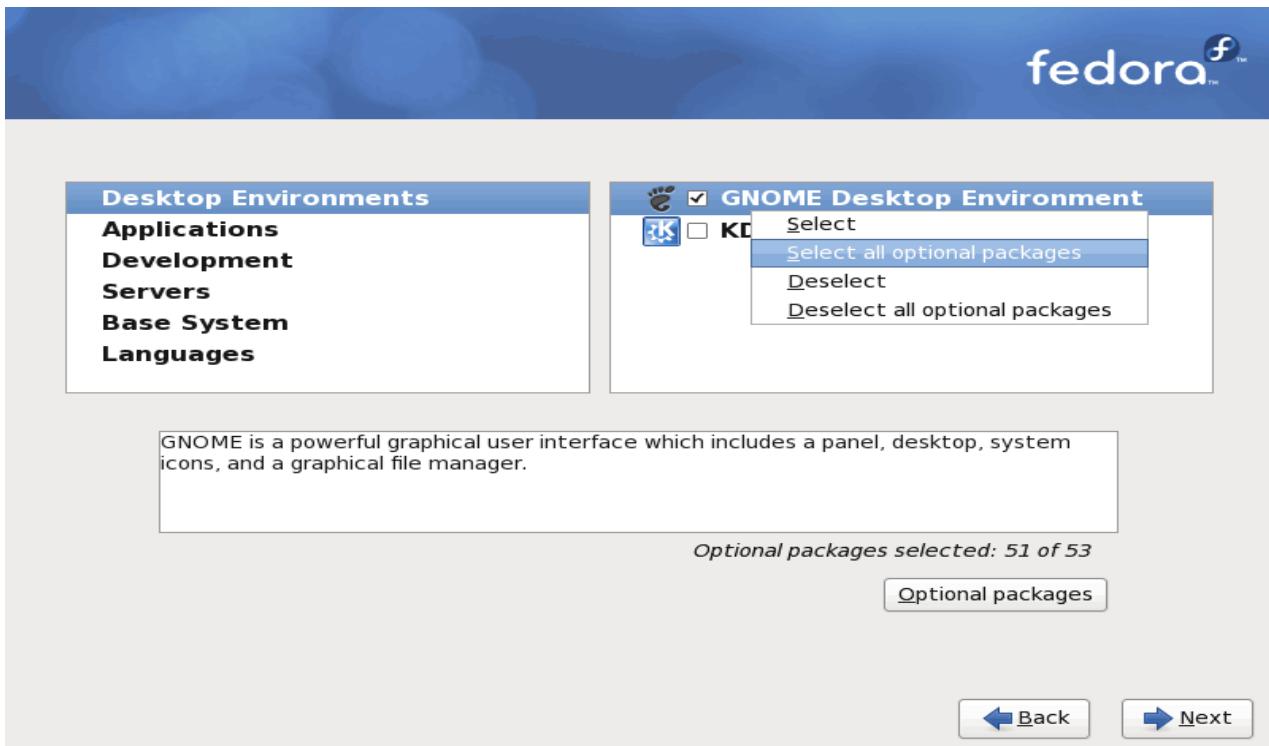


Leave default boot loader option as it is... or you can customize as you see fit. Click 'Next' button to proceed. Then Fedora 12 system installation wizard retrieve the installation repository from FLS server.

To customize or adding more software packages that you want to install or include on your Fedora 12 installation, select customize now button and then click on left box window that show Desktop Environments, Applications, Development, Servers, Base System and Languages. Tick the check box on the right window to adding and include the selected package group software.

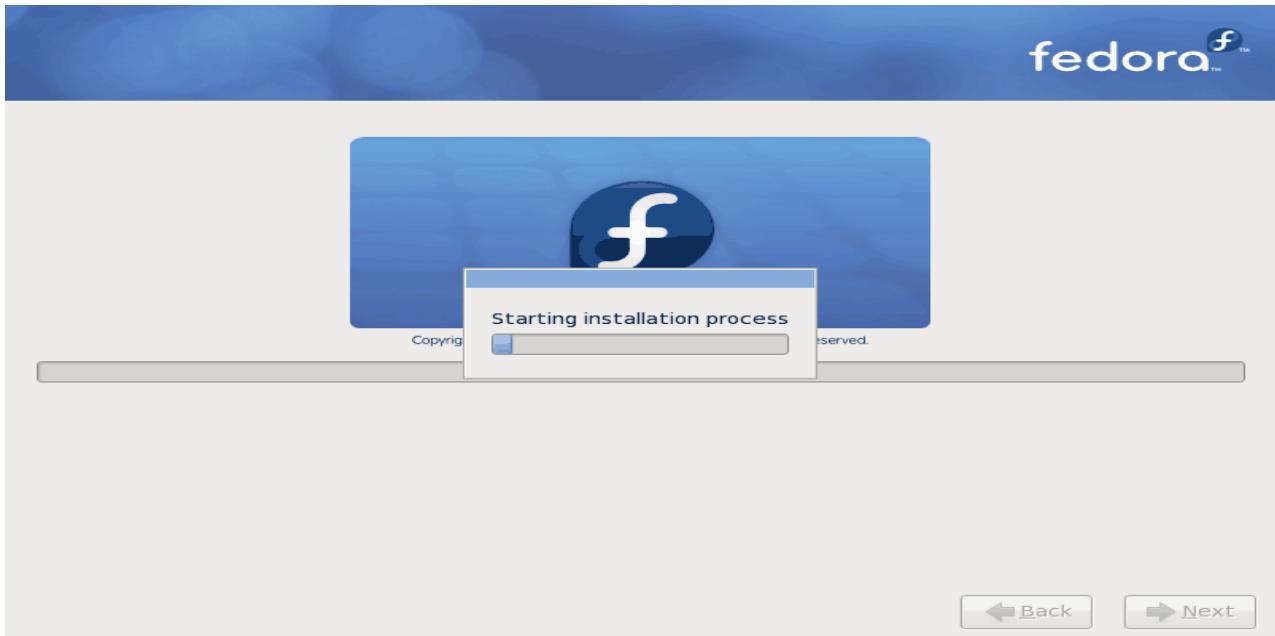


You can further customize the installation packages, by selecting optional packages in group package. To select all optional packages, click on right mouse button to open drop down menu option... the menu option include Select the package, Select all optional packages, De-select packages and De-select all optional packages. To install packages and all their optional packages. choose and click Select all optional packages as show on the screen shot below.

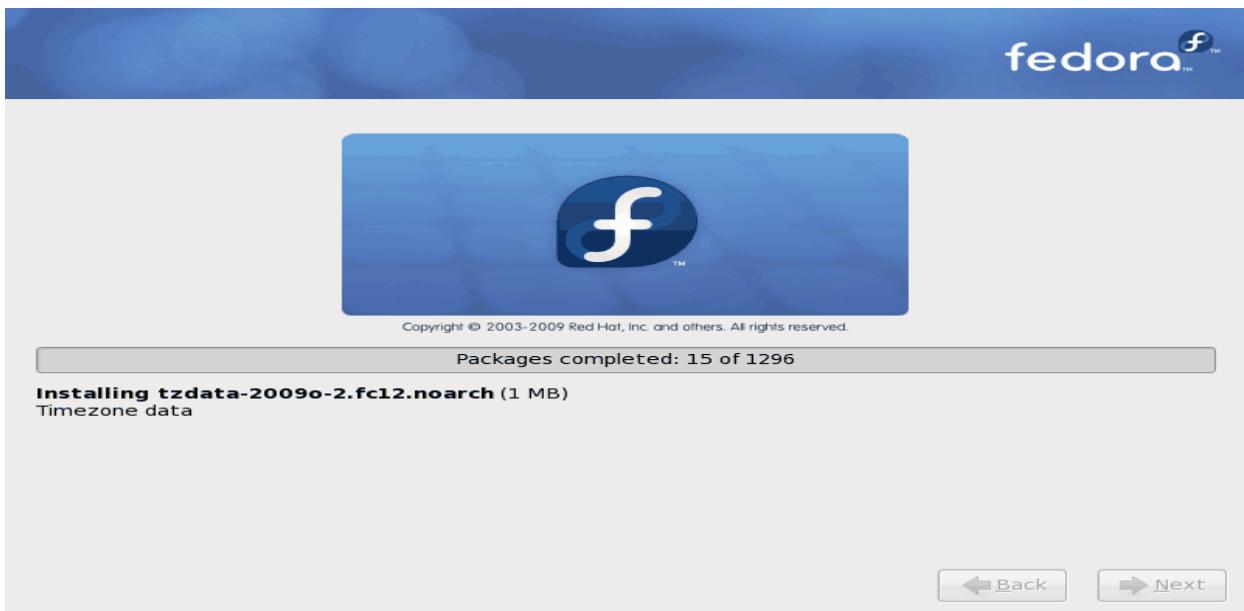


Click on 'Next' button to proceed with the installation process

After the package dependencies checking is done, the initial process of installation Fedora 12 operating system will start. The configuration process may take several minutes to complete, this is shown by the progress bar indication box.



The actual process of installing software packages to your system's hard disk . The process may take time to complete based on your overall packages selection.



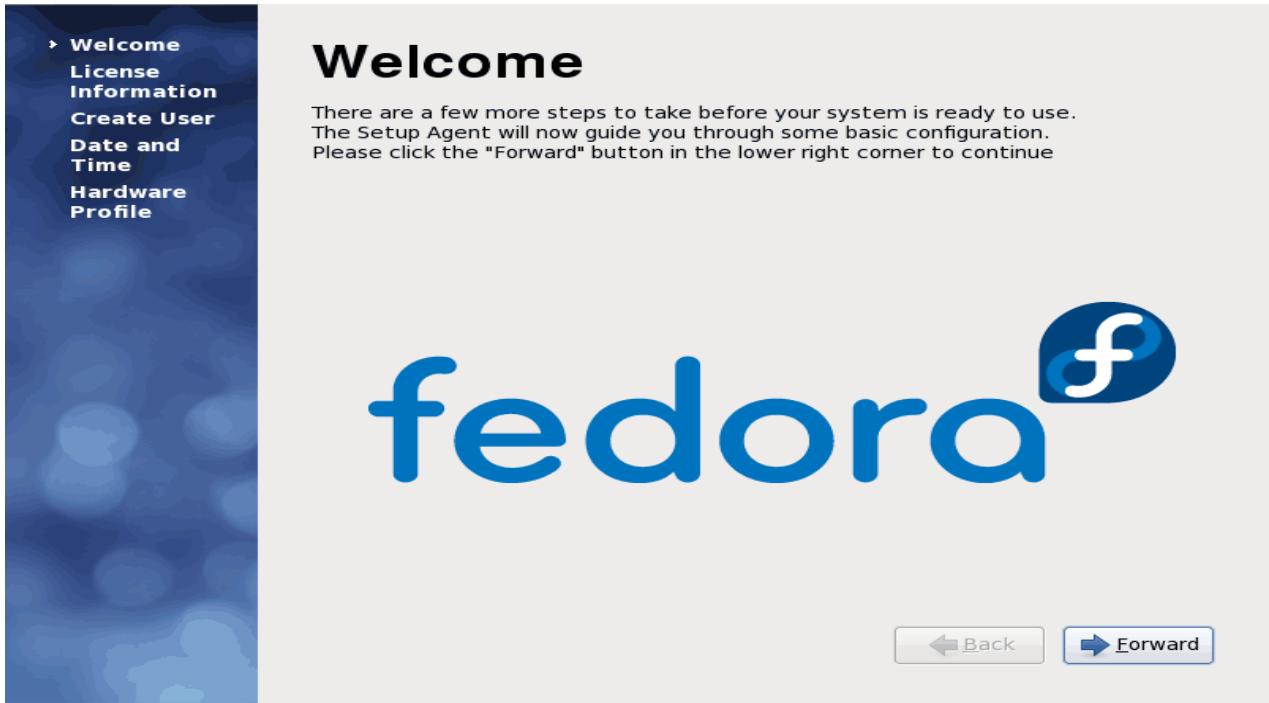
Please wait for installation process to finish installing all packages.

To finish up the installation process you need to click on the 'Reboot' button to reboot the system and proceed with the Fedora 12 basic system configuration.



Screen shot below shows the Fedora 12 boot up screen.

1.



Here is the Fedora 12 First boot configuration welcome screen. This is First Boot Configuration screen kick in only on the first boot of Fedora 12 after the installation process completed. The Fedora 12 Setup Agent (First Boot Configuration) process, enable us to do some basic configuration of Fedora 12 system before we log in on the Fedora 12 system for the first time. The screen shot below just show first screen of Setup Agent First Boot Configuration process. click on the 'Forward' button to proceed with the Fedora 12 configuration process.

This screen display the Fedora 12 License Information. License Information screen. No configuration or setting need to be done on this screen just click 'Forward' button to continue with the installation process.



The Fedora 12 configuration continue with Creating User. This user account is non administrative user account (normal user account). You need to create new user account by filling the Username, Full Name, Password and Confirm Password on the text box provided on this screen.

Next screen helps in the date and Time configuration . Then the Fedora 12 installation setup agent proceed with the Hardware profiler. Fedora 12 hardware profile detect your system hardware and make an profile out of it. This hardware profile contains all your system hardware information and information about the Linux Fedora 12 operating system installed. Please note that on this screen, Fedora uses molt as fedora 12 hardware profiler to enable you send your Fedora 12 system information profile anonymously.



Click on 'Finish' button to proceed, please note that if you want to send your system hardware profile to the Fedora Project.. just click on the Send Profile radio button before you click on the 'Finish' button. Then just key in the user password and hit the Enter key or click on 'Log In' button to start log in to the system.



## References:

[http://docs.fedoraproject.org/en-US/Fedora/12/html/Installation\\_Guide/index.html](http://docs.fedoraproject.org/en-US/Fedora/12/html/Installation_Guide/index.html)

[http://www.labtestproject.com/fedora\\_screenshot/001\\_fedora\\_12\\_screenshot\\_step\\_by\\_step\\_installation.html](http://www.labtestproject.com/fedora_screenshot/001_fedora_12_screenshot_step_by_step_installation.html)

## 2. Installing Software Packages on Gnu/Linux

**Aim:** The aim of this session is to learn how to install, update and remove software on Gnu/Linux.

### Introduction:

Software usually come in "packages". In the Gnu/Linux world, there are several kinds of packages, and each distribution has its own preferred package format.

The standard Linux package format (according to the Linux Standard Base) is **RPM**. RPM which stands for “Red Hat Package Manager” is a packaging system originally developed by Red Hat Inc., and widely used in the Linux community. Distributions using it include Redhat, Fedora, Mandriva, and SUSE. An RPM package file normally will be named something like “*program-version-other.rpm*”

Another popular package format is **DEB**, the Debian software package. Debian packages and the Advanced Packaging Tool (APT) were the first to introduce several advanced features that are now common, such as automatic dependency resolution and signed packages. Debian packages are used by Debian GNU/Linux (naturally), and distributions based on it, including Ubuntu and Knoppix. A Debian package file normally will be named something like “*program-version-other.deb*”

Packages on Gnu/Linux can be installed using command line and GUI methods.

### 1. Command Line Method

- Installing RPM's using the Redhat Package Manager
- Installing with Fedora/yum
- Installing using Debian's apt-get

### 2. Graphical(GUI) Process

- Using Fedora's Add/Remove Software
- Ubuntu's Synaptic

### 1. Command Line Method:

#### 1.1- Installing RPM's using the Redhat Package Manager:

RedHat RPM's offer a flexible and easy method to install new software. RPM's are files that have a ".rpm" extension. When one uses RPM for installing a software package, RPM checks if the system is suitable for the software the RPM package contains, figures out where to install files the package provides, install them on the system, and adds the piece of software into its database of installed RPM packages. The good point about RPM's is that the installation of new software, and maintaining the software currently installed is much easier than doing so for individual packages compiled from source. The downside to RPM's is that you don't have as much choice about where software is installed on your system, how it is compiled, and how it is configured. Login as the “**root**” user.

To install a package, use the following command:

```
# wget http://foss-lab-server-ip/content/packages/Fedora\_Packages/os/Packages/geany-0.18-6.fc12.i686.rpm
```

#**rpm -i geany-0.18-6.fc12.i686.rpm** flag in the above command means "*install*".

To upgrade an already installed package, - U option is used (where U stands for upgrade):

```
# rpm -U software-2.6.33.rpm
```

If everything goes well, the files in the package will get installed into the system and one can run the new program. The different files from the package gets placed in appropriate existing directories on the Linux system, executable programs go usually to /bin, /usr/bin, /usr/X11/bin or /usr/X11R6/bin after installing with rpm. The program automatically gets added to the menu or one can run the program by typing its name in the command prompt. One main reason for installed programs to go wrong is *failed dependencies* where RPM gives a list of files the program needs. One has to install or upgrade these dependency files one by one before installing the program.

Removing/Un-installing software that is installed with rpm:

```
# rpm -e software-geany.rpm
```

The "-e" option used here means "*erase*" (un-install).

One can just give the software name (without mentioning .rpm or version no) to remove the software

```
# rpm -e software
```

Note that <package> is different from <filename> used when installing. For example, for installing an application called "mysoftware", one may use a command like "rpm -i mysoftware-1.0.2-i386.rpm" to install "mysoftware", when removing we don't follow the filename for installation, but rather the name of the software itself is enough.

Querying the RPM Database: One can query the database to get information of the packages on the Linux system. To query a single package one has to use -q option.

```
# rpm -q software, The system gives the version of the package or a message that the package is not installed.
```

To list all the packages installed on the system, one has to query all with -qa

```
# rpm -qa
```

For further uses of RPM, please use "rpm --help" and "man rpm"

## 1.2 - Installing software with yum (fedora)

**yum** (Yellowdog Updater, Modified) unlike rpm, takes care of all the dependencies while installing a program on a fedora or rpm based system. Login as “root” user.

To see a list of available software

```
# yum list
```

To install a software

```
# yum install packagename
```

To update a software

```
# yum update packagename
```

Note: If you don't mention any "packagename", yum will update all your software.

To see what updates are available, you can do

```
# yum check-update
```

To search for a package, you can do

```
# yum search packagename
```

### 1.3 - Installing software with apt-get (Ubuntu)

APT (Advance Packaging Tool) is a package management system for Debian software package. It consists of different tools, whose names usually begins with "apt-" : apt-get, apt-cache, apt-cdrom, etc. Unlike RPM, whose equivalent in a Debian system would probably be DPKG, apt-get handles dependencies resolution and takes care of downloading the software for you (much like YUM in a Fedora/Red Hat system). One must have "*root*" privileges to execute apt-get or aptitude commands. Execute '*su*' in Debian and prefix '*sudo*' in Ubuntu to gain "*root*" privileges.

"apt-get" depends on Debian packages repositories (where are stored both sources and binary packages) that can be configured in the file /etc/apt/sources.list. Once the user has a sources.list adapted to his/her needs, the local list of packages needs to be updated and only then can the repositories be browsed with apt-cache.

```
# apt-get update
```

To search a package from its text description :

```
# apt-cache search <something>
```

Replace <something> with an application name or word. For example,

```
# apt-cache search irc client, will display a list of several irc clients.
```

To know more about a package and its description (dependencies, functionalities, maintainer's identity, etc.), type:

```
# apt-cache show <package_name>
```

In this case you have to replace <package\_name> with the exact package name.

Installing a binary package is done in one single step :

```
# apt-get install <package_name>
```

Un-installing/Removing a package is done like this :

```
# apt-get remove <package_name>
```

Or if you wish to remove the package along with all of its configuration files (essentially doing a clean un-install):

```
# apt-get remove --purge <package_name>
```

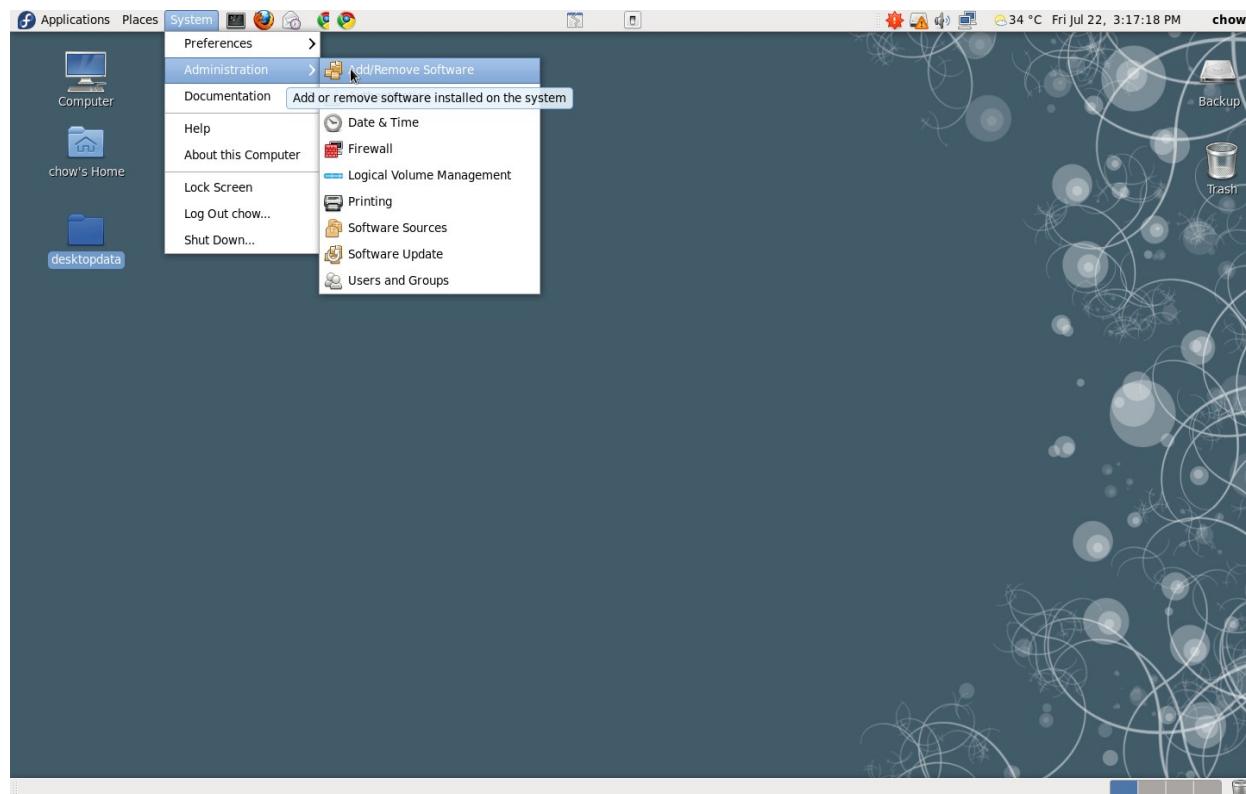
## 2. Installing software using GUI Tool:

### 2.1 - Installing software using Fedora's Add/Remove Software

One can use the “*Add/Remove software*” option from the system menu to install, upgrade or remove a package.

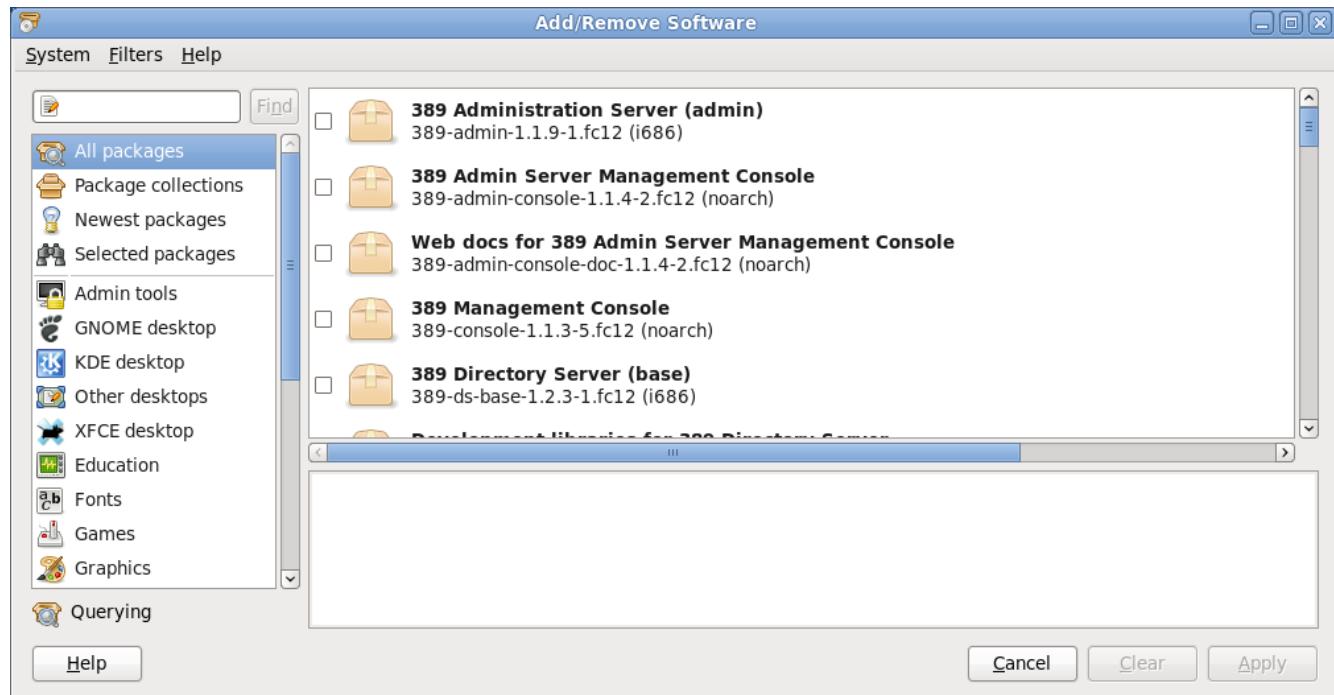
**System > Administration > “Add/Remove Software”.** This can be seen in **Figure 1**.

The Add/Remove Software window will open on the screen. This window is divided into four sections: a “package browser” on the left, the “package list” on the right upper half of the screen, “package description” on the lower left and “package details” (project’s license ,category, download size and source) on the right lower half of the screen.



**Figure 1.**

Lets install GIMP a graphics editing tool in Gnu/Linux. Select “Graphics” in the package browser, it will give a list of packages under “Graphics”, browse through this list and select the GNU Image Manipulation Program (GIMP). To select just check the check-box preceding GIMP and then click “*apply*”. This can be seen in **Figure 2**.

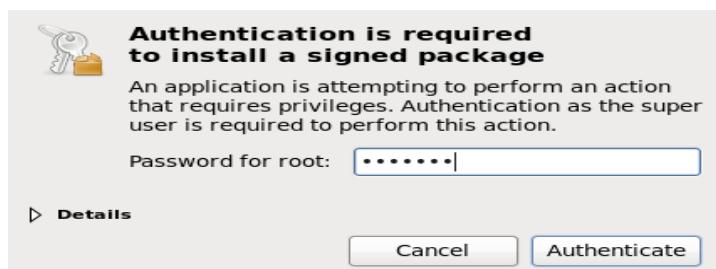


**Figure 2.**

When selecting the Graphics option in package browser no software packages to be listed on the right side window

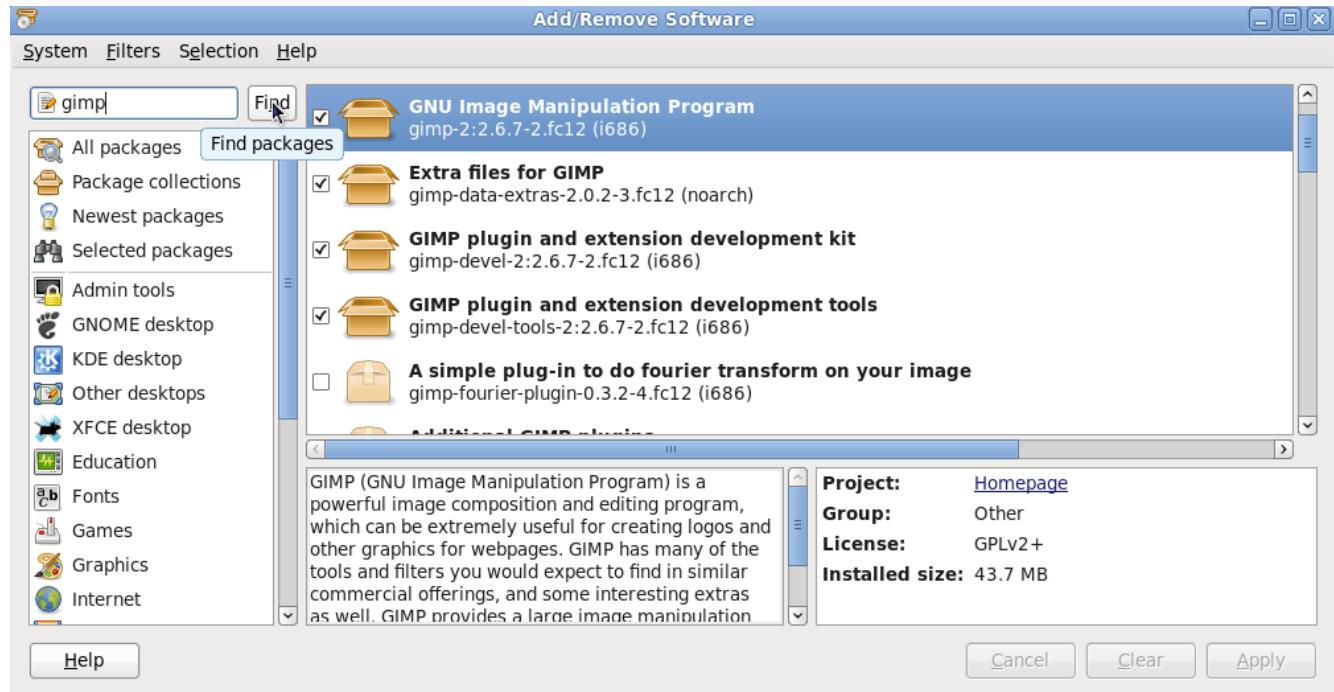
This will prompt the authenticate screen asking for root password. Type the root user password correctly and it will start downloading and installing the program. This can be seen in **Figure 3**.

.It may ask for the password



**Figure 3.**

The “downloading packages” message is displayed just above the “help” button on the bottom left of the screen along with a progress bar. This will take care of any dependencies the package might have. This can be seen in **Figure 4**.



**Figure 4.**

After installation a window with “close” and “run” button appears. To launch the program click “run” or click “close” button to close the window. This can be seen in the **Figure 5**.



**Figure 5.**

## 2.2 – Installing software using Ubuntu's Synaptic

Synaptic is a graphical front-end to apt, the package management system in Ubuntu. It combines the point-and-click simplicity of the graphical user interface with the power of the apt-get command line tool. You can install, remove, configure, or upgrade software packages, browse, sort and search the list of available software packages, manage repositories, or upgrade the whole system. You can queue up a number of actions before you execute them. Synaptic will inform you about dependencies (additional packages required by the software package you have chosen) as well as conflicts with other packages that are already installed on your system.

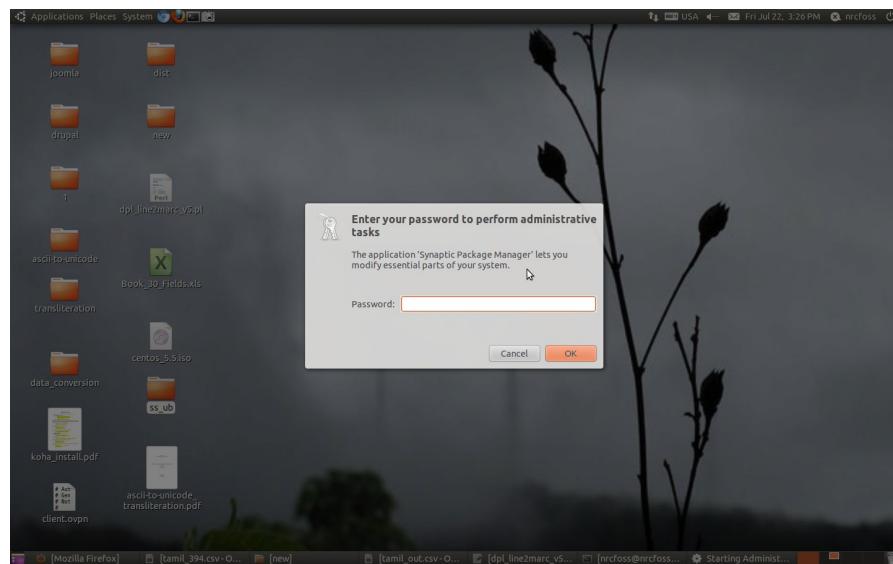
To launch Synaptic, choose (see **Figure 6**)

**System > Administration > Synaptic Package Manager**



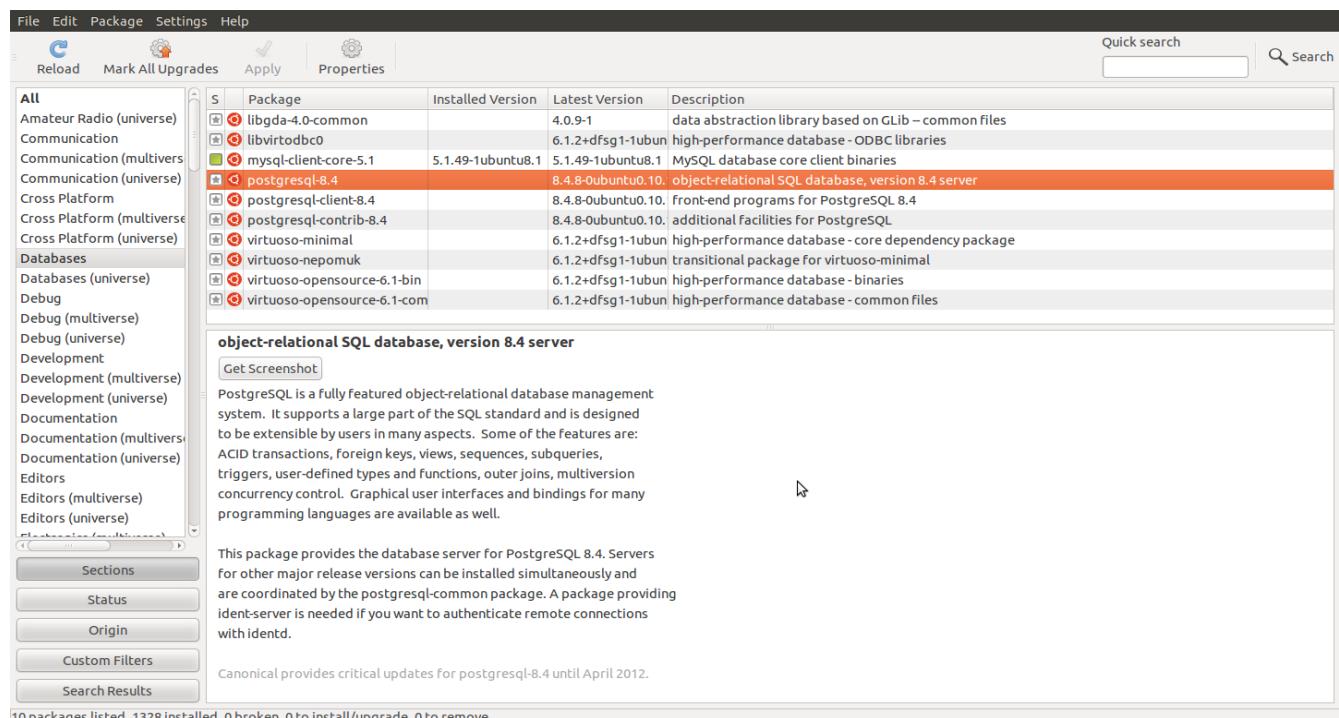
**Figure 6.**

This prompts the authentication screen asking for password. If the password entered is correct it will open the “Synaptic Package Manager” window, see **Figure 7**.



**Figure 7.**

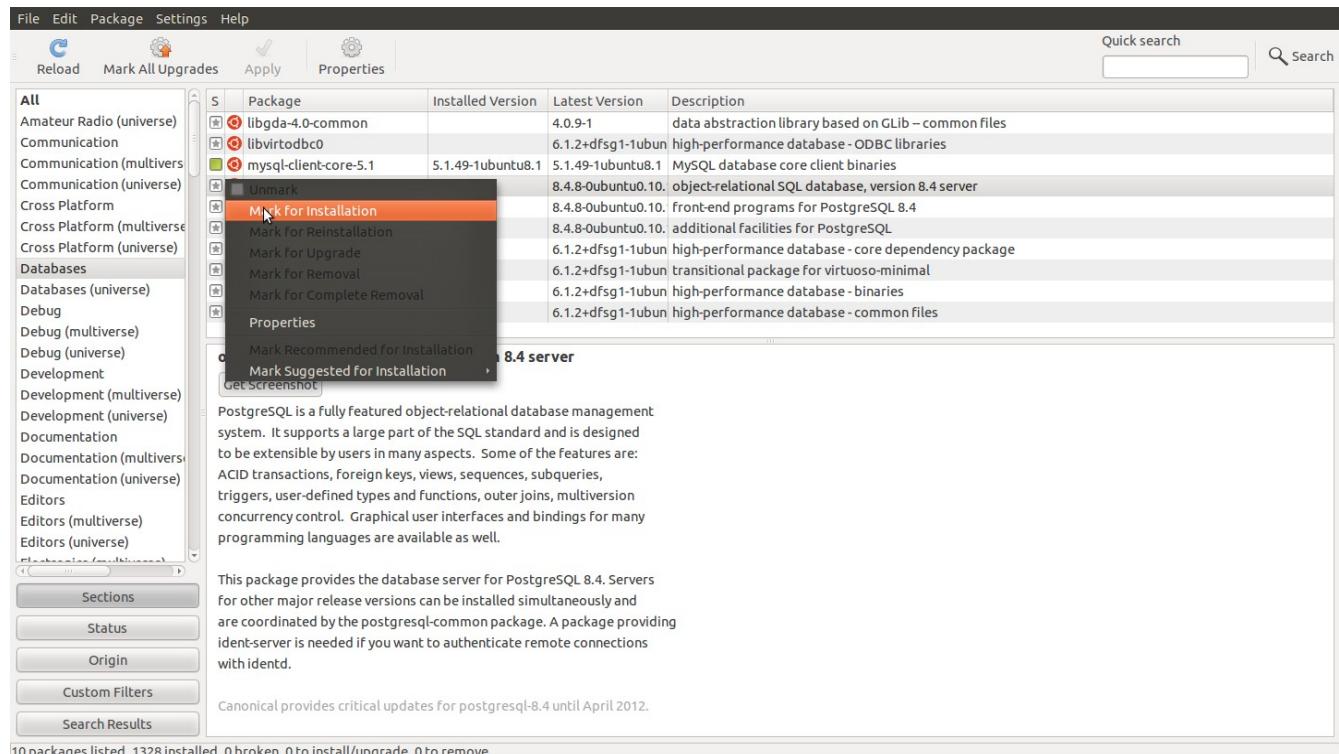
This window is divided into three sections: a “package browser” on the left, the “package list” on the upper right, and “package details” on the lower right. The status bar shows you the system state at a glance as in **Figure 8**.



**Figure 8.**

## CS2406 – Lab Manual – NRCFOSS/AU-KBC Centre, Anna University Chennai

Select a category from the “package browser” and it will give a list of packages belonging to the category selected. One can directly type the software package directly in the “Quick search” text box on the top right corner of the screen and it will display the software package from the package list. The package has its respective check-box which indicates (based on its color) whether the software is already installed or not. Select the package by right-clicking on the check-box and selecting “**mark for installation**”. The package is selected for installation (a tick mark is placed against the check-box) as shown in **Figure 9.**



**Figure 9.**

This will open a window (**Figure 10**) showing all the additional software (dependencies) that will be installed with this package. Select the “**Mark**” button and it will lead back to the “Synaptic Package Manager” window (**Figure 11**).

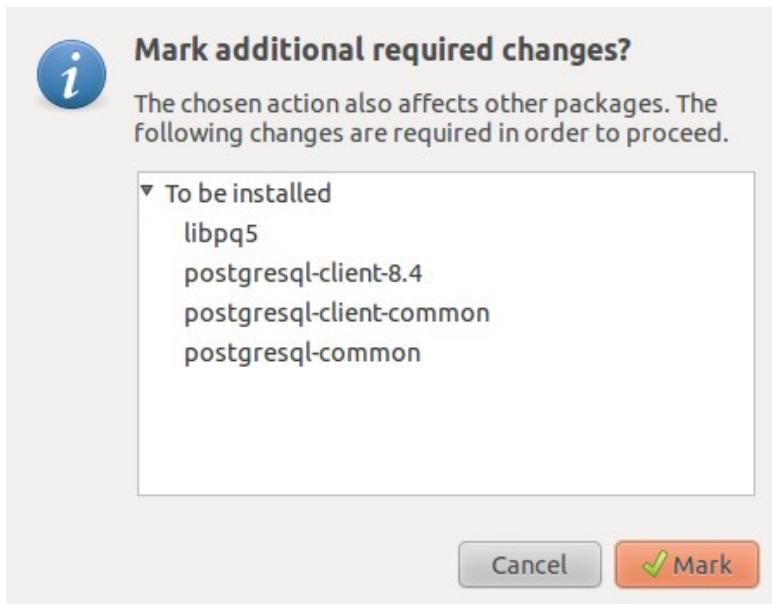


Figure 10.

Now click the “Apply” Tab.

File Edit Package Settings Help

Quick search  Search

S	Package	Installed Version	Latest Version	Description
<input type="checkbox"/>	libgda-4.0-common	4.0.9-1	data abstraction library based on GLib – common files	
<input type="checkbox"/>	libvirtodbc0	6.1.2+dfsg1-1ubuntu	high-performance database - ODBC libraries	
<input type="checkbox"/>	mysql-client-core-5.1	5.1.49-1ubuntu8.1	MySQL database core client binaries	
<input checked="" type="checkbox"/>	postgresql-8.4	8.4.8-0ubuntu0.10	object-relational SQL database, version 8.4 server	
<input checked="" type="checkbox"/>	postgresql-client-8.4	8.4.8-0ubuntu0.10	front-end programs for PostgreSQL 8.4	
<input type="checkbox"/>	postgresql-contrib-8.4	8.4.8-0ubuntu0.10	additional facilities for PostgreSQL	
<input type="checkbox"/>	virtuso-minimal	6.1.2+dfsg1-1ubuntu	high-performance database - core dependency package	
<input type="checkbox"/>	virtuso-nepomuk	6.1.2+dfsg1-1ubuntu	transitional package for virtuso-minimal	
<input type="checkbox"/>	virtuso-opensource-6.1-bin	6.1.2+dfsg1-1ubuntu	high-performance database - binaries	
<input type="checkbox"/>	virtuso-opensource-6.1-com	6.1.2+dfsg1-1ubuntu	high-performance database - common files	

**object-relational SQL database, version 8.4 server**

[Get Screenshot](#)

PostgreSQL is a fully featured object-relational database management system. It supports a large part of the SQL standard and is designed to be extensible by users in many aspects. Some of the features are: ACID transactions, foreign keys, views, sequences, subqueries, triggers, user-defined types and functions, outer joins, multiversion concurrency control. Graphical user interfaces and bindings for many programming languages are available as well.

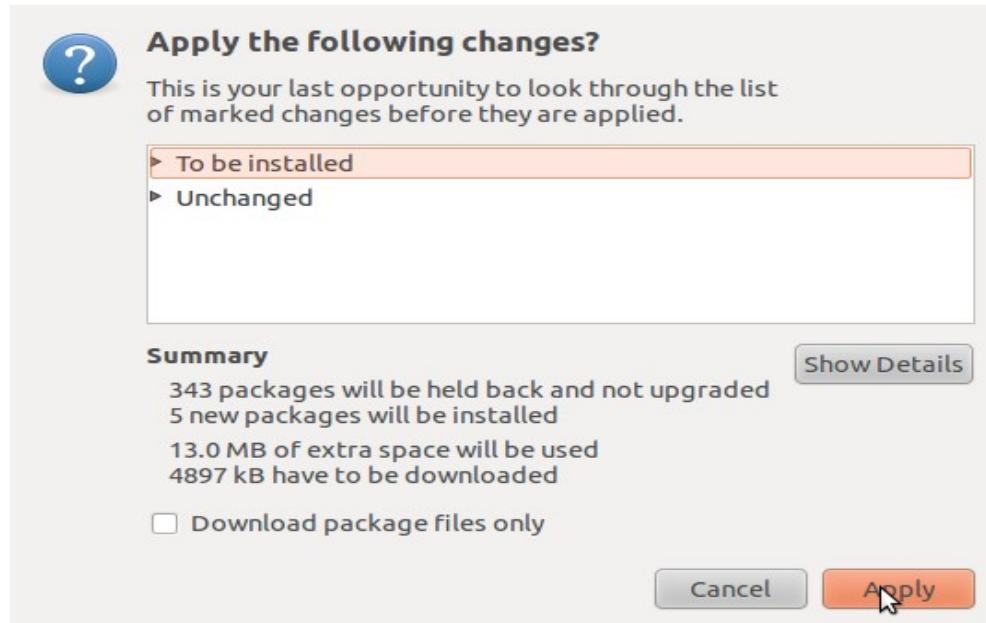
This package provides the database server for PostgreSQL 8.4. Servers for other major release versions can be installed simultaneously and are coordinated by the postgresql-common package. A package providing ident-server is needed if you want to authenticate remote connections with identd.

Canonical provides critical updates for postgresql-8.4 until April 2012.

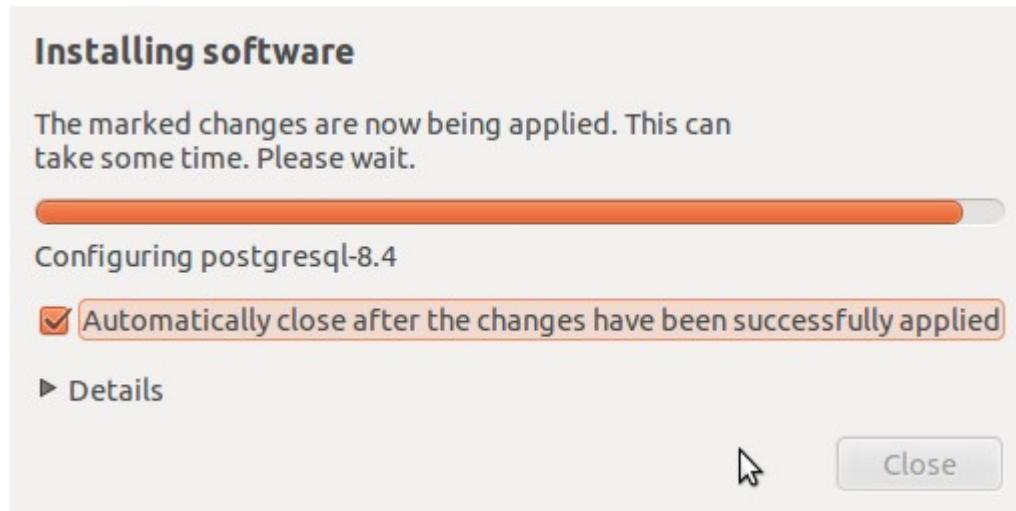
10 packages listed, 1328 installed, 0 broken. 5 to install/upgrade, 0 to remove; 13.0 MB will be used

Figure 11.

This will open a window (**Figure 12**) showing the list of changes along with a summary of number of packages upgraded, installed and held back along with the amount of space that will be used and the size of software that will be downloaded. Click “Apply” to start downloading and installing the software (**Figure 13**).

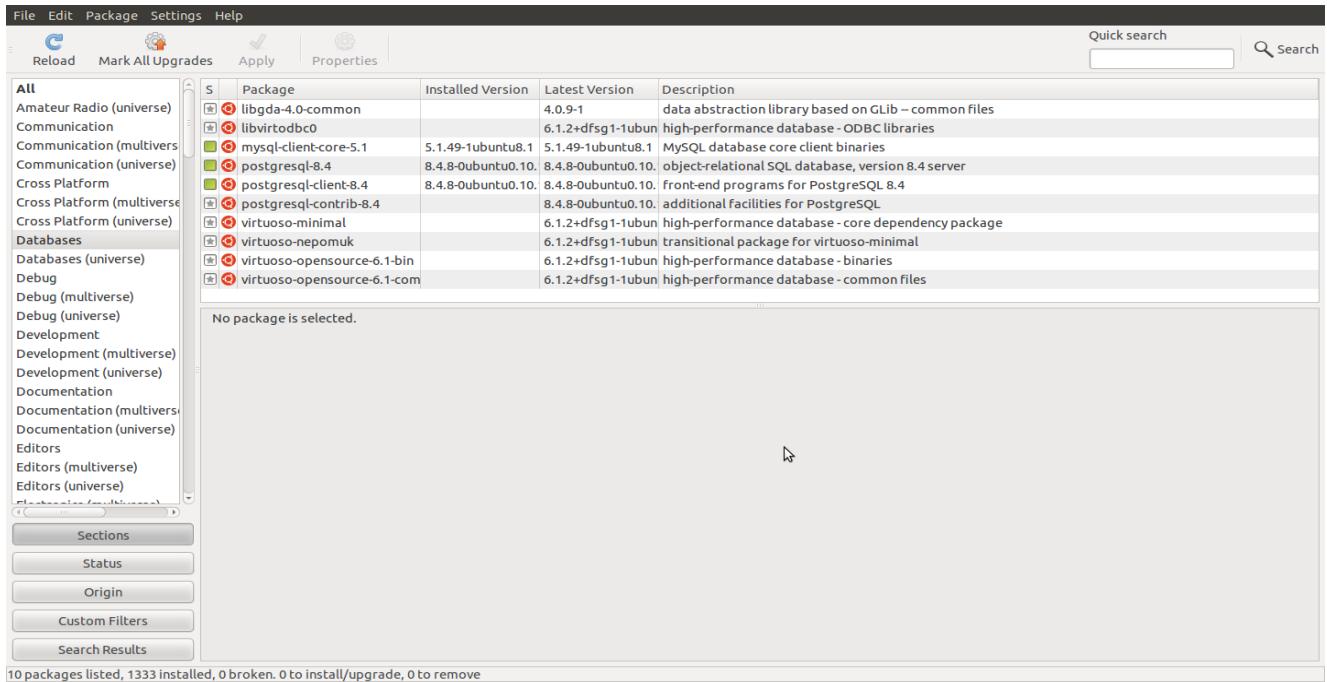


**Figure 12.**



**Figure 13.**

Once the software package is installed, the check-box preceding the software package will be in green color (**Figure 14**).



**Figure 14.**

For un-installing or removing a package, one has to identify the package and right click on check-box, select the “Mark for Remove” option and click on the “Apply” tab. Similarly one can re-install and the selected package by selecting “Mark for re-installation”.

[http://docs.fedoraproject.org/en-US/Fedora\\_Draft\\_Documentation/0.1/html/RPM\\_Guide/](http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/)

<http://yum.baseurl.org/wiki/YumCommands>

<http://www.debian.org/doc/manuals/apt-howto/>

<https://help.ubuntu.com/community/SynapticHowto>

## **5.b : Installing various software packages - Advanced Samba, CUPS, Django.**

### **Aim:**

To know the how to install the packages like Samba, Common Unix Printing System (CUPS) and Django.

### **1. Samba Installation and Configuration**

#### **1.1 Introduction to Samba**

Samba can be used with Linux to provide transparent access between machines running Linux and machines running Windows. Samba itself runs on a Linux machine and makes shared files and printers available to Windows machines, as if they are available on a Window machine or server. Thus, Samba has several practical applications which can generally be categorized as follows:

1. Using a Linux server as a simple peer-to-peer server. There is no user authentication involved and no need for passwords.
2. Using a Linux server as a member server on an existing Windows NT domain. The existing Windows domain controller will use NT authentication tools to control file permissions and access.
3. Using a Linux server as a primary domain controller with its own user authentication and control mechanisms.

Which of these three applications of Samba is used, determines how Samba is configured on the Linux machine.

#### **1.2 Installing Samba**

```
$ su -  
Enter Password:  
(Here enter the root password)
```

```
# yum install samba  
Checking the samba running.  
# /etc/init.d/smb status
```

Starting the samba service

```
#/etc/init.d/smb start
```

Restart the samba service

```
#/etc/init.d/smb restart
```

Creating a normal user

```
# useradd samba-test
```

## Configuring samba

Creating a normal user

```
# useradd samba-test
```

Setting the password

```
# passwd samba-test
```

Changing password for user samba-test.

New password:

Retype new password:

```
passwd: all authentication tokens updated successfully.
```

Adding the user to access samba shares with a password

```
# smbpasswd -a samba-test
```

New SMB password:

Retype new SMB password:

Added user samba-test.

Login into the samba as using the smbclient to add the files and folders to share with windows.

```
# smbclient \\\localhost\\samba-test -Usamba-test
```

Here enter smb password for login

Enter samba-test's password:

You gets the following output to see the what are files you are sharing

Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.8-68.fc15]

smb: \>

In this to see what are files you sharing enter 'ls' command

smb: \> ls

.	D	0	Tue Aug 9 19:08:44 2011
..	D	0	Tue Aug 9 19:08:44 2011
.bash_logout	H	18	Wed Jun 22 19:19:05 2011
.mozilla	DH	0	Wed Feb 9 03:52:39 2011
.bash_profile	H	193	Wed Jun 22 19:19:05 2011
.gnome2	DH	0	Tue Feb 8 14:08:01 2011

.bashrc H 124 Wed Jun 22 19:19:05 2011

56633 blocks of size 2097152. 7531 blocks available

To see the where your current directory located enter the ‘pwd’ command

smb: \> pwd

Current directory is \\localhost\samba-test\

To create a folder in sharing folder enter following command

smb: \> mkdir test

To check created directory enter the following command

smb: \> ls

.	D	0	Tue Aug 9	19:10:50	2011
..	D	0	Tue Aug 9	19:08:44	2011
.bash_logout	H	18	Wed Jun 22	19:19:05	2011
.mozilla	DH	0	Wed Feb 9	03:52:39	2011
.bash_profile	H	193	Wed Jun 22	19:19:05	2011
.gnome2	DH	0	Tue Feb 8	14:08:01	2011
test	D	0	Tue Aug 9	19:10:50	2011
.bashrc	H	124	Wed Jun 22	19:19:05	2011

56633 blocks of size 2097152. 7532 blocks available

If you want to come out from samba client enter the ‘exit’

smb: \> exit

## How to access the shared files from windows.

Open the samba location as following.

Start-> Run..

Enter the ip address of linux machine as following

eg: \\192.168.1.2

It displays log in window to log into the samba sever.

Here username and password should be what you given to while creating samba username and smb password.

eg:     username: samba-test  
          password:test123

Now you can see the what are files shared.

## 2. Common Unix Printing System (CUPS)

Printing within UNIX has historically been done using one of two printing systems - the Berkeley Line Printer Daemon ("LPD") [RFC1179] and the AT&T Line Printer system. These printing systems were designed in the 70's for printing text to line printers; vendors have since added varying levels of support for other types of printers.

### Linux Printing features

- All printing systems on Linux make use of the excellent PostScript system called GhostScript ([www.ghostscript.org](http://www.ghostscript.org)).
  - Ghostscript is a PostScript interpreter that is most commonly used on Linux.
  - Implements an excellent PostScript engine that can take as inputs formats like JPEG,
- TIFF, PS & Text and output data in formats like X Windows output, raster formats and PDF.
  - Also handles conversion of PS output for non-PS printers; can also be used a basic, spooler less printing system.
  - Most printing systems today use a combination of multiple tools (postscript interpreters, filters, rasterises etc) to process and print documents.
  - The printing system converts PostScript into a raster format and then converts that into a printer specific language to send commands to the printer.

### Installing CUPS

To Install CUPS Enter the Following command as *root* user

```
$ su -  
# yum install cups
```

Install the A printer administration tool  
# yum install system-config-printer

### Configuring CUPS

Open the cups by typing following URL  
<http://localhost:631/>

**CUPS 1.4.1**

CUPS is the standards-based, open source printing system developed by [Apple Inc.](#) for Mac OS® X and other UNIX®-like operating systems.

**CUPS for Users**

- [Overview of CUPS](#)
- [Command-Line Printing and Options](#)
- [What's New in CUPS 1.4](#)
- [User Forum](#)

**CUPS for Administrators**

- [Adding Printers and Classes](#)
- [Managing Operation Policies](#)
- [Printer Accounting Basics](#)
- [Server Security](#)
- [Using Kerberos Authentication](#)
- [Using Network Printers](#)
- [cupsd.conf Reference](#)
- [Find Printer Drivers](#)

**CUPS for Developers**

- [Introduction to CUPS Programming](#)
- [CUPS API](#)
- [Filter and Backend Programming](#)
- [HTTP and IPP APIs](#)
- [PPD API](#)
- [Raster API](#)
- [PPD Compiler Driver Information File Reference](#)
- [Developer Forum](#)

In this click on the [Adding Printers and Classes](#)

Here click on the Add Printer.

**Printers**

- [Add Printer](#)
- [Find New Printers](#)
- [Manage Printers](#)

**Classes**

- [Add Class](#)
- [Manage Classes](#)

**Jobs**

- [Manage Jobs](#)

**RSS Subscriptions**

- [Add RSS Subscription](#)

Name	Events	Queue Name
(notify_recipient_name)	printer-state-changed, printer-restarted, printer-shutdown, Printer Paused, Printer Added, Printer Deleted, job-state-changed, Job Created, Job Completed, Job Stopped	All Queues
<a href="#">Cancel RSS Subscription</a>		

**Server**

- [Edit Configuration File](#)
- [View Access Log](#)
- [View Error Log](#)
- [View Page Log](#)

**Server Settings:**

**Advanced ▾**

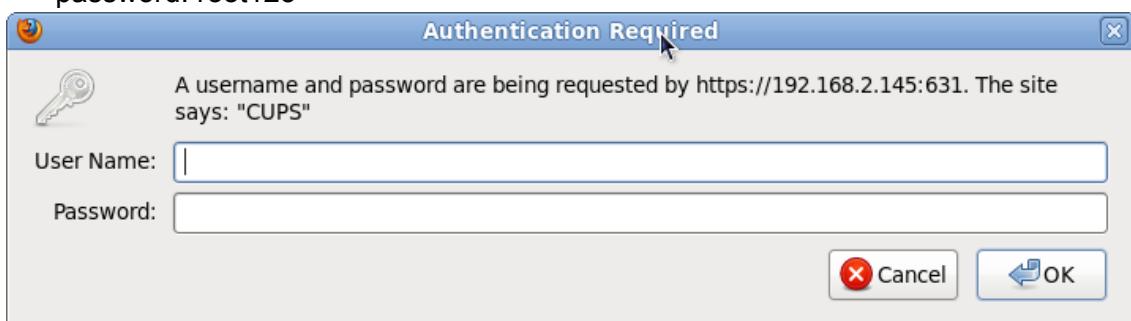
- Show printers shared by other systems
- Share printers connected to this system
  - Allow printing from the Internet
- Allow remote administration
- Use Kerberos authentication ([FAQ](#))
- Allow users to cancel any job (not just their own)
- Save debugging information for troubleshooting

[Change Settings](#)

After clicking on the Add Printer It will ask you for permission. Here you have to enter the system root username and password

eg: User name: root

password: root123



Then CUPS search for the printers available in your network.

A screenshot of a Mozilla Firefox browser window titled 'Add Printer - CUPS 1.4.1 - Mozilla Firefox'. The address bar shows 'http://localhost:631/admin/'. The main content area is titled 'Add Printer' and contains the following text:

\* Looking for printers...

**Local Printers:**  SCSI Printer  
 Serial Port #1  
 Serial Port #2  
 HP Printer (HPLIP)  
 HP Fax (HPLIP)

**Discovered Network Printers:**  PS-978183-U1  
 HP LaserJet P1505n (Hewlett-Packard HP LaserJet P1505n)

**Other Network Printers:**  AppSocket/HP JetDirect

At the bottom left, there is a progress bar labeled 'Read localhost'.

If your printer is connected to the network it shows in the Discovered Network Printers. Select the printer and click on **Continue** button.

**Add Printer**

**Local Printers:**

- SCSI Printer
- Serial Port #1
- Serial Port #2
- HP Printer (HPLIP)
- HP Fax (HPLIP)

**Discovered Network Printers:**

- PS-978183-U1
- HP LaserJet P1505n (Hewlett-Packard HP LaserJet P1505n)

**Other Network Printers:**

- AppSocket/HP JetDirect
- Internet Printing Protocol (ipp)
- Internet Printing Protocol (https)
- Internet Printing Protocol (http)
- Backend Error Handler
- LPD/LPR Host or Printer
- Windows Printer via SAMBA

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

Done

In following window give the name of printer and also description and location. At the location give your Lab name.

**Add Printer**

**Name:**   
(May contain any printable characters except ":", "#", and space)

**Description:**   
(Human-readable description such as "HP LaserJet with Duplexer")

**Location:**   
(Human-readable location such as "Lab 1")

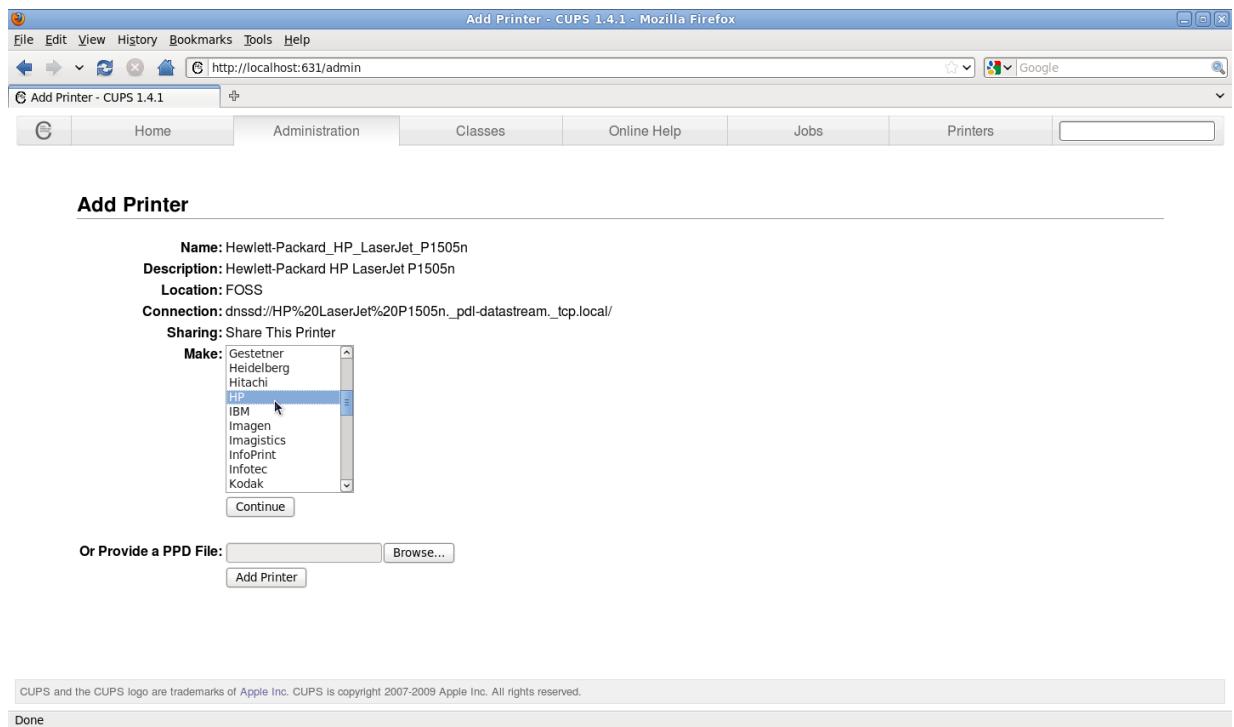
**Connection:**

**Sharing:**  Share This Printer

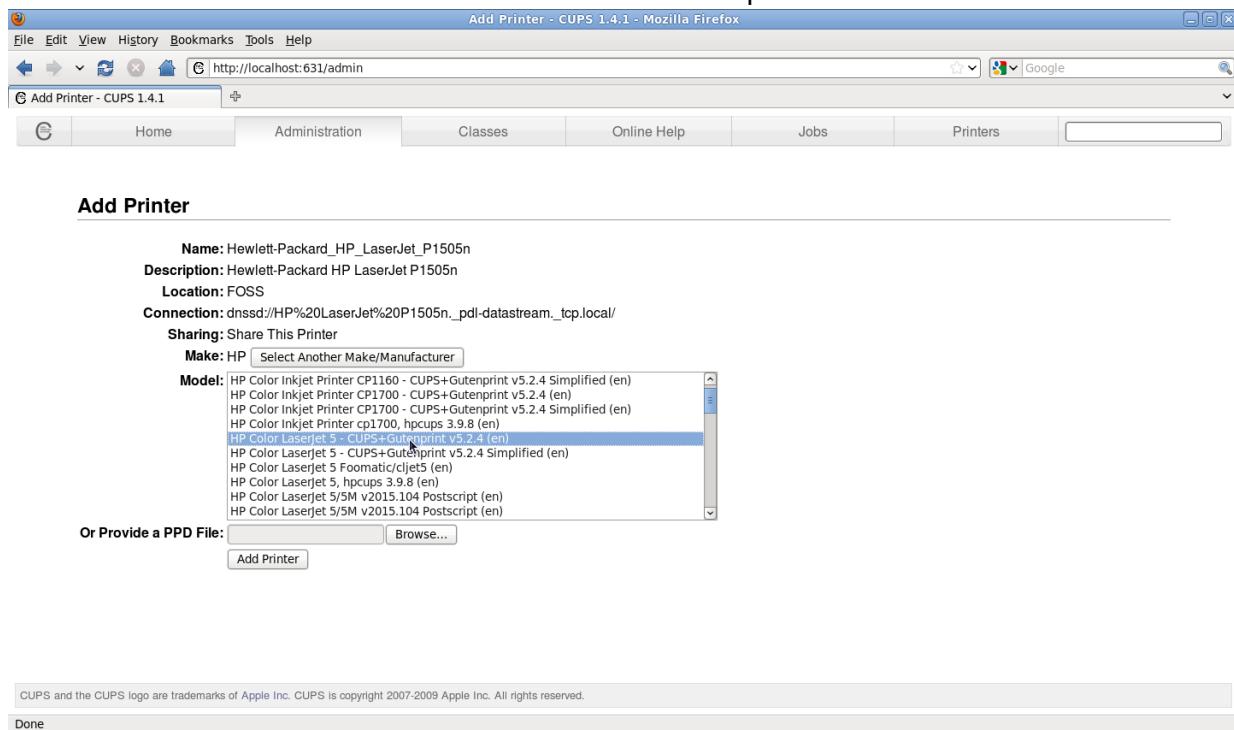
CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

Done

Here select the Maker of the Printer and click on the add printer.



Here select the Model of the Printer and click on the add printer.



Here you have to set the default options for printer. To set it click on the set default options. This is to manage paper size and also color and Quality.

**Set Default Options for Hewlett-Packard\_HP\_LaserJet\_P1505n**

**General   Output Control Common   Output Control Extra 1   Output Control Extra 2   Output Control Extra 4   Banners   Policies**

**General**

Media Size: Letter  
 Color Model: Grayscale  
 Color Precision: Normal  
 Media Source: Standard  
 Print Quality: Standard  
 Resolution: Automatic  
 Shrink Page If Necessary to Fit Borders: Shrink (print the whole page)

**Set Default Options**

After setting the options you can see following screen. In this you can see the maintenance, in that drop down you can delete the printer and also Edit the printer.

Below that you can see the Jobs. In that you can see Jobs you did till now means if you took any print from your machine that status it shows.

**Hewlett-Packard\_HP\_LaserJet\_P1505n (Idle, Accepting Jobs, Shared)**

**Maintenance   Administration**

Description: Hewlett-Packard HP LaserJet P1505n  
 Location: FOSS  
 Driver: HP Color LaserJet 5 - CUPS+Gutenprint v5.2.4 (grayscale)  
 Connection: dnssd://HP%20LaserJet%20P1505n.\_pdll-datastream.\_tcp.local/  
 Defaults: job-sheets=none, none media=na\_letter\_8.5x11in

**Jobs**

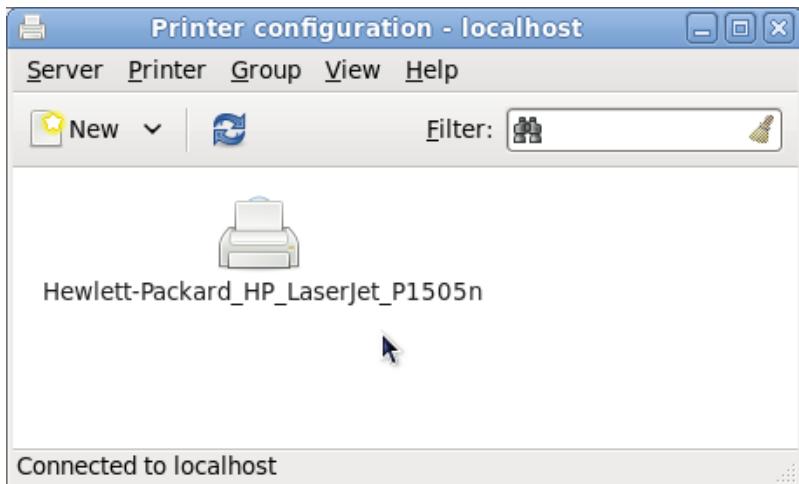
Search in Hewlett-Packard\_HP\_LaserJet\_P1505n:

No jobs.

CUPS and the CUPS logo are trademarks of Apple Inc. CUPS is copyright 2007-2009 Apple Inc. All rights reserved.

**Done**

If you want to see if your printer is configured correctly, Go to the menu name called System -> Administration -> Printing .



## 3. DJANGO

### 3.1 Introduction to DJANGO

Django is an web application framework, written in Python, which follows the **Model-View-Controller** (MVC) Pattern. It is named after **Django Reinhardt**, a gypsy jazz guitarist from the 1930s to early 1950s. To this day, he's considered one of the best guitarists of all time. Django is pronounced **JANG-oh**. Rhymes with FANG-oh. It was originally developed to manage several news-oriented sites for The World Company of Lawrence, Kansas, etc., and was released publicly under **BSD license** by July 2005. In June 2008, It was announced that a newly formed Django Software Foundation will take care of Django in the future.

### 3.2 Installing DJANGO

#### Installation

First, download **Django-x.x.tar.gz** from the following link

<http://FOSS-LAB-SERVER-IP/content/code/Django/Django-x.x.x.tar.gz>

Go to Applications -> System Tools -> **Terminal** and type the following commands

\$su -

Enter Password:

(Here enter the root user password)

# tar xzvf Django-x.x.tar.gz

# cd Django-x.x

#python setup.py install

**References:**

<http://www.samba.org/samba/docs/>

<http://www.cups.org/articles.php?L274>

<https://docs.djangoproject.com/en/dev/topics/install/?from=olddocs>

## 3.GUI Programming

### Aim :

Learn to write GUI programs using FOSS tools in Linux.

### Introduction :

GUI programming on Linux is based on the X Window system. X provides windowing on computer displays and manages keyboard, pointing device control functions and touch-screens. X provides the basic framework, or primitives, for building such GUI environments: drawing and moving windows on the display and interacting with a mouse, keyboard or touchscreen.

X uses GUI toolkits to help the development of GUI programs. **GUI toolkit** is a set of widgets for use in designing applications with graphical user interfaces (GUIs). GUI toolkits can be based on different languages. The most commonly used languages are C/C++. Often the toolkits provide higher level language bindings which allow the user to program GUIs using an easier to use higher level language like Python, Ruby etc.

There are two popular GUI toolkits which are widely used today. They are Qt and GTK+. Qt is written in C++ while GTK+ is written in C. Both toolkits provide bindings in a wide variety of languages.

For users who are familiar with Visual basic, Gambas is a full-featured object language and development environment built on a BASIC interpreter which runs on top of the X window system. It in turn relies on the underlying GUI toolkits (both Qt and GTK+).

This exercise consists of developing GUI programs using the Qt toolkit.

### Description :

Students will learn to setup their systems to develop using Qt and write 5 programs and test their results. The programs are of increasing complexity and introduce the students to more and more concepts.

All the programs will be purely code driven, ie. the user interface is developed entirely using C++ code. None of the visual GUI builders will be used for these exercises.

**Pre-requisites:**

1. In the commandline as root, type  
> yum install qt-devel qt-demos qt-examples qt-doc

**The Programs:**

**A GUI Hello World.**

- o Create a new directory for the program.

```
> mkdir qthello
```

- o Go to the newly created directory.

```
> cd qthello
```

- o create the file in the qthello directory.

```
> gedit qthello.cpp
```

```
//qthello.cpp
//Include file for Qt
#include <QtGui>
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QWidget window;
    //resize window to 320x240
    window.resize(320, 240);
    window.setWindowTitle("Hello World!");
    //Show the window
    window.show();
    //Start the event loop
    return app.exec();
}
```

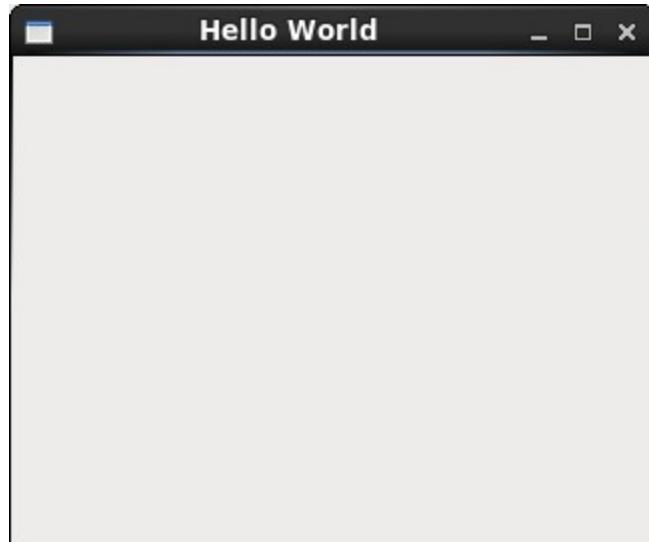
- o Once the file has been saved, use the following commands to compile and execute the program.

```
> qmake-qt4 -project
```

```
> qmake-qt4
```

```
> make  
> ./qthello
```

- o The output will be a window like this:



**Window with a button.**

- o Use the following code to create a window with a button:
- o Create a new directory for the program.

```
> mkdir qtbutton  
  
          o Go to the newly created directory.  
  
> cd qtbutton  
  
          o create the file in the qtbutton directory.  
  
> gedit qtbutton.cpp
```

```
#include <QtGui>  
  
int main(int argc, char *argv[])  
{  
    QApplication app(argc, argv);  
    QWidget window;
```

```
window.resize(320, 240);
window.setWindowTitle("Window with a button");
window.show();

//Create a button with the text "Press me"
//as a child of the main window
QPushButton *button = new QPushButton("Press me", &window);
//move the button into position
button->move(100, 100);
button->show();
return app.exec();
}
```

- Once the file has been saved, use the following commands to compile and execute the program.

```
> qmake-qt4 -project
> qmake-qt4
> make
> ./qtbutton
```

- The output will be a window like this:



### **Using Layouts.**

- Create a window with two widgets managed by a layout manager.

- Create a new directory for the program.

```
> mkdir qtlayout
```

- Go to the newly created directory.

```
> cd qtlayout
```

- create the file in the qtlayout directory.

```
> gedit qtlayout.cpp
```

```
//qtlayout.cpp
#include <QtGui>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QWidget window;

    //Create a label an a single line text box
    QLabel *label = new QLabel("Name:");
    QLineEdit *lineEdit = new QLineEdit();

    //Create a layout. Add the label and the lineedit to it.
    QHBoxLayout *layout = new QHBoxLayout();
    layout->addWidget(label);
    layout->addWidget(lineEdit);
    //Apply the layout to the main window.
    //Since the widgets are part of the layout,
    //they are now children of the window.
    window.setLayout(layout);
    window.setWindowTitle("Window layout");
    window.show();
    return app.exec();
}
```

- Once the file has been saved, use the following commands to compile and execute the program.

```
> qmake-qt4 -project
> qmake-qt4
> make
> ./qtlayout
```

- Compile and execute the program to get the following:



### Signals and Slots

- This program demonstrates the use of signals and slots to make two widgets interact with each other.

The entire application is divided into 3 files:

1. communicate.h
2. communicate.cpp
3. main.cpp

- Create a new directory for the program.

```
> mkdir qtsignals
```

- Go to the newly created directory.

```
> cd qtsignals
```

- create the file in the qtsignals directory.

```
> gedit communicate.h
```

```
//communicate.h
#include <QWidget>
#include <QApplication>
#include <QPushButton>
#include <QLabel>

class Communicate : public QWidget
{
//The Q_OBJECT macro causes the moc tool to initialise
//code for signals and slots, run time type information
//and dynamic property system
    Q_OBJECT

public:
    Communicate(QWidget *parent = 0);
```

```
//add a lot which allows widget communications
private slots:
    void add();

private:
    QLabel *label;

};

        o  create the file in the qtsignals directory.
> gedit communicate.cpp

//communicate.cpp
#include "communicate.h"
#include <QDesktopWidget>

Communicate::Communicate(QWidget *parent)
    : QWidget(parent)
{
    resize(180, 140);

    QPushButton *plus = new QPushButton("+", this);
    plus->setGeometry(50, 40, 50, 30);

    label = new QLabel("0", this);
    label->setGeometry(120, 40, 20, 30);

//Connect the clicked event of the button to
//the add method of the class
    connect(plus, SIGNAL(clicked()), this, SLOT(add()));
}

void Communicate::add()
{
    //Change the text displayed in the label
    int val = label->text().toInt();
    val++;
    label->setText(QString::number(val));
}

        o  create the file in the qtsignals directory.
> gedit main.cpp
```

```
//main.cpp
#include "communicate.h"

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    Communicate window;

    window.setWindowTitle("Communicate");
    window.show();

    return app.exec();
}
```

- Once the file has been saved, use the following commands to compile and execute the program.

```
> qmake-qt4 -project
> qmake-qt4
> make
> ./qtsignals
```

On compiling and running the program, the following window is displayed. The number can be incremented by pressing the plus button.



#### **Menus and Toolbars.**

- This program will display a menu which can be used to close the program. Once again we have 3 files as part of the program:

1. mymenu.h

2. mymenu.cpp
3. main.cpp

o Create a new directory for the program.

> mkdir qtmenu

o Go to the newly created directory.

> cd qtmenu

o create the file in the qtmenu directory.

> gedit mymenu.h

```
//mymenu.h
#include < QMainWindow >
```

```
class MyMenu : public QMainWindow
{
public:
    MyMenu(QWidget *parent = 0);
};
```

o create the file in the qtmenu directory.

> gedit mymenu.cpp

```
//mymenu.cpp
#include "mymenu.h"
#include <QMenu>
#include <QMenuBar>
#include <QApplication>
```

```
MyMenu::MyMenu(QWidget *parent)
    : QMainWindow(parent)
{
    //create the quit action object
    QAction *quit = new QAction("&Quit", this);

    //create the file menu
    QMenu *file;
    file = menuBar()->addMenu("&File");
```

```
//add the quit action to the new menu  
file->addAction(quit);  
  
//connect the triggered signal from the quit action menu  
//to the global quit method which closes the application  
connect(quit, SIGNAL(triggered()), qApp, SLOT(quit()));  
}
```

- o create the file in the qtmenu directory.

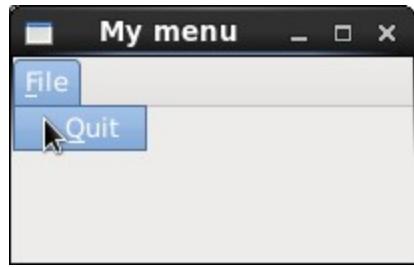
```
> gedit main.cpp
```

```
//main.cpp  
#include "mymenu.h"  
#include <QDesktopWidget>  
#include <QApplication>  
  
int main(int argc, char *argv[]){  
    QApplication app(argc, argv);  
  
    MyMenu window;  
  
    window.setWindowTitle("My menu");  
    window.show();  
  
    return app.exec();  
}
```

- o Once the file has been saved, use the following commands to compile and execute the program.

```
> qmake-qt4 -project  
> qmake-qt4  
> make  
> ./qtmenu
```

- o On running the program we get a window with a menu which allows us to close the program.



## References:

1. <http://doc.qt.nokia.com/> -- Qt documentation site.
2. <http://zetcode.com/tutorials/qt4tutorial/> -- A Qt4 tutorial

## 4 .Version Control system setup and usage

### I. Aim

- To setup a version control system for managing files using svn. The system should be able to track changes to the files/folders and a period of time and revert if necessary

### II. Introduction of the concept/tool

- Version control is the concept of managing all changes to a repository. The repository could be containing any type of file.
- In computer [software engineering](#), revision control is any practice that tracks and provides control over changes to source code. [Software developers](#) sometimes use revision control software to maintain documentation and [configuration files](#) as well as source code.
- As teams design, develop and deploy software, it is common for multiple versions of the same software to be deployed in different sites and for the software's developers to be working simultaneously on updates. [Bugs](#) or features of the software are often only present in certain versions (because of the fixing of some problems and the introduction of others as the program develops). Therefore, for the purposes of locating and fixing bugs, it is vitally important to be able to retrieve and run different versions of the software to determine in which version(s) the problem occurs. It may also be necessary to develop two versions of the software concurrently (for instance, where one version has bugs fixed, but no new features ([branch](#)), while the other version is where new features are worked on ([trunk](#))).
- At the simplest level, developers could simply retain multiple copies of the different versions of the program, and label them appropriately. This simple approach has been used on many large software projects. While this method can work, it is inefficient as many near-identical copies of the program have to be maintained. This requires a lot of self-discipline on the part of developers, and often leads to mistakes. Consequently, systems to automate some or all of the revision control process have been developed.
- Moreover, in software development, legal and business practice and other environments, it has become increasingly common for a single document or snippet of code to be edited by a team, the members of which may be geographically dispersed and may pursue different and even contrary interests. Sophisticated revision control that tracks and accounts for ownership of changes to documents and code may be extremely helpful or even necessary in such situations.
- Revision control may also track changes to configuration files, such as those typically stored in /etc or /usr/local/etc on Unix systems. This gives system administrators another way to easily track changes made and a way to roll back to earlier versions should the need arise.[1]

### III. Description of the exercise

- In this exercise, we would be setting up version control using svn. We would create a repository and then access that repository over http.
- Briefly, we would be doing the following
  - Install/setup svn repository
  - Configure apache for mod\_svn access. We would also password-protect svn access
  - Check in code/files using svn commands
  - Check out codes to another location
  - Modify/commit codes
  - Revert to an older version of the code

#### **IV. Pre-requisite**

- Fedora 12 installed with root access (root access for installation/configuration purposes)
- LAN connection
- Apache Webserver
- selinux disabled - “ setenforce 0 ” can be given at command line to disable it runtime and **reboot** the machine
- FOSS Lab server or internet connectivity for downloading packages from internet using yum

#### **V. Detailed steps for the exercise-**

**Note:** In the description (steps), all greyed out lines are actual commands/configurations done on the server

##### **Step 1 - Installation of packages**

The first thing to do is to install the packages required for the setup of svn. We also need to install/configure packages required for letting svn command be accessed over http

```
yum install mod_dav_svn subversion
```

Its recommended to install packages using yum as it will pull in any dependencies required for the installation

##### **Step 2 - Configuration of the repository**

a. The following commands create the necessary folder structure for the repository

```
mkdir /svn  
mkdir /svn/repos  
cd /svn/repos  
chown -R apache.apache ..../repos
```

b. Now we need to initialize the svn structure and create a repository

```
# svnadmin create svn-test
```

With this, the repository is configured

##### **Step 3 - Creation of a local area for code**

The purpose of this step is to have a staging area for code from where you would be doing some development. This code can then be committed to the svn

###### **a. Create the local folder structure**

```
# mkdir /home/<user>/svn-test  
# cd /home/<user>/svn-test
```

###### **b. Create local directories as per requirement - This can be customized to suit your needs**

```
# mkdir configurations options main
```

###### **c. Edit any file as required**

```
vim configurations/test.cfg
```

Edit and save any other file required.

#### **Step 4 : Import of the code area onto svn**

##### **a. Import the initial structure onto svn**

```
# svn import /home/<user>/svn-test/ file:///svn/repos/svn-test/ -m "Initial Import"
Adding    /root/svn-test/main
Adding    /root/svn-test/configurations
Adding    /root/svn-test/configurations/test.cfg
Adding    /root/svn-test/options
```

Committed revision 1.

This imports the files/folders onto the repo with a log message of “Initial Import”

```
# chown -R apache.apache /svn/repos/
for fixing the permissions of the newly committed repo to be accessible by apache
```

#### **Step 5 : Configuring Apache**

##### **a. Edit subversion.conf at /etc/httpd/conf.d/subversion.conf and add the following lines**

```
# gedit /etc/httpd/conf.d/subversion.conf
```

```
<Location /repos>
  DAV svn
  SVNParentPath /svn/repos
  AuthType Basic
  AuthName "Subversion repos"
  AuthUserFile /etc/svn-auth-conf
  Require valid-user
</Location>
```

The above line allow for a repo on /svn/repos to be accessible with a password contained in /etc/svn-auth-conf

##### **b. Restart apache to apply the changes**

```
# /etc/init.d/httpd restart
```

##### **c. configure password for authentication**

```
htpasswd -cm /etc/svn-auth-conf <username>
```

for example

```
# htpasswd -cm /etc/svn-auth-conf fedora
```

Please enter the password twice to create the user

#### **Step 6 - Checkout the repo**

##### **a. We can check out the repo into a third party area by doing the following steps**

```
#cd /tmp/
# svn --username=fedora co http://localhost:80/repos/svn-test/
A  svn-test/main
A  svn-test/configurations
A  svn-test/configurations/test.cfg
A  svn-test/options
Checked out revision 1.
```

## Step 7 : Editing and Committing

### a. Editing

```
# cd /tmp/svn-test/
# ls
configurations main options
# vim configurations/test.cfg
```

Edit the file and make any changes/save them

### b. Committing the changes

```
# svn commit -m "Added a line to testconf1.cfg."
Sending      configurations/test.cfg
Transmitting file data .
Committed revision 2.
```

### c. Adding/Deleting Items

```
# ls
configurations main options
# cd configurations/
# cp /etc/yum.conf .
# svn add yum.conf
A      yum.conf
# svn commit -m "Added yum conf"
Adding      configurations/yum.conf
Transmitting file data .
Committed revision 3.
```

### d. Reverting Items

#### i. Listing out all revisions.

This step will give you all the checkpoints

```
# svn log http://localhost:80/repos/svn-test/
Added the fedora repo
-----
r2 | fedora | 2011-07-21 20:30:22 +0530 (Thu, 21 Jul 2011) | 1 line
Added a line to testconf1.cfg.
-----
r1 | fedora | 2011-07-21 20:24:40 +0530 (Thu, 21 Jul 2011) | 1 line
```

Initial Import

#### ii) reverting/checkout to revision 1 -

```
# svn co -r 1 http://localhost:80/repos/svn-test/
A   svn-test/main
A   svn-test/configurations
A   svn-test/configurations/test.cfg
A   svn-test/options
Checked out revision 1.
```

**VI) References:**

- [1] - [http://en.wikipedia.org/wiki/Revision\\_control#Overview](http://en.wikipedia.org/wiki/Revision_control#Overview)
- [2] - <http://svnbook.red-bean.com/en/1.1/index.html>

## 5: Text Processing with Perl

### Aim:

To learn how to do some common text processing tasks using Perl.

### Introduction:

**Perl** is a high-level, general-purpose, interpreted, dynamic programming language. Perl was originally developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier.

Perl borrows features from other programming languages including C, shell scripting (sh), AWK, and sed. The language provides powerful text processing facilities without the arbitrary data length limits of many contemporary Unix tools, facilitating easy manipulation of text files.

Though originally developed for text manipulation, Perl is used for a wide range of tasks including system administration, web development, network programming, games, bioinformatics, and GUI development.

The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). Its major features include support for multiple programming paradigms (procedural, object-oriented, and functional styles), reference counting memory management , built-in support for text processing, and a large collection of third-party modules.

**CPAN**, the **Comprehensive Perl Archive Network**, is an archive of over 90,000 modules of software written in Perl, as well as documentation for it. It has a presence on the World Wide Web at [www.cpan.org](http://www.cpan.org) and is mirrored worldwide at more than 200 locations. CPAN can denote either the archive network itself, or the Perl program that acts as an interface to the network and as an automated software installer (somewhat like a package manager). Most software on CPAN is free and open source software.

This exercise consists of 7 programs of increasing complexity in Perl.

### Description:

Students will write seven programs in Perl and test their results. The programs will also use 3rd party modules to the language. The third party modules will be installed from the distribution packages rather than through CPAN.

## **Pre-requisites:**

Perl is installed by default in all Linux distributions. So the students can start programming with any text editor of their choice. When a program requires a third-party module and support files it will be mentioned with instructions on how to install them.

## **The Programs:**

The seven programs to be done in this exercise are:

1. Hello World
2. Greeting the user
3. Analysing text from a file and printing some statistics
4. Proper command line processing and analysing a text file to get word frequency, word size frequency and the type-token ratio.
5. Text analysis and outputting the result to another text file with proper formatting.
6. Read data from a flat file using Perl's database interface and performing SQL queries on the data.
7. Read rainfall data from a csv file, do some computations and produce a graph based on the results.

Create a new directory for the programs.

```
> mkdir perl_exercises  
> cd perl_exercises
```

Download the supporting materials zip file to the newly created directory and unzip the contents to the directory.

### **1. Hello World**

Create a new file using the gedit text editor.

```
> gedit hello.pl
```

Use the following code:

```
#!/usr/bin/env perl  
#The above statement tells the system that this is a perl program.  
  
print "Hello World!\n"; #print the text Hello World and a newline.
```

Save the file.

Now run the program as follows:

```
> perl hello.pl  
Hello World!  
>
```

The above command asks the perl interpreter to load the file called hello.pl and execute it. On execution the text “Hello World” is printed on the screen.

## 2. Greeting the user

This program asks the user’s name and the year of birth. It then greets the user and tells the age of the user.

```
> gedit name.pl
```

The Code:

```
#!/usr/bin/env perl  
#  
# name.pl  
  
print "Enter your name and press return:";  
$name=<STDIN>; #read the data  
chomp($name); #remove the newline  
print "\nEnter your birth year and press return:";  
$byear=<STDIN>;  
chomp($byear);  
  
#localtime gives the data with 9 distinct values. Collect them.  
my ($sec, $min, $hour, $mday, $mon, $year, $wday, $yday, $dst) =  
localtime time;
```

```
$age=($year + 1900) - $byear; #the year starts from 1900 according to  
localtime  
  
print "\nHello, $name!\n";  
print "You are $age years old.\n";
```

On execution:

```
> perl name.pl  
Enter you name and press return:Mickey Mouse  
  
Enter your birth year and press return:1928  
  
Hello, Mickey Mouse!  
You are 83 years old.  
>
```

### 3. Analysing text and printing the statistics

This programs read the text file given in the command line, asks the user for the word to search in the text and prints some statistics about the text. Note that the program will hang if the user fails to give the name of the file when the program is run. Proper handling of commandline arguments is explored in the next exercise.

```
> gedit words.pl
```

The Code:

```
#!/usr/bin/env perl  
#  
#words.pl word FILE  
#  
#if no data filename is given, this program will hang  
  
print "Enter the word you want to search for and press return:";  
$word=<STDIN>;
```

```
chomp($sword);

$count = 0;          #search counter
$bcnt = 0;           #blank line counter

while(<>)    #continue reading as long as there is input
{
    chomp;          #remove newline from each line
    foreach $w (split)      #split each line into words
    {
        if ($w eq $sword)
        {
            $count++;   #search hit counter
        }
        $words++;
        $char += length($w);
    }
    #if the length of the current line is 0, we have a blank line
    if (length($_) == 0)
    {
        $bcnt++;
    }
}

$avgw = $words/$.;      #average words per line including blank lines
$avgc = $char/$words;   #average characters per word

print "There are $. lines in this file including $bcnt blank
lines.\n";
print "There are $words words in this file.\n";
print "There are $char characters in this file.\n";
print "The average number of words per line is $avgw.\n";
print "The average number of characters per word is $avgc.\n";
print "the word $sword occurs in the text $count times.\n";
```

On execution:

```
> perl words.pl constitution_preamble.txt
```

Enter the word you want to search for and press return:the  
There are 13 lines in this file including 6 blank lines.  
There are 85 words in this file.  
There are 470 characters in this file.  
The average number of words per line is 6.53846153846154.

The average number of characters per word is 5.52941176470588.  
the word the occurs in the text 4 times.

The file constitution\_preamble.txt is part of the support file archive which was unzipped at the beginning.

#### 4. Command line processing and more text analysis

This program also reads from a text file and analyses the text. Proper command line handling is now performed. The program converts all input text into lower case and strips off all the punctuation marks in the text. The use of hashes is introduced.

```
> gedit wordcount.pl
```

The Code:

```
#!/usr/bin/env perl
#
#wordcount.pl FILE
#
#if no filename is given, print help and exit
if (length($ARGV[0]) < 1)
{
    print "Usage is : words.pl word filename\n";
    exit;
}

my $file = $ARGV[0];      #filename given in commandline

open(FILE, $file);        #open the mentioned filename
while(<FILE>)            #continue reading until the file ends
{
    chomp;
    tr/A-Z/a-z/;          #convert all upper case words to lower case
    tr/.,:;!?"(){}//d;     #remove some common punctuation symbols

    #We are creating a hash with the word as the key.
    #Each time a word is encountered, its hash is incremented by 1.
    #If the count for a word is 1, it is a new distinct word.
    #We keep track of the number of words parsed so far.
    #We also keep track of the no. of words of a particular length.
```

```
foreach $wd (split)
{
    $count{$wd}++;
    if ($count{$wd} == 1)
    {
        $dcount++;
    }
    $wcount++;
    $lcount{length($wd)}++;
}

#To print the distinct words and their frequency,
#we iterate over the hash containing the words and their count.
print "\nThe words and their frequency in the text is:\n";
foreach $w (sort keys%count)
{
    print "$w : $count{$w}\n";
}

#For the word length and frequency we use the word length hash
print "The word length and frequency in the given text is:\n";
foreach $w (sort keys%lcount)
{
    print "$w : $lcount{$w}\n";
}

print "There are $wcount words in the file.\n";
print "There are $dcount distinct words in the file.\n";

$tratio = ($dcount/$wcount)*100;    #Calculating the type-token ratio.

print "The type-token ratio of the file is $tratio.\n";
```

On execution:

```
> perl wordcount.pl constitution_preamble.txt
```

```
The words and their frequency in the text is:
1949 : 1
a : 1
adopt : 1
all : 2
among : 1
```

CS2406-Lab Manual - NRCFOSS/AU-KBC Centre, Anna University Chennai

```
and : 8
assembly : 1
assuring : 1
belief : 1
citizens : 1
constituent : 1
constitute : 1
constitution : 1
day : 1
democratic : 1
dignity : 1
do : 1
economic : 1
enact : 1
equality : 1
expression : 1
faith : 1
fraternity : 1
give : 1
having : 1
hereby : 1
in : 1
india : 2
individual : 1
integrity : 1
into : 1
its : 1
justice : 1
liberty : 1
nation : 1
november : 1
of : 7
opportunity : 1
our : 1
ourselves : 1
people : 1
political : 1
promote : 1
republic : 1
resolved : 1
secular : 1
secure : 1
social : 1
socialist : 1
solemnly : 1
```

```
sovereign : 1
status : 1
the : 5
them : 1
this : 2
thought : 1
to : 5
twenty-sixth : 1
unity : 1
we : 1
worship : 1
The word length and frequency in the given text is:
1 : 1
10 : 5
11 : 2
12 : 2
2 : 15
3 : 18
4 : 6
5 : 7
6 : 8
7 : 7
8 : 9
9 : 5
There are 85 words in the file.
There are 61 distinct words in the file.
The type-token ratio of the file is 71.7647058823529.
```

## 5. Text analysis with results output to another file

This program analyses the text of a file and outputs the results to another file after formatting the output.

```
> gedit freqcount.pl
```

The Code:

```
#!/usr/bin/env perl
#
#freqcount.pl FILE
```

```
#  
  
use strict; #using strict mode to help us find errors easily  
  
#all variables being used are declared  
my $file;  
my $wd;  
my %count;  
my $w;  
  
if (@ARGV) #Check if the ARGV array exists. This array is populated with  
#the arguments passed to the program.  
{  
    $file = $ARGV[0]; #First argument is the data file name.  
}  
else  
{  
    die "Usage : freqcount.pl FILE\n"; #Bail out if no data filename  
#is given  
}  
  
open(FILE, $file);  
  
open(RESULTS, ">freqcount.txt"); #Open the file where the results  
#will be written. If it exists it will be overwritten.  
  
while(<FILE>)  
{  
    chomp;  
    tr/A-Z/a-z/;  
    tr/.,:;!?"(){}//d;  
  
    foreach $wd (split)  
    {  
        $count{$wd}++;  
    }  
}  
  
print RESULTS "Word\t\tFrequency\n"; #Writing to newly opened file  
  
foreach $w (sort by_number keys%count) #The result will be sorted  
#using our by_number function  
{  
    write(RESULTS);  
}
```

```
close(RESULTS);

#Our sorting function.
#The <=> is used to sort the result in a descending order of frequency.
#The second <=> is used to sort the result based on the length of the
#word if the frequency is same.
sub by_number
{
    $count{$b} <=> $count{$a} || length($b) <=> length($a);
}

#Formatting the results.
#A @ denotes the values to be printed.
#A < stands for left justify text in that position, > stand for right
#justify.
#The formatting ends with a final .
format RESULTS=
@<<<<<<<<<< @>>
$w,    $count{$w}
.
```

#### On Execution:

```
> perl freqcount.pl constitution_preamble.txt
> cat freqcount.txt
Word          Frequency
and            8
of              7
the             5
to              5
india           2
this            2
all             2
twenty-sixth   1
constitution   1
opportunity    1
constituent    1
individual     1
constitute     1
expression     1
democratic     1
fraternity     1
ourselves      1
```

integrity	1
socialist	1
political	1
sovereign	1
solemnly	1
assembly	1
citizens	1
resolved	1
november	1
economic	1
equality	1
assuring	1
republic	1
thought	1
dignity	1
worship	1
liberty	1
promote	1
justice	1
secular	1
secure	1
social	1
people	1
belief	1
nation	1
status	1
having	1
hereby	1
unity	1
among	1
faith	1
adopt	1
enact	1
give	1
them	1
1949	1
into	1
our	1
day	1
its	1
in	1
do	1
we	1
a	1

## 6. Connecting to databases

This program shows how to connect to a database using perl DBI. In place of the database we use a csv file which is accessed using the csv database driver.

The database driver is a third party module which needs to be installed. The driver name is DBD::CSV and it is installed by the following command.

```
> yum install perl-DBD-CSV
```

Once the driver is installed we can start. We will be using the Indian\_capitals.csv from the support files zip.

```
> gedit dbdcsv.pl
```

The Code:

```
#!/usr/bin/env perl
#
#dbdcsv.pl
#
# Shows connecting to databases using perl. The database here is a CSV
file.
# The same principle applies to any database connection.

use strict;
use warnings;
use DBI;      #using perl database interface

# Connect to the database, (the directory containing our csv file)
my $dbh = DBI->connect("DBI:CSV:f_dir=.;csv_eol=\n");

# Associate our csv file with the table name 'Indian_capitals.csv' and
# manually declare names for each of the columns
$dbh->{ 'csv_tables' }->{ 'Indian_capitals.csv' } = {
    'col_names' => [ "state", "admin_c", "legis_c", "year", "old_c" ] };

# Run a SQL command on the database which gives the list of all states
#whose capital city was changed.
# The SQL statement is prepared first before being executed.
# This provides us with a place to verify that the sql is proper and
#valid.
# A limited set of SQL operations can be performed here.
# If we are actually connecting to a database, all the SQL supported by
```

```
#the DB can be done here.  
# This statement needs to be vetted for any SQL injection vulnerabilities  
#in a real life scenario.  
my $sth = $dbh->prepare("SELECT * FROM Indian_capitals.csv WHERE old_c  
NOT LIKE '-!');  
$sth->execute();  
  
print ("\\nThe list of states which had their capitals changed:\\n");  
  
# The output from the SQL statement is fetched and the results are  
#printed.  
while (my $row = $sth->fetchrow_hashref)  
{  
    print($row->{'state'}, "\\n");  
}  
  
# The statement is closed.  
$sth->finish();
```

On Execution:

```
> perl dbdcsv.pl
```

```
The list of states which had their capitals changed:  
State_or_UT  
Andhra Pradesh  
Assam  
Gujarat  
Karnataka  
Kerala  
Madhya Pradesh  
Orissa  
Punjab
```

## 7. Charting data

This programs reads rainfall data from a number of stations for a period of 2 months. It then calculates how far from the normal, the actual rainfall varies and plots a nice bar chart based on the data. The data is once again in a csv file. The csv file is read as it is ie. it is not considered as a database. The file rainfall.csv is in the support files zip.

First the charting package has to be installed. the package is GD::Graph. The package for reading the CSV file is Text::CSV.

```
> yum install perl-GDGraph perl-Text-CSV  
> gedit rainfall.pl
```

The Code:

```
#!/usr/bin/env perl  
#  
#rainfall.pl  
#  
# This program reads data from a csv file containing a list of stations,  
# actual rainfall and the average rainfall in each station over a period  
# of two months from June to July 2011 for the states of Tamil Nadu and  
# Pondicherry  
  
use strict;  
use warnings;  
use Text::CSV;      #for reading the csv file  
use GD::Graph::hbars;    #for drawing the horizontal bar graph  
  
my $file = "rainfall.csv";  
  
my $csv = Text::CSV->new();  
  
open (CSV, "<", $file) or die $!;    #open the specified file for reading  
  
my @columns;        #array holding the data from the csv file  
my $station;        #individual station  
my $rainfall;       #rainfall in each station  
my @gprecip;        #array holding the charting data  
my $err;  
my $s;  
my @station;        #array of stations  
my @prainfall;     #array of positive rainfall
```

```

my @rainfall;      #array of negative rainfall
my $nr;
my $ns;
my $my_graph;     #chart variable

while(<CSV>)
{
    next if ($. == 1);           #ignore the first line since it will have
#the headings
    if ($csv->parse($_))
    {
        @columns = $csv->fields();      #load values into array from
#the file
        $station = $columns[0];
        $ns = push(@station, $station);   #build the array of
#stations
        $rainfall = (((($columns[1]/$columns[2])*100) - 100 );

#To draw the chart with different colors we need the
#rainfall values in 2 arrays
#One for the positive values of rainfall
#Another for the negative values of rainfall

        if ($rainfall > 0)
        {
            $nr = push(@prainfall, int($rainfall));
            $nr = push(@nrainfall, undef);
        }
        else
        {
            $nr = push(@nrainfall, int($rainfall));
            $nr = push(@prainfall, undef);
        }
    }
    else
    {
        $err = $csv->error_input;
        print "Failed to parse line: $err";
    }
}
close CSV;

@gprecip = (@station, @prainfall, @nrainfall);      #combine the
#arrays to form the big array for GD::Graph
$my_graph = GD::Graph::hbars->new(480, 640);      #create the chart

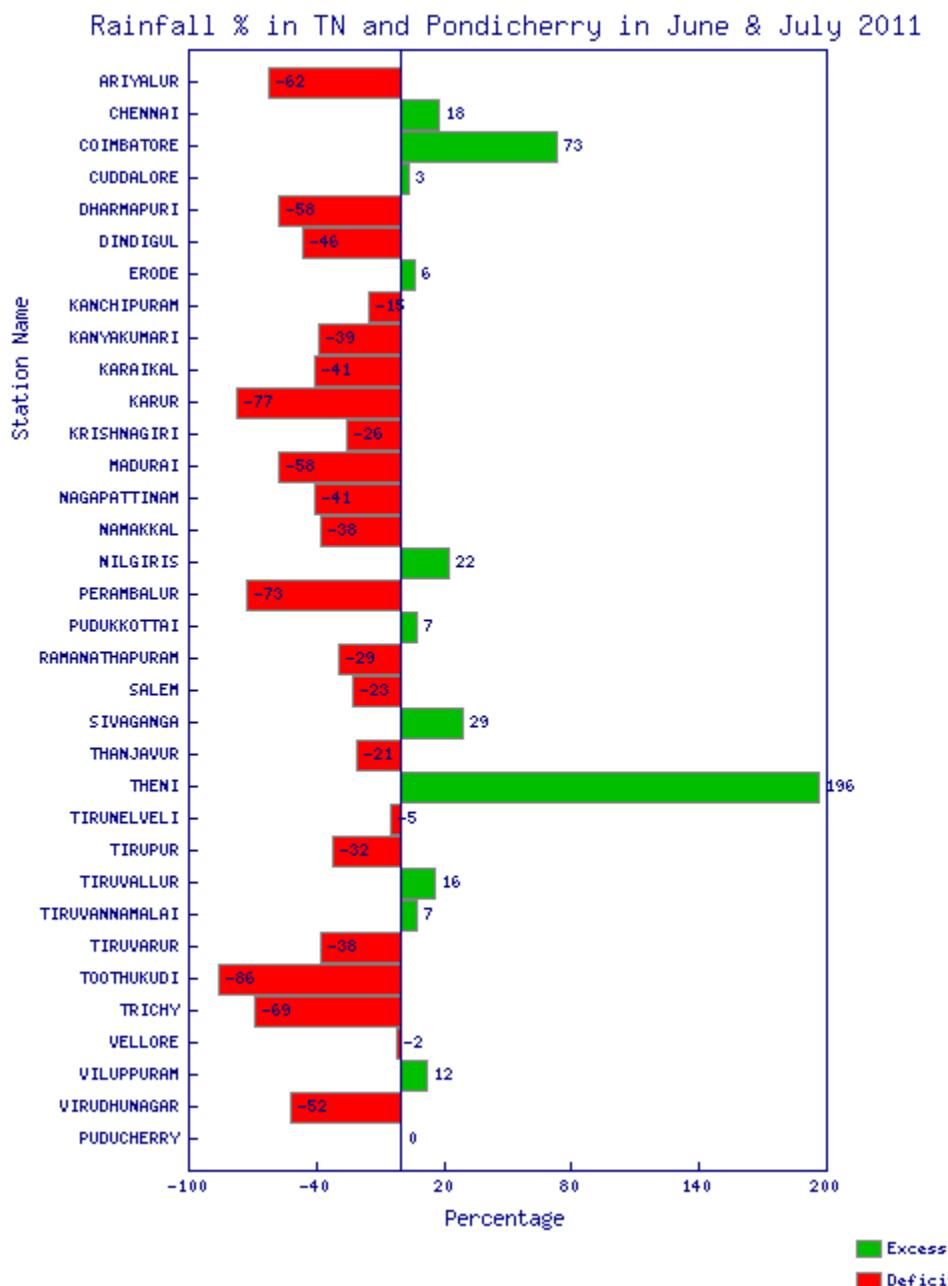
```

```
#Set parameters of the graph. See GD::Graph documentation for
#details.
$my_graph->set(
    x_label => 'Station Name',
    y_label => 'Percentage',
    title => 'Rainfall % in TN and Pondicherry in June & July 2011',
    y_max_value => 200,
    y_min_value => -100,
    overwrite => 1,
    dclrs => [qw (green lred) ],
    legend_placement => 'RB',
    show_values => 1,
    transparent=>0,
);
$my_graph->set_legend('Excess', 'Deficit');

#Write the bar graph into a png file
open(IMG, ">rainfall.png") or die $!;
binmode IMG;
print IMG $my_graph->plot(\@gprecip)->png;
close IMG;
```

On execution, a new file rainfall.png will be created in the current directory.

```
> perl rainfall.pl
```



## References:

1. [www.perl.org](http://www.perl.org) - Perl home page.
2. <http://www.llc.manchester.ac.uk/intranet/ug/useful-links/perl/> - Perl for Text processing
3. [http://perlmeme.org/tutorials/parsing\\_csv.html](http://perlmeme.org/tutorials/parsing_csv.html) - Parsing CSV
4. <http://gdgraph.com/samples.html> - GDGraph examples

## 6. Running PHP

### Aim:

To learn simple programs in php after setting up a LAMP stack

### Introduction:

PHP is a general-purpose scripting language originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP can be deployed on most web servers and as a standalone interpreter, on almost every operating system and platform free of charge. A competitor to Microsoft's Active Server Pages(ASP) server-side script engine and similar languages, PHP is installed on more than 20 million websites and 1 million web servers.[5]

PHP was originally created by Rasmus Lerdorf in 1995. The main implementation of PHP is now produced by The PHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.

This exercise consists of 7 programs of increasing complexity in PHP.

### Description:

Students will write seven programs in PHP and test their results.

### Pre-requisites:

PHP does not come with the basic installation so it requires a web server,php and mysql to be installed on the machine.

#### Steps for installation

##### Installing MySQL 5

open a terminal and execute the following commands

```
 yum install mysql mysql-server
```

Then we create the system startup links for MySQL (so that MySQL starts automatically whenever the system boots) and start the MySQL server

```
 chkconfig --levels 235 mysqld on  
 /etc/init.d/mysqld start
```

Run

```
 mysqladmin -u root password yourrootsqlpassword  
 to set a password for the user root (otherwise anybody can access your MySQL database!).
```

## Installing Apache2

Apache2 is available as a Fedora package, therefore we can install it like this

```
yum install httpd
```

Now configure your system to start Apache at boot time.

```
chkconfig --levels 235 httpd on
```

and start Apache:

```
/etc/init.d/httpd start
```

Now direct your browser to `http://localhost`, and you should see the Apache2 placeholder page  
Apache's default document root is `/var/www/html` on Fedora, and the configuration file is  
`/etc/httpd/conf/httpd.conf`. Additional configurations are stored in the  
`/etc/httpd/conf.d/` directory.

## Installing PHP5

```
yum install php
```

We must restart Apache afterwards:

```
/etc/init.d/httpd restart
```

## The Programs:

The seven programs to be done in this exercise are:

1. Hello World
2. Concatenation of Strings
3. PHP Control Structures
4. Add User Input to Array
5. PHP Forms
6. Simple Registration Form with out database.
7. Creating a Login page

Create all your programs in `/var/www/html/` and the output can be viewed in a web browser by giving the url as "`http://localhost/yourfilename.php`"

Note: if you couldnot create a file at `/var/www/html` then login as root user by issuing a command `su`

### 1. Hello World

Create a PHP page, with the standard HTML `<head>`, `<title>` and `<body>` tags. This is not strictly necessary but is good practice and should the the first step for each exercise. Consider creating a template file with these tags already in place. Remember to save your file with the `.php` extension.

Inside the <body> tag, create a PHP section that will show the text “Hello World!” in the browser. Check your work. If you get a blank page, check to be sure that you ended each statement with “;” and that you included the opening and closing PHP markers.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>Hello, World!</title>
</head>
<body>
<?php
echo "Hello, World!";
// You could have used the print statement here.
?>
</body>
</html>
```

## 2. Concatenation of Strings

For this PHP exercise, write a program using the following variable:

\$around="around";

Single quotes and double quotes don't work the same way in PHP. Using single quotes (' ') and the concatenation operator, echo the following to the browser, using the variable you created:

**What goes around, comes around.**

**Code:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>Concatenation of Strings</title>
</head>
<body>
<?php
$around = "around";
echo 'What goes ' . $around . ' comes ' . $around . '.';
?>
```

```
</body>  
</html>
```

The point of this exercise is to use the period (".") concatenation operator. Using double quotes (" "), you could have included the variables in your string, and PHP would print them as intended. If your output doesn't look right, be sure you have included the appropriate spaces inside the quotes.

### 3. PHP Control Structures

HTML tables involve a lot of repetitive coding - a perfect place to use for loops. You can do even more if you nest the for loops.

In this PHP exercise, use two for loops, one nested inside another. Create the following multiplication table:

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	15	18	21
4	8	12	16	20	24	28
5	10	15	20	25	30	35
6	12	18	24	30	36	42
7	14	21	28	35	42	49

#### Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
<head>  
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />  
  <title>Nested Loop Multiplication Table</title>  
</head>  
  
<body>  
  <h2>Nested Loop Multiplication Table</h2>  
  
  <?php  
    //Generate an HTML table  
    echo "<table border=\"1\">" ;
```

```
//Generate table data showing the numbers 1-7 multiplied by each other,
//starting with the rows.

for ($row=1; $row<=7; $row++) {
    echo "<tr>\n";
    //Generate each entry in the row to create the columns.
    for ($col=1; $col<=7; $col++) {
        //First, do the math.
        $x=$col * $row;
        //Then send the value to the table with the table data tags.
        echo "<td>$x</td>\n";
    }
    echo "</tr>";
}
echo "</table>";
?>

</body>
</html>
```

#### 4.Add User Input to Array

In this PHP exercise, you will create an array, print it to the browser, then ask the user to add to it.

Create an array of modes of transportation, including Automobile, Jet, Ferry, Subway. Print the following statement to the browser:"Travel takes many forms, whether across town, across the country, or around the world. Here is a list of some common modes of transportation:" Follow this with an unordered list created by iterating through the array variable you created.

Next, give the user a text box and ask the user to add other modes of transportation to the list, separated by commas. When the user clicks 'Go', process the input with array functions to send back the original list with the user's additions. Include another text box with the text "Add more?" and another submit button. When the user clicks this button, the page should reload with the new additions added to the previously expanded list. Your code should allow the user to add items as many times as they like

#### Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>User Input Added to Array</title>
</head>
<body>
<h2>How Are You Traveling?</h2>
<?php
//If form not submitted, display form.
if (!isset($_POST['submit'])) {
$travel=array(
"Automobile",
"Jet",
"Ferry",
"Subway"
);
?>
<p>Travel takes many forms, whether across town, across the country, or
around the world. Here is a list of some common modes of
transportation:</p>
<ul>
<?php
foreach ($travel as $t){
echo "<li>$t</li>\n";
}
?>
</ul>
<form method="post" action="yourfile.php">
<p>Please add your favorite, local, or even imaginary modes of travel
to the list, separated by commas:</p>
<input type="text" name="added" size="80" />
<p />

<?php
//Send current travel array as hidden form data.
foreach ($travel as $t){
```

```
echo "<input type=\"hidden\" name=\"travel[]\" value=\"$t\" />\n";
}

?>

<input type="submit" name="submit" value="Go" />
</form>

<?php
//If form submitted, process input.
}else{
//Retrieve established travel array.
$travel=( $_POST['travel']);
//Convert user input string into an array.
$added=explode(',',$_POST['added']);

//Add to the established array.
array_splice($travel, count($travel), 0, $added);
//This could also be written $travel=array_merge($travel, $added);

//Return the new list to the user.
echo "<p>Here is the list with your additions:</p>\n<ul>\n";
foreach($travel as $t){
    //The trim functions deletes extra spaces the user may have entered.
    echo "<li>".trim($t)."</li>\n";
}
echo "</ul>";

?>
<p>Add more?</p>
<form method="post" action="yourfile.php">
<input type="text" name="added" size="80" />
<p />
<?php
```

```
//Send current travel array as hidden form data.  
foreach ($travel as $t) {  
    echo "<input type=\"hidden\" name=\"travel[]\" value=\"$t\" />\n";  
}  
?  
<input type="submit" name="submit" value="Go" />  
</form>  
<?php  
}  
?  
</body>  
</html>
```

## 5.PHP Forms

One very useful thing you can do with PHP is include the request for user input and the response in the same file, using conditional statements to tell PHP which one to show. For this PHP exercise, rewrite the two files of the previous exercise into one file using an if-else conditional statement.

Hint: You'll need some way to tell if the form has been submitted. The function to determine if a variable has been set and is not null is `isset()`.

### Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
<head>  
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />  
    <title>Form Response with If-Else Statement</title>  
</head>  
<body>  
    <h2>Favorite City</h2>  
<?php
```

```
//If form not submitted, display form.  
if (!isset($_POST['submit'])) {  
?>  
  
<form method="post" action="yourfile.php">  
<!--Make sure you have entered the name you gave the file as the action.-->  
Please enter your favorite city: <br />  
<input type="text" name="city" />  
<p />  
<input type="submit" name="submit" value="Go" />  
</form>  
  
<?php  
//If form submitted, process input.  
} else {  
    //Retrieve string from form submission.  
    $city = $_POST['city'];  
    echo "Your favorite city is $city.";  
}  
?>  
</body>  
</html>
```

## 6. Simple Registration Form with out a database

In this PHP exercise, you will build the beginnings of a user registration form.

First of all, write an array that includes browser types: Firefox, Chrome, Internet Explorer, Safari, Opera, Other.

Then write the class Select and then next comes the HTML. Write a user registration form asking for name, username, email, browser. Use text fields to collect the user data for the first three, then instantiate an object based on your class for the select field. When the user clicks the submit button, return the data as confirmation.

If you were creating a registration form to use on the Web, you would want to collect the data in a database. However, using PHP with mySQL or other databases is beyond the scope of this program.

```
<?php  
//Create array.  
$browsers=array(  
    "Firefox",  
    "Chrome",  
    "Internet Explorer",  
    "Safari",  
    "Opera",  
    "Other"  
) ;  
  
class Select{  
    //Property  
    private $name;      //String variable.  
    private $value;     //Array variable.  
  
    //Methods  
    //The string set by this method will be the name of the select field.  
    public function setName($name){  
        $this->name = $name;  
    }  
  
    public function getName(){  
        return $this->name;  
    }  
  
    //This method provides the values used for the options  
    //in the select field and checks to be sure the value is an array.  
    public function setValue($value){  
        if (!is_array($value)){  
            die ("Error: value is not an array.");  
    }  
}
```

```
$this->value = $value;  
}  
  
public function getValue(){  
    return $this->value;  
}  
  
//This method creates the actual select options. It is private,  
//since there is no need for it outside the operations of the class.  
private function makeOptions($value){  
    foreach($value as $v){  
        echo "<option value=\"$v\">" .ucfirst($v). "</option>\n";  
    }  
}  
  
//This method puts it all together to create the select field.  
public function makeSelect(){  
    echo "<select name=\"\" ".$this->getName()."\">\n";  
    //Create options.  
    $this->makeOptions($this->getValue());  
    echo "</select>" ;  
}  
}//end class  
  
?>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
<head>  
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
```

```
<title>Class Select - Browsers</title>
</head>

<body>
<h2>User Registration - Browser<br /></h2>

<?php
//If form not submitted, display form.
if(!isset($_POST['submit'])){
?>

<form method="post" action="yourfile.php">
<p>Name:<br />
<input type="text" name="name" size="60" /> </p>
<p>Username:<br />
<input type="text" name="username" size="60" /></p>
<p>Email:<br />
<input type="text" name="email" size="60" /></p>
<p>Browser:<br />

<?php
//Instantiate object.
$browser = new Select();
//Set properties.
$browser->setName('browser');
$browser->setValue($browsers);
//The object has the data it needs from the preceding commands.
//Tell it to make the select field.
$browser->makeSelect();
?>
</p>
<input type="submit" name="submit" value="Go" />
```

```
</form>

<?php
    //If form submitted, process input.
} else{
    //Could include code to send data to database here.
    //Retrieve user responses.
    $name=$_POST['name'];
    $username=$_POST['username'];
    $email=$_POST['email'];
    //The following variable has an altered name to avoid confusion.
    $selBrowser=$_POST['browser'];
    //Confirm responses to user.
    echo "The following data has been saved for $name: <br />";
    echo "Username: $username<br />";
    echo "Email: $email<br />";
    echo "Browser: $selBrowser<br />";
}
?>

</body>
</html>
```

## 7.Creating a Login page

Here We are going to create a login page and the input values will be check with the existing mysql database value and based on the validation the page will be redirected.

Login in to mysql

mysql -p -u root

Create a database named “members”

Create database members;

use members

### Create table "members"

```
CREATE TABLE `members` (
  `id` int(4) NOT NULL auto_increment,
  `username` varchar(65) NOT NULL default '',
  `password` varchar(65) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=2 ;

-- 
-- Dumping data for table `members`
-- 

INSERT INTO `members` VALUES (1, 'john', '1234');
```

### Create file main\_login.php

```
<html>
<head> </head>
<body>

<table width="300" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#CCCCCC">
<tr>
<form name="form1" method="post" action="checklogin.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#FFFFFF">
<tr>
<td colspan="3"><strong>Member Login </strong></td>
</tr>
<tr>
<td width="78">Username</td>
<td width="6">:</td>
<td width="294"><input name="myusername" type="text" id="myusername"></td>
</tr>
<tr>
<td>Password</td>
<td>:</td>
<td><input name="mypassword" type="text" id="mypassword"></td>
</tr>
<tr>
```

```
<td>&nbsp;</td>
<td>&nbsp;</td>
<td><input type="submit" name="Submit" value="Login"></td>
</tr>
</table>
</td>
</form>
</tr>
</table>
</body>
</html>
```

**Create file checklogin.php**

```
<?php
$host="localhost"; // Host name
$username=""; // Mysql username
$password=""; // Mysql password
$db_name="members "; // Database name
$tbl_name="members"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// username and password sent from form
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];

// To protect MySQL injection (more detail about MySQL injection)
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
$mypassword = mysql_real_escape_string($mypassword);
```

```
$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and  
password='$mypassword';  
  
$result=mysql_query($sql);  
  
// Mysql_num_row is counting table row  
$count=mysql_num_rows($result);  
// If result matched $myusername and $mypassword, table row must be 1 row  
  
if($count==1){  
// Register $myusername, $mypassword and redirect to file  
"login_success.php"  
session_register("myusername");  
session_register("mypassword");  
header("location:login_success.php");  
}  
else {  
echo "Wrong Username or Password";  
}  
?>
```

### Create file login\_success.php

```
<?  
session_start();  
if(!session_is_registered(myusername)){  
header("location:main_login.php");  
}  
?>
```

```
<html>  
<body>  
Login Successful  
</body>  
</html>
```

## **References:**

1. <http://en.wikipedia.org/wiki/PHP>
2. <http://www.php.net>
3. <http://www.howtoforge.com/installing-apache2-with-php5-and-mysql-support-on-fedora-12-lamp>
4. <http://phpexercises.com>
5. <http://www.phpeasystep.com/phptu/6.html>

## 8.Running Python & Django

### Aim

Learn to write Python and Django programs using FOSS tools in Linux.

### Introduction

**Python** is a high-level programming language. It is an easy to learn, powerful programming language. It has efficient, high-level data structures and a simple yet effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, makes it an ideal language for scripting and rapid application development in many areas and on most platforms. The Creator of this Language is **Guido van Rossum**. He named this language after the BBC show "**Monty Python's Flying Circus**". But he doesn't particularly like snakes that kill animals for food by winding their long bodies around them and crushing them.

### Pre-requisites

By Default, Most of the Linux distributions come with Python. If your distribution doesn't have it, then you can install it by either downloading from the python website or from your distribution's repository.

```
# yum install python
```

After Python Installation, go to **Terminal** and Type "**python**" without double quotes. You must get the below output

```
$ python
Python 2.6.2 (r262:71600, Aug 21 2009, 12:22:21)
[GCC 4.4.1 20090818 (Red Hat 4.4.1-6)] on linux2
Type "help", "copyright", "credits" or "license" for more
information.

>>>
```

If you get any error, your python might not installed properly, then reinstall the python and try again.

## The Programs

### 1. Beginning of the Python Code

**Python** has a dynamic environment called **IDLE**. In which, either you can get the info about a command and it's working, or you can execute the command. Hence, we can execute a complete program through IDLE instead of writing in a file and executing it. This helps in learning and reduces program development time.

#### Writing Code in the Interpreter:

Type “**python**” in the Terminal

```
$ python
Python 2.6.2 (r262:71600, Aug 21 2009, 12:22:21)
[GCC 4.4.1 20090818 (Red Hat 4.4.1-6)] on linux2
Type "help", "copyright", "credits" or "license" for more
information.

>>>
```

Next, we are going to execute a “Hello World” program.

```
>>> print "Hello World!"
Hello World!
```

#### Writing Code in a file:

First, we need to create a file with the extension **.py**. This **.py** file is the source file which can be created by using any editor like **vi**.

For instance, we create a file called **helloworld.py** and type the below code and save it

```
#!/usr/bin/env python
print "Hello World!"
```

You can run this code in two ways.

1. By using “**python**” command

```
$python Helloworld.py
```

2. By executing the file

First, you have to make the file executable. In Linux, you can do that by giving the following command in terminal

```
$chmod +x Helloworld.py  
$./helloworld.py
```

**Note :**

1. In Python, Indentation is an important thing. Wrong indentation will result in an indentation error.

```
>>> print "Hello World!"  
>>>   print "Hello World!"  
  
      File "<stdin>", line 1  
  
        print "Hello World!"  
          ^  
  
IndentationError: unexpected indent
```

There is an extra space in the beginning of the second line which is causing the error. Hence, always look for proper indentation.

2. You will run into indentation errors, even if you mix up tabs and spaces. For instance, if you use spaces and only use spaces for indentation, don't use tabs in the middle of the program instead of couple of spaces. Even though the code looks the same, it will result in an indentation error.

**Note :**

**In Python, Indentation is an important thing. Wrong indentation will result in an indentation error. So you can use GUI editor like geany. Because the geany editor keeps the indentation on each line automatically. You can use other editors also. In that case you have care yourself about that code indentation and all.**

Type the following command in **terminal**. If you dont have **geany** installed in your system.

```
# yum install geany
```

## 2. Sending Birthday invitation to friends

```
#!/usr/bin/python
def get_name():
    """
        Invitation for Birthday party.
        Enter your 5 friend's name :
    """
    friends_name = [] # initializing list
    for x in range(5):
        name = raw_input("Friend %d: " % (x + 1))
        friends_name.append(name)
        # Appending names to list one by one
    return friends_name

def print_name():
    """
        def print_name
        Gets the name one by one and displays them in
        uppercase
    """
    namelist = get_name()
    print "\n    Sending invitation to your friends...\n"
    for name in namelist:
        print "Invitation sent to " + name.upper() + " "
    # upper() function for converting string to upper case

print get_name.__doc__
print_name()
```

### 3. Collect and Printing Biodata in a terminal by importing modules

Create a new module named **biodata.py**

**#biodata.py**

```
#!/usr/bin/python
## Sample python module

def Message():
    print "This program will collect your Biodata and display through
terminal"

# define a class
class Biodata:
    def __init__(self):
        print "\n Enter your details \n "
        self.name = raw_input("Name : ")
        self.dob = raw_input("Date of birth : ")
        self.age = raw_input("Age : ")
        self.email = raw_input("Email id : ")
        self.mobile = raw_input("Mobile Number : ")

    def print_biodata(self):
        print "\n Your Biodata \n\n",
        print "Name : "+self.name+"\nDate of Birth : "+self.dob+"\nAge :
"+self.age+"\nEmail Id : "+self.email+"\nMobile Number : "+self.mobile
```

Next create a file named **test.py**

**Note :** **test.py** and **biodata.py** should be placed in same folder.

**#test.py**

```
#!/usr/bin/python
import biodata      # importing existing biodata
module
biodata.Message()
data = biodata.Biodata()
data.print_biodata()
```

#### 4. Drawing Triangle in python using Turtle graphics module

```
#!/usr/bin/python
import turtle
turtle.setup(800, 600)      # set the window size to 800 by 600
pixels
wn = turtle.Screen()        # set wn to the window object
wn.bgcolor("lightgreen")    # set the window background color
wn.title("Hello, This is triangle!")  # set the window title
tess = turtle.Turtle()
tess.color("blue")          # make tess blue
tess.pensize(3)             # set the width of her pen
tess.forward(300)
tess.left(120)
tess.forward(300)
tess.left(120)
tess.forward(300)
wn.exitonclick()
```

## 5. Moving Turtle using Turtle graphics module

```
#!/usr/bin/python
import turtle
turtle.setup(800, 600)
wn = turtle.Screen()
wn.bgcolor("lightgreen")
wn.title("Tess's Spiral")
tess = turtle.Turtle()
tess.shape("turtle")
tess.color("blue")
tess.penup()                      # this is new
size = 20
for i in range(30):
    tess.stamp()                  # leave an impression on
the canvas
    size = size + 3              # increase the size on
every iteration
    tess.forward(size)           # move tess along
    tess.right(24)               # and turn her
wn.exitonclick()
```

## Introduction

**Django** is an web application framework, written in Python, which follows the **Model-View-Controller (MVC)** Pattern. It is named after **Django Reinhardt**, a gypsy jazz guitarist from the 1930s to early 1950s. To this day, he's considered one of the best guitarists of all time. Django is pronounced **JANG-oh**. Rhymes with FANG-oh. It was originally developed to manage several news-oriented sites for The World Company of Lawrence, Kansas, etc., and was released publicly under **BSD license** by July 2005. In June 2008, It was announced that a newly formed Django Software Foundation will take care of Django in the future.

## The Django framework components

### Object-relational mapper

Define your **data models** entirely in Python. You get a rich, **dynamic database-access API** for free — but you can still write SQL if needed.

### Automatic admin interface

Save yourself the tedious work of creating interfaces for people to add and update content. **Django does that automatically**, and it's production-ready.

### Elegant URL design

Design pretty, **cruft-free URLs** with no framework-specific limitations. Be as flexible as you like.

### Template system

Use Django's powerful, extensible and designer-friendly **template language** to separate design, content and Python code.

### Cache system

Hook into memcached or other cache frameworks for **super performance** — caching is as granular as you need.

### Internationalization

Django has full support for multi-language applications, letting you specify translation strings and providing hooks for language-specific functionality.

## Pre-requisites

First, download **Django-x.x.tar.gz** from the following link

**<http://FOSS-LAB-SERVER-IP/content/code/Django/Django-x.x.x.tar.gz>**

Go to the directory where the **Django-x.x.tar.gz** file has been downloaded and type the following commands through **terminal**

```
# tar xzvf Django-x.x.tar.gz  
# cd Django-x.x  
# sudo python setup.py install
```

### Note:

Before installing Django, make sure that whether python has installed properly.

For testing :

Type the command **python** in terminal. Then you get something like this

```
$ python  
Python 2.6.2 (r262:71600, Aug 21 2009, 12:22:21)  
[GCC 4.4.1 20090818 (Red Hat 4.4.1-6)] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

## The Programs

### 1. Sample Django application for displaying “Hello world” message with Current time.

#### Create a project

```
# django-admin.py startproject mysite
```

Get into project folder which you created now.

```
# cd mysite
```

#### Note:

When you run **django-admin.py startproject mysite**, Django will automatically create a new directory containing four files:

- \* **\_\_init\_\_.py**, which will be empty. This file is required to tell Python that the directory is a Python module and can be imported (and imported from).
- \* **manage.py**, which provides a number of convenience functions for working with the project.
- \* **settings.py**, which will be the project’s settings file.
- \* **urls.py**, which will be the project’s root URL configuration.

Next you could check the project whether you have installed successfully or not by running the django's inbuilt development server named **runserver**

```
# python manage.py runserver
Validating models...
0 errors found

Django version 1.2.5, using settings
'mysite.settings'

Development server is running at
http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Now open the url <http://127.0.0.1:8000/> in your browser. You will get the following output.

## It worked!

Congratulations on your first Django-powered page.

Of course, you haven't actually done any work yet. Here's what to do next:

- If you plan to use a database, edit the `DATABASE_*` settings in `mysite/settings.py`.
- Start your first app by running `python mysite/manage.py startapp [appname]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

## Create an application

Get into project “**mysite**”. And create an application

```
# django-admin.py startapp web
```

### Note:

When you run **django-admin.py startapp web**, Django creates a sub-directory of your project directory, and creates the following files:

**web/**

`__init__.py`  
`models.py`  
`tests.py`  
`views.py`

`__init__.py`, which serves the same purpose as at the project level.

`models.py`, which should hold the application's model classes.

`views.py`, which is for any custom views the application wants to provide.

## Edit **views.py**

Open **views.py** (which is in **mysite/web** directory) in terminal.

```
# vi views.py
```

Type the below code and save it.

```
# Create your views here.

from django.http import HttpResponseRedirect
import datetime

def current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body>Hello World! It is now %s.</body></html>" % now
    return HttpResponseRedirect(html)
```

#### Note :

In **Python**, Indentation is an important thing. Wrong indentation will result in an indentation error. So you can use **GUI editor** like **geany**. Because the **geany** editor keeps the indentation on each line automatically. You can use other editors also. In that case you have care yourself about that code indentation and all.

#### Edit urls.py

Open urls.py (which is in **mysite** directory) in terminal.

```
# vi urls.py
```

Do the changes as per the below code and save it.

```
from django.conf.urls.defaults import *
from web.views import current_datetime

urlpatterns = patterns('',
    (r'^$', current_datetime),
)
```

#### Run the Server

Make sure your current directory is **mysite** project directory or get in to the project directory.

```
# cd mysite
```

Now, run the django server using the following command

```
# python manage.py runserver
Validating models...
0 errors found

Django version 1.2.5, using settings
'mysite.settings'

Development server is running at
http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

if there is no errors found in terminal. You can check the output in browser with the url  
<http://127.0.0.1:8000/>

## 2. Simple blog application using Django

### Create a project

```
# django-admin.py startproject blog
```

Get in to project folder which you created now.

```
# cd blog
```

### Create an application

```
# django-admin.py startapp posts
```

### Edit models.py

Open **models.py** (which is in **blog/posts** directory) in terminal.

```
# vi models.py
```

#### Note :

In **Python**, Indentation is an important thing. Wrong indentation will result in an indentation error. So you can use **GUI editor** like **geany**. Because the **geany** editor keeps the indentation on each line automatically. You can use other editors also. In that case you have care yourself about that code indentation and all.

Type the below code and save it.

```
from django.db import models
import datetime

# Create your models here.

class Posts(models.Model):
    """
    Model that holds blog posts
    """
```

```
title = models.CharField(max_length=200)
body = models.CharField(max_length=1000)
posted_date = models.DateTimeField(
    default= datetime.datetime.now, blank=True)
```

## Create a new file named forms.py in blog/posts directory

### Edit forms.py

Open **forms.py** (which is in **blog/posts** directory) in terminal.

```
# vi forms.py
```

Type the below code and save it.

```
from django import forms

class BlogPostForm(forms.Form):
    """
    Form to get user posts
    """
    title = forms.CharField(label="Title", required=True)
    body = forms.CharField(label="Body",
        widget= forms.Textarea, required=True)
```

### Edit views.py

Open **views.py** (which is in **blog/posts** directory) in terminal.

```
# vi views.py
```

Type the below code and save it.

```
# Create your views here.
from django.http import HttpResponseRedirect
from django.views.generic.simple import direct_to_template
from django.http import HttpResponseRedirect
from posts.models import Posts
from posts.forms import BlogPostForm

def get_new_blog_post_form(request):
    """
    Method that renders the new post form
    """
    old_posts = Posts.objects.all()
    if request.method == "GET":
        return direct_to_template(request,'post.html',
{'posts':old_posts,'postform':BlogPostForm()})

def save_blog_post(request):
    """
    Storing posts to database
    """
    if request.method == 'POST':
        form = BlogPostForm(request.POST)
        if form.is_valid():
            blogtable = Posts(
                title = request.POST['title'],
                body = request.POST['body'],
            )
            blogtable.save()
            return HttpResponseRedirect('/posts/new/')
    else:
        form = BlogPostForm()
    variables= RequestContext(request, {
        'form': form
    })
    return render_to_response('post.html',variables)
```

## Edit urls.py

Open urls.py (which is in **blog** directory) in terminal.

```
# vi urls.py
```

Do the changes as same as below code and save it.

```
from django.conf.urls.defaults import *
from blog.posts.views import *

urlpatterns = patterns('blog.posts',
    (r'^posts/new/$', 'views.get_new_blog_post_form'),
    (r'^posts/add/$', 'views.save_blog_post'),
)
```

## Create a new directory named templates in blog directory

```
# mkdir templates
```

Get into project folder which you created now.

```
# cd templates
```

## Create two files named base.html, post.html in blog/templates directory

### Edit base.html

Open **base.html** (which is in **blog/templates** directory) in terminal.

```
# vi base.html
```

Type the below code and save it.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
{%
  block title %
}
{%
  endblock %
}</head>
<body>
{%
  block content %
}
{%
  endblock %
}</body>
</html>
```

## Edit post.html

Open **post.html** (which is in **blog/templates** directory) in terminal.

```
# vi post.html
```

Type the below code and save it.

```
{% extends 'base.html' %}  
{% block title %}  
<title> My Blog Post </title>  
{% endblock %}  
{% block content %}  
<center><h3><font color="green">My Blog Post </font></h3></center>  
<form action="/posts/add/" method="POST">  
{% csrf_token %}  
<table>  
{{ postform.as_table }}  
<tr>  
<td>  
<input type="submit" id="post" value="Add Post">  
</td>  
</tr>  
</table>  
<hr>  
<center><h3><font color="red">Old Blog Post </font></h3></center>  
{% if posts %}  
<table>  
<!-- &lt;tr&gt;&lt;td width="40%"&gt;&lt;strong&gt;Title&lt;/strong&gt;&lt;/td&gt;&lt;td<br/>width="60%"><strong><font color="blue">Body</font></strong></td></tr>  
-->  
{% for post in posts %}  
<tr><td>  
<h5><font color="blue">{{post.title }}</font><h5><br>  
<h6>&nbsp;{{post.body }}</h6>  
</td></tr>  
{% endfor %}  
</table>  
{% else %}  
<br>  
<span><strong>No posts available<strong></span><br>  
{% endif %}  
  
{% endblock %}
```

## Edit settings.py

Open settings.py (which is in **blog** directory) in terminal.

```
# vi settings.py
```

Do the changes as same as below code and save it.

### Database Part:

Type the Database **ENGINE** and **NAME**

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'blog.db',
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': ''
    }
}
```

### TEMPLATE\_DIRS Part:

Add the first five lines of code before **TEMPLATE\_DIRS** block. And add the last line within the **TEMPLATE\_DIRS** block.

```
import os
try:
    ROOT_DIR
except NameError:
    ROOT_DIR = os.path.dirname(__file__)

TEMPLATE_DIRS = (
    # Put strings here, like "/home/html/django_templates"
    # or "C:/www/django/templates".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative
    paths.
    os.path.join(ROOT_DIR, 'templates'),
)
```

## INSTALLED\_APPS Part:

Add a line '**blog.posts**' next to the '**django.contrib.messages**' line

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'blog.posts',
    # Uncomment the next line to enable the admin:
    #'django.contrib.admin',
    # Uncomment the next line to enable admin
documentation:
    # 'django.contrib.admindocs',
)
```

Finally **save** the **settings.py** file

All the customization has been finished. You can run the django server now.

## Run the Server

Make sure your current directory is **blog** project directory or get in to the project directory.

```
# cd blog
```

Next execute the below command for database creation. When executing the **syncdb** command, it will ask for **Yes/No** option for creating admin user. At this time give **No** as option.

```
# python manage.py syncdb
```

Now, run the django server using the following command

```
# python manage.py runserver
Validating models...
0 errors found

Django version 1.2.5, using settings 'blog.settings'
Development server is running at
http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

if there is no errors found in terminal. You can check the output in browser with the url  
<http://127.0.0.1:8000/>

### **Page Not Found error :**

if You check the output in browser with the django's default url <http://127.0.0.1:8000/> you will get **Page Not Found error** only. That is because we gave the **main page url** as **/posts/new/** in **urls.py**.

So check the main page by giving the url <http://127.0.0.1:8000/posts/new/>

That's it. you have created your own blog application now.

## **Python References**

<http://diveintopython.org/toc/index.html>

<http://www.ibiblio.org/g2swap/byteofpython/read/>

<http://openbookproject.net/thinkcs/python/english3e/index.html>

## **Django References**

<https://www.djangoproject.com/>

<http://www.djangobook.com/en/2.0/>

## 9 : Setting Up of Complete Network Interface

### Aim:

To set up Network Interface using 'ifconfig' command and the related system files.

### Introduction:

A **Network** interface is the point of interconnection between a computer and a private or public network. A network interface is generally a network interface card (NIC).

**Ethernet** is a family of [computer networking](#) technologies for [local area networks](#) (LANs) commercially introduced in 1980. Standardized in [IEEE 802.3](#), Ethernet has largely replaced competing wired LAN technologies.

**Gateway** is a node that allows you to gain entrance into a network and vice versa. On the Internet the node which is the stopping point can be a gateway or a host node. A computer that controls the traffic your network or your ISP (Internet Service Provider) receives is a node.

**DNS (Domain name server)**, the system that automatically translates Internet addresses to the numeric machine addresses that computers use

**Iptables** is a [user space](#) application program provided by the [Linux kernel firewall](#) (implemented as different [Netfilter](#) modules), that allows us to configure the tables and the chains and rules it stores. Iptables requires elevated privileges to operate and must be executed by user [root](#), otherwise it fails to function. On most Linux systems, iptables is installed as /usr/sbin/iptables and documented in its [man page](#), which can be opened using man iptables when installed.

### Prerequisites :

To enable networking, you must configure your network interface card or cards with an IP address and netmask.

The kernel must have support for your cards compiled in, either as modular support or direct support. To set your cards up, do the following. Assume your network as 192.168.2.0, IP=192.168.x.y, broadcast=192.168.x.y, netmask=255.255.255.0, gateway=192.168.x.y, nameserver=192.168.x.y

Note:- Please replace 'x' & 'y' with the relevant numbers from your network.

## Procedure :

### Step 1 : Learn about your network & tools required

#### a) Basic Information

1. Determine your machines IP address from your network administrator
  2. Your network mask. This determines which portion of the IP address specifies the subnetwork number and which portion specifies the host.  
Class C (most networks) 255.255.255.0
  3. Your network address which is your IP address bit wise anded with the network mask.
  4. Your broadcast address. Used to broadcast packets to every machine on your subnet.
  5. Your gateway address. The address of the machine that is your gateway to the outside world.
- In many cases: Ex: IP: 192.168.2.100 Gateway: 192.168.2.201
6. Your nameserver address. Translates host names into IP addresses. 192.168.2.205

#### b) Configuration tools

There are many network configuration tools today. They are:

netconf	A GUI interactive interface
linuxconf	A GUI interactive interface which includes netconf configuration.
netconfig	A GUI step by step interface
ifconfig	A text based program to configure the network interface. Type "man ifconfig" for info.

These programs will modify values in the following files:

- /etc/sysconfig/network - Defines your network and some of its characteristics.
- /etc/HOSTNAME - Shows the host name of this host. If your name is "myhost" then that is exactly the text this file will contain.
- /etc/resolv.conf - Specifies the domain to be searched for host names to connect to, the nameserver address, and the search order for the nameservers.
- /etc/hosts - Shows addresses and names of local hosts.
- /etc/networks - Provides a database of network names with network addresses similar to the /etc/hosts file. This file is not required for operation.
- /etc/sysconfig/network-scripts/ifcfg-eth\* - There is a file for each network interface. This file contains the IP address of the interface and many other setup variables.

#### c) Analysis Tools

- netstat - Displays information about the systems network connections, including port connections, routing tables, and more. The command "netstat -r" will display the routing table.
- traceroute - This command can be used to determine the network route from your computer to some other computer on your network or the

internet. To use it you can type "route IPaddress" of the computer you want to see the route to.

- nslookup - Used to query DNS servers for information about hosts.
- arp - This program lets the user read or modify their arp cache.
- tcpdump - This program allows the user to see TCP traffic on their network.
- dig(1) - Send domain name query packets to name servers for debugging or testing.

Try the above tools.

## Step 2 : Configuring the Network

You can use one of the above tools or configure the network the old fashioned way as follows:

1. First to use networking on any permanent basis you should setup the file /etc/sysconfig/network similar to the example shown below.
2. Assign an ip address with "ifconfig eth0 192.168.x.y netmask 255.255.255.0 up".
3. Tell your machine that a hub is ready for information with the command "route add -net 192.168.0.0 netmask 255.255.255.0 eth0"
4. To contact hosts outside your network if a machine with IP address 192.168.x.y is the gateway use the command "route add default gw 192.168.x.y eth0"
5. If using a dialup connection use the command "route add default ppp0" The word default says if the packet is not for a machine on your local network, send it to the default device.

These settings are not permanent, but go away the next time you boot. They are normally set up in the directory /etc/sysconfig/network-scripts. Add the network interface to the file /etc/sysconfig/network-scripts/ifcfg-eth\*. For example the file ifcfg-eth0 if for the first ethernet interface, ifcfg-eth1 for the second, and so on. The lines given below are from ifcfg-eth0: (Assuming 192.168.2.100 as your machine IP)

```
DEVICE="eth0"
IPADDR="192.168.2.100"
NETMASK="255.255.0.0"
ONBOOT="yes"
BOOTPROTO="none"
```

Unless you know what you're doing it is best to use a network configuration tool.

## Step 3 : Configuring an interface for multiple IP addresses

If you want to configure your network card to act as more than one IP address, issue the following command:

```
ifconfig dummy 192.168.1.102 netmask 255.255.255.0
```

This uses the dummy system interface capability supported in the kernel to setup another virtual interface which operates at IP address 192.168.1.102. Substitute the IP address that you want your virtual interface to be with an appropriate netmask for your network. To disable this, issue the following command.

```
ifconfig dummy down
```

Another way to use multiple IP addresses on one ethernet card is to set up a new file in your /etc/sysconfig/network-scripts directory. Copy your ifcfg-eth0 role to ifcfg-eth0:0. Edit that file and rename the device to "eth0:0" and the IP address to the desired IP address. You may also want to modify BROADCAST, NETWORK, or NETMASK. You can continue adding IP addresses by using :1, :2, etc such as ifcfg-eth0:2. Sample of ifcfg-eth0:1 for the IP : 192.168.2.101

```
DEVICE="eth0:1"
IPADDR="192.168.2.101"
NETMASK="255.255.0.0"
ONBOOT="yes"
BOOTPROTO="none"
```

To make it effective, you must reboot your system or issue the command  
"/etc/rc.d/init.d/network restart" as root.

#### Step 4: Gaining familiarity with Networking files and configuring for DNS

- /etc/sysconfig/network  
The /etc/inittab file contains the entry  
"si::sysinit:/etc/rc.d/rc.sysinit" which causes the system at startup to run the rc.sysinit script. The rc.sysinit file expects to find the file /etc/sysconfig/network if networking is to be enabled.  
The network file looks like this: Make the changes as per the highlighted lines.

```
NETWORKING=yes
FORWARD_IPV4=false
HOSTNAME=yourname.collegename.com
DOMAINNAME=collegename.com
GATEWAY=192.168.x.y
GATEWAYDEV=eth0
```

Where GATEWAYDEV is the network interface card that is attached to the network the gateway machine is on. The GATEWAY is the actual IP address of the gateway machine.(Assuming 192.168.2.100 as your machine IP)

- /etc/hosts - Defines local hosts.  
127.0.0.1 localhost localhost.localdomain  
192.168.1.100 yourname.collegename.com yourname
- /etc/named.conf - Used for domain name service to configure named. Other files used are dependent on this file. This file is explained further in the DNS section
- /etc/resolv.conf - Specifies the domain to be searched for host names to connect to, the nameserver address, and the search order for the nameservers.

```
domain collegename.com
search collegename.com mynet.net
nameserver 192.168.1.100
nameserver 192.168.199.1
nameserver 192.168.1.10
```

The third line specifies that DNS should be tried on your machine first then use the normal nameserver on the fifth line. The fourth line specifies that my machine is running nameservices on another network which is using interface 192.168.199.1. This assumes the nameserver is set up on your machine.

You can add additional nameservers like google DNS

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

## Step 5: Routing – Setting up of Gateway

Routing table information is used to route incoming and outgoing network diagrams to other machines. On most simple configurations, there are three routes. One for sending packets to your own machine, one for sending packets to other machines on your network and one for sending packets to other machines outside your network through the gateway. Two programs (ifconfig and route) are used to configure these parameters. They are described in more detail in the routing section.

Static routes: IP (Internet Protocol) uses a routing table to determine where packets should be sent. First the packet is examined to see if its' destination is for the local or remote network. If it is to be sent to a remote network, the routing table is consulted to determine the path. If there is no information in the routing table then the packet is sent to the default gateway. Static routes are set with the `route` command and with the configuration file:

- `/etc/sysconfig/network-scripts/route-eth0`

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.x.y
NETMASK=255.255.255.0
GATEWAY=XXX.XXX.XXX.XXX
HOSTNAME=node-name.name-of-domain.com
DOMAIN=name-of-domain.com
```

- Define route with with the '`route`' command:

Examples:

```
route add default gw XXX.XXX.XXX.XXX dev eth0
route add -net XXX.XXX.XXX.0 netmask 255.255.255.0 gw
XXX.XXX.XXX.XXX dev eth0
```

Where **XXX.XXX.XXX.XXX** is the gateway to the internet as defined by your ISP or network operator.

Note :- If a mistake is made just repeat the route command substituting "del" in place of "add".

## Step 6: Managing iptables

Try the following .. as root user...

- a) You can start, stop, and restart iptables after booting by using the commands:

```
# service iptables start  
# service iptables stop  
# service iptables restart
```

- b) To get iptables configured to start at boot, use the chkconfig command:.

```
# chkconfig iptables on
```

- c) You can determine whether iptables is running or not via the service iptables status command. Fedora Core will give a simple status message. For example

```
# service iptables status
```

Firewall is stopped.

### References :

- 1.<http://www.yolinux.com/TUTORIALS/LinuxTutorialNetworking.html>
- 2.[http://www.linuxhomenetworking.com/wiki/index.php/Quick\\_HOWTO:\\_Ch03:\\_Linux\\_Networking](http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO:_Ch03:_Linux_Networking)
- 3.<http://whatismyipaddress.com/gateway>
- 4.<http://en.wikipedia.org/wiki/Ethernet>

## 10. Package Management system

### Aim:

To build, maintain and serve rpm package repository over http and to configure the client machines to access this repository.

### Introduction:

In software, a package management system, also called package manager, is a collection of software tools to automate the process of installing, upgrading, configuring, and removing software packages for a computer's operating system in a consistent manner. It typically maintains a database of software dependencies and version information to prevent software mismatches and missing prerequisites.

Packages are distributions of software, applications and data. Packages also contain metadata, such as the software's name, description of its purpose, version number, vendor, checksum, and a list of dependencies necessary for the software to run properly. Upon installation, metadata is stored in a local package database.

Operating systems based on Linux and other Unix-like systems typically consist of hundreds or even thousands of distinct software packages; in the former case, a package management system is a convenience, in the latter case it becomes essential.

### Description:

Students will build, maintain and serve rpm package repository to the client machines

### Pre-requisites:

- I) LAN setup
- II) Fedora 12
- III) Firewall should be disabled

## Steps:

To Create a repository for a given rpm packages . Please follow these steps

- 1) Create directory

```
mkdir -p /var/www/html/RPMS
```

- 2) Create a temporary to download few rpm files from foss lab server

```
mkdir tmp
```

```
cd tmp
```

- 3) To download 20 MB of rpm files from foss-lab server , execute the following command. This file limit can be changed by increasing the value in the command (ex -Q100m)

- 4) wget -rnd -Q20m [http://foss-lab-server-ip/content/packages/Fedora\\_Packages/os/Packages/](http://foss-lab-server-ip/content/packages/Fedora_Packages/os/Packages/)

- 5) cp \*.rpm /var/www/html/RPMS/

- 6) we have copied few rpm files to /var/www/html/RPMS from foss lab server instead of that you can use your own rpms also to build a repository.

- 7) Install createrepo to generate the metadata of the software packages

```
yum install createrepo
```

- 8) Now, we will have to create repo of the directory in which we have copied the RPMS. Following is the command to create the repo

```
createrepo /var/www/html/RPMS
```

Once the above command gets completed you will find repodata directory in /var/www/html/RPMS folder

- 9) **Setting up HTTP server to serve the content**

```
yum install httpd
```

```
service httpd restart
```

- 10) configuring YUM in **client machine** to work with our repository. Create a new file in /etc/yum.repo.d/

```
vim /etc/yum.repo.d/test.repo
```

and enter the following text in the file and save it.

```
[repo]
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://192.168.2.68/RPMS/
#give your server IP here
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora file:///etc/pki/rpm-gpg/RPM-GPG-KEY
```

11) Now Try to install any package using " yum " command

for example:

```
yum install 3proxy.i686
```

Note : if you want to serve these packages using FTP then install **vsftpd** and then copy /var/www/html/RPMS/ folder to /var/ftp/RPMS/

## References:

1. <http://www.bestdesigns.co.in/blog/create-yum-repository-local-machine>
2. [http://en.wikipedia.org/wiki/Package\\_management\\_system](http://en.wikipedia.org/wiki/Package_management_system)
3. <http://www.phpsuperblog.com/fedora-12/fedora-12-setup-ftp-server-with-vsftpd/>

## 11: Writing userspace drivers using FUSE

### Aim:

To understand the concepts behind Filesystems in Userspace and to write a simple userspace driver.

### Introduction:

A **file system (filesystem)** is means to organize data expected to be retained after a program terminates by providing procedures to store, retrieve and update data, as well as manage the available space on the device(s) which contain it. A file system organizes data in an efficient manner and is tuned to the specific characteristics of the device. There is usually a tight coupling between the operating system and the file system. Some filesystems provide mechanisms to control access to the data and metadata. Insuring reliability is a major responsibility of a filesystem. Some filesystems provide a means for multiple programs to update data in the same file nearly at the same time.

Without a filesystem programs would not be able to access data by file name or directory and would need to be able to directly access data regions on a storage device.

File systems are used on data storage devices such as magnetic storage disks or optical discs to maintain the physical location of the computer files. They may provide access to data on a file server by acting as clients for a network protocol (e.g., NFS, SMB, or 9P clients), or they may be virtual and exist only as an access method for virtual data (e.g., procfs).

Generally filesystem development was the job of the kernel developer. Creating filesystems required knowledge of kernel programming and technologies. The developer needed a good expertise in of C and C++. If other programmers wanted to add features to a filesystem, they had to gain all the knowledge which was a time consuming task.

A userspace filesystem is a filesystem in which data and metadata are provided by an ordinary userspace process.

**Filesystem in Userspace (FUSE)** is a loadable kernel module for Unix-like computer operating systems that lets non-privileged users create their own file systems without editing kernel code. This is achieved by running file system code in user space while the FUSE module provides only a "bridge" to the actual kernel interfaces.

FUSE is particularly useful for writing virtual file systems. Unlike traditional file systems that essentially save data to and retrieve data from disk, virtual filesystems do not actually store data themselves. They act as a view or translation of an existing file system or storage device. In principle, any resource available to a FUSE implementation can be exported as a file system.

FUSE is made up of three main parts:

- A kernel filesystem modules
- A userspace library
- A mount/unmount program

The fuse distribution contains a small example program which implements a simple filesystem called the hello world filesystem which is less than a 100 lines long. This exercise consists of studying the sample code and building it and executing it on a machine.

## Description:

Students will study the given simple user space filesystem and implement it on their machines.

## Pre-requisites:

The fuse package and its development files have to be installed in the system.

```
> yum install fuse-devel
```

## The Program:

First we create a new directory for this exercise.

```
> mkdir fuseprog  
> cd fuseprog
```

```
> gedit hello.c
```

The Code:

```
#define FUSE_USE_VERSION 26

#include <fuse.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

```
#include <errno.h>
#include <fcntl.h>

static const char *hello_str = "Hello World!\n";
static const char *hello_path = "/hello";

static int hello_getattr(const char *path, struct stat *stbuf)
{
    int res = 0;

    memset(stbuf, 0, sizeof(struct stat));
    if(strcmp(path, "/") == 0) {
        stbuf->st_mode = S_IFDIR | 0755;
        stbuf->st_nlink = 2;
    }
    else if(strcmp(path, hello_path) == 0) {
        stbuf->st_mode = S_IFREG | 0444;
        stbuf->st_nlink = 1;
        stbuf->st_size = strlen(hello_str);
    }
    else
        res = -ENOENT;

    return res;
}

static int hello_readdir(const char *path, void *buf, fuse_fill_dir_t
filler,
                        off_t offset, struct fuse_file_info *fi)
{
    (void) offset;
    (void) fi;

    if(strcmp(path, "/") != 0)
        return -ENOENT;

    filler(buf, ".", NULL, 0);
    filler(buf, "..", NULL, 0);
    filler(buf, hello_path + 1, NULL, 0);

    return 0;
}

static int hello_open(const char *path, struct fuse_file_info *fi)
{
```

```
if(strcmp(path, hello_path) != 0)
    return -ENOENT;

if((fi->flags & 3) != O_RDONLY)
    return -EACCES;

return 0;
}

static int hello_read(const char *path, char *buf, size_t size, off_t offset,
                      struct fuse_file_info *fi)
{
    (void) fi;
    size_t len;
    if(strcmp(path, hello_path) != 0)
        return -ENOENT;

    len = strlen(hello_str);
    if (offset < len) {
        if (offset + size > len)
            size = len - offset;
        memcpy(buf, hello_str + offset, size);
    } else
        size = 0;

    return size;
}

static struct fuse_operations hello_oper = {
    .getattr      = hello_getattr,
    .readdir      = hello_readdir,
    .open         = hello_open,
    .read         = hello_read,
};

int main(int argc, char *argv[])
{
    return fuse_main(argc, argv, &hello_oper, NULL);
}
```

We will look at the code in more detail.

We specify the api level of the code we are compiling.

```
#define FUSE_USE_VERSION 26
```

To be able to program a FUSE filesystem you need to include a series of headers:

```
#include <fuse.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <fcntl.h>
```

- fuse.h contains definitions for basic functions needed to implement a filesystem
- errno.h contains definitions of error numbers
- fcntl.h contains definitions of file options (the ones used with fcntl() and open() )

### Global Variables:

The next thing to do in our example will be the the definition of global variables containing path and data of a /hello file:

```
static const char *hello_str = "Hello World!\n";
static const char *hello_path = "/hello";
```

### getattr() function

This function returns metadata concerning a file specified by path in a special [stat](#) structure. It has to be declared static, as all functions passed in the [fuse\\_operations](#) structure to [fuse\\_main\(\)](#), to work properly.

```
static int hello_getattr(const char *path, struct stat *stbuf)
{
    int res = 0;
```

Reset memory for the stat structure (set it's contents to 0's)

```
memset(stbuf, 0, sizeof(struct stat));
```

Now we have to check which file attributes we have to return, our filesystem will have only two files (one regular file and one directory) so we can use simple if here.

```
if(strcmp(path, "/") == 0) {
```

st\_mode contains file attributes, S\_IFDIR is equal to 0040000 and it marks a file as a directory, then it is binary added to 0755 value, being normalfile permissions written in octal format.

```
stbuf->st_mode = S_IFDIR | 0755;
```

st\_nlink contains number of hardlinks, this file being a directory and not having another directories inside will have 2 links

```
stbuf->st_nlink = 2;
}
```

If the user requests permissions for /hello file

```
else if(strcmp(path, hello_path) == 0) {
```

Set the file attributes to a regular file (S\_IFREG) and give it 0444 permissions

```
stbuf->st_mode = S_IFREG | 0444;
```

The /hello file is one of two files in this filesystem so it will have only one hard link pointing to it

```
stbuf->st_nlink = 1;
```

File size is equal to the size of the hello\_str

```
stbuf->st_size = strlen(hello_str);
}
```

If the user asks for permissions for any other file, return negated ENOENT value. ENOENT is declared in the errno.h and means that "A component of the path does not exist, or the path is an empty string.", it is negated to identify it as a FUSE error.

```
else
    res = -ENOENT;

    return res;
}
```

## readdir() function

Next very important function is used to read directory contents. Because our implementation is very simple, we will be interested only in path, buf and filler arguments passed to it. Path is the path to the directory from which we will have to read our contents, buf will hold them, and filler is [afuse\\_fill\\_dir\\_t](#) function which we will use to add contents to directory. Offset and fi are not important now.

```
static int hello_readdir(const char *path, void *buf, fuse_fill_dir_t
filler,
                         off_t offset, struct fuse_file_info *fi)
{
    (void) offset;
    (void) fi;
```

Check if user is asking for the content's of the root directory, if it's not the case, return "Directory does not exist" error number.

```
if(strcmp(path, "/") != 0)
    return -ENOENT;
```

Add "." and ".." directory entries.

```
filler(buf, ".", NULL, 0);
filler(buf, "..", NULL, 0);
```

Add hello file, because its name should show as "hello", not "/hello" we have to omit first character in hello\_path.

```
filler(buf, hello_path + 1, NULL, 0);

return 0;
}
```

## [open\(\)](#) function

This function checks whatever user is permitted to open the /hello file with flags given in the [fuse\\_file\\_info](#) structure.

```
static int hello_open(const char *path, struct fuse_file_info *fi)
{
```

If user is asking for anything besides /hello we tell him, that such file does not exist.

```
    if(strcmp(path, hello_path) != 0)
        return -ENOENT;
```

If the user wants to open the file for anything else than reading only, we tell him that he does not have sufficient permissions.

```
    if((fi->flags & 3) != O_RDONLY)
        return -EACCES;
```

If neither of the above things happened we tell the user that he can open the file

```
    return 0;
}
```

## read() function

Read function is used to feed the user with data from the file. Path is the path to the file to read, buf is a buffer to which we will copy information from the file, size is at the same time the size of the buffer and tells how much of the data we have to read, offset is the place from which we have to start reading and fi contains additional information about the type of access. Since this procedure will be called only after open() (and because this particular open allows only read-only operations on /hello file) we don't have to pay any attention to it's contents.

```
static int hello_read(const char *path, char *buf, size_t size, off_t
offset,
                      struct fuse_file_info *fi)
{
    (void) fi;
```

len will contain number of read bytes

```
    size_t len;
```

User cannot open any file besides /hello

```
if(strcmp(path, hello_path) != 0)
    return -ENOENT;
```

Set the number of chars to read as length of hello\_str

```
len = strlen(hello_str);
```

Check if the amount of data to read does not exceed the length of data, if yes, return this smaller size, then copy requested amount of data from hello\_str

```
if (offset < len) {
    if (offset + size > len)
        size = len - offset;
    memcpy(buf, hello_str + offset, size);
} else
    size = 0;

return size;
}
```

## declaring structure containing implemented fuse operations

Functions you have implemented must be contained in [fuse\\_operations](#) structure

```
static struct fuse_operations hello_oper = {
    .getattr      = hello_getattr,
    .readdir      = hello_readdir,
    .open         = hello_open,
    .read         = hello_read,
};
```

## main() function

main() function can have only one function, [fuse\\_main\(\)](#):

```
int main(int argc, char *argv[])
{
    return fuse_main(argc, argv, &hello_oper, NULL);
```

}

### Running the program:

To run the program, compile it with gcc:

```
> gcc -Wall `pkg-config fuse --cflags --libs` hello.c -o hello
```

Run the program as root user with a single parameter - a mount point:

```
> su  
# mkdir tmp  
# ./hello tmp/  
#
```

After running the program you should have a single file in the tmp/ directory named hello, its content should be Hello World!:

```
# ls tmp/  
# ./hello tmp/  
# ls tmp/  
hello  
# cat tmp/hello  
Hello World!  
#
```

Unmount the file system.

```
# umount tmp/
```

### References:

1. <http://fuse.sourceforge.net/> - FUSE homepage. The wiki also contains some documentation.
2. [http://en.wikipedia.org/wiki/Filesystem\\_in\\_Userspace](http://en.wikipedia.org/wiki/Filesystem_in_Userspace) - List of various FUSE projects

## 12: Compiling software from source

### Aim:

To learn about the common build systems available in Linux and to use them.

### Introduction:

Open source software is distributed in source code form. In case of popular software Linux distributions will often have the software packaged in their repositories. If the package is not package is not in the repository the user has to compile the software from source. To do this the user has to understand about the build system used in the project.

The **GNU build system**, also known as the **Autotools**, is a suite of programming tools designed to assist in making source-code packages portable to many Unix-like systems. It can be difficult to make a software program portable: the C compiler differs from system to system; certain library functions are missing on some systems; header files may have different names. One way to handle this is write conditional code, with code blocks selected by means of preprocessor directives (#ifdef); but because of the wide variety of build environments this approach quickly becomes unmanageable. The GNU build system is designed to address this problem more manageably.

### Tools included in the GNU build system

The GNU build system comprises the GNU utility programs Autoconf, Automake, and Libtool. Other related tools frequently used with the GNU build system are GNU's make program, GNU gettext, pkg-config, and the GNU Compiler Collection, also called GCC.

#### GNU Autoconf

Autoconf generates a configure script based on the contents of a config.ac file which characterizes a particular body of source code. The configure script, when run, scans the build environment and generates a subordinate config.status script which, in turn, converts other input files and most commonly Makefile.in into output files (Makefile) which are appropriate for that build environment. Finally the make program uses Makefile to generate executable programs from source code.

The complexity of the GNU build system reflects the variety of circumstances under which a body of source code may be built.

- If a source code file is changed then it suffices to re-run make which only re-compiles that part of the body of the source code affected by the change.
- If a .in file has changed then it suffices to re-run config.status and make.
- If the body of source code is copied to another computer then it is suffices to re-run configure (which runs config.status) and make. (For this reason source code using the GNU build system is normally distributed without the files that configure generates.)
- If the body of source code is changed more fundamentally then configure.ac and the .in files need to be changed and all subsequent steps also followed.

To process files, autoconf uses the GNU implementation of the m4 macro system.

Autoconf comes with several auxiliary programs such as Autoheader, which is used to help manage [C](#) header files; Autoscans, which can create an initial input file for Autoconf; and ifnames, which can list C pre-processor identifiers used in the program.

### **GNU Automake**

Automake helps to create portable Makefiles, which are in turn processed with the make utility. It takes its input as Makefile.am, and turns it into Makefile.in, which is used by the configure script to generate the file Makefile output.

### **GNU Libtool**

Libtool helps manage the creation of static and dynamic libraries on various Unix-like operating systems. Libtool accomplishes this by abstracting the library-creation process, hiding differences between various systems (e.g. GNU/Linuxsystems vs. Solaris).

### **Gnulib**

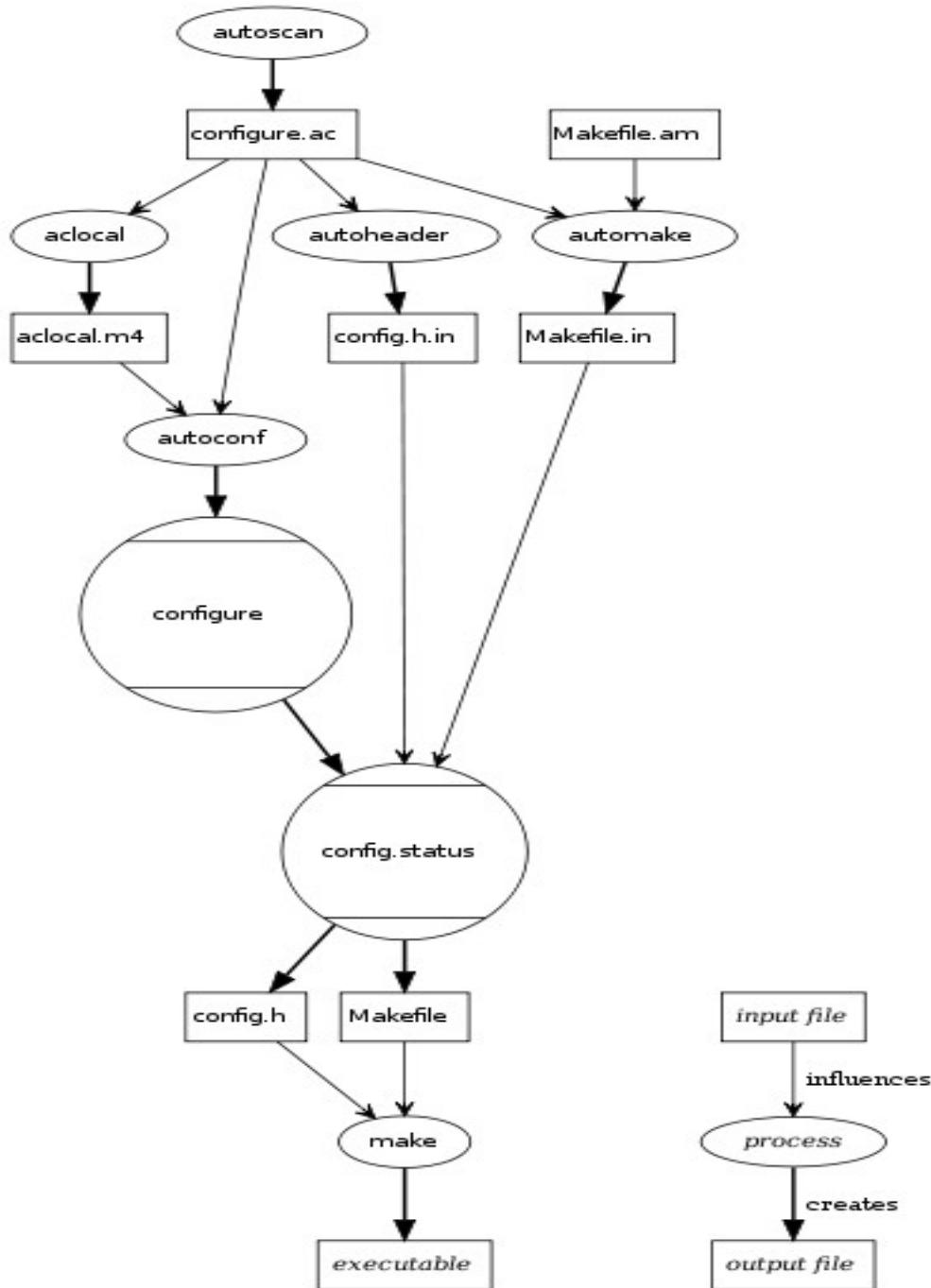
Gnulib simplifies the process of making software that uses Autoconf and Automake portable to a wide range of systems.

### **Make**

In software development, **make** is a utility that automatically builds executable programs and libraries from source code by reading files called **makefiles** which specify how to derive the target program. Make can decide where to start through topological sorting. Though integrated development environments and language-specific compiler features can also be used to manage the build process in modern systems, *make* remains widely used, especially in Unix.

Make is typically used to build executable programs and libraries from source code. Generally though, any process that involves transforming a dependency file to a target result (by executing some number of arbitrary commands) is applicable to make. To cite an example, make could be

used to detect a change made to an image file (the dependency) and the target actions that result might be to convert the file to some specific format, copy the result into a content management system, and then send e-mail to a predefined set of users that the above actions were performed.



Flow diagram of autoconf and automake

## CMake

**CMake** is a unified, cross-platform, open-source build system that enables developers to build, test and package software by specifying build parameters in simple, portable text files. It works in a compiler-independent manner and the build process works in conjunction with native build environments, such as make, Apple's Xcode and Microsoft Visual Studio. It also has minimal dependencies, C++ only. CMake is open source software.

CMake can:

- Create libraries
- Generate wrappers
- Compile source code
- Build executables in arbitrary combinations

## Apache Ant

**Apache Ant** is a software tool for automating software build processes. It is similar to Make but is implemented using the Java language, requires the Java platform, and is best suited to building Java projects.

The most immediately noticeable difference between Ant and Make is that Ant uses XML to describe the build process and its dependencies, whereas Make uses Makefile format. By default the XML file is named build.xml.

Ant is an Apache project. It is open source software, and is released under the Apache Software License.

## Description:

In this exercise students will learn how to use the GNU Autotools, Plain makefiles, CMake and Ant build scripts. Students will write the build scripts for the various build tools except for the Autotools where they will study an already existing project as a sample of how to download and compile opensource software from the net.

## Pre-requisites:

The computers need the development tools to be installed. Instructions for installing them will be given along with each exercise.

## Exercises:

Create a directory for all the programs in the exercise.

```
> mkdir build_systems  
> cd build_systems
```

### 1. Make

We will be using a simple program written in C and write a makefile to compile the program.

```
> mkdir gnumake  
> cd gnumake  
  
> gedit squareroot.c
```

The Code:

```
// A simple program that computes the square root of a number  
#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
int main (int argc, char *argv[])  
{  
    if (argc < 2)  
    {  
        fprintf(stdout,"Usage: %s number\n",argv[0]);  
        return 1;  
    }  
    double inputValue = atof(argv[1]);  
    double outputValue = sqrt(inputValue);  
    fprintf(stdout,"The square root of %g is %g\n",  
            inputValue, outputValue);  
    return 0;  
}
```

Test it by compiling it once:

```
> gcc squareroot.c -o squareroot -lm  
> ./squareroot 49  
> The square root of 49 is 7
```

Now we write a simple makefile to compile the program.

```
> gedit Makefile
```

The Code:

```
# Commands start with TAB not spaces
CC      = gcc
CFLAGS  = -g
LDFLAGS = -lm

all: squareroot

squareroot: squareroot.o

squareroot.o: squareroot.c

clean:
    rm -f squareroot squareroot.o
```

Now we test the Makefile:

```
> make
make: Nothing to be done for `all'.
> make clean
rm -f squareroot squareroot.o
> make
gcc -g -c -o squareroot.o squareroot.c
gcc -lm squareroot.o -o squareroot
> ./squareroot 49
The square root of 49 is 7
>
```

## 2. CMake

We will now write a simple script for CMake to compile the previously written program.

First we install CMake

```
> yum install cmake
```

We create a new directory and copy the source code to it.

```
> cd ..  
> mkdir cmake  
> cd cmake  
> cp ../gnumake/squareroot.c .
```

Now we create configuration files for CMake.

```
> gedit CMakeLists.txt
```

The Code:

```
cmake_minimum_required (VERSION 2.6)  
project (squareroot)  
add_executable(squareroot squareroot.c)  
TARGET_LINK_LIBRARIES(squareroot m)
```

CMake is commonly used with out of source builds ie, we build the program in a directory separate from the source. We use the generated makefile to compile the program.

```
> mkdir build  
> cd build  
> ls  
> cmake ..  
-- The C compiler identification is GNU  
-- The CXX compiler identification is GNU  
-- Check for working C compiler: /usr/bin/gcc  
-- Check for working C compiler: /usr/bin/gcc -- works  
-- Detecting C compiler ABI info  
-- Detecting C compiler ABI info - done  
-- Check for working CXX compiler: /usr/bin/c++  
-- Check for working CXX compiler: /usr/bin/c++ -- works  
-- Detecting CXX compiler ABI info  
-- Detecting CXX compiler ABI info - done  
-- Configuring done  
-- Generating done  
-- Build files have been written to:  
/home/<user>/projects/buildsystems/cmake/build  
> ls  
CMakeCache.txt  CMakeFiles  cmake_install.cmake  Makefile
```

```
> make
Scanning dependencies of target squareroot
[100%] Building C object CMakeFiles/squareroot.dir/squareroot.c.o
Linking C executable squareroot
[100%] Built target squareroot
>ls
CMakeCache.txt  CMakeFiles  cmake_install.cmake  Makefile  squareroot
> ./squareroot 49
The square root of 49 is 7
```

Study the CMake generated Makefile.

### 3. Apache Ant

Apache ant uses java. So we need to install java.

```
> yum install java-1.6.0-openjdk-devel
> yum install ant
```

We need a new directory for the ant exercise.

```
> cd ../../
> mkdir ant
> cd ant
> mkdir -p src/hello

> gedit src/hello/HelloWorld.java
```

The Code:

```
package hello;

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

We now check to see if the code is working correctly

```
> mkdir -p build/classes
> javac -sourcepath src -d build/classes/ src/hello/HelloWorld.java
```

```
> java -cp build/classes hello.HelloWorld
Hello World
>
```

We now write the ant build script.

```
> gedit build.xml
```

The Code:

```
<project>

    <target name="clean">
        <delete dir="build"/>
    </target>

    <target name="compile">
        <mkdir dir="build/classes"/>
        <javac srcdir="src" destdir="build/classes"/>
    </target>

    <target name="jar">
        <mkdir dir="build/jar"/>
        <jar destfile="build/jar/HelloWorld.jar"
            basedir="build/classes">
            <manifest>
                <attribute name="Main-Class" value="hello.HelloWorld"/>
            </manifest>
        </jar>
    </target>

    <target name="run">
        <java jar="build/jar/HelloWorld.jar" fork="true"/>
    </target>

</project>
```

Now the project can be compile and run using ant.

```
> ant clean
Buildfile: build.xml

clean:
[delete] Deleting
directory /home/<user>/projects/buildsystems/ant/build

BUILD SUCCESSFUL
Total time: 0 seconds

> ant compile jar run
Buildfile: build.xml

compile:
[mkdir] Created
dir: /home/<user>/projects/buildsystems/ant/build/classes
[javac] /home/<user>/projects/buildsystems/ant/build.xml:9:
warning: 'includeantruntime' was not set, defaulting to
build.sysclasspath=last; set to false for repeatable builds
[javac] Compiling 1 source file
to /home/<user>/projects/buildsystems/ant/build/classes

jar:
[mkdir] Created
dir: /home/<user>/projects/buildsystems/ant/build/jar
[jar] Building
jar: /home/<user>/projects/buildsystems/ant/build/jar/HelloWorld.jar

run:
[java] Hello World

BUILD SUCCESSFUL
Total time: 1 second
```

#### 4. GNU Autotools

Now we will learn how to use autotools. We will be using a program designed specifically to teach about using autotools -- GNU Hello. Copy the file hello-2.7.tar.gz to the buildsystems project directory and uncompress it.

```
> cd ..
> ls
```

```
ant cmake gnumake hello-2.7.tar.gz
> tar -xzf hello-2.7.tar.gz
> cd hello-2.7
> ./configure
> make
> src/hello
Hello, world!
>
```

The program will now reside in the src directory. To install the program log in as root.

```
> su
> make install
> exit
```

Now the program can be run from anywhere.

```
> hello
Hello, world!
> hello --help
      Usage: hello [OPTION]...
Print a friendly, customizable greeting.

-h, --help           display this help and exit
-v, --version        display version information and exit

-t, --traditional    use traditional greeting format
-n, --next-generation use next-generation greeting format
-g, --greeting=TEXT  use TEXT as the greeting message

Report bugs to: bug-hello@gnu.org
GNU Hello home page: <http://www.gnu.org/software/hello/>
General help using GNU software: <http://www.gnu.org/gethelp/>
```

To uninstall the program:

```
> su
> make uninstall
> exit
```

Instead of using software downloaded as an archive for compilation, source code can also be obtained from software repositories using version control systems like SVN. Build and installation instructions will be found along with the source code which can be used to build the software.

**References:**

1. <http://sources.redhat.com/autobook/> -- Autotools book
2. <http://www.gnu.org/software/make/manual> -- GNU Make manual
3. [http://www.cmake.org/cmake/help/cmake\\_tutorial.html](http://www.cmake.org/cmake/help/cmake_tutorial.html) -- CMake tutorial
4. <http://ant.apache.org/manual/tutorial-HelloWorldWithAnt.html> -- Apache Ant tutorial

## 13. Virtualization Environment

### I. Aim

- To learn the Virtualization setup with xen system and installing ubuntu in virtual machine

### II. Introduction of the concept/tool [1]

- Xen is a free open source software virtual machine monitor for Intel / PowerPC architectures. It is software that runs on a host operating system and allows several guest operating systems to be run on top of the host on the same computer hardware at the same time

Virtual machine monitors (also known as hypervisors) often operate on mainframes and large servers running IBM, HP, and other systems. Internet\_hosting\_service companies increasingly use hypervisors to provide virtual\_private\_servers. Both Amazon EC2 and Rackspace\_Cloud use Xen as the VM hypervisor.

Server virtualization can provide benefits such as:

- consolidation
- increased utilization
- rapid provisioning
- dynamic fault tolerance against software failures (through rapid bootstrapping or rebooting)
- hardware fault tolerance (through migration of a virtual machine to different hardware)
- the ability to securely separate virtual operating systems
- the ability to support legacy software as well as new OS instances on the same computer
- Types of virtualization
  - Paravirtualization, requiring porting of guest systems
  - Hardware-assisted virtualization, allowing for unmodified guests

### III. Description of the exercise

- In this exercise, we would be installing and configuring xen on the base system
- We would be installing and configuring libvirtd
- We would create a sample machine on top of xen+libvirtd
- Installing a ubuntu in a virtual machine

#### IV. Pre-requisite

- Fedora 12 installed with root access (root access for configuration purposes)
- LAN connection with a network setup to access network resources / internet
- selinux disabled - “ setenforce 0 “ can be given at command line to disable it runtime
- FOSS lab Server or Internet Connectivity for downloading/installing packages

#### V. Detailed steps for the exercise-

**Note:** In the description (steps), all greyed out lines are actual commands/configurations done on the server

##### **Step 1 - Preparing the base system for virtualization [2]**

Install the required packages for virtualization using the following command

```
# yum install xen virt-manager kernel-xen qemu libvirt
```

configure xend to start up on boot

```
# chkconfig xend on
```

Reboot the machine to boot into the new kernel

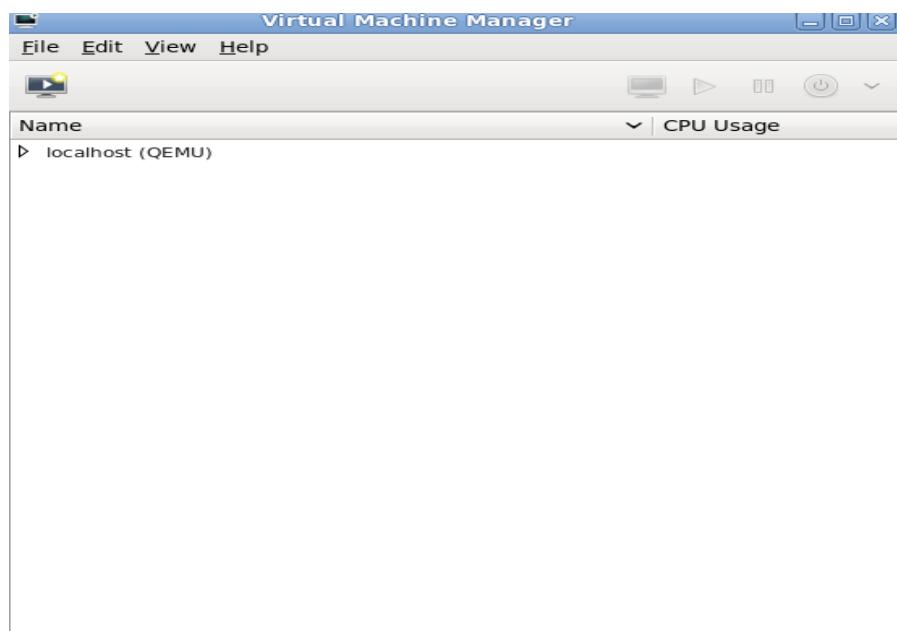
```
# reboot
```

##### **Step 2 - Creating a virtual machine**

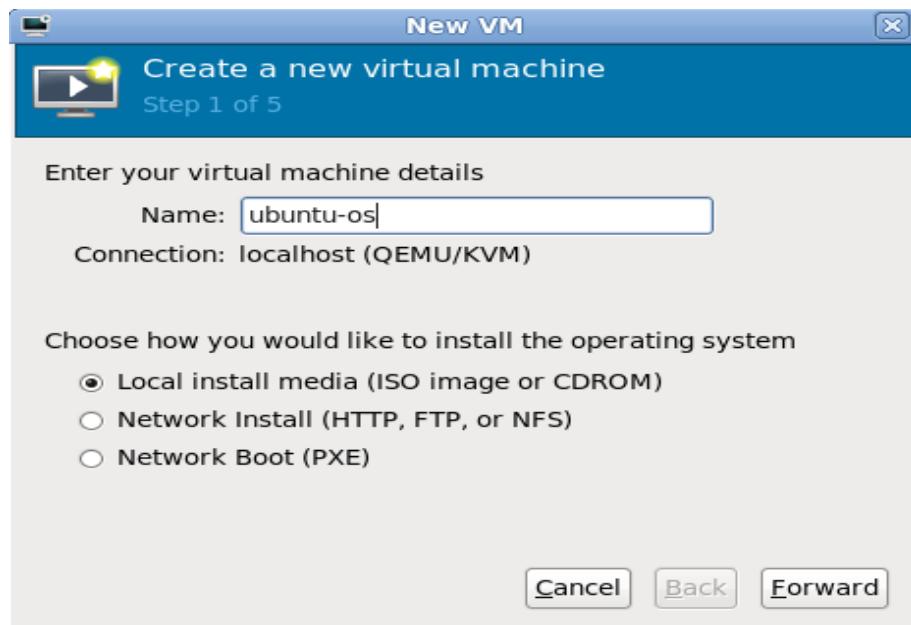
```
# virt-manager &
```

for ease in configuration please use GUI tool called virt-manager:

a) Click on file connect to connect to localhost

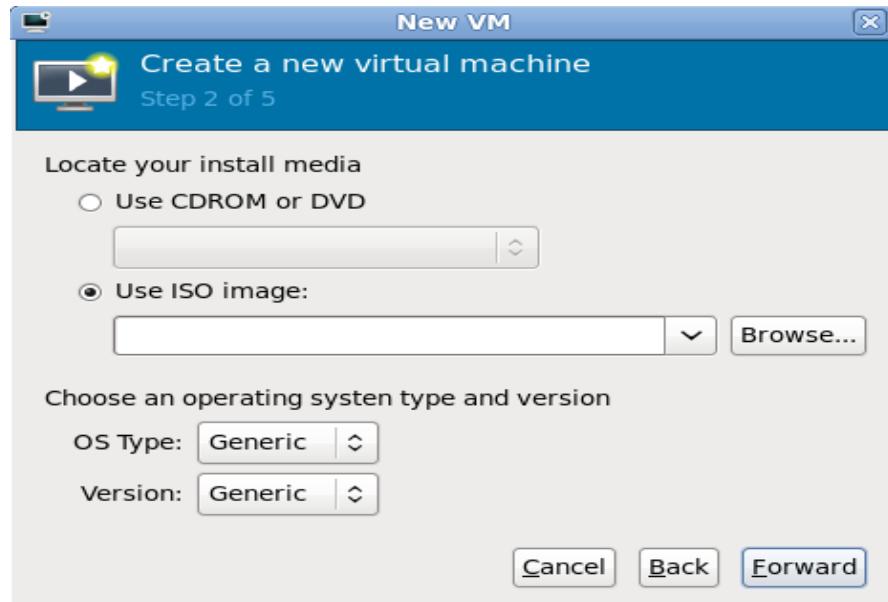


b) In the base menu, please right-click on the localhost (QEMU) to create a new VM. Please name the VM as per your choice

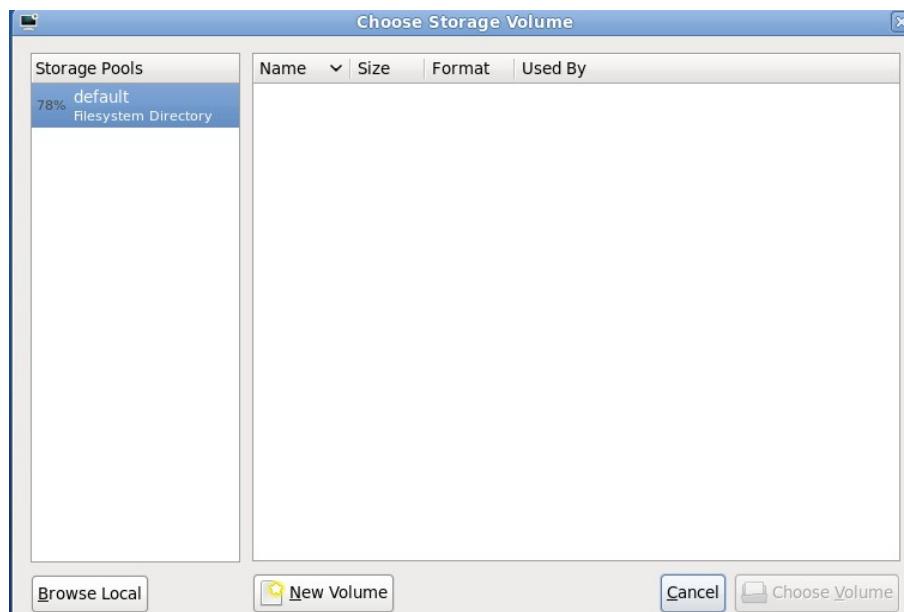


Once the name of the vm is entered, please click on “forward”

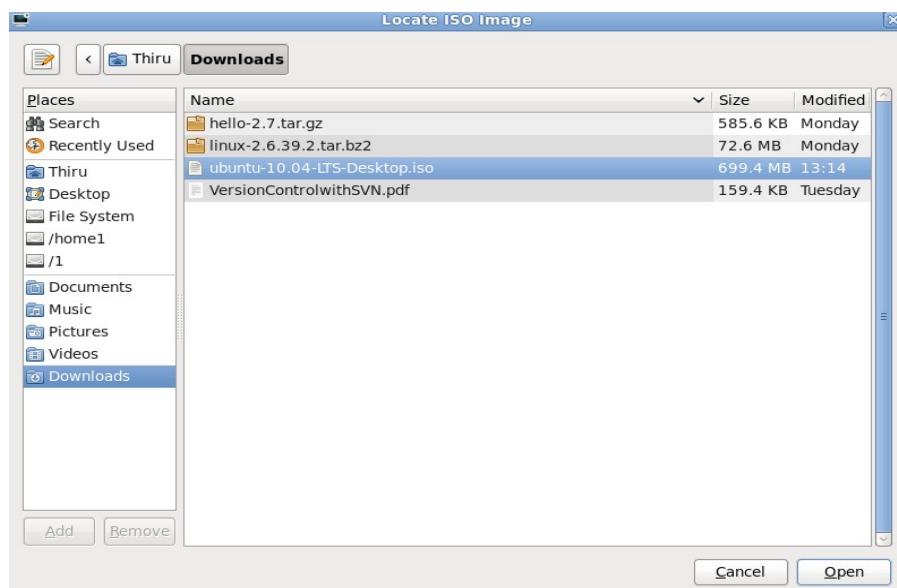
c ) In the next screen, select the Use ISO image option then click the Browse option.



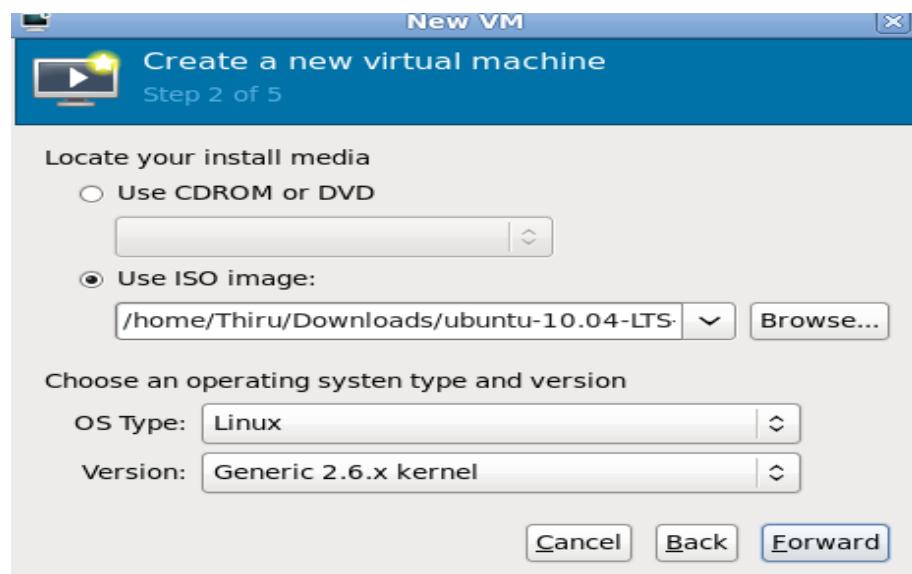
In the next screen click the Browse Local option on the bottom-left section of the screen.



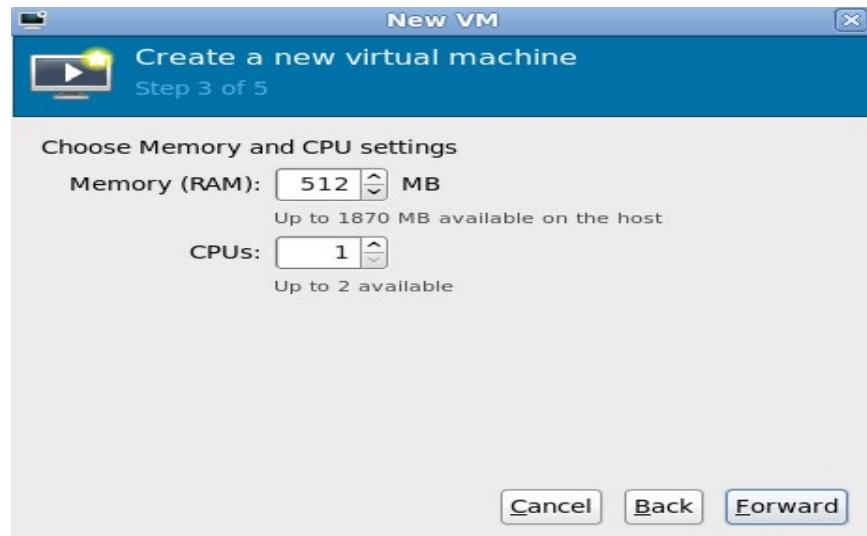
In the next screen please select the ISO image file of your own choice (Ubuntu, centos, fedora etc)



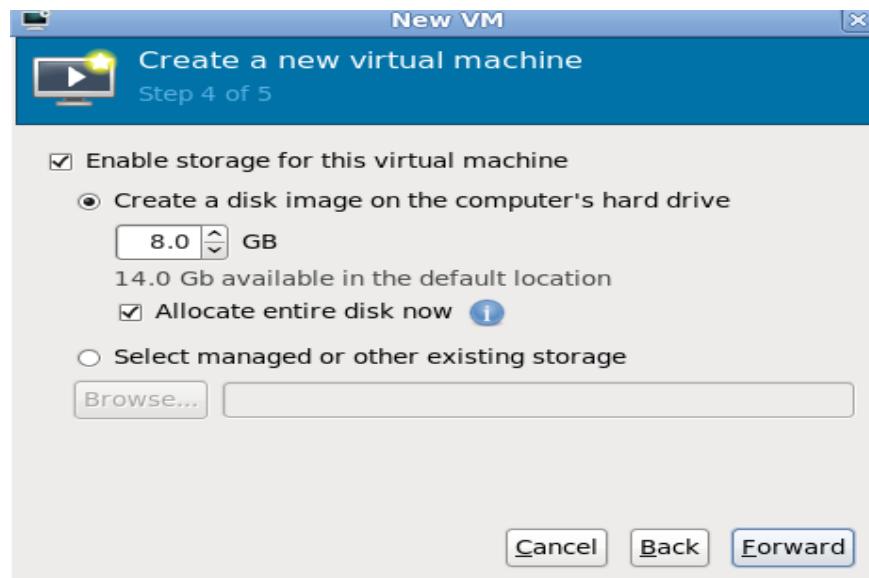
After selecting the ISO file please select OS Type as Linux and Version as Generic 2.6.x kernel.



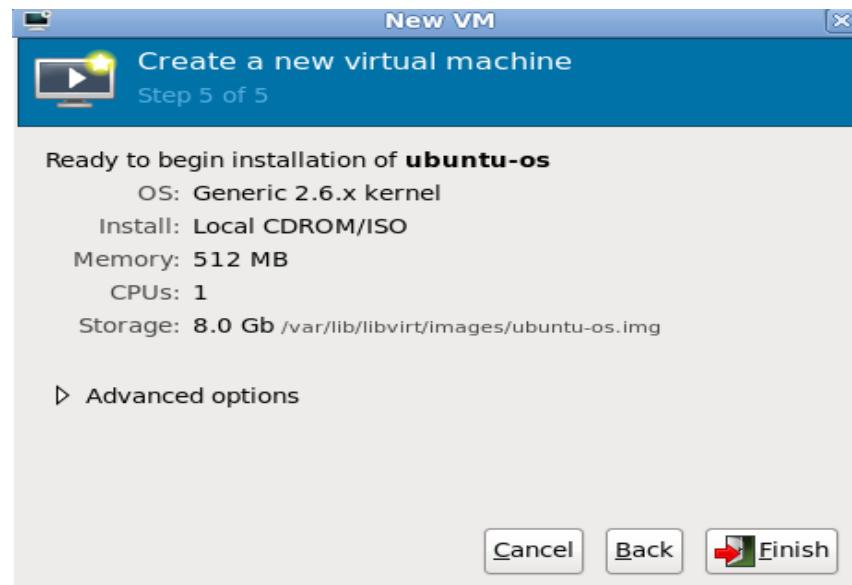
- d) In the next screen, please select the CPU and RAM limits for the VM. By default they are 512M and one CPU.



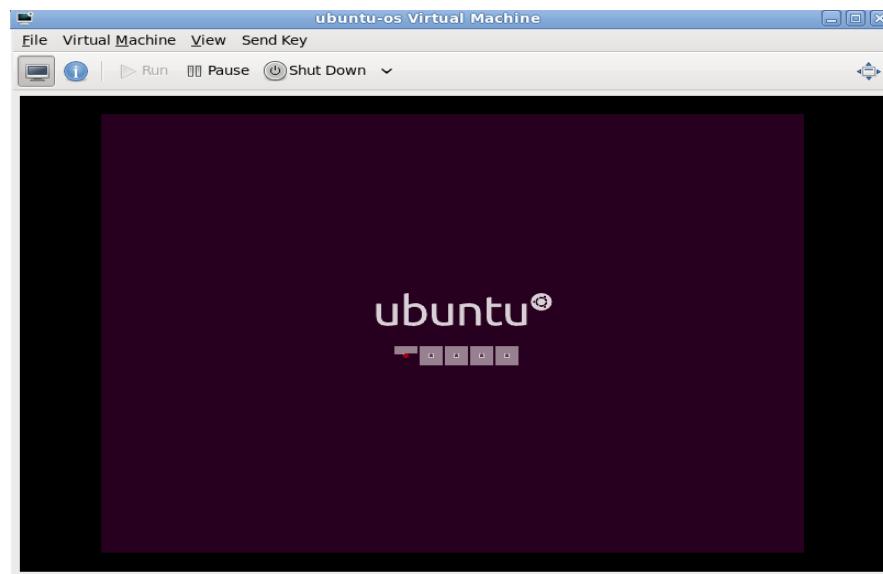
- e) In the next screen, please select the required size for the VM (By default 8GB)



f) In the next screen, by default no networking is selected without using a bridge. If required networking can be enabled for the VM by creating a bridge. Please click on Finish to start creating the machine. Once the machine is created, the same would be displayed on the virt-manager screen and the VM can be started, stopped by selecting the VM and right-clicking to select options.



g) Please wait until Ubuntu installation wizard appear.



- h) In the next screen select Install Ubuntu 10.04 LTS option to install the Ubuntu OS in your virtual manager.



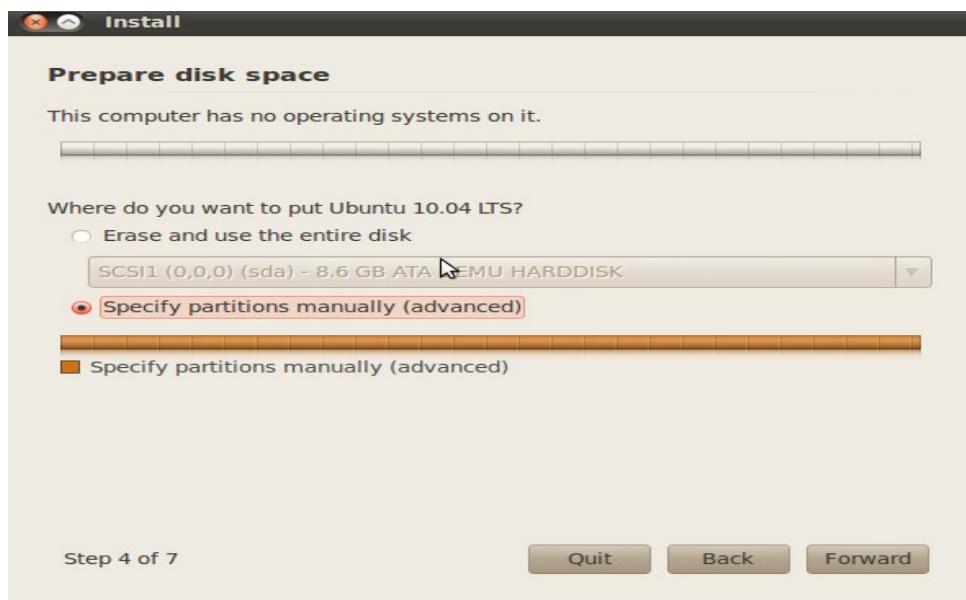
- i) In the next screen select Region and Time zone as India and India Time respectively.



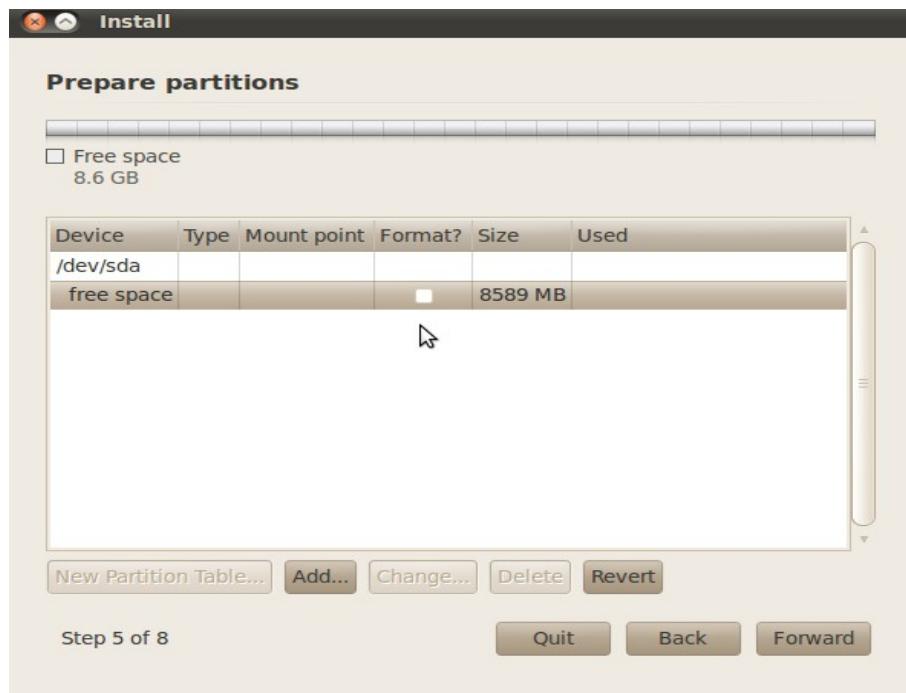
j) Please select the Keyboard layout as default.



k) In the next window select Specify partitions manually(advanced) option to do partition for your Ubuntu OS.



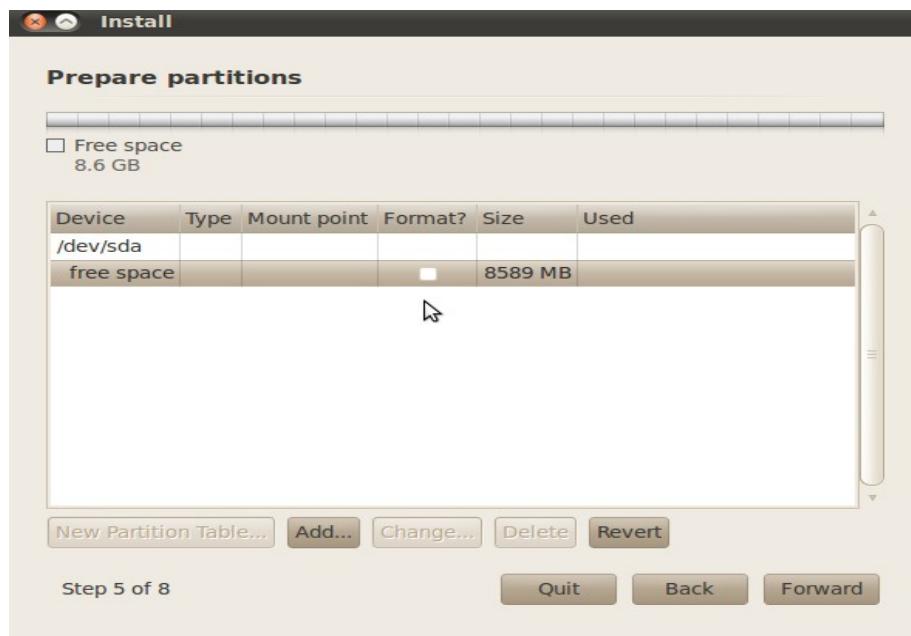
I) In the next screen please select the New Partition Table option.



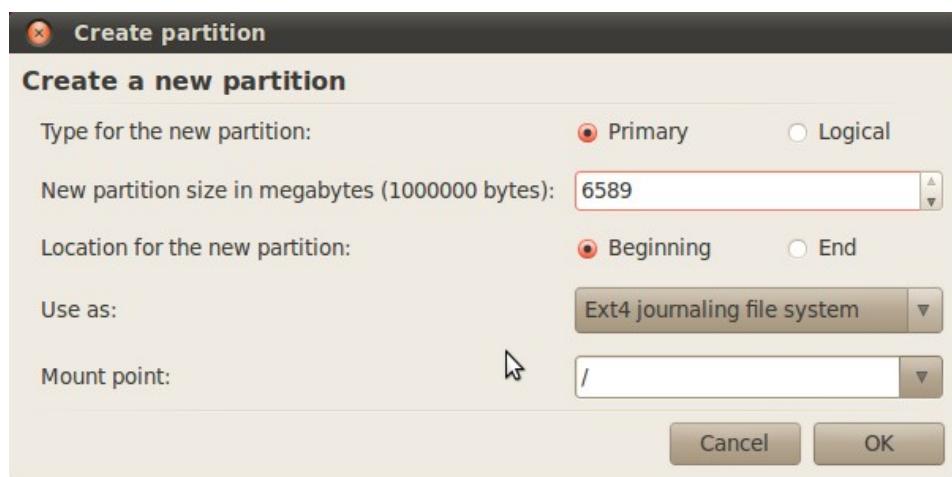
It will show the following confirmation window. Please select Continue option to in that window to continue the installation process.



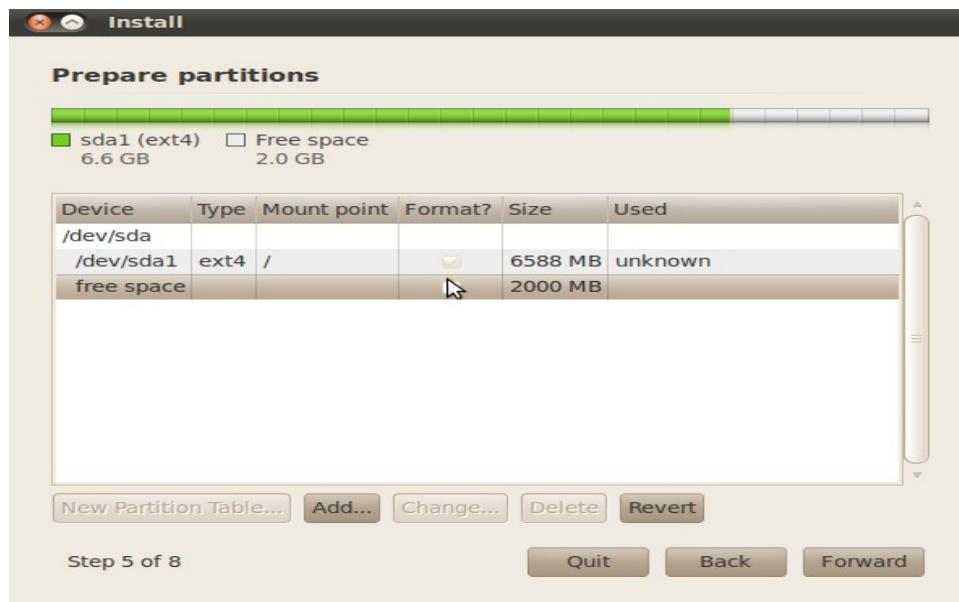
Then in the next screen select the free space and click add option to do partition.



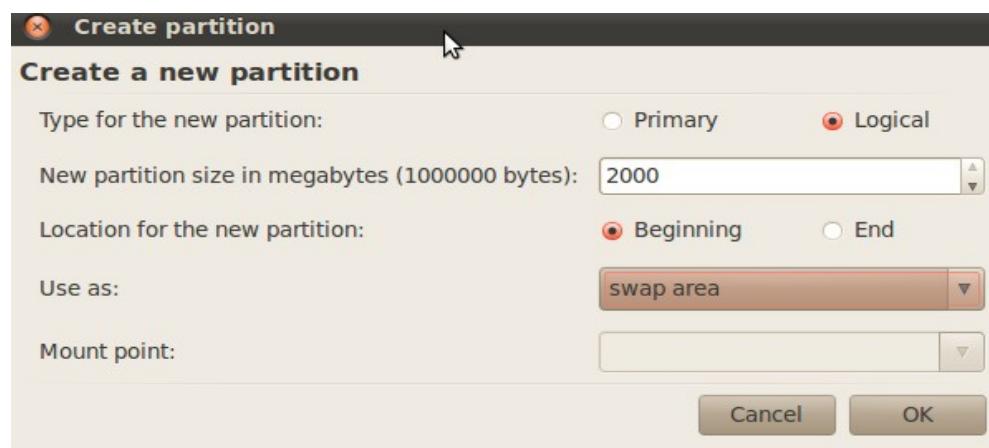
- m) In the next screen select the size for primary partition and mount point as root(/) and Use as Ext4 journaling file system.



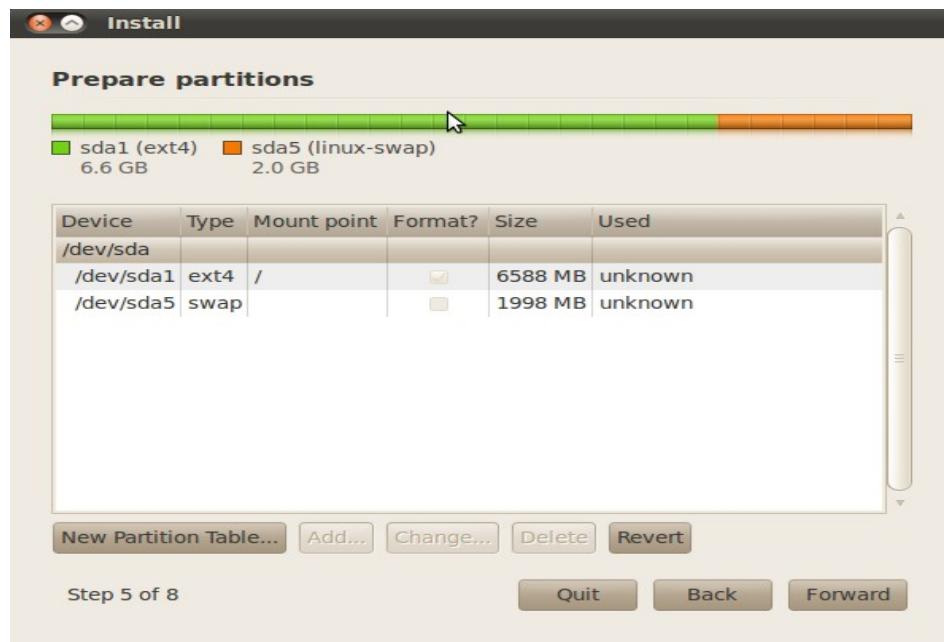
- n) In the next screen again select the free space and click add option to do partition.



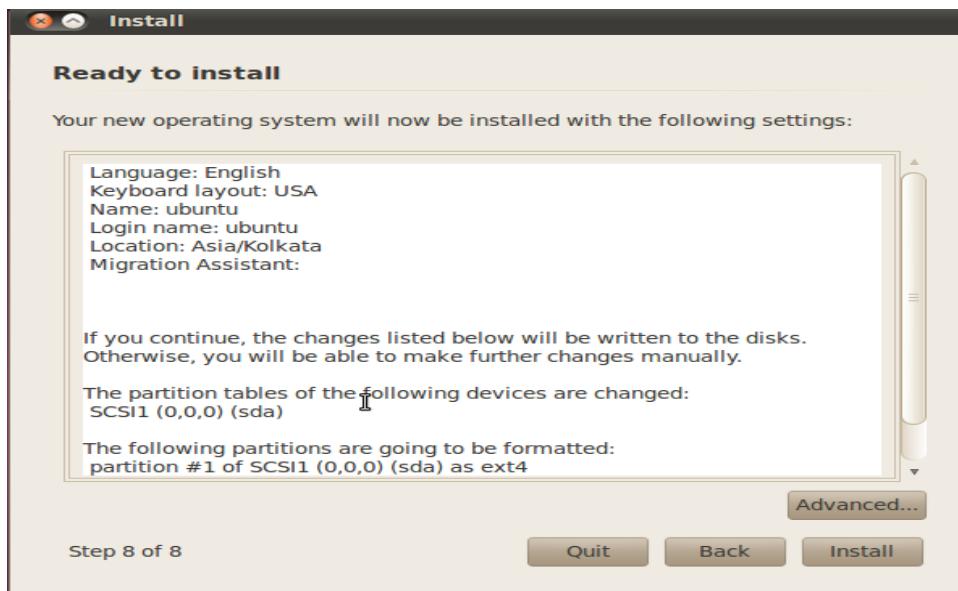
- o) To do partition for Swap select the size for swap area and use as Swap area and select the Logical option.



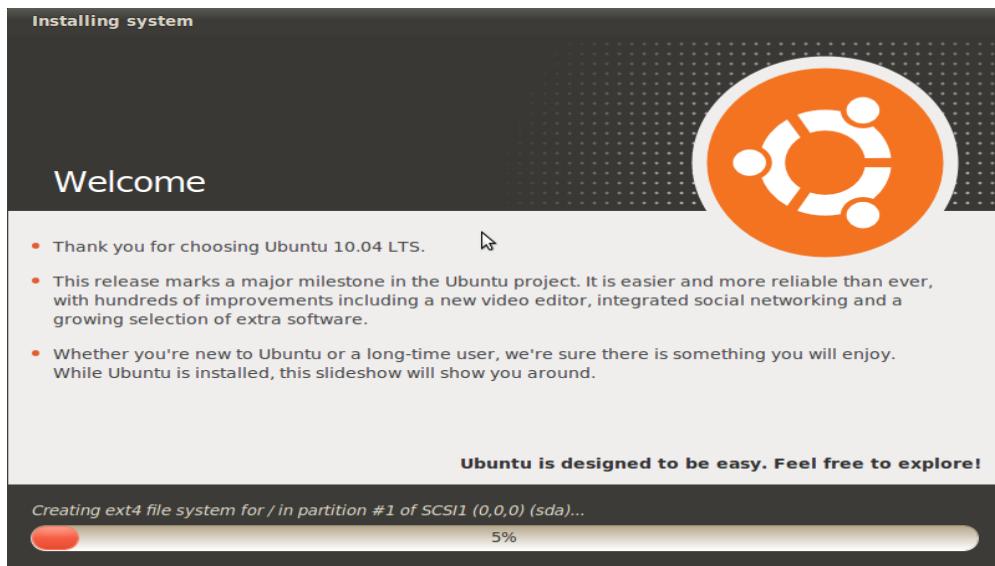
p) In the next window click Forward.



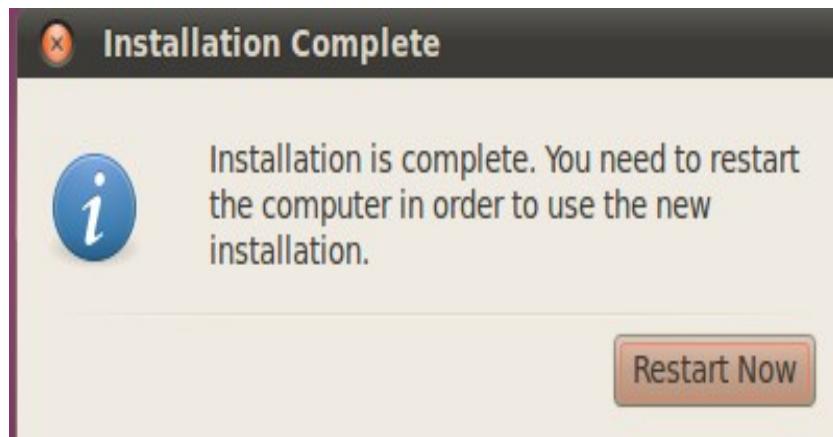
q) In the next window click Install to trigger the installation process.



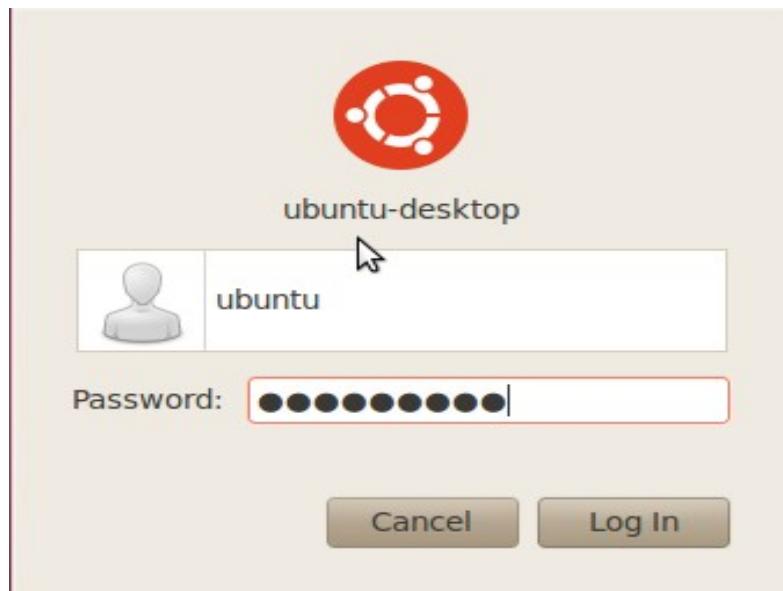
- r) The next screen shows the Ubuntu installation process. Please wait until the installation process is over.



- s) In the next screen click Restart Now option to restart the virtual manager.



- t) The next screen shows logon window. Please give password to logon into your new Virtual operating system.



## VI) References:

- [1] - <http://en.wikipedia.org/wiki/Xen>
- [2] - <http://www.cyberciti.biz/tips/rhel-centos-xen-virtualization-installation-howto.html>

## 14: Kernel configuration, compilation and installation

### Aim:

To learn how to configure, compile and install Linux kernel from source.

### Introduction:

The Linux kernel in the distributions are configured to work correctly in a wide variety of hardware and there is usually no need to use any other kernel.

A user may want to rebuild the kernel for various reasons. The main reason was once to optimize the kernel to the environment (hardware and usage patterns). With modern hardware there is rarely a need to recompile unless there is a particular feature of a new kernel that is required. The performance gains are probably not noticeable unless specific benchmarks are being run.

### Description:

Students will compile a custom kernel using the new kernel source available in the FOSS Lab server. The students should be able to boot the system using the newly compiled kernel.

### Pre-requisites:

The latest kernel source from the FOSS Lab server. It is located in [http://<fosslab-server ip>/content/packages/Linux\\_Kernel/v2.6/linux-2.6.39.2.tar.bz2](http://<fosslab-server ip>/content/packages/Linux_Kernel/v2.6/linux-2.6.39.2.tar.bz2)

### The exercise:

All actions are performed as root.

```
> su  
#
```

We need to ensure that all tools required for compiling the kernel are installed.

```
# yum install kernel-devel
```

This command will ensure that all packages required to compile the current running kernel will be installed. We will be using the same tools to compile the newer custom kernel.

Remove traces of old kernel source if they exist. Be very carefull with the rm command as you can completely trash the system if you are careless.

```
# rm -rf /usr/src/linux/  
# rm -rf /usr/src/linux-2.6/
```

The kernel source is usually kept under /usr/src. Copy the downloaded kernel source to /usr/src.

```
# cp linux-2.6.39.2.tar.bz2 /usr/src  
# cd /usr/src/  
# tar -xjvf linux-2.6.39.2.tar.bz2  
# cd linux-2.6.39.2
```

We now create two symlinks to the kernel tree.

```
# ln -s /usr/src/linux-2.6.39.2 /usr/src/linux  
# ln -s /usr/src/linux-2.6.39.2 /usr/src/linux-2.6
```

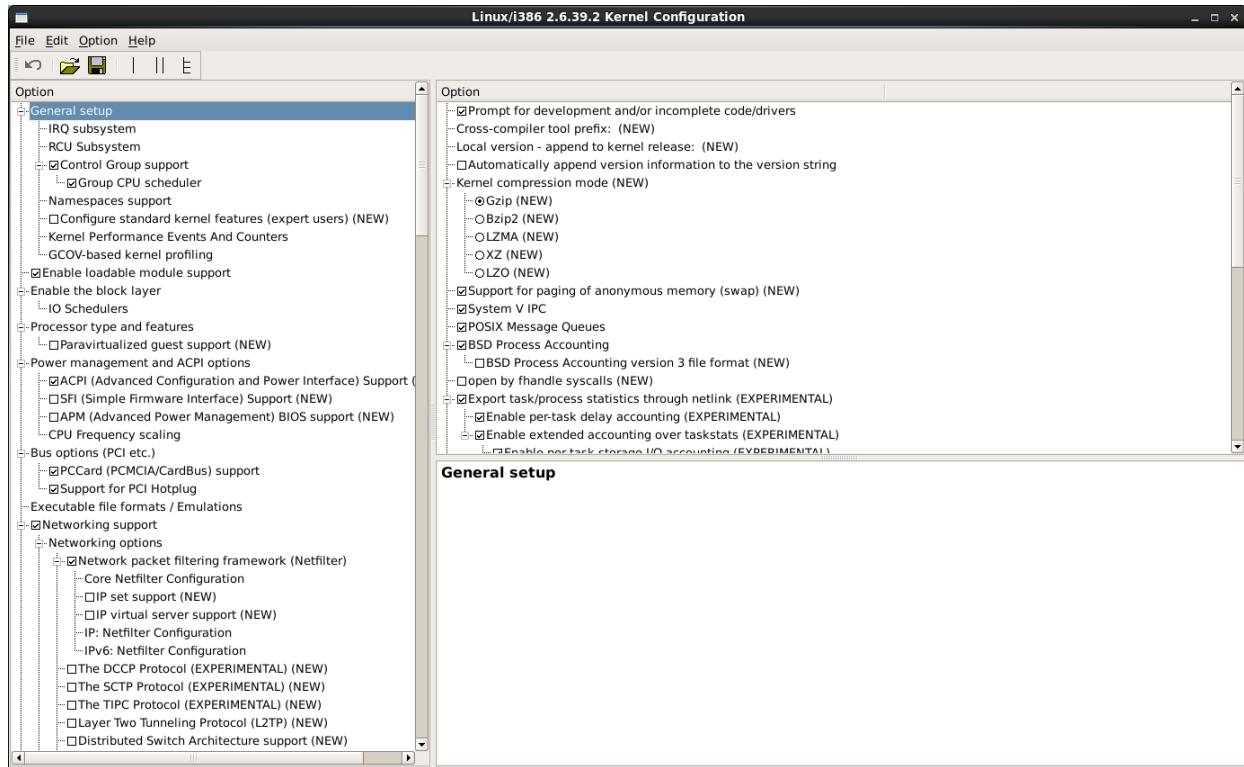
Now we clean out all previous configurations and reset the source directory to a pristine state. The main reason for doing this is that some files do not automatically get rebuilt, which can lead to failed builds, or at worst, a buggy kernel

```
# make mrproper
```

Now we configure the kernel. The build system is intelligent enough to take most of the current configuration from the currently running kernel. There are thousands of options and usually the current options will be suitable to create a working kernel. We can experiment with modifying the kernel options after compiling the kernel successfully with the default configuration.

```
# make xconfig  
      (or)  
# make menuconfig
```

## CS2406-Lab Manual - NRCFOSS/AU-KBC Centre, Anna University Chennai



Save and exit the tool.

Now we are ready to build the kernel.

```
# make clean  
# make all
```

This can take up anywhere from 10 minutes to upto 2 hours depending on the hardware. Once the compilation is completed we can install the kernel and its modules.

```
# make modules_install  
# make install
```

The newly created kernel will be in /boot

Now we need to check that the install process has configured the boot loader to point to the new kernel.

```
# vi /boot/grub/menu.lst
```

The new kernel will have an entry at the top of the kernel list. It can be identified by the kernel version number.

Change the lines containing default, timeout to the values shown and comment out the hiddenmenu entry.

```
default=0  
timeout=5  
#hiddenmenu
```

Now reboot the computer and the computer will boot into the new kernel. If it fails, reboot the machine and select the previously running kernel to boot successfully and redo the exercise carefully.

To check the version of the running kernel, use the uname command.

```
# uname -r  
2.6.39.2
```

Now the process can be repeated with different kernel configuration options.

## References:

1. <http://kernelnewbies.org/FAQ/KernelCompilation> - Kernel Compilation FAQ
2. <http://www.digitalhermit.com/linux/Kernel-Build-HOWTO.html> - Kernel build HOWTO



*User Manual*

## **Why Foss Lab Server**

To learn about GNU/Linux effectively, one of the most useful resources can be the Internet. Apart from the ready availability of all source code, documentation and examples, the Internet also has a wealth of other knowledge in form of mailing lists and newsgroups. Most of the times, a student approaching GNU/Linux for the first time will use a lot of Internet resources to enhance and augment his or her learning.

This could become a problem for students taking up this course. They either have no Internet access at the college (or at home / hostel) or Internet access could be very slow. This means that students would be at a loss to access resources commonly available on the Internet. They would not be able to download packages or source code from the Internet, get access to documentation, HOWTOs, FAQs, online books and guides or access / search mailing list archives for their problems, doubts or questions. And this severely impairs their ability to learn about GNU/Linux and Free Software in the best possible way.

One of the best ways to solve this problem is to provide the students with resident servers in the FOSS lab that would have an archive of all sorts of resources that the students want to access. These servers would contain software binaries and source code, a Debian repository, other GNU/Linux distributions, software documentation and manuals and even mailing list archives. Additionally all this data would be completely indexed and searchable via a simple web-based interface. So students will be able to still learn almost as effectively as they would have with high-speed, convenient and good Internet access.

## **Browsing the Repository Content**

The complete repository can be browsed (and even bookmarked) under /content for example:  
<http://fosslab/content/code/>

## **Hyperlinked and Cross-referenced source code**

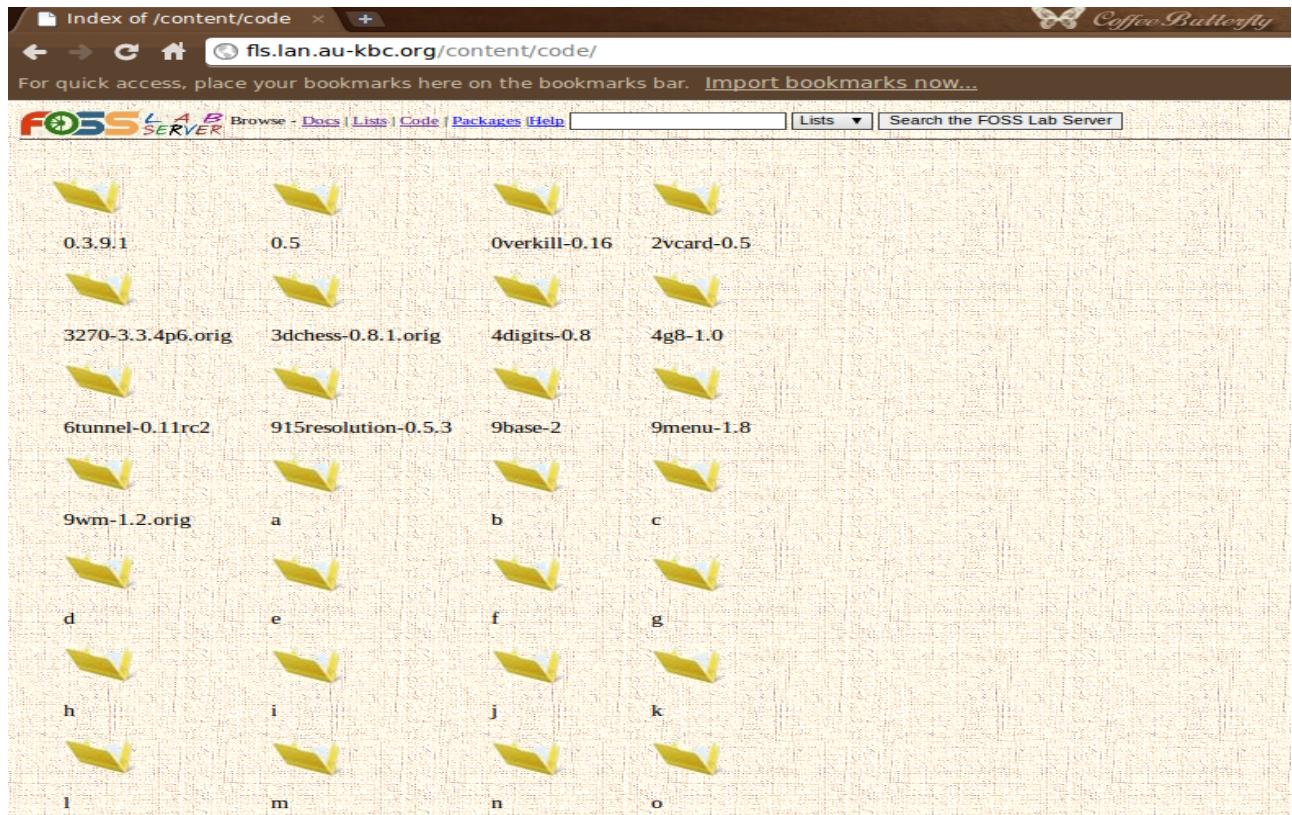
Source code is not always easy to read. Most programmers' editors contain features that allow you to browse source code easily - complete with syntax highlighting and possibilities of being able to jump to function and variable definitions and so on. However, such setups can be difficult to do for students initially. But then, there is no better way to learn to program than by reading and writing code.

To aid students in browsing and navigating the source code of popular software, the FOSS Lab Server contains completely hyperlinked and cross-referenced source code. This enables the following features: A student can browse all source code via a browser from anywhere on the network. The source code is beautifully formatted with syntax highlighting All macros, variables and function names are hyperlinks to their declarations. All local header files are links to their respective definitions Covers source code written in multiple languages - C, C++, Java, PHP, Python, Perl, Apache, MySql, scilab etc.

## Accessing source code

To get a listing of all source code available, you can visit your FOSS Lab Server as follows:

<http://SERVER-IP/content/code/>



## Packages

The following repositories are mirrored completely. This list will get updated more content gets added to the server. Packages are:

- CPAN
- Fedora Packages

- Kernel
- Ubuntu Packages
- Ubuntu 10.04 ISO
- Fedora 12 ISO

The screenshot shows a web browser window with the following content:

- Header:** A butterfly logo and the text "CoffeeButterfly".
- Address Bar:** "fis.lan.au-kbc.org/content/packages/"
- Toolbar:** Back, Forward, Stop, Refresh, Home, Bookmarks, etc.
- Bookmarks Bar:** "Import bookmarks now..."
- Right Sidebar:** "Other Bookmarks" folder icon.
- Main Content Area:**
  - Perl Modules:** A link labeled "READ".
  - LINUX KERNEL:** A link labeled "DOWNLOAD".
  - ISO IMAGES:** A link labeled "Click any of the iso image to download:" followed by two download links: "Fedora-12-i386-DVD" and "ubuntu-10.04-LTS-Desktop".
  - REPOSITORIES:** A link labeled "REPOSITORIES".
  - Fedora Packages:** A link labeled "DOWNLOAD".
  - Ubuntu Packages:** A link labeled "DOWNLOAD".

**Bottom Footer:**

© 2010, The National Resource Centre for Free and Open Source Software (NRCFOSS).  
The FOSS Lab Server is designed and developed by DeepRoot Linux.

## Comprehensive Perl Archive Network (CPAN)

CPAN consists of every single Perl module that has been released as Free Software. By mirroring the complete CPAN archive, we make it possible for students to do any sort of Perl-based development effectively and conveniently.

The screenshot shows the Comprehensive Perl Archive Network (CPAN) homepage. At the top, there's a navigation bar with links for File, Edit, View, History, Bookmarks, Tools, and Help. Below the bar is a toolbar with icons for back, forward, search, and other functions. The main title "Comprehensive Perl Archive Network" is displayed next to the CPAN logo, which features a stylized flame icon.

Key statistics are shown: "2008-09-12 online since 1995-10-26", "4690 MB 239 mirrors", and "6848 authors 14289 modules".

A welcome message says, "Welcome to CPAN! Here you will find All Things Perl." Below this, there are sections for "Browsing" and "Searching".

The "Browsing" section includes links to Perl modules, Perl scripts, Perl binary distributions ("ports"), Perl source code, Perl recent arrivals, recent Perl modules, CPAN sites list, and CPAN sites map.

The "Searching" section includes links to Perl core and CPAN modules documentation, Perl core documentation, CPAN modules, distributions, and authors, and CPAN modules documentation.

The "FAQ etc" section includes links to CPAN Frequently Asked Questions, Perl Mailing Lists, and Perl Bookmarks.

At the bottom, there's a note from "Jarkko Hietaniemi" and a link to the "Disclaimer". Logos for W3C XHTML 1.0 and W3C CSS are present. The footer also includes a link to the CPAN master site hosted by FUNET.

## Fedora Packages

Includes:

- The RPM packages for the Fedora 9(Sulphur) distribution.

This archive of Fedora Packages ensures that the students can easily download and install the .rpm packages from fosslab-server . This can be done in two ways

1. Search particular packages in web browser then download it and that can be installed by double clicking it.
2. Otherwise set these packages as a repository in your machine so that it can automatically search and install the required packages. Follow the given procedure to do that

## Fedora Repository Setup

**Step 1:** Go to /etc/repos.d/

```
#cd /etc/repos.d
```

**Step 2:** Create a file fls.repo

```
#vim fls.repo
```

**Step 3:** Type the given text in that file

```
[fls]
```

```
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://type-your-foss-lab-server-ip-here/content/packages/Fedora_Packages/os/Packages/
#give your Fosslab server IP here
#mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora file:///etc/pki/rpm-gpg/RPM-GPG-KEY
```

**Step 4:** execute “yum update” command to update your repository

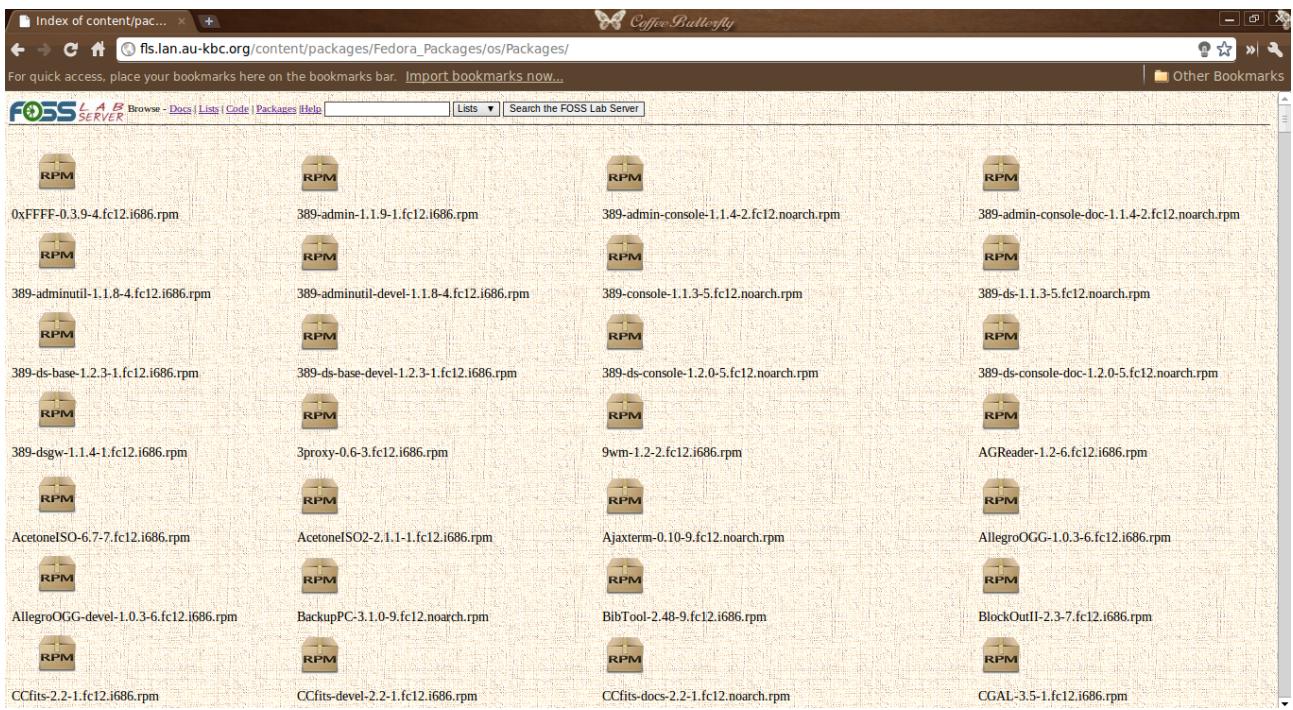
```
#yum update
```

**Step 5:** New packages can be installed by the following command.

```
#yum install [package-name]
```

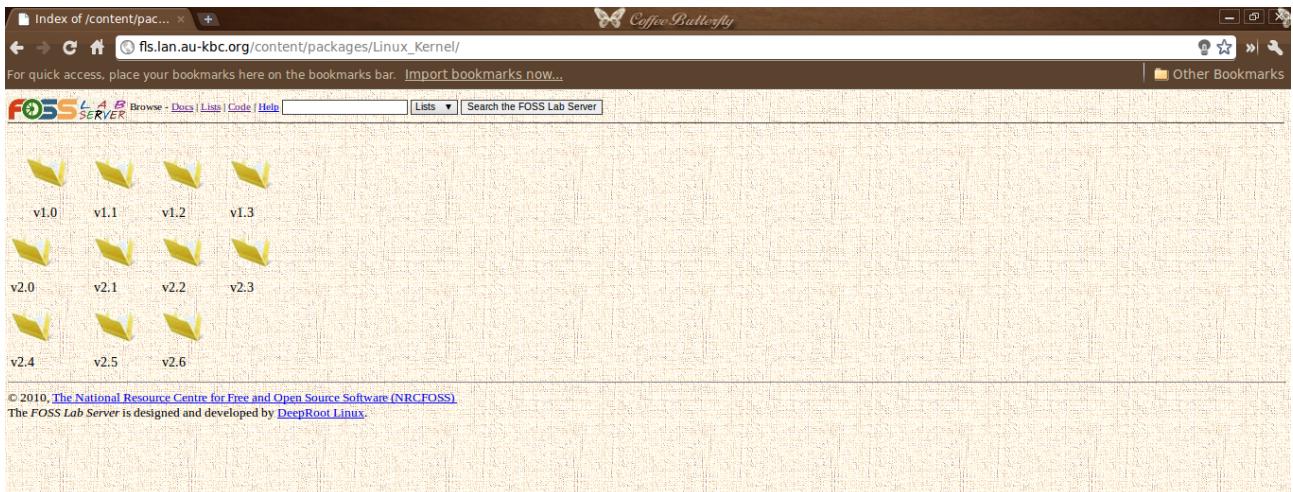
Eg: #yum install PolicyKit

**Fedora Packages through a Web Browser:**



## The Linux Kernel

This mirror consists of the source code for all the Linux kernels ever released. Even though most of us prefer to use the latest Linux kernels for our work, older kernels are provided for historical and academic reasons - so that students can study and appreciate how the Linux kernel has matured over the years.



## The Ubuntu(10.04 – Lucid Lynx) Repository

Includes:

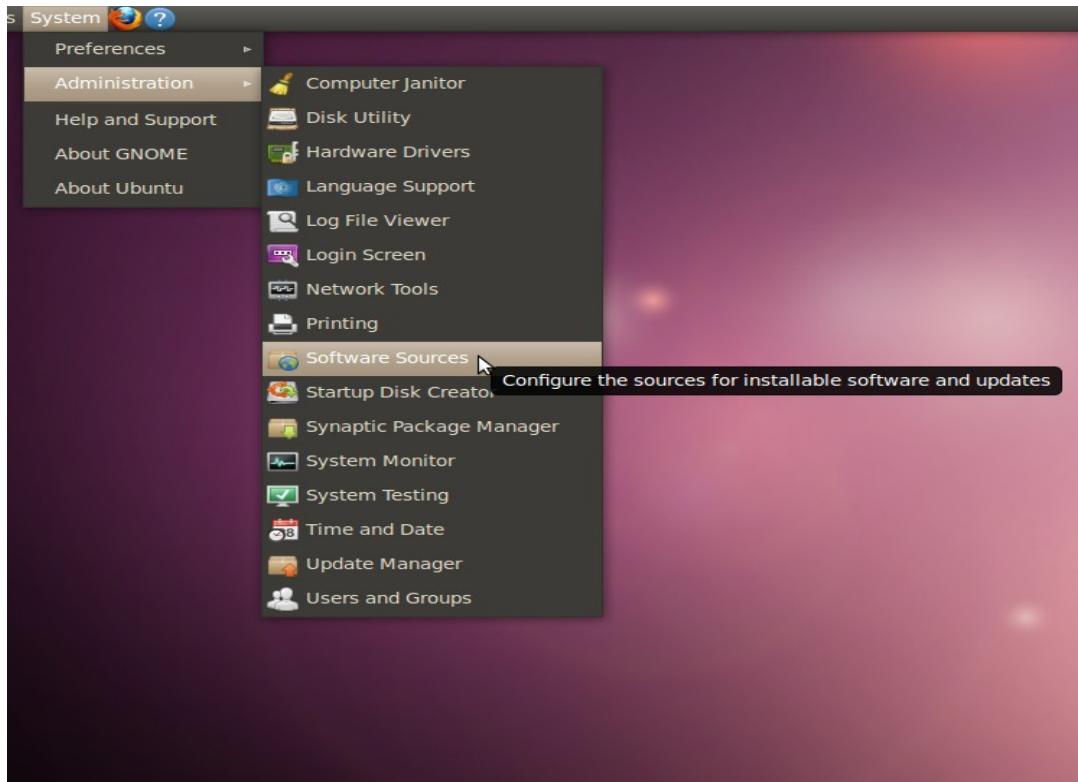
- The complete Ubuntu binary package repository for the 10.04-Lucid Lynx distribution
- Source code for all the Ubuntu packages

This archive of Ubuntu packages and source code ensures that the students have the source code of just about any free software package that Ubuntu ships. There is a document on how to use the Ubuntu Repository as well.

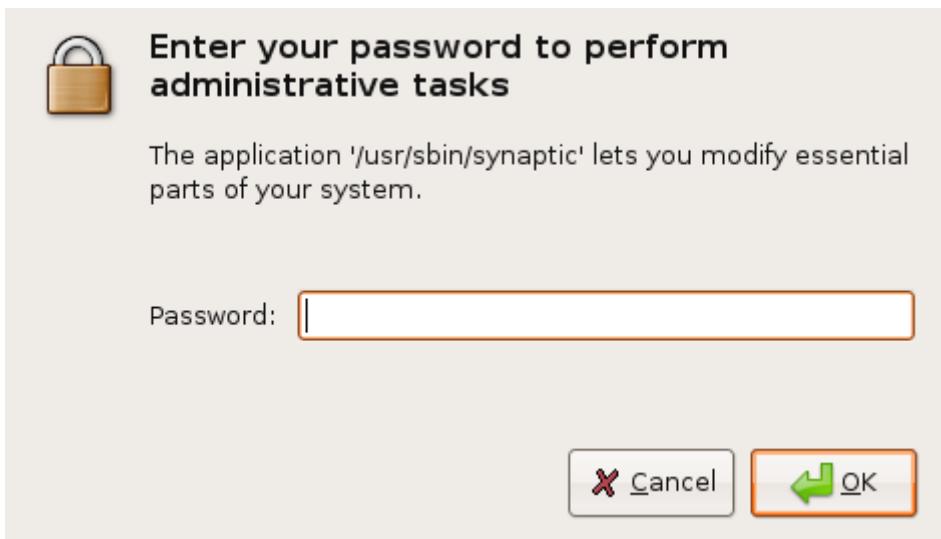
**Setting up the Ubuntu Package Manager to access the FOSS Lab Server in a client machine.**

Follow the given steps here

**Step 1: Go to System -> Administration -> Software Sources**



**Step 2:** After clicking the Software Sources it will ask for the root authentication just type the root password and proceed.

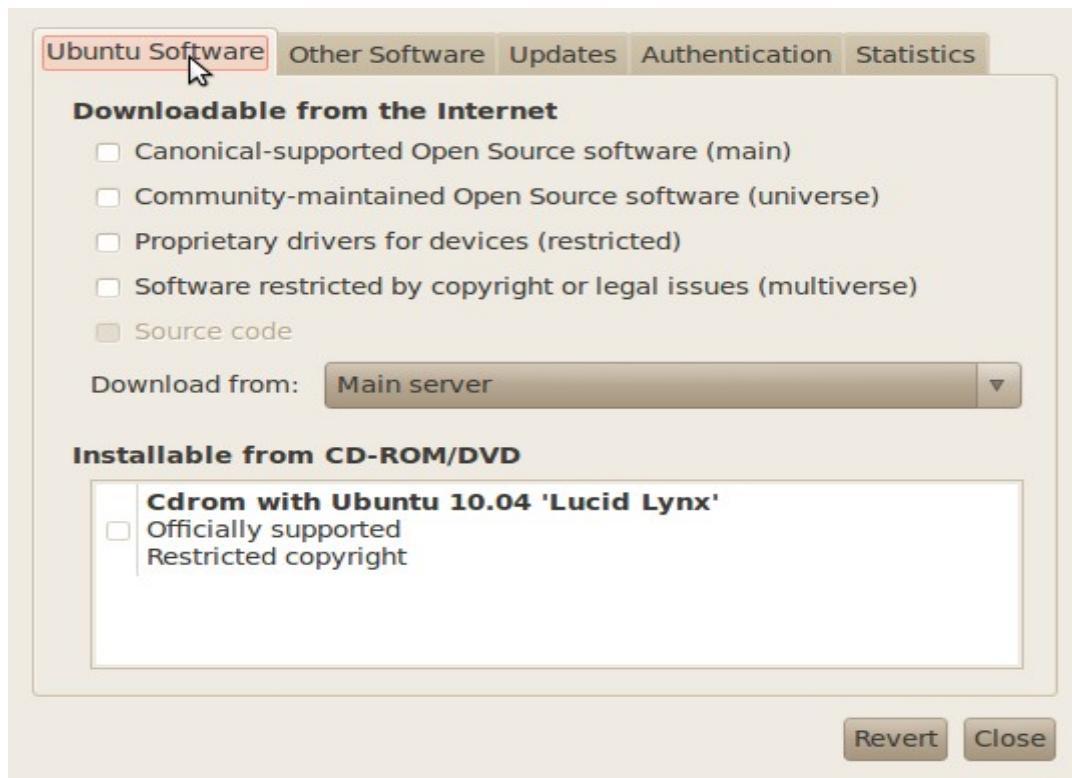


**Step 3:** Now the software sources window will be shown to you.

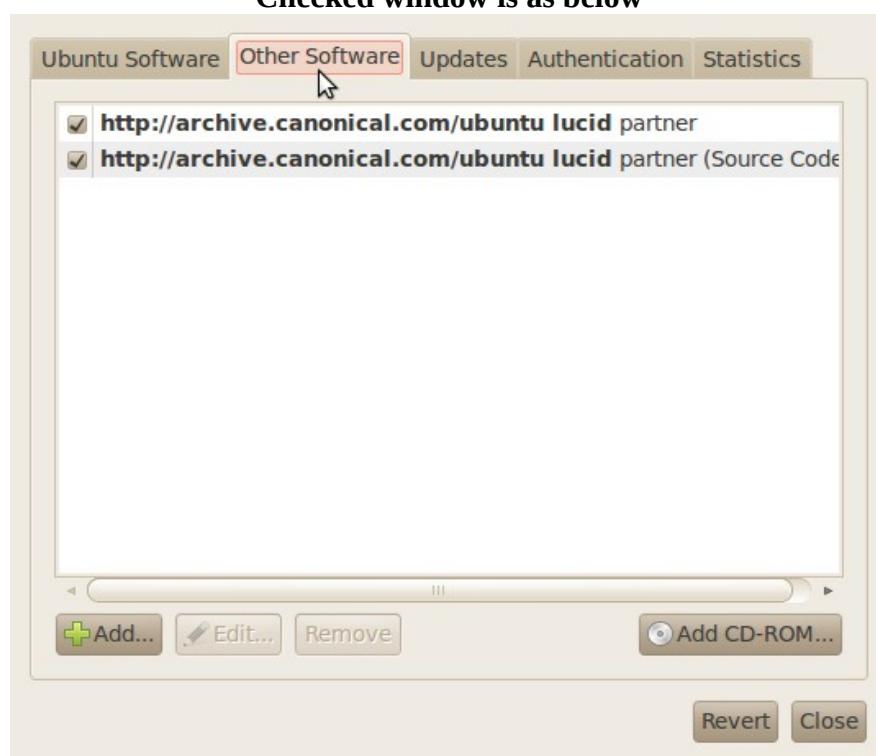


**Step 4:** Uncheck all the fields in **Ubuntu Software** and **Other Software** and **Updates** and **Statistics**.

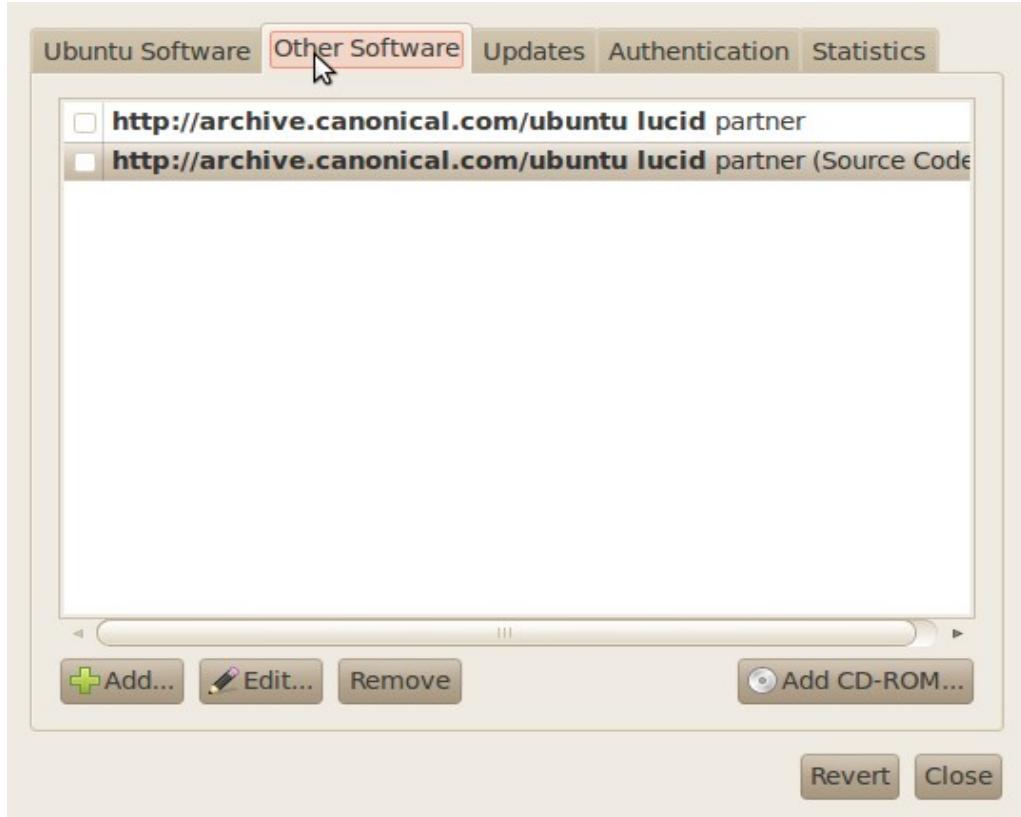
**Step 4.1:** Uncheck all the fields in **Ubuntu Software** tab as below



**Step 4.2: Uncheck all the fields in Other Software tab as below if it was checked**

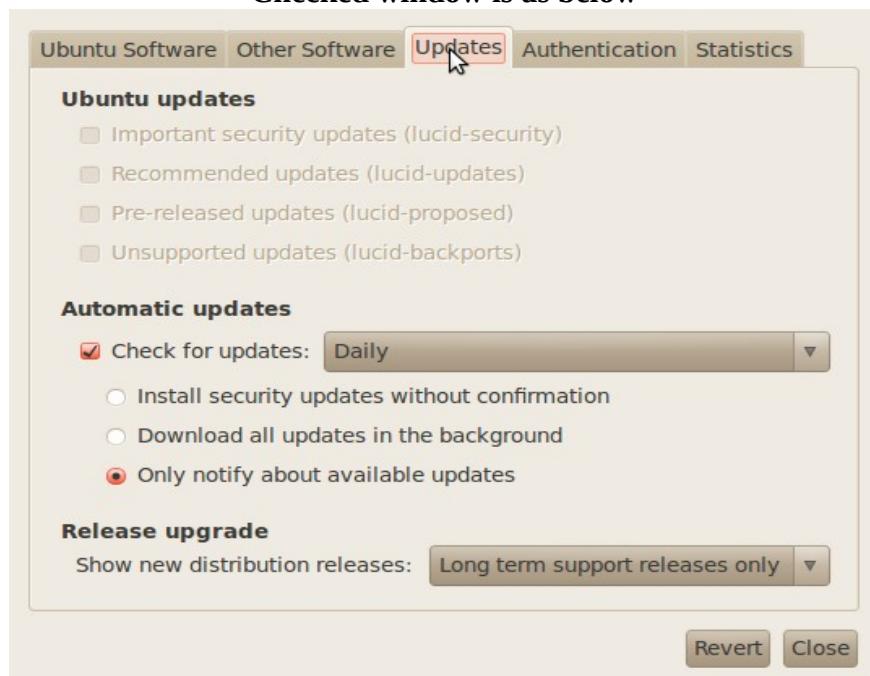


**Un-Checked window is as below**

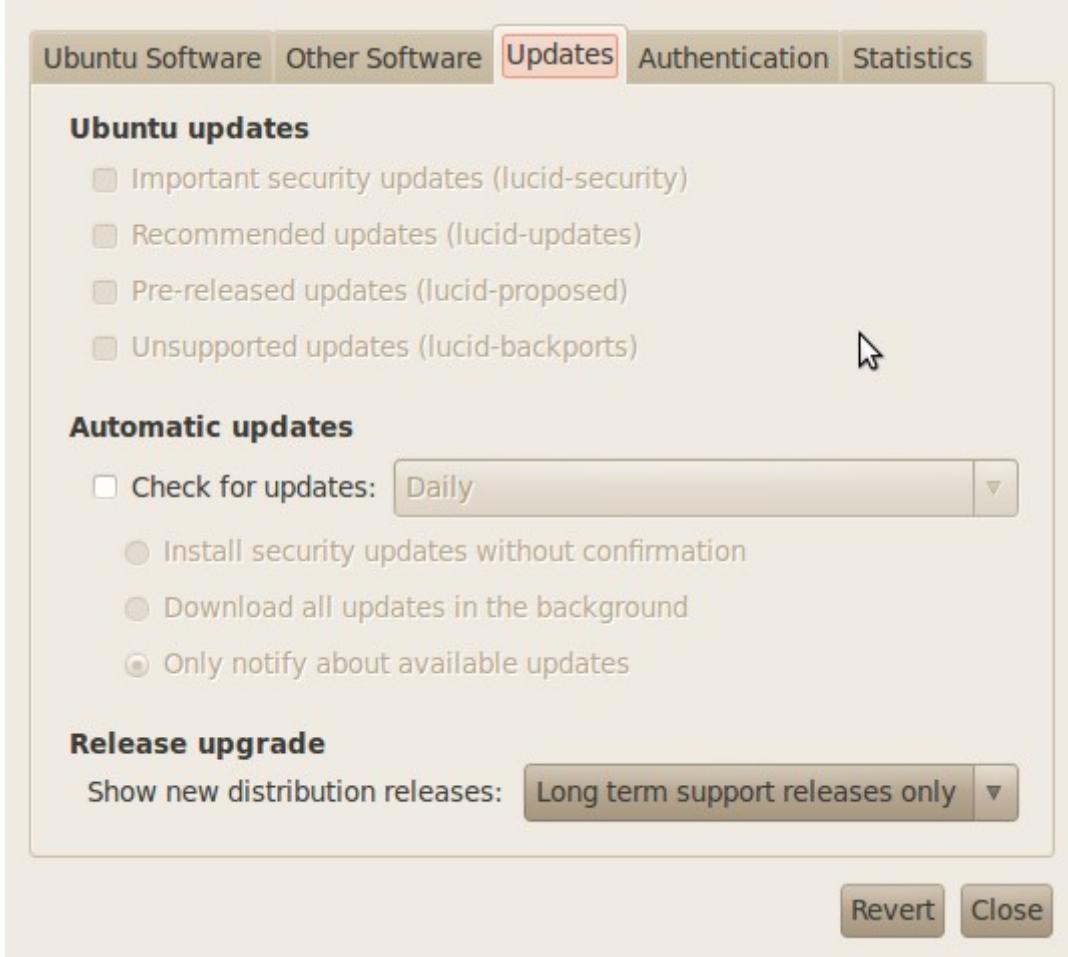


**Step 4.3: Uncheck all the fields in Updates tab as below if it was checked.**

**Checked window is as below**



**Checked window is as below**

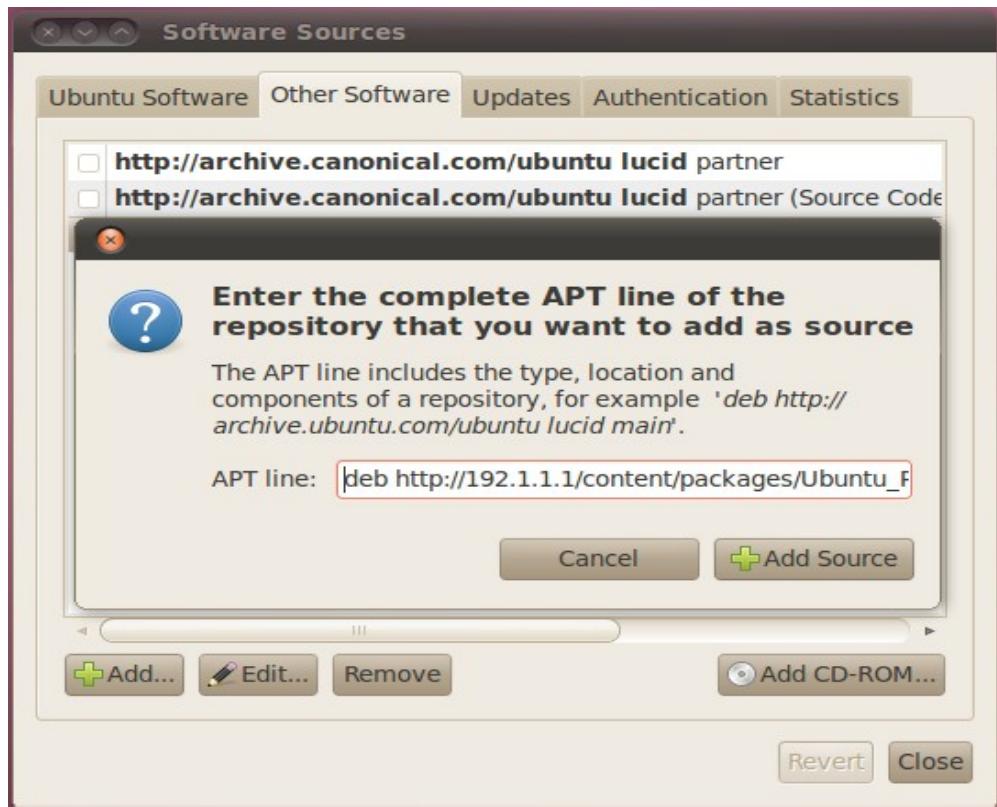


### **Step 5: Adding FOSS Lab Server repository**

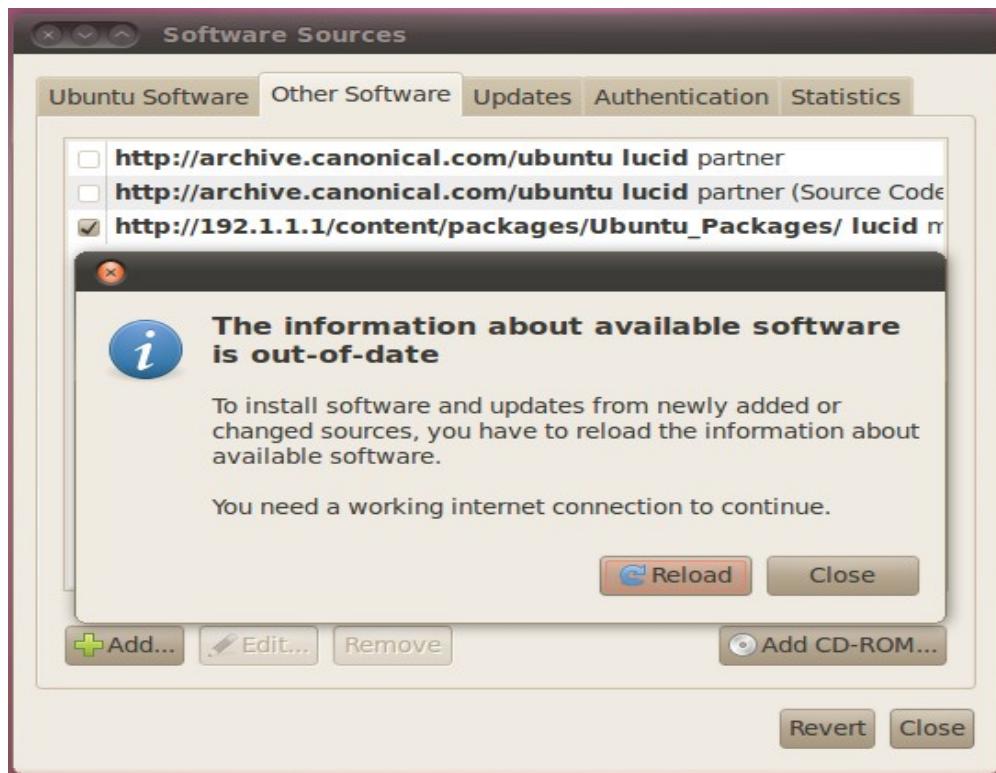
Go to “**other software**” tab in Software Sources and Click **Add** button then APT window will appear as shown below. Then enter the following line in the prompt as shown below.

**deb http://Foss-lab-server-ip-here/content/packages/Ubuntu\_Packages/ hardy main universe  
restricted multiverse**

**note: Foss-lab-server-ip-here -> Your FOSS-LAB Server IP (ex:192.1.1.1)**



then Click on  
the Add Source and finally click on the close button. A new window will pop up just click the reload button to update the repository.



The following window will appear and then the ubuntu repository will be updated from FOSS-LAB Server .

This process will take up some more time so it requires your patience.



If you want to see the details of the installation package you can see in the following screen

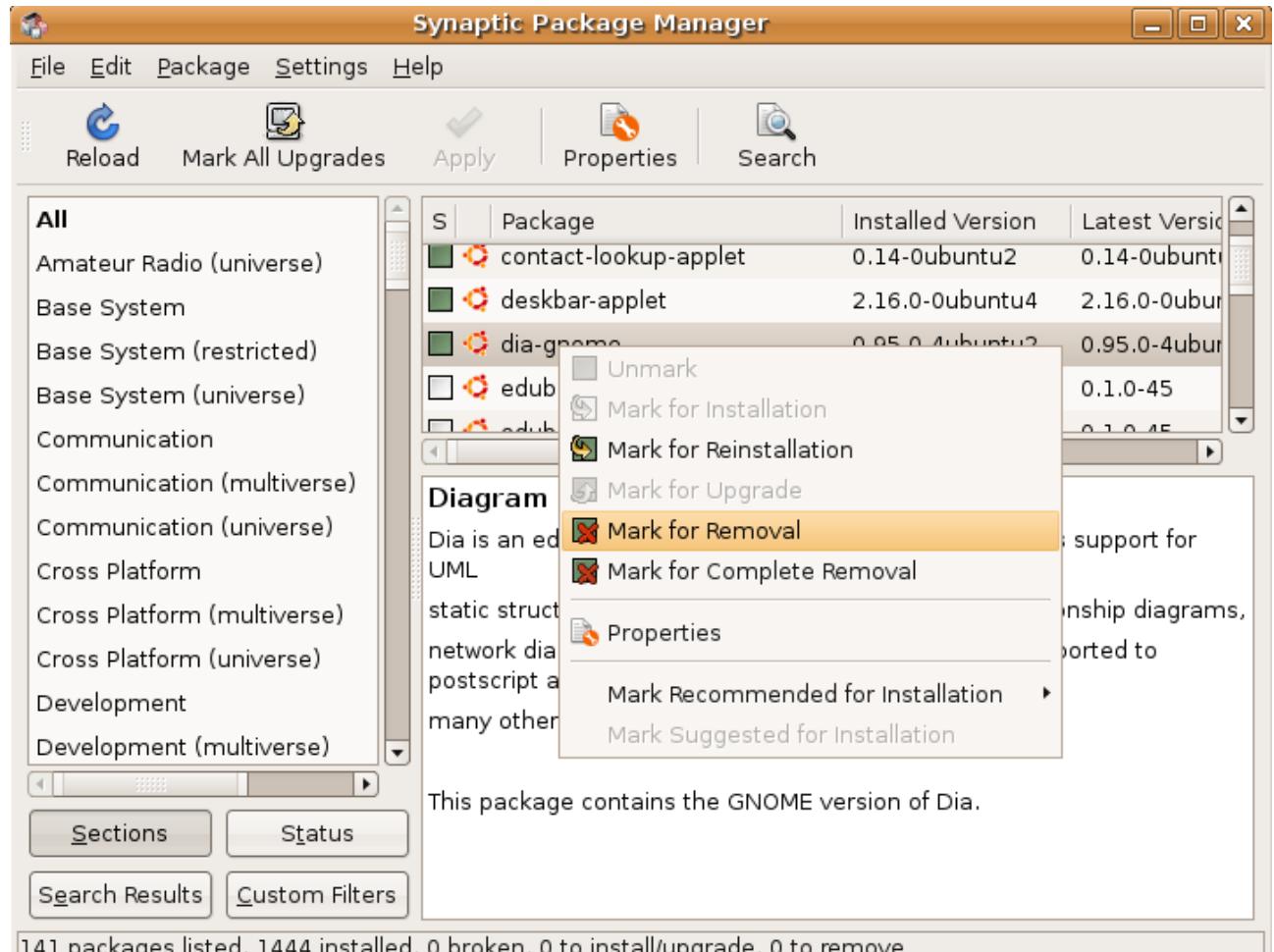


Changes are applied successfully windows and click on close

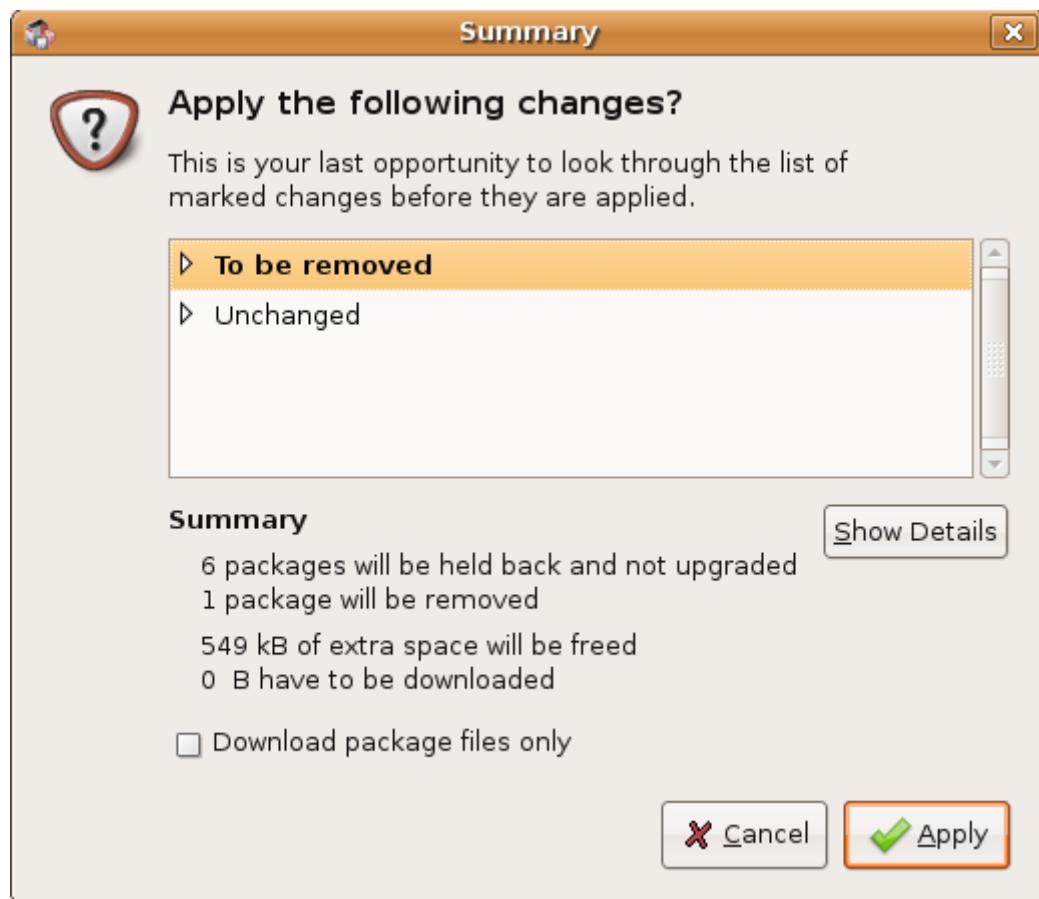


## Remove a Package Using Synaptic Package Manager

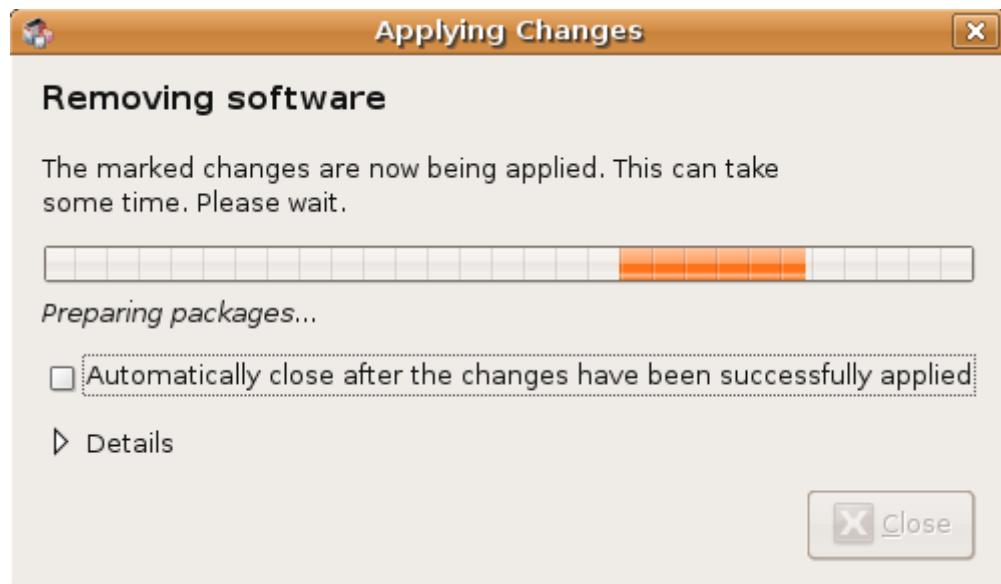
Now if you want to remove installed package you need to select that package and right click on that you need to select Mark for removal option



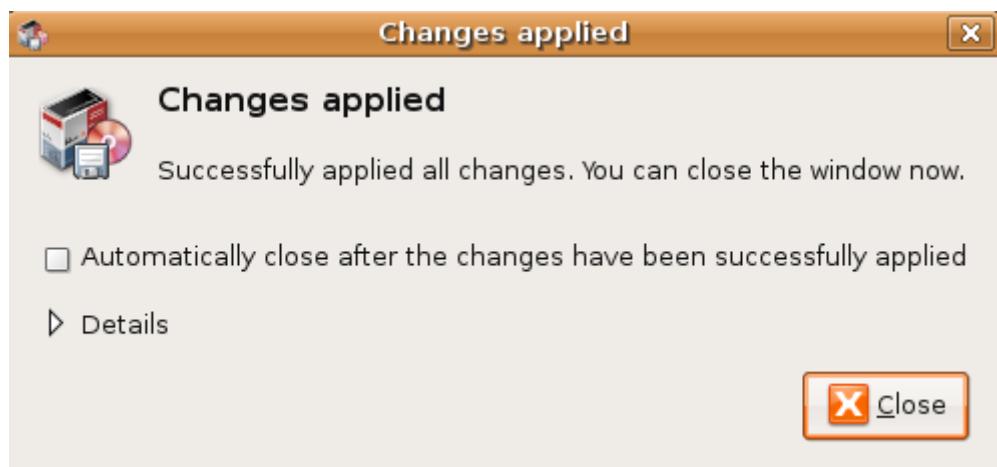
Here you can see the packages are going to be removed and other details here you need to click on apply or press Ctrl + P



Removing package is in progress



Changes are applied successfully windows and click on close



### Search a Package Using Synaptic Package Manager

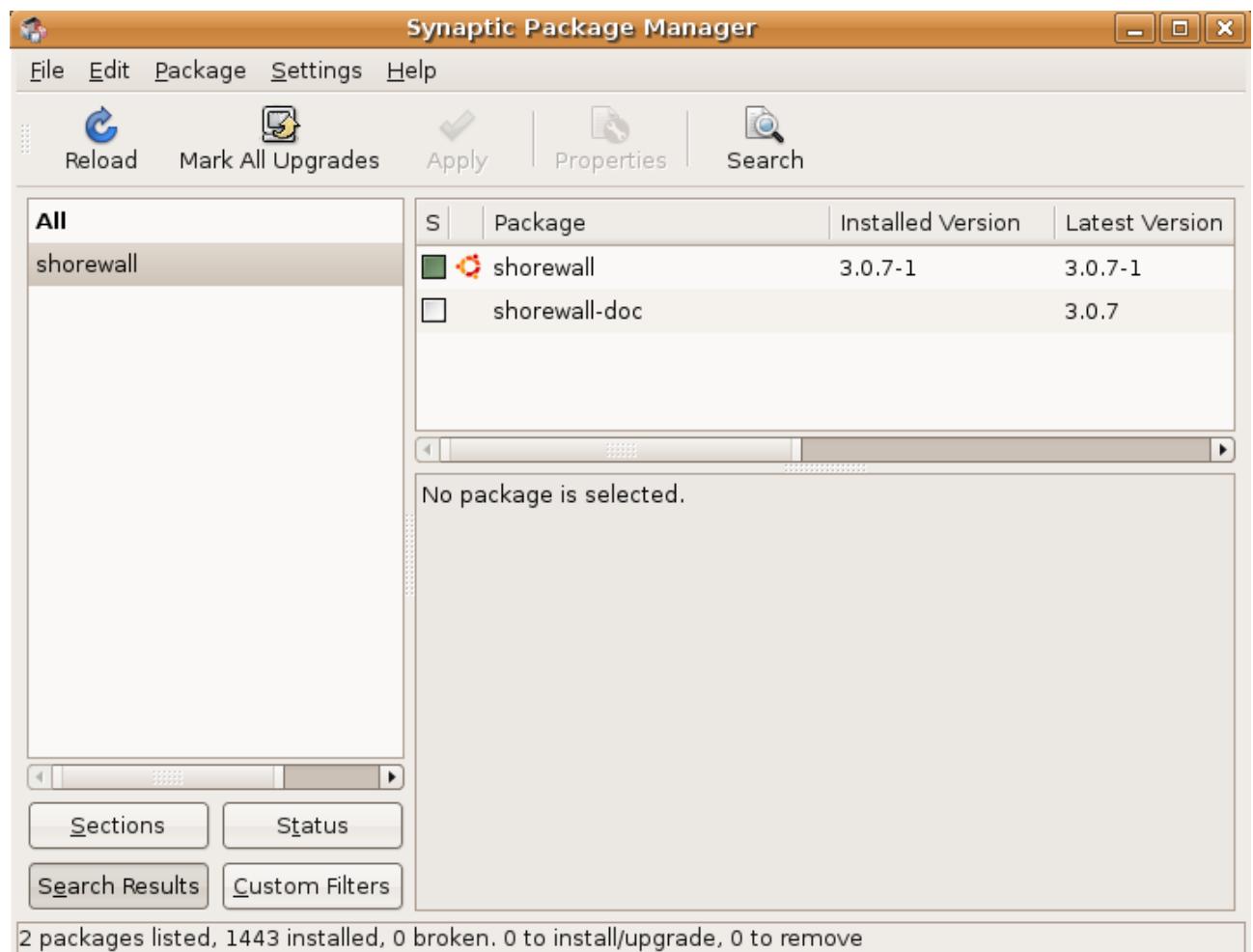
If you want to search a package you need to click on search option in synaptic tool bar



You should see the following screen asking for package name and select the the option like description or name or both and click on search in this example i am searching for shorewall



You should see the results in the following screen if you want to install these packages you can follow the above procedure



## Ubuntu Packages through Web Browser



## **Documentation**

The following documentation are archived and made browseable and searchable:

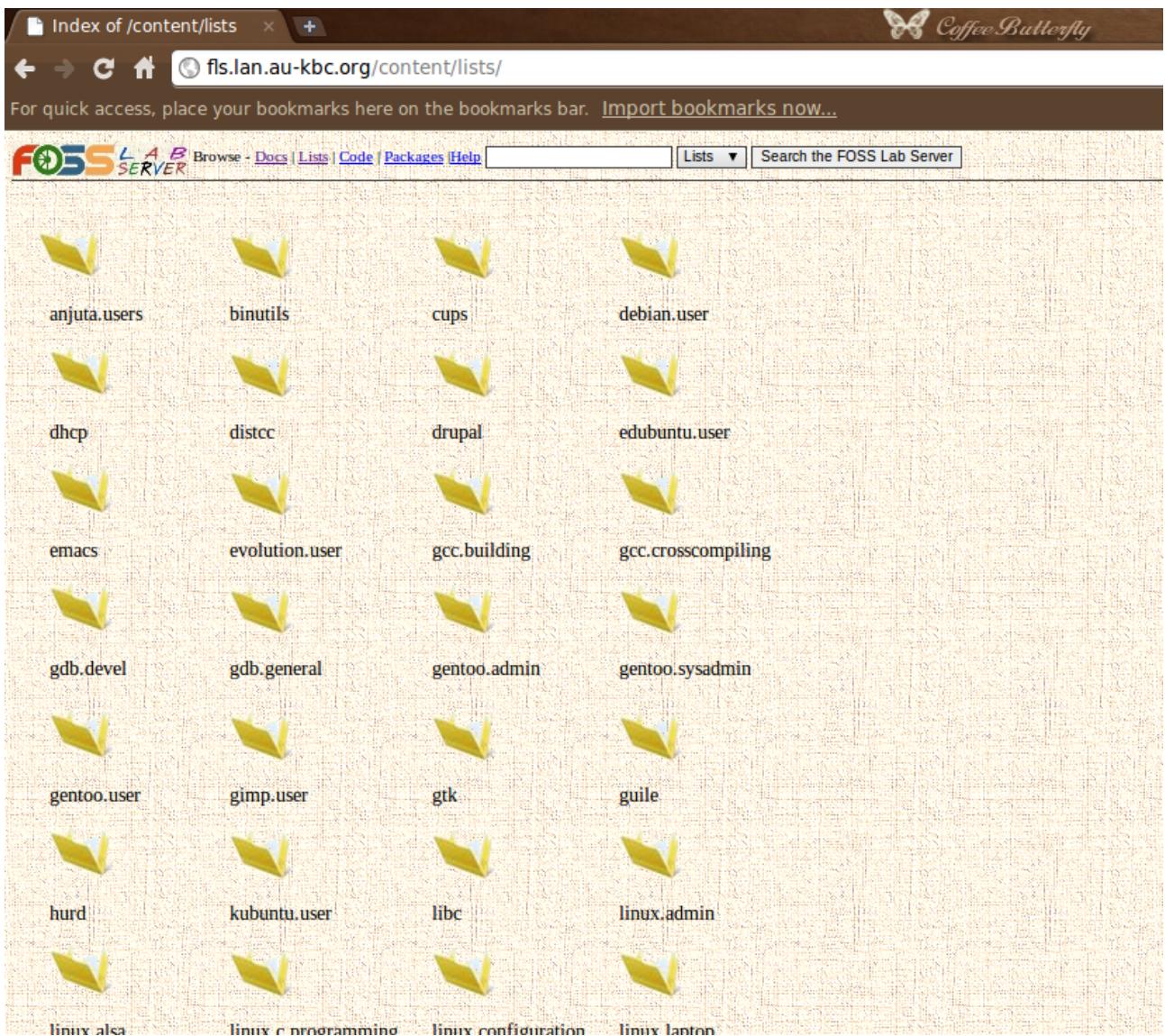
- MySql
- Gnome
- Django
- CakePHP
- PHP
- Perl
- Python
- JAVA2
- XML
- Subversion
- HTML
- Gnome
- CSS
- PostgreSQL
- Debian Document
- Gentoo-wiki
- Sugar CRM
- Joomla
- TLDP
- CUPS
- Trac
- Redmine

The screenshot shows a web browser window with the following details:

- Title Bar:** Index of /content/docs
- Address Bar:** fls.lan.au-kbc.org/content/docs/
- Header:** Coffee Butterfly logo
- Toolbar:** Back, Forward, Stop, Home, Refresh buttons.
- Bookmark Bar:** Placeholder for "For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now..."
- Navigation:** FOSS LAB SERVER logo, Browse - Docs | Lists | Code | Packages | Help, Lists dropdown, Search the FOSS Lab Server.
- Content:** A grid of 24 project links, each with a yellow book icon:
  - Row 1: Apache2-Http, CSS, CUPS, Cakephp, Codeigniter
  - Row 2: Debian\_Document, Django, Drupal, Gentoo-Wiki, Gnome, HTML
  - Row 3: Java, Java2, Joomla, Mediawiki, Moodle, Mysql
  - Row 4: Netfilter, Openbravo, Openerp, Perl, Php, Plone
  - Row 5: Postfix, PostgreSQL, Python, Qmail, Redmine, Rubyonrails
  - Row 6: Spree, Squid, Subversion, Sugarcrm, TLDp, Trac

## Mailing Lists

Some significant mailing lists for the last 2 years have been archived and made searchable. Access the mailing list archives at <http://fosslab/content/lists>. The archives are sorted and displayed month-wise. The default view for the lists is threaded by topic. One could also sort it by date. The mail archives are converted from mbox format to readable html format.



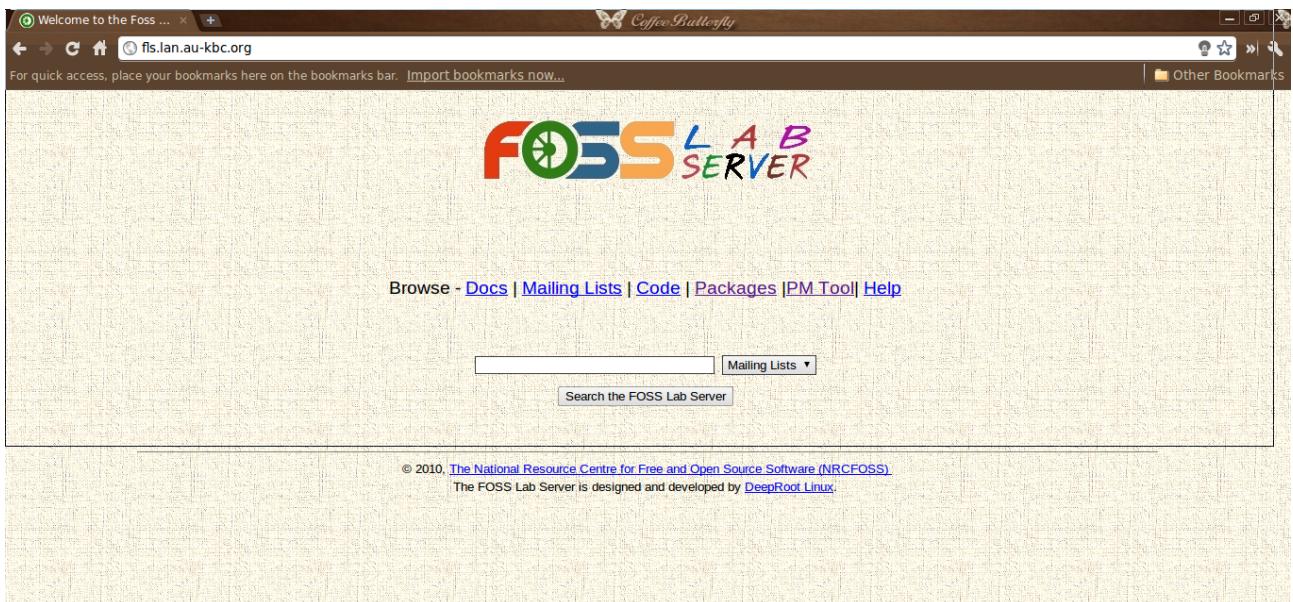
## Using the search engine

The FOSS Lab Server search engine provides one of the most effective ways to using the server. All content is indexed and made searchable using the xapian search engine library. Also the docs, lists, code and packages are searchable separately. This is possible because they are indexed separately and thus faster to query. A search result displays the following Information

- The Mime-type of the document is displayed as an icon. Currently the following mimetypes are supported: application/xml, application/pdf, text/plain, text/html, application/x-debian-package, text/x-perl, text/x-python, text/x-sh, text/x-c++src
- The title with a hyperlink to the corresponding document. If there is no title to show for the document the URL is displayed instead.

- The relevance of the search keyword in the corresponding document. The relevance is based on the proximity of the keyword to the beginning of the document
- A brief description of the search item.

The search results are sorted by rank. The search results are paginated and one can skip to any page of choice. There is no limit set to the number of search .



## Single keyword Search

- In this user can search docs, Lists, Packages and Code by using single word
- Results are hyperlinked so users can navigate easily to the particular page

## Single keyword search using Lists

File Edit View History Bookmarks Tools Help  
http://fosslab/search?query=php&ctg=lists  
Foss Lab Server - php

**FOSS LAB SERVER**  in

Browse - [Docs](#) | [Lists](#) | [Code](#) | [Packages](#) | [Help](#)

Search **php** in category **Lists** has returned approximately **54269** result(s). Showing matches **148103511 – 10**

[1] [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) ►

**Re: dev-php/php blocking dev-lang/php**  
Relevance : **99%**  
Re: dev-php/php blocking dev-lang/php Subject: Re: dev-php/php blocking dev-lang/php From: Darren Grant <dgrant@monstertech.ca> Date: Tue, 07 Feb 2006 13:50:18 -0800 James Ausmus wrote: On 2/7/06, \*Darren Grant\* <dgrant@monstertech.ca <mailto:dgrant@monstertech.ca>> wrote: I'm trying to install squirrelmail on a system that I've already compiled apache2 with dev-php/php and dev-php/mod\_php support. squirrelmail is complaining

**Re: dev-php/php blocking dev-lang/php**  
Relevance : **99%**  
Re: dev-php/php blocking dev-lang/php Subject: Re: dev-php/php blocking dev-lang/php From: James Ausmus <james.ausmus@gmail.com> Date: Tue, 7 Feb 2006 14:11:33 -0800 On 2/7/06, Darren Grant <dgrant@monstertech.ca> wrote: <snip> Ahh found it... just needed to change -D PHP to -D PHP5 ;) -- gentoo-user@gentoo.org mailing list Great, glad it's working for you! -James Follow-Ups: Re: dev-php/php blocking dev-lang/php From: Darren

**Re: dev-php/php blocking dev-lang/php**  
Relevance : **99%**  
Re: dev-php/php blocking dev-lang/php Subject: Re: dev-php/php blocking dev-lang/php From: Darren Grant <dgrant@monstertech.ca> Date: Tue, 07 Feb 2006 15:58:23 -0800 Yeah thanks for everyone's help. The only thing I'm having trouble with now is phpmyadmin. It's giving me a ... <> phpMyAdmin - Error Cannot load /session/ http://.../session/autologin. Please check your PHP configuration - I tried recompiling it after all

Done

## Single keyword search using Docs

File Edit View History Bookmarks Tools Help  
http://fosslab/search?query=php&ctg=docs  
Foss Lab Server - php

**FOSS LAB SERVER**  in

Browse - [Docs](#) | [Lists](#) | [Code](#) | [Packages](#) | [Help](#)

Search **php** in category **docs** has returned approximately **18711** result(s). Showing matches **148055487 – 10**

[1] [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) ►

**PHP: PHP input/output streams - Manual**  
Relevance : **99%**  
downloads | documentation | faq | getting help | mailing lists | reporting bugs | php.net sites | links | conferences | my php.net search for in the function list all php.net sites this mirror only online documentation bug database Site News Archive All Changelogs just pear.php.net just pecl.php.net just talks.php.net general mailing list developer mailing list documentation mailing list PHP Manual Appendices

**PHP: php.ini directives - Manual**  
Relevance : **99%**  
downloads | documentation | faq | getting help | mailing lists | reporting bugs | php.net sites | links | conferences | my php.net search for in the function list all php.net sites this mirror only online documentation bug database Site News Archive All Changelogs just pear.php.net just pecl.php.net just talks.php.net general mailing list developer mailing list documentation mailing list PHP Manual Appendices

**PHP: Migrating from PHP 4 to PHP 5 - Manual**  
Relevance : **98%**  
downloads | documentation | faq | getting help | mailing lists | reporting bugs | php.net sites | links | conferences | my php.net search for in the function list all php.net sites this mirror only online documentation bug database Site News Archive All Changelogs just pear.php.net just pecl.php.net just

Done

## Single keyword search using Code

The screenshot shows a web browser window with the following details:

- Address Bar:** http://fosslab/search?query=php&ctg=code
- Title Bar:** Foss Lab Server - php
- Search Bar:** A search bar with "php" in it, with a dropdown menu set to "Code". Below it is a button labeled "Search the FOSS Lab Server".
- Breadcrumbs:** Browse - [Docs](#) | [Lists](#) | [Code](#) | [Packages](#) | [Help](#)
- Search Results Summary:** Search php in category code has returned approximately **26248** result(s). Showing matches **148610039 – 10**.
- Page Number:** [1] [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) ►
- Results List:** The results are listed as follows:
  - [admin/xmldb/actions/view\\_table.php/view\\_table.php.class.php](#)  
Relevance : **100%**
  - [php/download.php](#)  
Relevance : **98%**
  - [php/index.php](#)  
Relevance : **98%**
  - [php/phpAiksaur.php](#)  
Relevance : **98%**
  - [php/showsource.php](#)  
Relevance : **98%**
- Bottom Navigation:** Done, with various browser extension icons.

## Multi-word or phrase search

Phrase search uses the default AND operator. If two words are separated by mere space, its assumed to be an AND. However an OR can also be specified explicitly. The search application does not support substring and wildcard search yet.

### Multi-word or phrase search using Lists

File Edit View History Bookmarks Tools Help

http://fosslab/search?query=kde+or+gnome&ctg=lists

Foss Lab Server - kde or gnome

**FOSS LAB SERVER**

kde or gnome in Lists

Search the FOSS Lab Server

Browse - [Docs](#) | [Lists](#) | [Code](#) | [Packages](#) | [Help](#)

Search kde or gnome in category lists Showing matches 148515667 – 10

**Re: Gnome or KDE?**  
Relevance : 99%

Re: Gnome or KDE? Subject: Re: Gnome or KDE? From: ajtim <ajtim@ctel.net> Date: Mon, 18 Apr 2005 13:34:53 -0400 On Monday 18 April 2005 13:24, audriusb@homelan.lt wrote: > Quoting mbuel76 <ulist@gsl.ubuntuforums.org>: > I was working with KDE long before GNOME was born and I agree that KDE was > better than GNOME but once upon time UNIX users community selected GNOME > as main desktop because of

**Re: Gnome or KDE?**  
Relevance : 99%

Re: Gnome or KDE? Subject: Re: Gnome or KDE? From: Karl <kaha@colug.org> Date: Mon, 18 Apr 2005 21:30:38 -0700 Xfce is one of those shining jewels that a lot of people haven't discovered yet. I actually flip back and forth between Gnome and xfce. On Mon, 2005-04-18 at 21:15 -0500, Drewcore wrote: > granted, gnome and kde both look nice... > > but i'm an xfce user... it's way faster on my machine, i haven't had > problems getting

**Re: Gnome or KDE?**  
Relevance : 99%

Re: Gnome or KDE? Subject: Re: Gnome or KDE? From: nautilus <ulist@gsl.ubuntuforums.org> Date: Mon, 18 Apr 2005 23:01:44 -0400 \*raised fist\* Vi, babay. -- nautilus References: Gnome or KDE? From: lluishc <lluishc@gmail.com> Re: Gnome or KDE? From: "Yannick Le Saint (kyncani)" <y.lesaint@gmail.com> Re: Gnome or KDE? From: Adam Membrev <membrev@ontusnet.com.au> Re: Gnome or KDE? From: "Yannick Le Saint (kyncani)"

Done

## Multi-word or phrase search using Docs

File Edit View History Bookmarks Tools Help

http://fosslab/search?query=kde+or+gnome&ctg=docs

Foss Lab Server - kde or gnome

**FOSS LAB SERVER**

kde or gnome in Docs

Search the FOSS Lab Server

Browse - [Docs](#) | [Lists](#) | [Code](#) | [Packages](#) | [Help](#)

Search kde or gnome in category docs Showing matches 149074215 – 10

**X Windows versus Windows 95/98/NT: No contest LG #36**  
Relevance : 100%

"Linux Gazette...making Linux just a little more fun!" X Windows versus Windows 95/98/NT: No contest By Paul Gregory Cooper In the December Issue of the Linux Journal Sergio Martinez wrote in asking for a (quick) article about the differences between X and Windows 95/98/NT (w95) -- see below. This is my attempt to answer his questions - I remember asking similar things when I started using UNIX 4 years ago. [More answers can be

**Two Desktop Systems**  
Relevance : 96%

"Linux Gazette...making Linux just a little more fun!" KDE and Gnome by Larry Ayers Introduction Watching the Linux operating system begin to mature is interesting these days. A couple of years ago much attention was devoted to incompatibilities with various hardware components, networking, and the development of the kernel itself. Though these activities continue, it's no longer necessary to follow these development efforts as

**Two Desktop Systems**  
Relevance : 96%

"Linux Gazette...making Linux just a little more fun!" KDE and Gnome by Larry Ayers Introduction Watching the Linux operating system begin to mature is interesting these days. A couple of years ago much attention was devoted to incompatibilities with various hardware components, networking, and the development of the kernel itself. Though these activities continue, it's no longer necessary to follow these development efforts as

Done

## Multi-word or phrase search using Code

The screenshot shows a web browser window with the following details:

- Address Bar:** http://fosslab/search?query=kde+or+gnome&ctg=docs
- Title Bar:** Foss Lab Server - kde or gnome
- Search Bar:** kde or gnome in Code
- Search Button:** Search the FOSS Lab Server
- Browse Links:** Docs | Lists | Code | Packages | Help
- Search Results Summary:** Search kde or gnome in category docs | Showing matches 149074215 – 10
- First Result:**
  - Title:** X Windows versus Windows 95/98/NT: No contest LG #36
  - Relevance:** 100%
  - Description:** "Linux Gazette...making Linux just a little more fun!" X Windows versus Windows 95/98/NT: No contest By Paul Gregory Cooper In the December Issue of the Linux Journal Sergio Martinez wrote in asking for a (quick) article about the differences between X and Windows 95/98/NT (w95) -- see below. This is my attempt to answer his questions - I remember asking similar things when I started using UNIX 4 years ago. [More answers can be
- Second Result:**
  - Title:** Two Desktop Systems
  - Relevance:** 96%
  - Description:** "Linux Gazette...making Linux just a little more fun!" KDE and Gnome by Larry Ayers Introduction Watching the Linux operating system begin to mature is interesting these days. A couple of years ago much attention was devoted to incompatibilities with various hardware components, networking, and the development of the kernel itself. Though these activities continue, it's no longer necessary to follow these development efforts as
- Third Result:**
  - Title:** Two Desktop Systems
  - Relevance:** 96%
  - Description:** "Linux Gazette...making Linux just a little more fun!" KDE and Gnome by Larry Ayers Introduction Watching the Linux operating system begin to mature is interesting these days. A couple of years ago much attention was devoted to incompatibilities with various hardware components, networking, and the development of the kernel itself. Though these activities continue, it's no longer necessary to follow these development efforts as

At the bottom of the browser window, there are several small icons and a "Done" button.

## CS2406-Open Source Laboratory

### **List of Questions:**

1. (i) Implement the Employee details application using PHP with MYSQL.  
(ii) Install the samba package for sharing files from Linux to windows.
2. (i) Implement the Bank application using python with MYSQL.  
(ii) Install the webbattery package and compiling from source for checking the status of battery level.
3. (i) Implement the Student details application using Perl with MYSQL.  
(ii) Develop a package resource pool and serve for clients to access it over http / or ftp like accessing atop-rpm.
4. (i) Implement the Electricity Board(EB) application using Perl with MYSQL.  
(ii) Install fuse package for achieving userspace drivers which is easier to debug and less dangerous to the system.
5. (i) Develop an On-line shopping GUI using Tk.  
(ii) Implement the setup of complement Network Interface using command mode.
6. (i) Implement Student name and phone number catalog application using CGI with MYSQL.  
(ii) Install the cups package for common printing system.
7. (i) Implement the virtualization size of 100M hard disk with 2-CPU each having 512M RAM space using qemu.  
(ii) Implement the following using PHP: (a) Prime number check (b) Distance calculation.
8. (i) Implement the kernel configuration, compilation and installation by downloading basic kernel.  
(ii) Implement the following using python: (a) Calculator (b) Temperature conversion.
9. (i) Implement the Library details application using PHP with MYSQL.  
(ii) Implement CGI program to give a message about CSE Department.
10. (i) Implement the Train Reservation application using python with MYSQL.  
(ii) Implement to give a Greeting Message when button pressed using python-TK.
  
11. (i) Implement the Student details application using PHP with MYSQL.  
(ii) Install the webbattery package and compiling from source for checking the status of battery level.
12. (i) Implement the Library application using python with MYSQL.  
(ii) Install the samba package for sharing files from Linux to windows.
13. (i) Implement the Student details application using Perl with MYSQL.  
(ii) Develop a package resource pool and serve for clients to access it over http / or ftp like accessing atop-rpm.

## CS2406-Open Source Laboratory

14. (i) Implement the Electricity Board(EB) application using Perl with MYSQL.  
(ii) Install fuse package for achieving userspace drivers which is easier to debug and less dangerous to the system.
15. (i) Develop an On-line Voting System GUI using Tk.  
(ii) Implement the setup of complement Network Interface using command mode.
16. (i) Implement Movies catalog application using CGI with MYSQL.  
(ii) Install the cups package for common printing system.
17. (i) Implement the virtualization size of 100M hard disk with 2-CPU each having 512M RAM space using qemu.  
(ii) Implement the gmail Sign-up registration form design using PHP.
18. (i) Implement the kernel configuration, compilation and installation by downloading basic kernel.  
(ii) Implement the following using python: (a) Calculator (b) Temperature conversion.
19. (i) Implement the On-line course registration application using PHP with MYSQL.  
(ii) Implement CGI program to give a message about CSE Department.
20. (i) Implement the Bus Reservation application using python with MYSQL.  
(ii) Implement to give a Greeting Message when button pressed using python-TK.
21. (i) Implement the Population Survey application using PHP with MYSQL.  
(ii) Install the webbattery package and compiling from source for checking the status of battery level.
22. (i) Implement the Medical Database application using python with MYSQL.  
(ii) Install the samba package for sharing files from Linux to windows.
23. (i) Implement the Student details application using Perl with MYSQL.  
(ii) Develop a package resource pool and serve for clients to access it over http / or ftp like accessing atop-rpm.
24. (i) Implement the Electricity Board(EB) application using Perl with MYSQL.  
(ii) Install fuse package for achieving userspace drivers which is easier to debug and less dangerous to the system.
25. (i) Develop an On-line Voting System GUI using Tk.  
(ii) Implement the setup of complement Network Interface using command mode.
26. (i) Implement Movies catalog application using CGI with MYSQL.  
(ii) Install the cups package for common printing system.
27. (i) Implement the virtualization size of 100M hard disk with 2-CPU each having 512M RAM space using qemu.  
(ii) Implement the gmail Sign-up registration form design using PHP.
28. (i) Implement the kernel configuration, compilation and installation by

CS2406-Open Source Laboratory

downloading basic kernel.

(ii) Implement the following using python: (a) Factorial (b) Matrix addition using array (2X2).

29. (i) Implement the On-line course registration application using PHP with MYSQL.

(ii) Implement CGI program to give a message about CSE Department.

30. (i) Implement the Bus Reservation application using python with MYSQL.

(ii) Implement to give a Greeting Message when button pressed using python-TK.

31. (i) Implement the Bank details application using PHP with MYSQL.

(ii) Install the samba package for sharing files from Linux to windows.

32. (i) Implement the Employee details using python with MYSQL.

(ii) Install the webattery package and compiling from source for checking the status of battery level.

33. (i) Implement the Train Reservation application using Perl with MYSQL.

(ii) Develop a package resource pool and serve for clients to access it over http / or ftp like accessing atop-rpm.

34. (i) Implement the Electricity Board(EB) application using Perl with MYSQL.

(ii) Install fuse package for achieving userspace drivers which is easier to debug and less dangerous to the system.

35. (i) Develop an On-line shopping GUI using Tk.

(ii) Implement the setup of complement Network Interface using command mode.

36. (i) Version Control System setup and usage using RCS, CVS, SVN for accessing repositories.

(ii) Implement a simple application to send a message to parent' Mobile for Student absent on that day using Gambas.

INTERNAL EXAMINER

EXTERNAL EXAMINER

*CS2406-OPEN SOURCE LAB*

***LAB MANUAL: 2012-2013(ODD SEMESTER)***

***CS2406-OPEN SOURCE LAB***

**CS 2406**

**OPEN SOURCE LAB**

**L T P C 0 0 3 2**

**OBJECTIVE:**

To expose students to FOSS environment and introduce them to use open source packages.

1. **Kernel configuration, compilation and installation :** Download / access The latest kernel source code from kernel.org,compile the kernel and install it in the local system.Try to view the source code of the kernel
2. **Virtualisation environment** (e.g., xen, kqemu or lguest) to test an applications,new kernels and isolate applications. It could also be used to expose students to other alternate OSs like \*BSD
3. **Compiling from source :** learn about the various build systems used like the auto\* family, cmake, ant etc. instead of just running the commands. This could involve the full process like fetching from a cvs and also include autoconf,automake etc.,
4. **Introduction to packet management system :** Given a set of RPM or DEB, how to build and maintain, serve packages over http or ftp. and also how do you configure client systems to access the package repository.
5. **Installing various software packages** Either the package is yet to be installed or an older version is existing. The student can practice installing the latest version. Of course, this might need internet access
  - 5.1 Install samba and share files to windows
  - 5.2 Install Common Unix Printing System(CUPS)
6. **Write userspace drivers using fuse** — easier to debug and less dangerous to the system (Writing full-fledged drivers is difficult at student level)
7. **GUI programming** : a sample programme – using Gambas since the students have VB knowledge. However, one should try using GTK or QT
8. **Version Control System** setup and usage using RCS, CVS, SVN
9. **Text processing with Perl**: simple programs, connecting with database e.g., MYSQL
10. **Running PHP** : simple applications like login forms after setting up a LAMP stack
11. **Running Python** : some simple exercise – e.g. Connecting with MySql database

12. Set up the complete network interface using ifconfig command like setting gateway, DNS, IP tables, etc.,

TOTAL: 45 PERIODS

**LIST OF EQUIPMENTS:**

**Hardware:**

**Minimum Requirements:**

- 700 Mhz X86 Processor
- 384 MB of system memory (RAM)
- 40 GB of disk space
- Graphics card capable of 1024\*768 resolution
- Sound Card
- Network or Internet Connection

**Software:**

Latest distribution of Linux(FEDERO 14)

## **LIST OF EXPERIMENTS**

- 1) Kernel configuration, compilation and installation**
- 2) Virtualisation environment**
- 3) Compiling from source**
- 4) Introduction to packet management system**
- 5) Installing various software packages**
- 6) Write userspace drivers using fuse**
- 7) GUI programming**
- 8) Version Control System**
- 9) Text processing with Perl**
- 10) Running PHP**
- 11) Running Python**
- 12) Set up the complete network interface**

## **EX. No : 1    KERNEL CONFIGURATION, COMPILEDATION AND INSTALLATION**

**DATE:**

**Aim :**

To download / access The latest kernel source code from kernel.org,compile the kernel and install it in the local system and to Try viewing the source code of the kernel.

**Algorithm:**

Follow the following Steps:

**Step 1:**

select the kernel source code (ex no 1)->display the linux-2.6.35.7.tar.gz->select linux-2.6.35.7.tar.gz link->save the folder in Downloads directory->go to Download directory from places in menubar->extract linux-2.6.35.7.tar.gz folder on that same directory.

**Step 2:Display the user name**

```
[fosslab@fosslab ~]$ uname -r  
2.6.35.6-45.fc14.i686
```

**Step 3: Enter the Download Directory and Display password of the Directory and display the list**

```
[fosslab@fosslab ~]$ cd Downloads/  
[fosslab@fosslab Downloads]$ pwd  
/home/fosslab/Downloads  
[fosslab@fosslab Downloads]$ ls -l  
total 93332  
-rw-rw-r-- 1 fosslab fosslab 344011 Jun 28 01:50 04524284.pdf  
-rw-rw-r-- 1 fosslab fosslab 251225 Jun 28 01:36 91-US-31-1_Cloud_Computing.pdf  
-rw-rw-r-- 1 fosslab fosslab 556032 Jun 28 01:41 ABSTRACT and pro_vidhya.doc  
-rw-rw-r-- 1 fosslab fosslab 429466 Jun 28 01:37 computing-whitepaper.pdf  
-rw-rw-r-- 1 fosslab fosslab 99019 Aug 1 01:03 Criterion-8.docx  
-rw-rw-r-- 1 fosslab fosslab 110194 Aug 1 01:03 Criterion 9(2).docx  
-rw-rw-r-- 1 fosslab fosslab 110194 Aug 1 01:03 Criterion 9.docx  
-rw-rw-r-- 1 fosslab fosslab 512000 Aug 1 01:03 criter XP12-chandran.doc  
-rw-rw-r-- 1 fosslab fosslab 422400 Aug 1 01:03 criter XP12_Mech.doc
```

```
-rwxrwxr-x 1 fosslab fosslab 493564 Jun  7 2011 Downloads.exe
-rw-rw-r-- 1 fosslab fosslab 634100 Jun 28 00:13 EJSR_64_2_05.pdf
-rw-rw-r-- 1 fosslab fosslab 237418 Jun 28 00:13 EJSR_64_2_14.pdf
-rw-rw-r-- 1 fosslab fosslab 54227 Jun 28 01:39 EJSR_74_3_04.pdf
-rw-rw-r-- 1 fosslab fosslab 527523 Jun 28 01:44 EJSR_77_1_06.pdf
-rw-rw-r-- 1 fosslab fosslab 147175 Jul  9 01:55 foss-lab-manual-p1-1.0-rc1.pdf
-rw-rw-r-- 1 fosslab fosslab 20228 Jun 28 01:50 Gartner Data Mining Addtl.pdf
-rw-rw-r-- 1 fosslab fosslab 12253 Aug 11 14:02 HP-LaserJet-laserjet.ppd
-rw-rw-r-- 1 fosslab fosslab 219237 Jun 28 00:10 kdd98_elder_abott_nopics_bw.pdf
-rw-rw-r-- 1 fosslab fosslab 88323744 Aug 16 14:27 linux-2.6.35.7.tar.gz
-rw-rw-r-- 1 fosslab fosslab 60416 Jul 14 04:56 newFOC LP.doc
-rw-rw-r-- 1 fosslab fosslab 43520 Jun 28 01:40 ProjectTitles.doc
-rw-rw-r-- 1 fosslab fosslab 180964 Jun 28 01:57 sensor-route-security.pdf
-rw-rw-r-- 1 fosslab fosslab 56320 Jul 18 01:55 ssLABabet outcome.doc
-rw-rw-r-- 1 fosslab fosslab 142336 Jul 18 01:53 ss_Lesson-plan.doc
-rw-rw-r-- 1 fosslab fosslab 14540 Jul 30 04:55 st_newmark3(2).jsp
-rw-rw-r-- 1 fosslab fosslab 14702 Jul 30 04:45 st_newmark3.jsp
-rw-rw-r-- 1 fosslab fosslab 1517376 Jul 28 02:36 wrar420.exe
```

**Step 4: Enter the linux-2.6.35.7 Directory and Display the password of the linux-2.6.35.7 and Display the list of that linux-2.6.35.7 directory**

```
[fosslab@fosslab Downloads]$ cd linux-2.6.35.7/
[fosslab@fosslab linux-2.6.35.7]$ pwd
/home/fosslab/Downloads/linux-2.6.35.7
[fosslab@fosslab linux-2.6.35.7]$ ls -l
total 456
drwxrwxr-x 25 fosslab fosslab 4096 Sep 28 2010 arch
drwxrwxr-x 2 fosslab fosslab 4096 Sep 28 2010 block
-rw-rw-r-- 1 fosslab fosslab 18693 Sep 28 2010 COPYING
-rw-rw-r-- 1 fosslab fosslab 94031 Sep 28 2010 CREDITS
drwxrwxr-x 3 fosslab fosslab 4096 Sep 28 2010 crypto
drwxrwxr-x 85 fosslab fosslab 12288 Sep 28 2010 Documentation
drwxrwxr-x 89 fosslab fosslab 4096 Sep 28 2010 drivers
drwxrwxr-x 36 fosslab fosslab 4096 Sep 28 2010 firmware
```

```
drwxrwxr-x 72 fosslab fosslab 4096 Sep 28 2010 fs
drwxrwxr-x 20 fosslab fosslab 4096 Sep 28 2010 include
drwxrwxr-x 2 fosslab fosslab 4096 Sep 28 2010 init
drwxrwxr-x 2 fosslab fosslab 4096 Sep 28 2010 ipc
-rw-rw-r-- 1 fosslab fosslab 2440 Sep 28 2010 Kbuild
drwxrwxr-x 8 fosslab fosslab 4096 Sep 28 2010 kernel
drwxrwxr-x 6 fosslab fosslab 4096 Sep 28 2010 lib
-rw-rw-r-- 1 fosslab fosslab 174535 Sep 28 2010 MAINTAINERS
-rw-rw-r-- 1 fosslab fosslab 51251 Sep 28 2010 Makefile
drwxrwxr-x 2 fosslab fosslab 4096 Sep 28 2010 mm
drwxrwxr-x 50 fosslab fosslab 4096 Sep 28 2010 net
-rw-rw-r-- 1 fosslab fosslab 17459 Sep 28 2010 README
-rw-rw-r-- 1 fosslab fosslab 3371 Sep 28 2010 REPORTING-BUGS
drwxrwxr-x 7 fosslab fosslab 4096 Sep 28 2010 samples
drwxrwxr-x 12 fosslab fosslab 4096 Sep 28 2010 scripts
drwxrwxr-x 7 fosslab fosslab 4096 Sep 28 2010 security
drwxrwxr-x 21 fosslab fosslab 4096 Sep 28 2010 sound
drwxrwxr-x 4 fosslab fosslab 4096 Sep 28 2010 tools
drwxrwxr-x 2 fosslab fosslab 4096 Sep 28 2010 usr
drwxrwxr-x 3 fosslab fosslab 4096 Sep 28 2010 virt
```

**Step 5:Open the Makefile in vi editor and type the extraversion directory**

[fosslab@fosslab linux-2.6.35.7]\$ vi Makefile

VERSION = 2

PATCHLEVEL = 6

SUBLEVEL = 35

EXTRAVERSION = .7-veni

NAME = Yokohama

**Step 6:To make the menuconfig**

[fosslab@fosslab linux-2.6.35.7]\$ make menuconfig

HOSTCC scripts/basic/fixdep

HOSTCC scripts/basic/docproc

HOSTCC scripts/basic/hash

HOSTCC scripts/kconfig/conf.o

```
HOSTCC scripts/kconfig/kxgettext.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
HOSTCC scripts/kconfig/lxdialog/textbox.o
HOSTCC scripts/kconfig/lxdialog/util.o
HOSTCC scripts/kconfig/lxdialog/yesno.o
HOSTCC scripts/kconfig/mconf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/lex.zconf.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf arch/x86/Kconfig
#
# using defaults found in /boot/config-2.6.35.6-45.fc14.i686
#
#
# configuration written to .config
#
*** End of Linux kernel configuration.
*** Execute 'make' to build the kernel or try 'make help'.
```

### **Step 7:Display the General setup screen and do the following steps**

file systems->DOS/FAT/NT FILE SYSTEM->NTFS FILE SYSTEMS press space bar->NTFS  
FILE SUPPORT press the space bar->exit

### **Step 8: To make the bootImage from the x86 boot directory**

[fosslab@fosslab linux-2.6.35.7]\$make bzImage

### **Step 9: Login in to the super user(root)**

[fosslab@fosslab linux-2.6.35.7]\$ su

Password: (admin123)

### **Step 10: To copy the bootlinuzImage from the x86 boot directory in to the vmlinuz-2.6.35.7 directory**

```
[root@fosslab linux-2.6.35.7]# cp arch/x86/boot/bzImage /boot/vmlinuz-2.6.35.7-veni
```

**Step 11: Exit from the super user**

```
[root@fosslab linux-2.6.35.7]# exit
```

```
exit
```

**Step 12: To make the modules in linux-2.6.35.7 and Display list of modules are available in modules**

```
[fosslab@fosslab linux-2.6.35.7]$ make modules
```

```
[fosslab@fosslab linux-2.6.35.7]$ ls -l /lib/modules
```

```
total 8
```

```
drwxr-xr-x. 6 root root 4096 May 29 02:54 2.6.35.6-45.fc14.i686
```

```
drwxr-xr-x. 6 root root 4096 May 29 02:48 2.6.35.6-45.fc14.i686.PAE
```

**Step 13: Login in to the super user(root)**

```
[fosslab@fosslab linux-2.6.35.7]$ su
```

```
Password:(admin123)
```

**Step 14: Modules are install into the linux-2.6.35.7 on the super user and Display the available directories on the linux-2.6.35.7**

```
[root@fosslab linux-2.6.35.7]# make modules_install
```

```
[root@fosslab linux-2.6.35.7]# ls -l /lib/modules
```

```
total 12
```

```
drwxr-xr-x. 6 root root 4096 May 29 02:54 2.6.35.6-45.fc14.i686
```

```
drwxr-xr-x. 6 root root 4096 May 29 02:48 2.6.35.6-45.fc14.i686.PAE
```

```
drwxr-xr-x 3 root root 4096 Aug 17 11:49 2.6.35.7-veni
```

**Step 15: stored all the hardwares and directories in to th e initram directory**

```
///////////[root@fosslab linux-2.6.35.7]# mkinitrd /boot/initramfs-2.6.35.7-veni.img 2.6.35.7-veni
```

**Step 16: Enter in to the boot directory and Display the list of files are avilable on the boot directoy**

```
[root@fosslab linux-2.6.35.7]# cd /boot
```

```
[root@fosslab boot]# ls -l
```

```
total 61848
```

```
-rw-r--r--. 1 root root 114968 Oct 18 2010 config-2.6.35.6-45.fc14.i686
```

```
-rw-r--r--. 1 root root 115205 Oct 18 2010 config-2.6.35.6-45.fc14.i686.PAE
```

```
drwxr-xr-x. 3 root root 4096 May 29 01:06 efi
```

```
drwxr-xr-x. 2 root root 4096 May 29 00:53 extlinux
```

```
drwxr-xr-x. 2 root root 4096 May 29 02:56 grub
-rw-r--r--. 1 root root 13507699 May 29 02:46 initramfs-2.6.35.6-45.fc14.i686.img
-rw-r--r--. 1 root root 13502690 May 29 02:47 initramfs-2.6.35.6-45.fc14.i686.PAE.img
-rw-r--r-- 1 root root 11005350 Aug 17 11:51 initramfs-2.6.35.7-veni.img
-rw-r--r--. 1 root root 1106328 May 29 01:52 initrd-plymouth.img
-rw-r--r--. 1 root root 1681526 Oct 18 2010 System.map-2.6.35.6-45.fc14.i686
-rw-r--r--. 1 root root 1709576 Oct 18 2010 System.map-2.6.35.6-45.fc14.i686.PAE
-rwxr-xr-x. 1 root root 3696448 Oct 18 2010 vmlinuz-2.6.35.6-45.fc14.i686
-rwxr-xr-x. 1 root root 3761568 Oct 18 2010 vmlinuz-2.6.35.6-45.fc14.i686.PAE
-rw-r--r-- 1 root root 3677056 Aug 17 10:58 vmlinuz-2.6.35.7-veni
-rw-r--r--. 1 root root 571311 Oct 12 2010 xen-4.0.1.gz
lrwxrwxrwx. 1 root root 12 May 29 00:40 xen.gz -> xen-4.0.1.gz
-rw-r--r--. 1 root root 8840980 Oct 12 2010 xen-syms-4.0.1
```

**Step 17: Enter in to the grub file on the boot directory**

```
[root@fosslab boot]# cd /boot/grub
```

**Step 18: open the grub.conf file from grub directory**

```
[root@fosslab grub]# vi grub.conf
```

**add end of the file line**

```
title Fedora (2.6.35.7-veni)
root (hd0,0)
kernel /boot/vmlinuz-2.6.35.7-mykernel ro root=UUID=6c37c0aa-4b4c-4bbe-a235-d9149be80d24
rd_NO_LUKS rd_NO_LVM rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 SYSFONT=latarcyrheb-
sun16 KEYTABLE=us rhgb quiet

initrd /boot/initramfs-2.6.35.7-veni.img
```

**copy this statement form title fedora**

```
ro root=UUID=6c37c0aa-4b4c-4bbe-a235-d9149be80d24 rd_NO_LUKS rd_NO_LVM rd_NO_MD
rd_NO_DM LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYTABLE=us rhgb quiet
```

**Step 19: Reboot the system**

```
[root@fosslab grub]# reboot
```

**Result:**

Thus to download / access the latest kernel source code from kernel.org, compiling the kernel and install it in the local system and trying to view the source code of the kernel is done successfully.

**EX.NO.2:**

## **VIRTUALISATION ENVIRONMENT**

**DATE:**

**Aim:**

To test an applications, new kernels and isolate applications.

**Algorithm:**

**Follow the following steps :**

**Step1:** In Application Menu-->System Tools-->Terminals.

**Step2: Display the qemu packages are available on the resource package manager**

```
[fosslab@fosslab ~]$ rpm -qa |grep qemu
```

qemu-user-0.13.0-0.7.rc1.fc14.i686

qemu-system-sh4-0.13.0-0.7.rc1.fc14.i686

qemu-system-ppc-0.13.0-0.7.rc1.fc14.i686

qemu-system-sparc-0.13.0-0.7.rc1.fc14.i686

qemu-img-0.13.0-0.7.rc1.fc14.i686

qemu-kvm-0.13.0-0.7.rc1.fc14.i686

qemu-common-0.13.0-0.7.rc1.fc14.i686

gpxe-roms-qemu-1.0.1-1.fc14.noarch

kmod-kqemu-2.6.35.6-45.fc14.i686-1.4.0-0.2.pre1.fc14.17.i686

qemu-0.13.0-0.7.rc1.fc14.i686

qemu-system-cris-0.13.0-0.7.rc1.fc14.i686

qemu-system-m68k-0.13.0-0.7.rc1.fc14.i686

kmod-kqemu-2.6.35.6-45.fc14.i686.PAE-1.4.0-0.2.pre1.fc14.17.i686

akmod-kqemu-1.4.0-0.2.pre1.fc14.17.i686

qemu-system-mips-0.13.0-0.7.rc1.fc14.i686

kqemu-1.4.0-0.5.pre1.fc14.noarch

qemu-system-x86-0.13.0-0.7.rc1.fc14.i686

qemu-system-arm-0.13.0-0.7.rc1.fc14.i686

**Step3: Display the details about qemu**

[fosslab@fosslab ~]\$ rpm -qi qemu

Name : qemu	Relocations: (not relocatable)
Version : 0.13.0	Vendor: Fedora Project
Release : 0.7.rc1.fc14	Build Date: Tue 21 Sep 2010 06:14:54 PM EDT
Install Date: Tue 29 May 2012 01:38:39 AM EDT Build Host: x86-09.phx2.fedoraproject.org	
Group : Development/Tools	Source RPM: qemu-0.13.0-0.7.rc1.fc14.src.rpm
Size : 0	License: GPLv2+ and LGPLv2+ and BSD
Signature : RSA/SHA256, Wed 22 Sep 2010 01:42:58 PM EDT, Key ID 421caddb97a1071f	
Packager : Fedora Project	
URL : <a href="http://www.qemu.org/">http://www.qemu.org/</a>	
Summary : QEMU is a FAST! processor emulator	

Description :

QEMU is a generic and open source processor emulator which achieves a good emulation speed by using dynamic translation. QEMU has two operating modes:

\* Full system emulation. In this mode, QEMU emulates a full system (for example a PC), including a processor and various peripherals. It can be used to launch different Operating Systems without rebooting the PC or to debug system code.

\* User mode emulation. In this mode, QEMU can launch Linux processes compiled for one CPU on another CPU.

As QEMU requires no host kernel patches to run, it is safe and easy to use.

**Step4: Open Firefox and type the following address:**

<http://192.168.105.254/tools/freedos/>

**output:**

**Index of /tools/freedos**

<a href="#"><u>Name</u></a>	<a href="#"><u>Last modified</u></a>	<a href="#"><u>Size</u></a>	<a href="#"><u>Description</u></a>
<a href="#">Parent Directory</a>		-	
<a href="#">fdbasecd.iso</a>	28-Mar-2011 09:17	7.9M	
<a href="#">freedos.qcow2</a>	28-Mar-2011 09:18	9.2M	

**Step 5: Select fdbasecd.iso and save the file**

Now Goto Places -> Downloads and check whether the .iso file have been downloaded

**Step6: Goto to terminal and diaplay the password of fosslab**

[fosslab@fosslab ~]\$ pwd

/home/fosslab

**Step7: To make the directory name virtual**

[fosslab@fosslab ~]\$ mkdir virtual

**Step8: Get the directory virtual**

[fosslab@fosslab ~]\$ cd virtual/

[fosslab@fosslab virtual]\$

**Step 9:To display the password of virtual directory**

[fosslab@fosslab virtual]\$ pwd

/home/fosslab/virtual

**Step10: Move the fdbasecd.iso. folder from download directory into virtual directory**

[fosslab@fosslab virtual]\$ mv /home/fosslab/Downloads/fdbasecd.iso .

**Step11: [fosslab@fosslab virtual]\$ ls -l**

total 8140

**-rw-rw-r-- 1 fosslab fosslab 8333312 Aug 22 08:38 fdbasecd.iso**

### Step 12: Create virtual disk image size is 100M

```
[fosslab@fosslab virtual]$ qemu-img create virtualdisk.img 100M
```

Formatting 'virtualdisk.img', fmt=raw size=104857600

### Step13:[fosslab@fosslab virtual]\$ ls -l

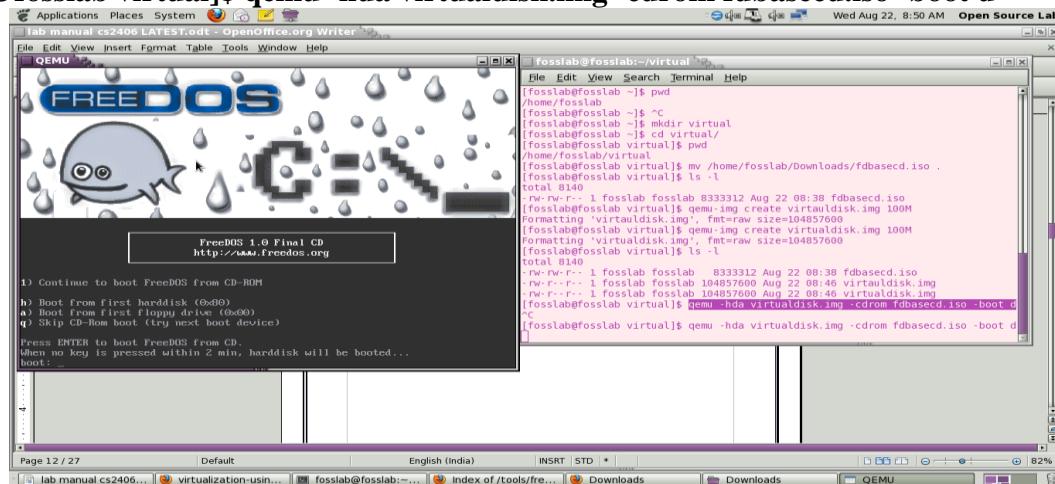
total 8140

```
-rw-rw-r-- 1 fosslab fosslab 8333312 Aug 22 08:38 fdbasecd.iso
```

```
-rw-r--r-- 1 fosslab fosslab 104857600 Aug 22 08:46 virtualdisk.img
```

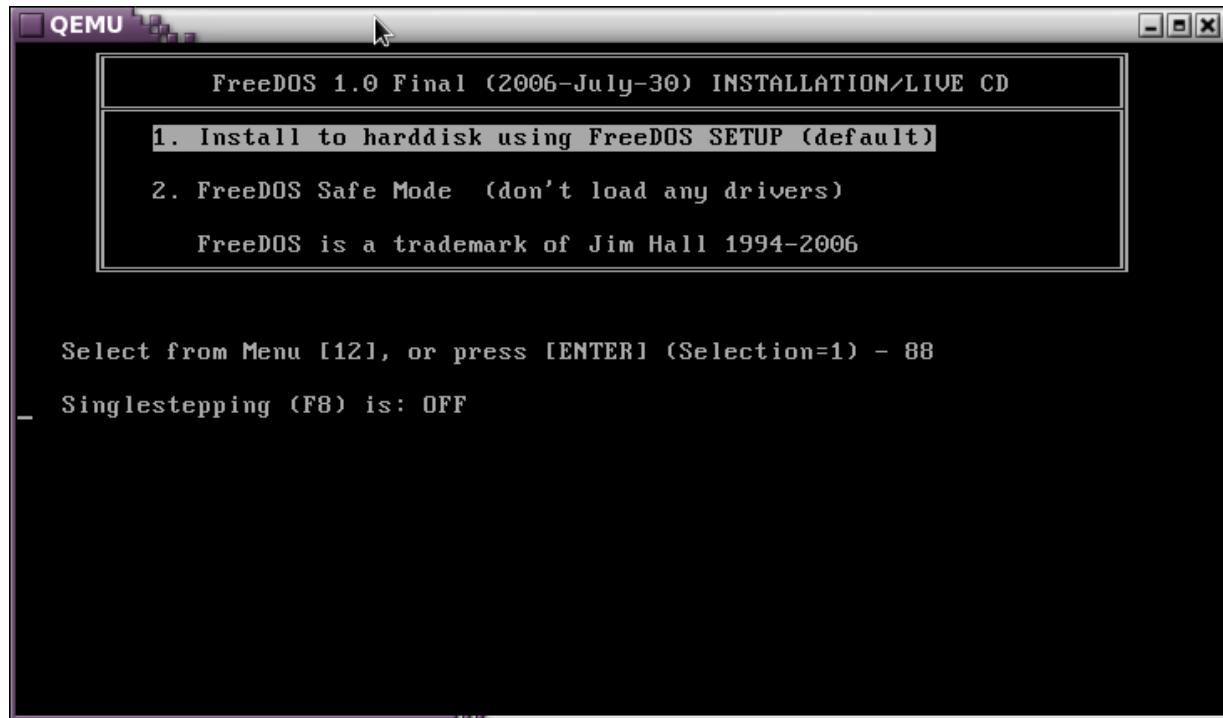
### step14:Boot the virtual disk

```
[fosslab@fosslab virtual]$ qemu -hda virtualdisk.img -cdrom fdbasecd.iso -boot d
```



Now Press Enter key

### Step15:



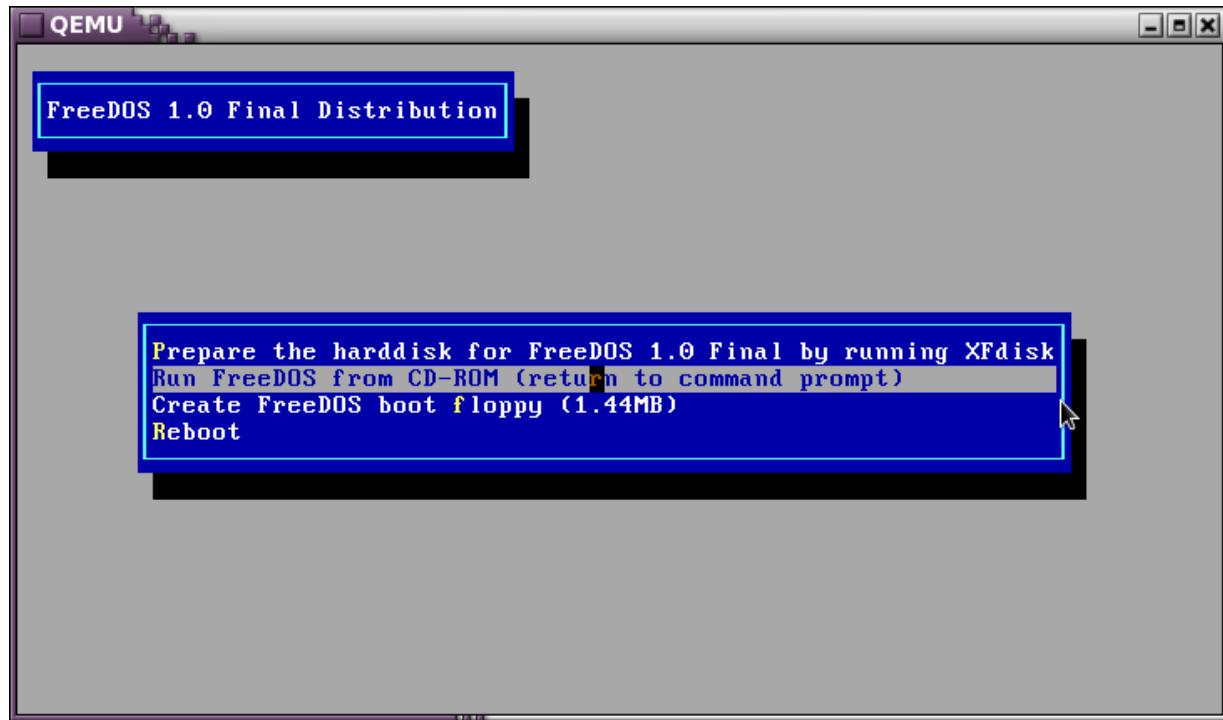
Now Press Enter

Step16:



Press Enter

Step17:



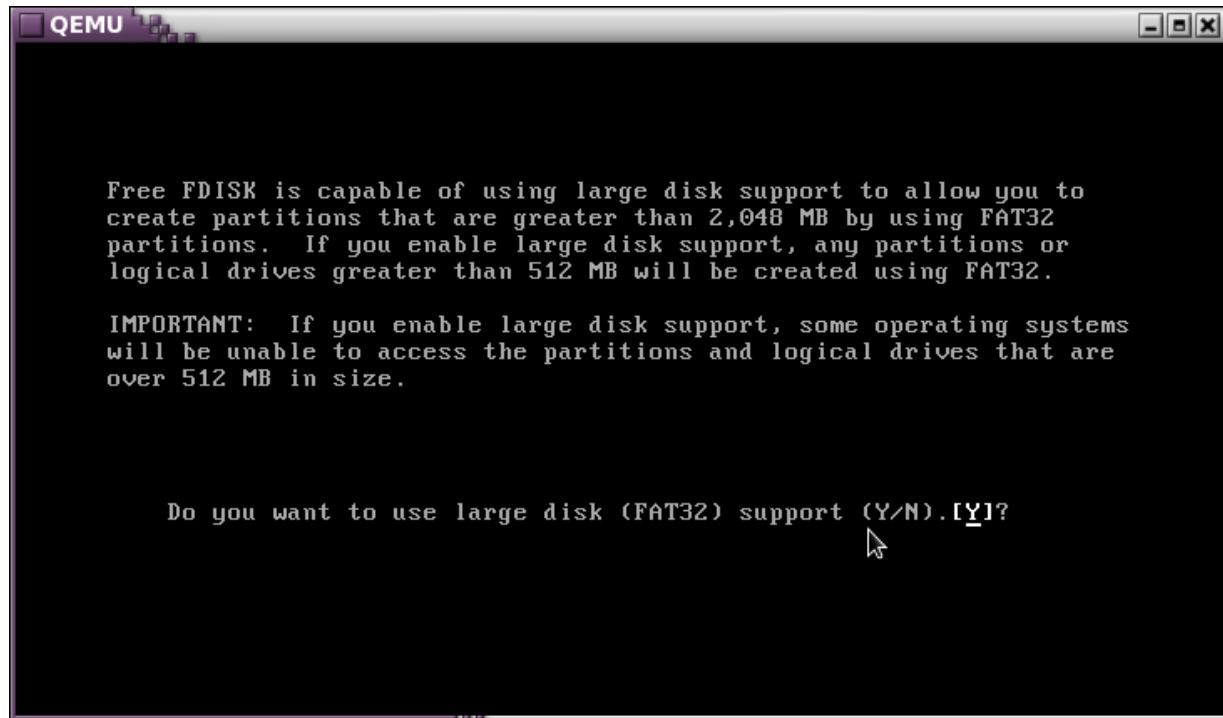
Press Enter

Step18:

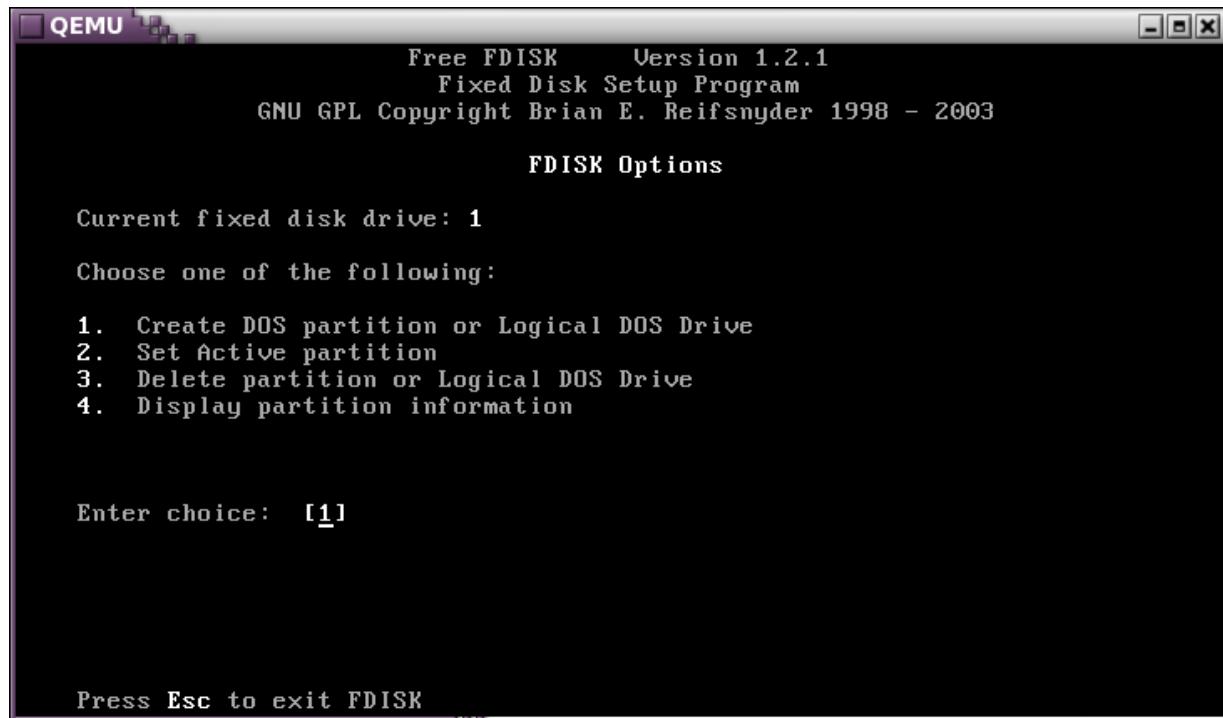


press Enter

Step19: Type fdisk

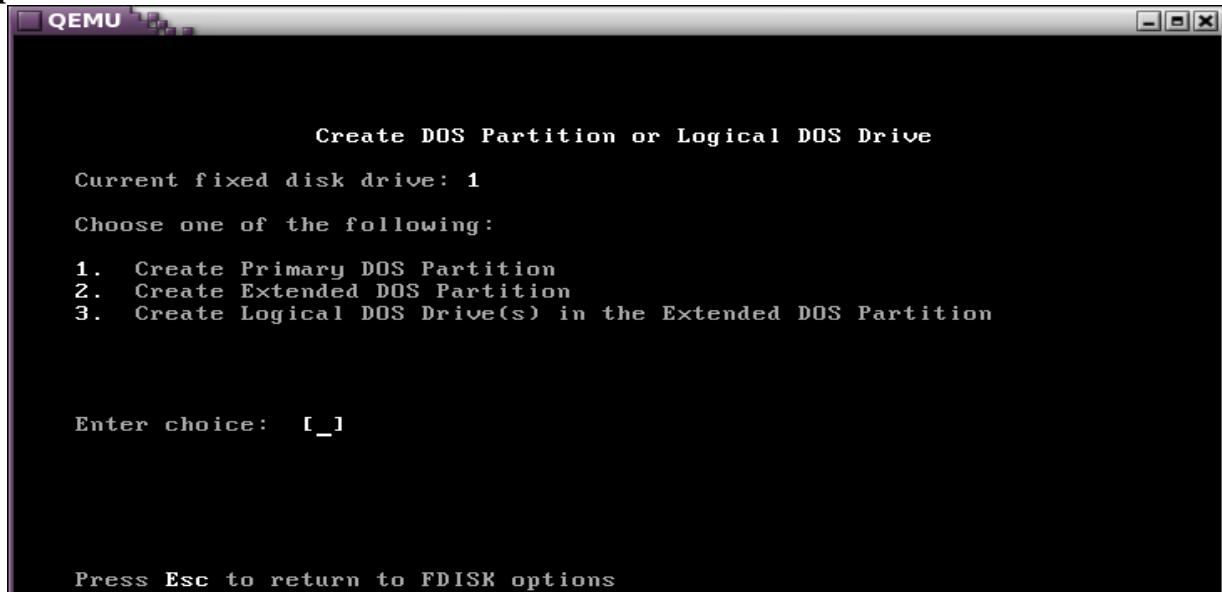


**Step20:**



Press [1]

Step21 :

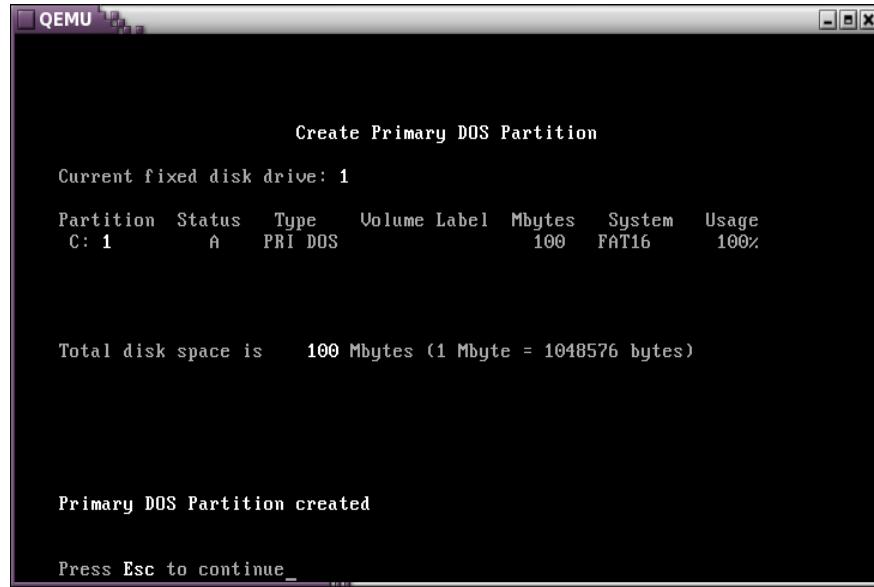


Now Press [1] option

Step22:

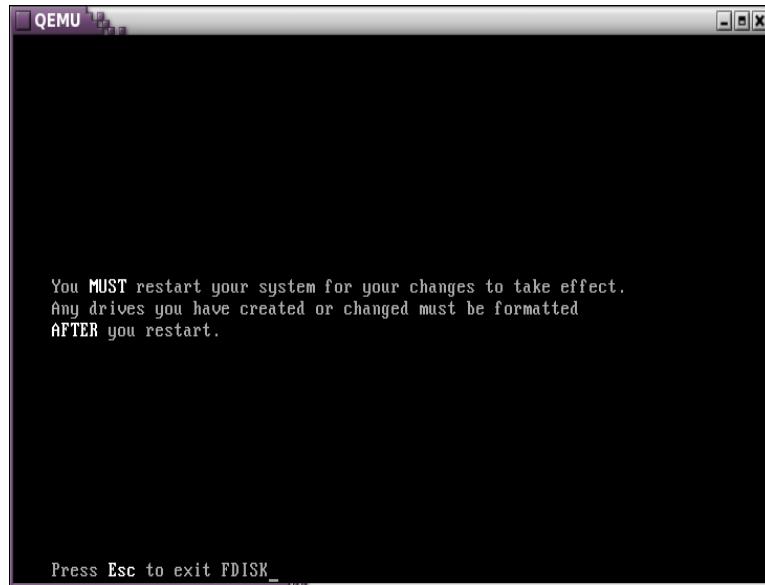


Step23:

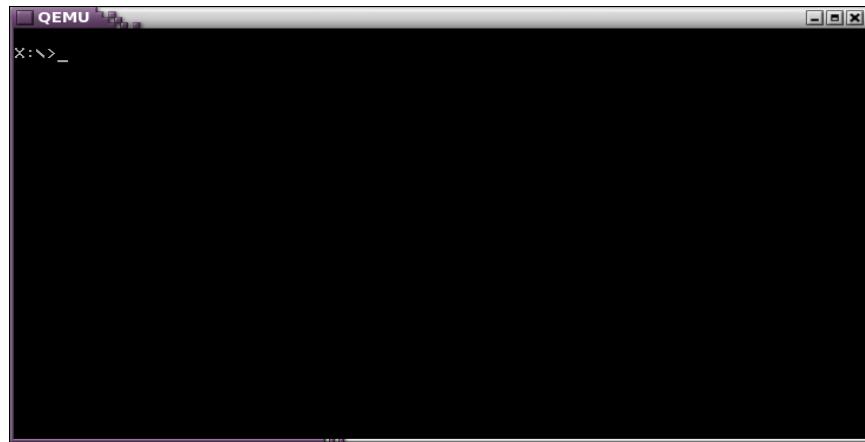


**Now press Esc to Continue**

**Step24:**



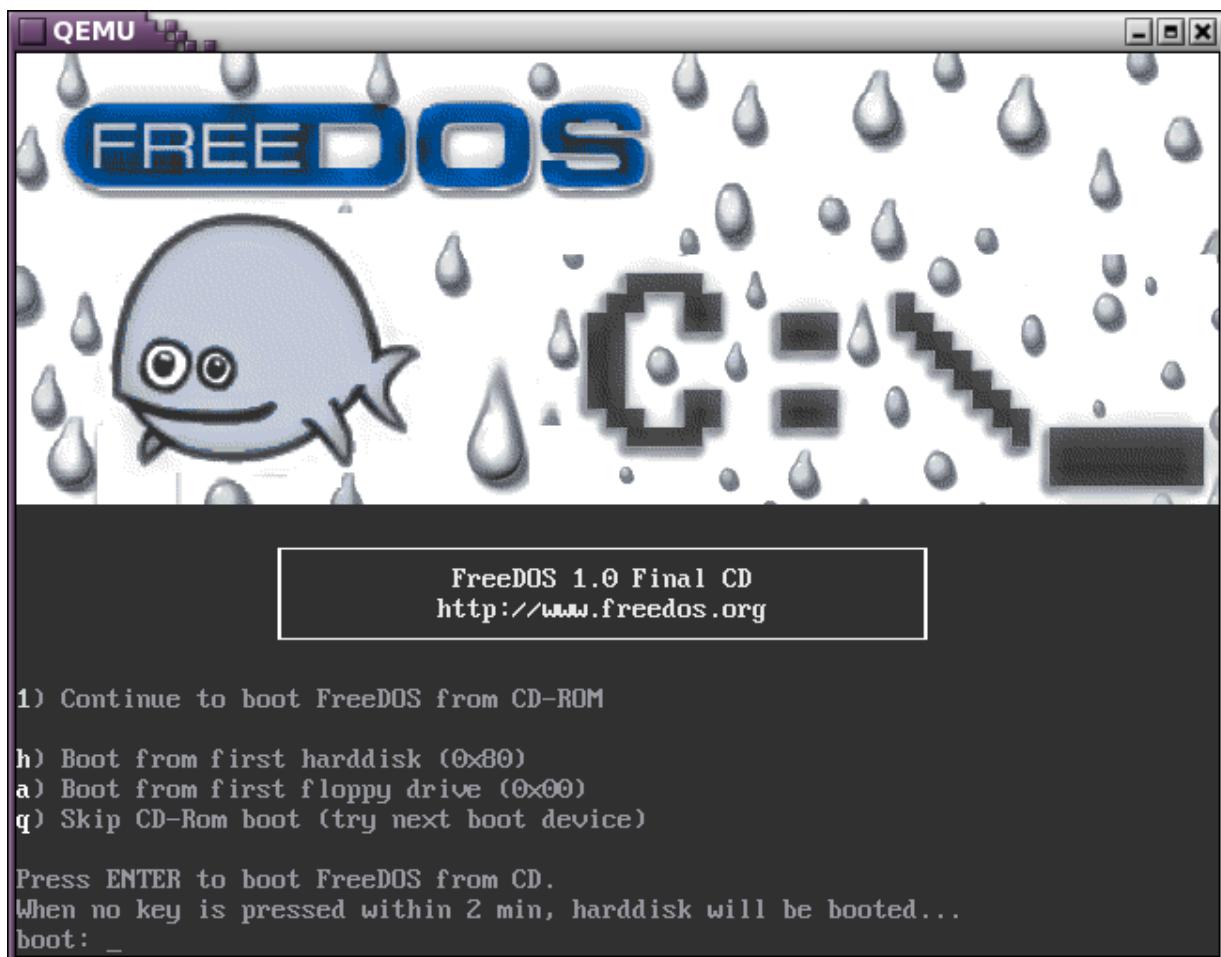
**Press ESC**



Press Esc and Quit Qemu- by pressing Cancel.

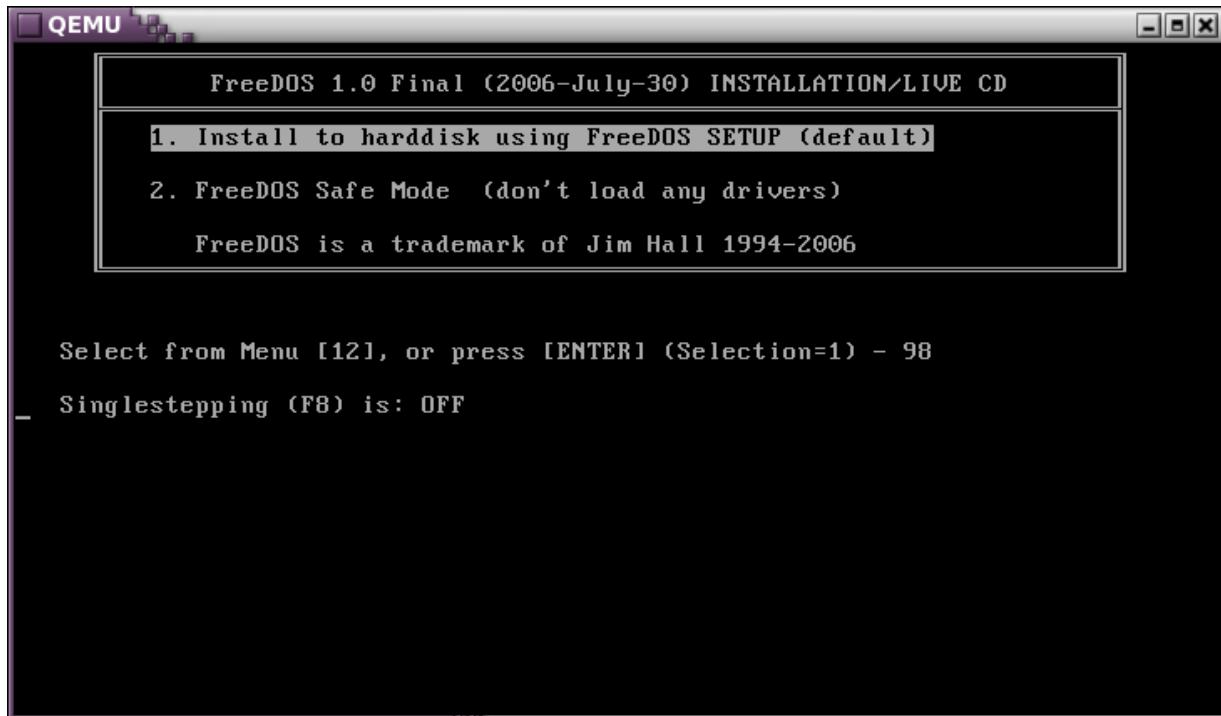
**Step25:** Now in terminal type

```
[fosslab@fosslab virtual]$ qemu -hda virtualdisk.img -cdrom fdbasecd.iso -boot d
```

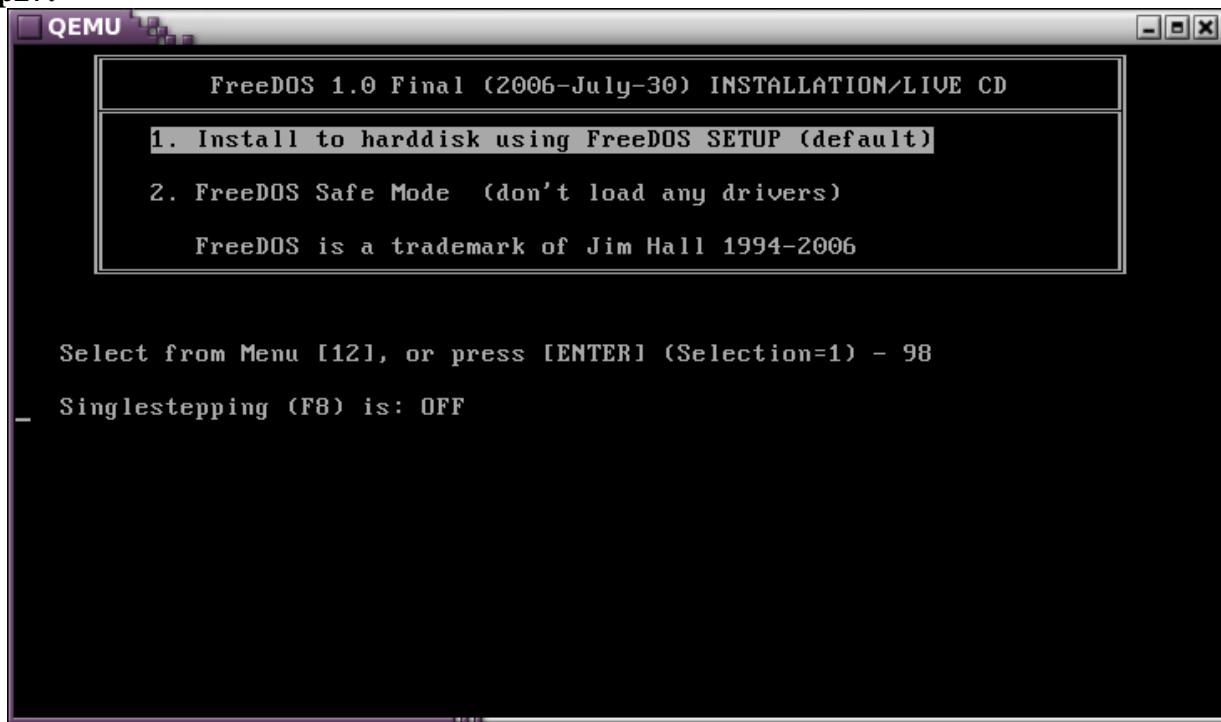


**Step26:**

press 1 now

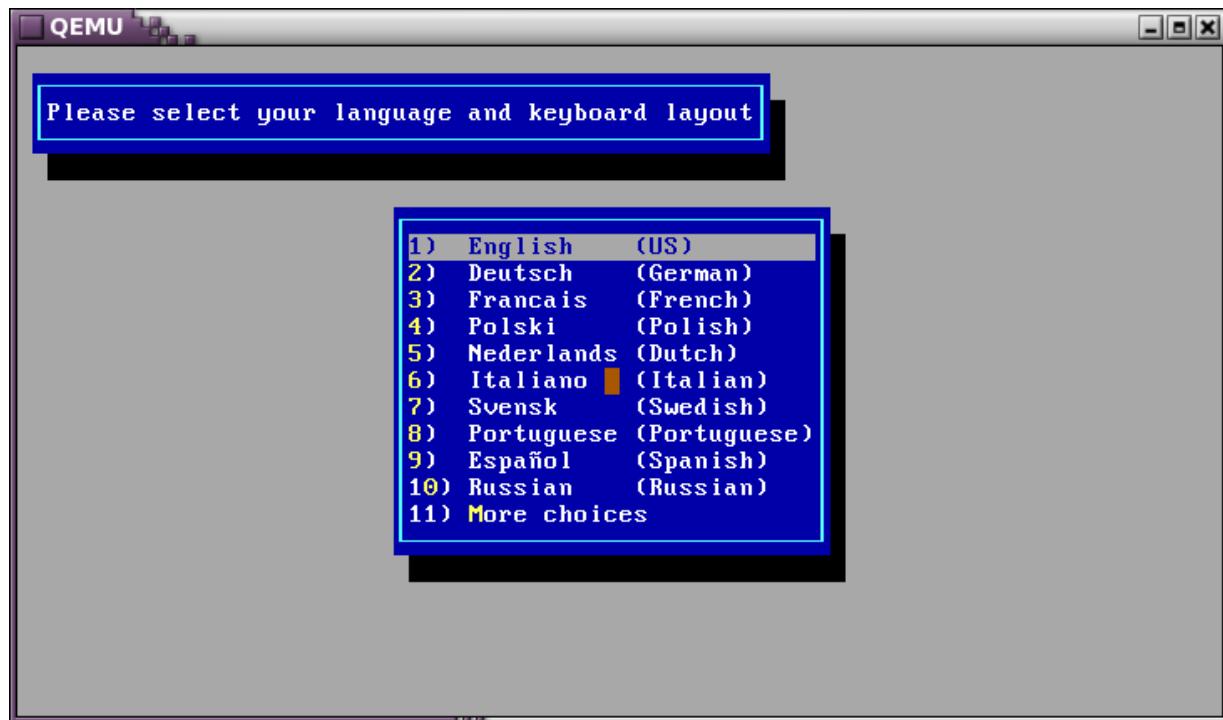


Step27:



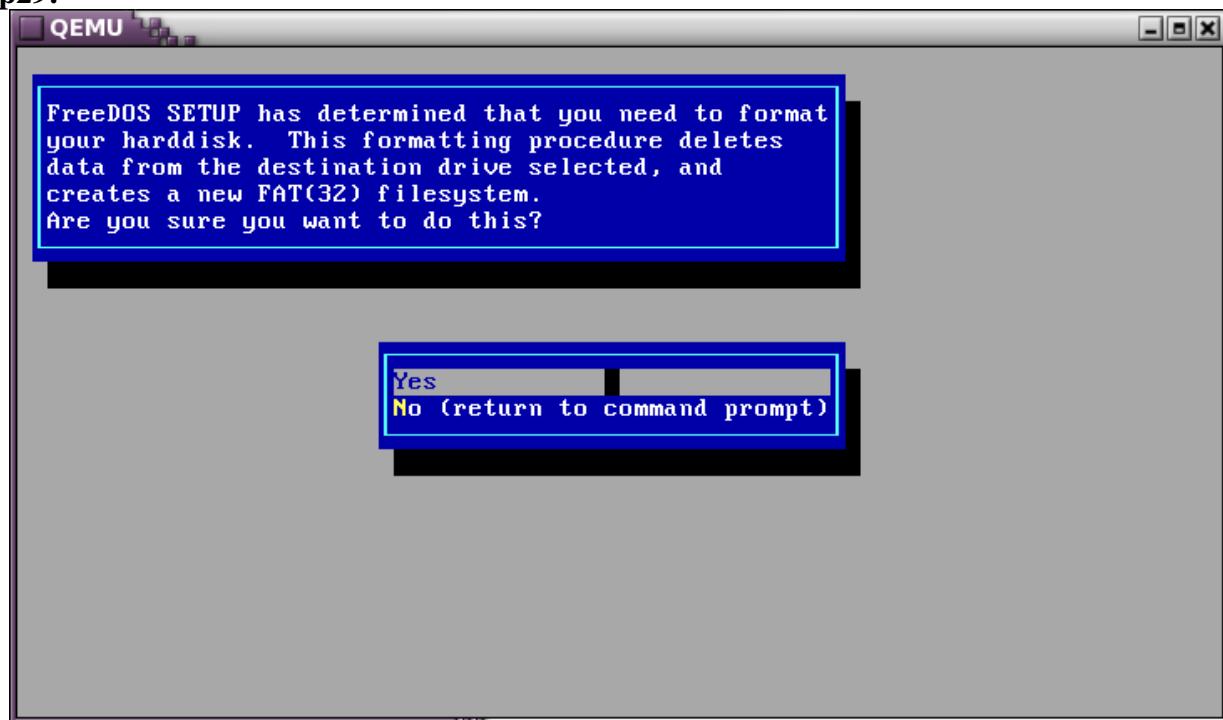
Press Enter

Step28:



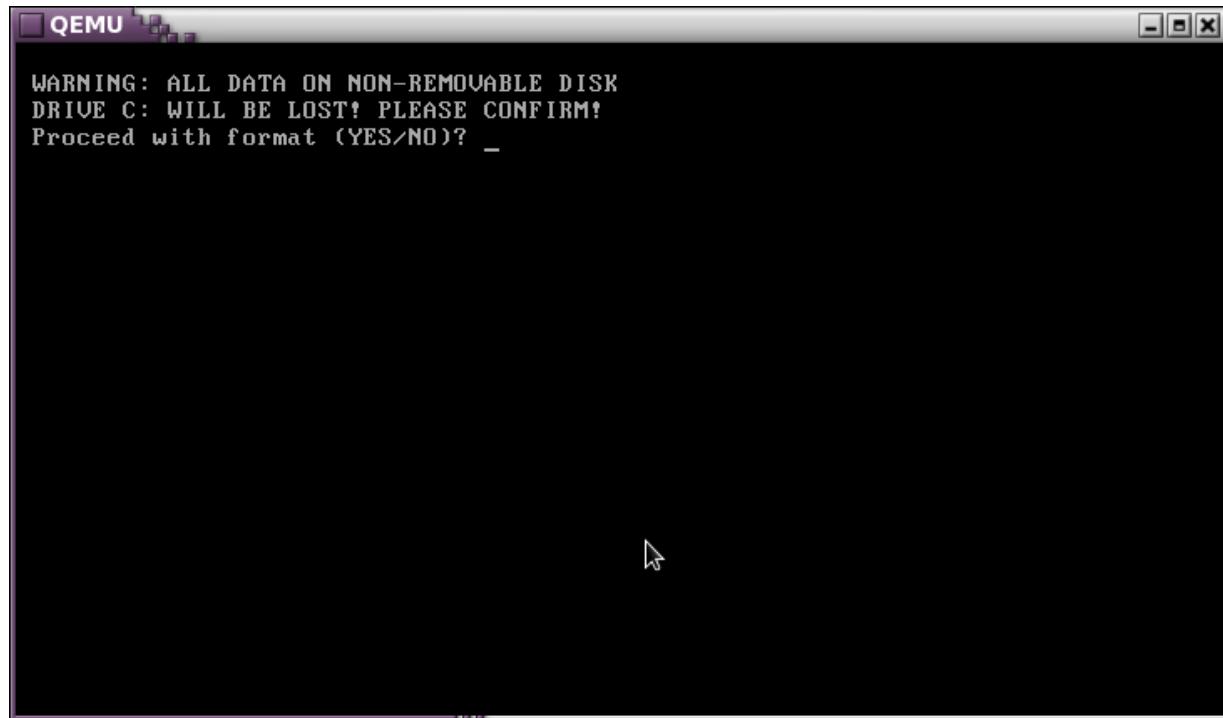
press Enter

Step29:



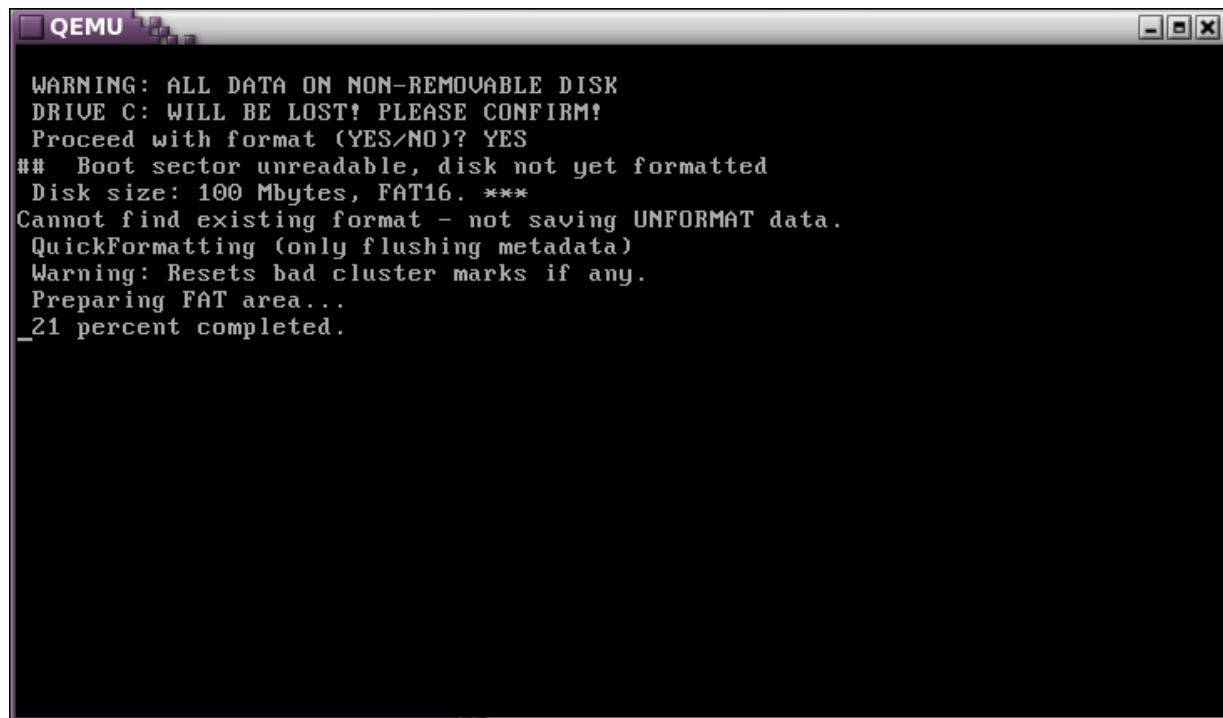
Press Enter

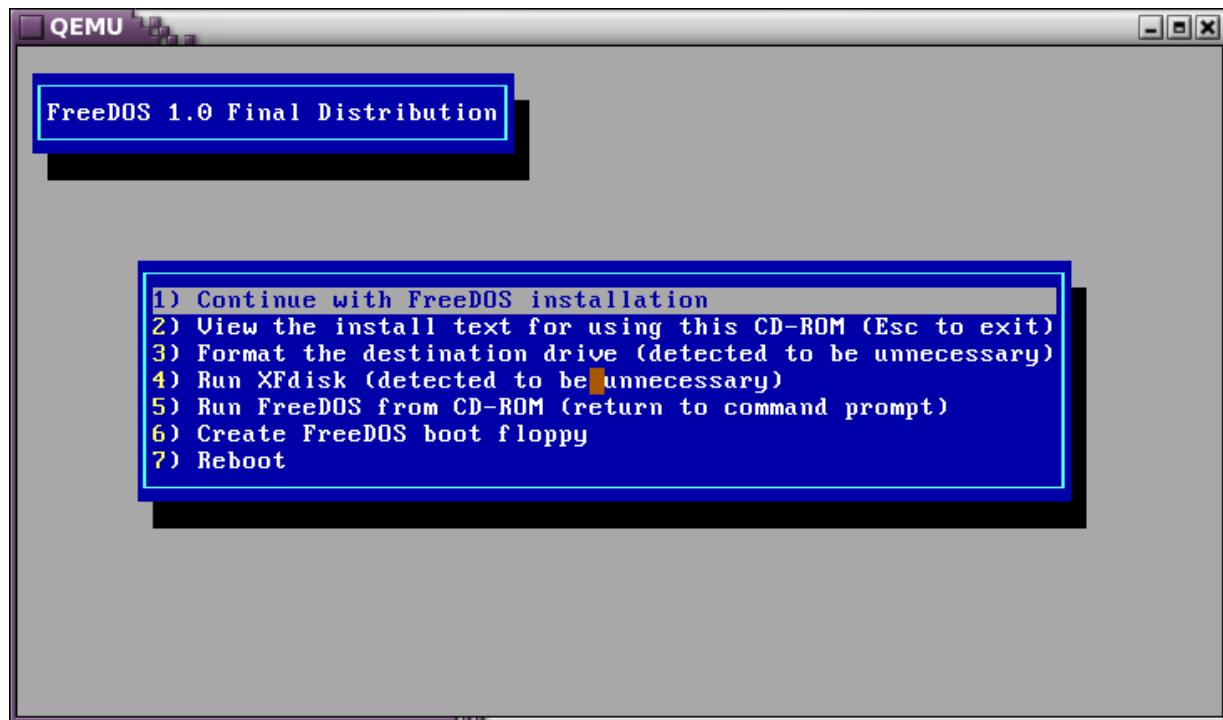
Step30:



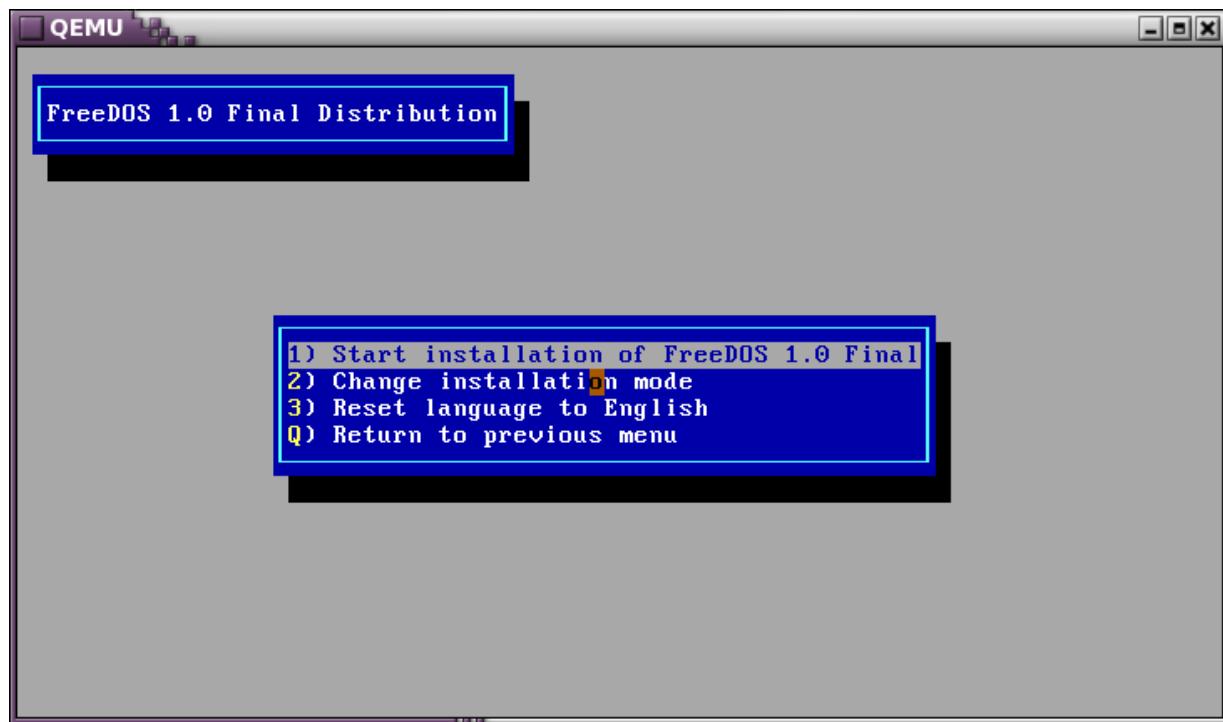
Press YES

Step31:

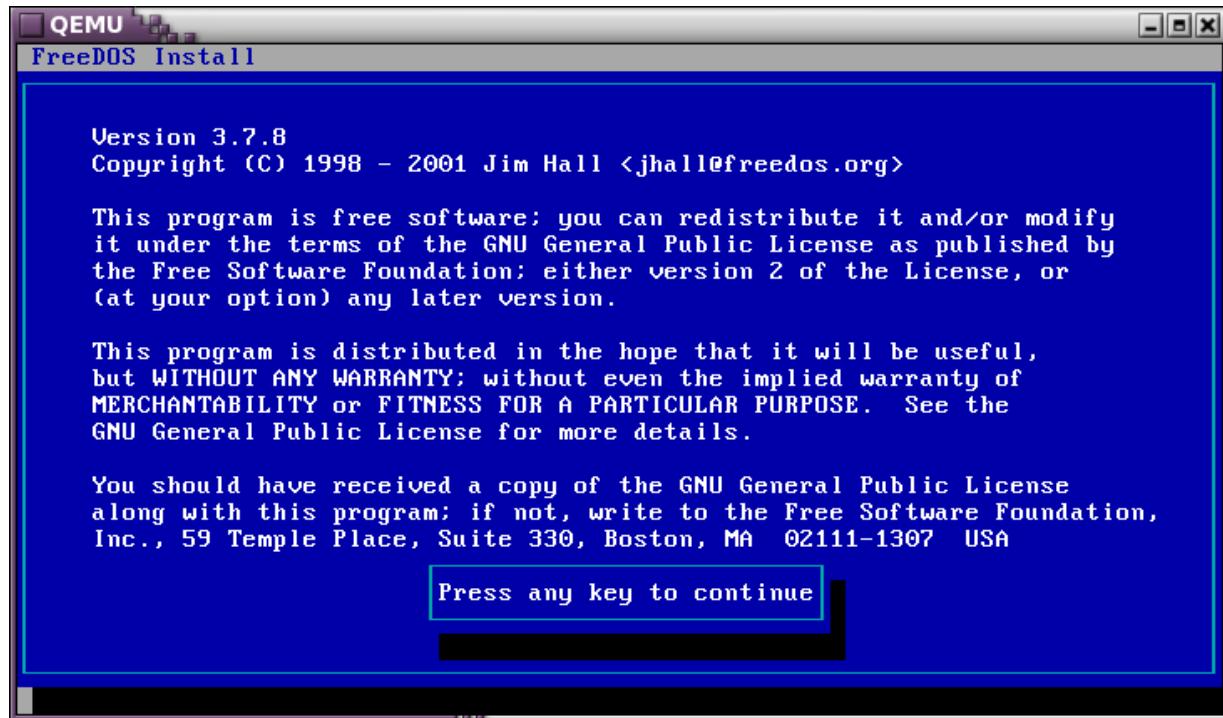




Step32 :Press Enter



Step33: press Enter

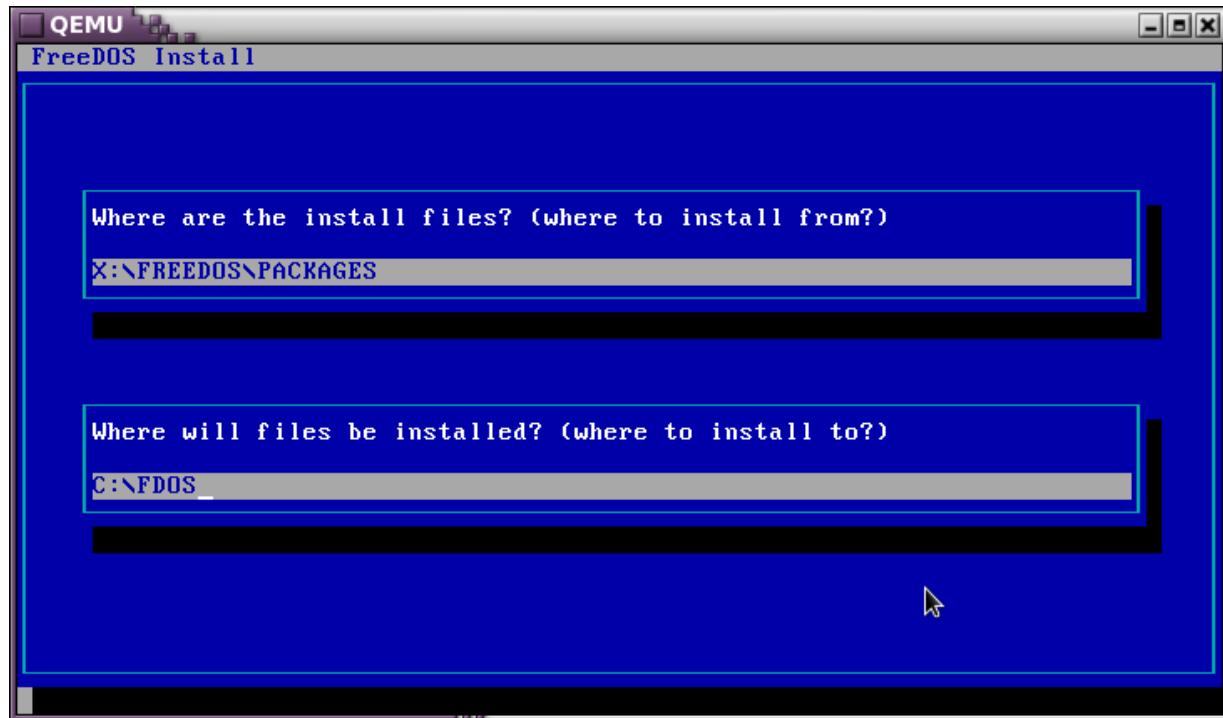


Step34:

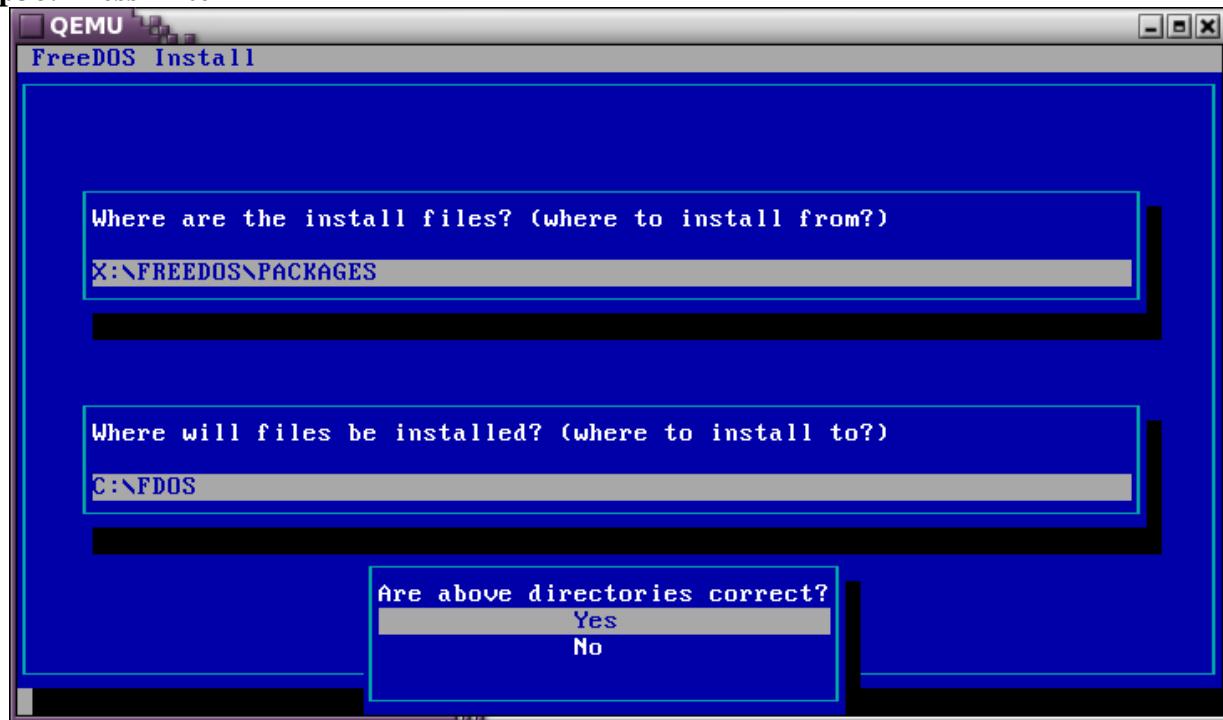
press Enter



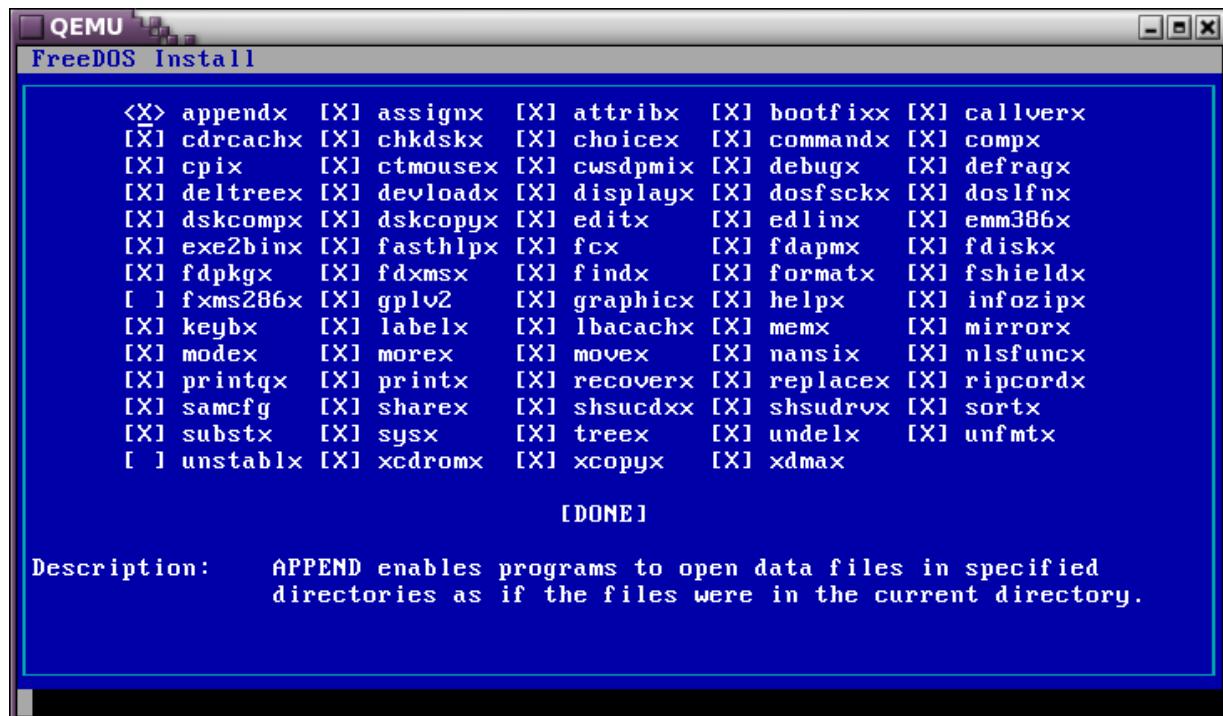
Step35: press Enter



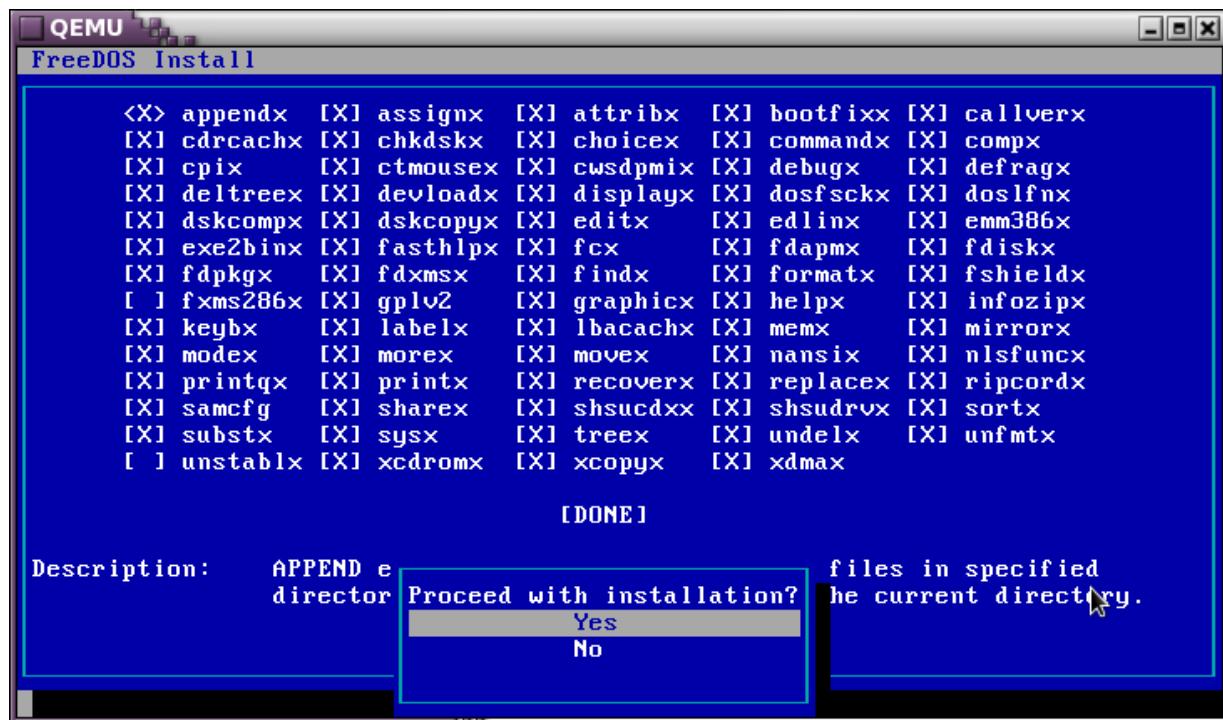
Step36: Press Enter



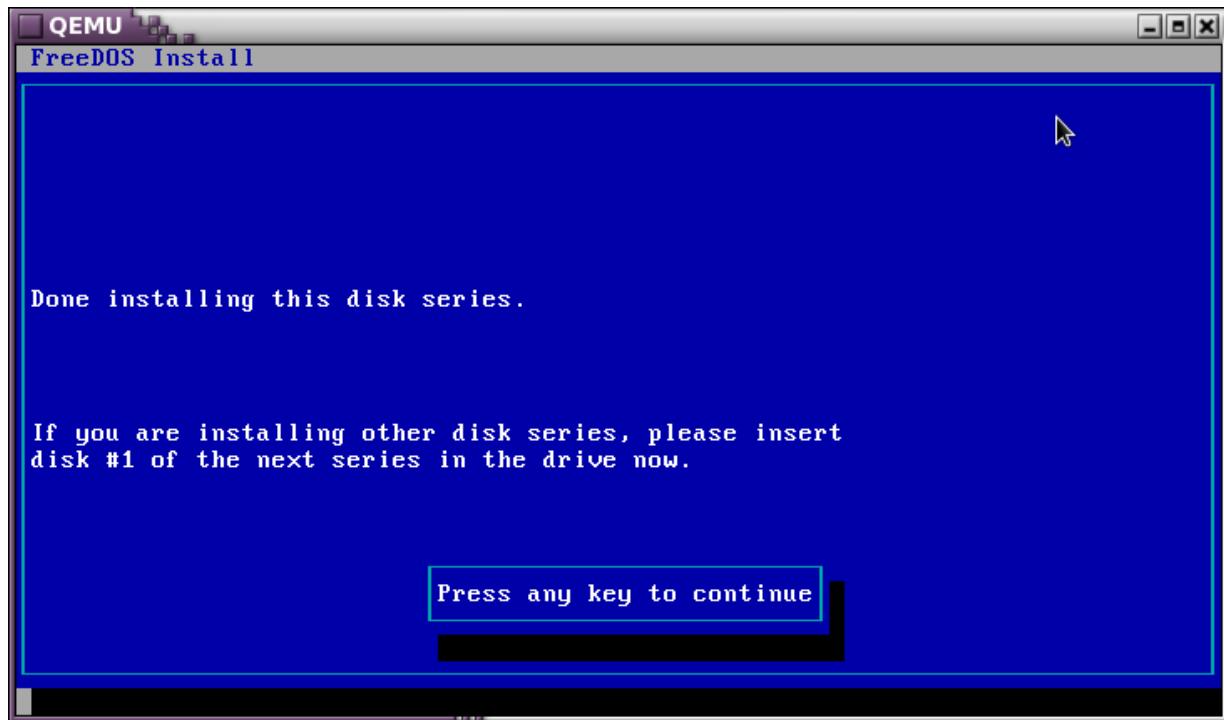
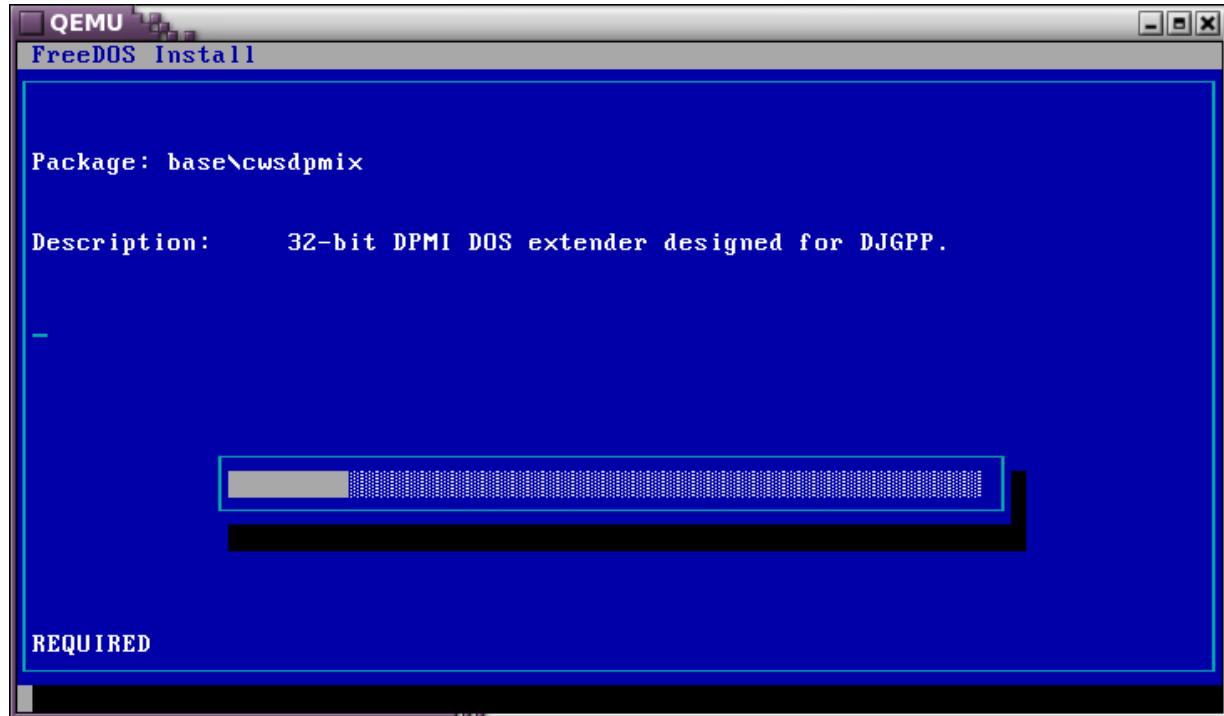
Press 37: Press Enter



Press 38:Press Enter



Press 39:press Enter



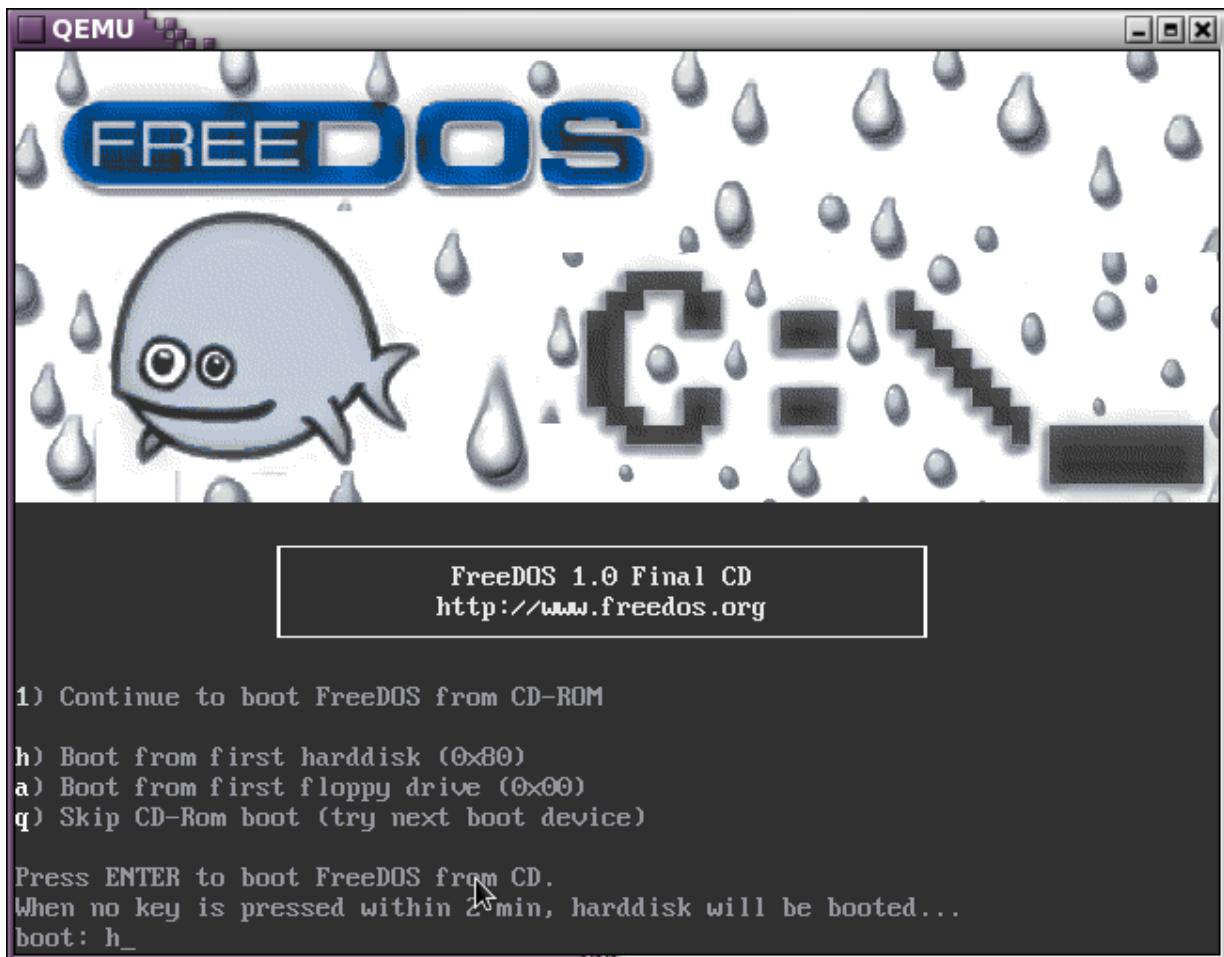
Press 40:press Enter

```
[QEMU] FreeDOS 1.0 Final cdrom distribution.

Performing post installation steps:
[+] Stage 1 : Install FAT32-enabled kernel
[-] Stage 2 : Create startup configuration file
[-] Stage 3 : Create startup automation file
[-] Stage 4 : Setting up translated programs.
[-] Stage 5 : Setup all packages.
[-] Stage 6 : Done, reboot system?

-
```

process Continues



**Press 41 :**Press h

Press Enter when asking Load FreeDOS

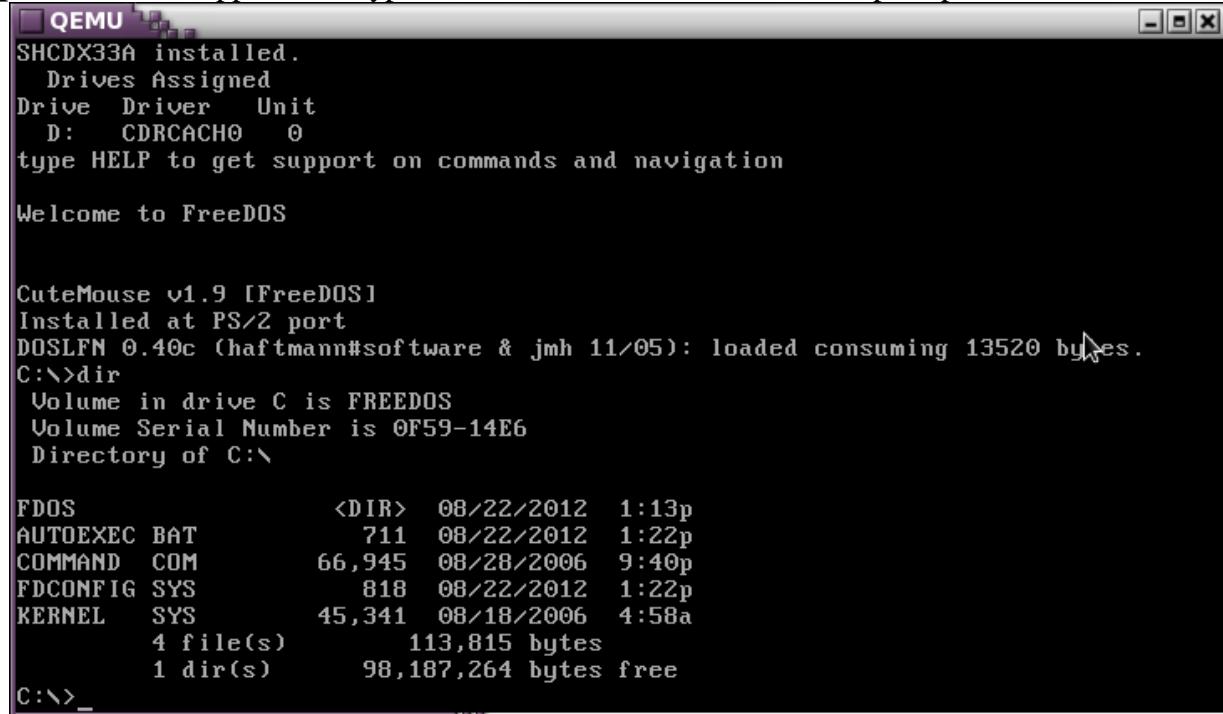
It take few minutes .....

**Step42:** In terminal type **qemu -hda virtualdisk.img -boot c** and Press enter

**Step43:**

Now press option 3 and enter i.e Load FreeDOS including HIMEM XMS-memory driver

**Step44:** Now C:/> appears can type dir command, it will work as DOS prompt.



```
QEMU
SHCDX33A installed.
Drives Assigned
Drive   Driver   Unit
D:     CDRCACH0   0
type HELP to get support on commands and navigation

Welcome to FreeDOS

CuteMouse v1.9 [FreeDOS]
Installed at PS/2 port
DOSLFN 0.40c (haftmann#software & jmh 11/05): loaded consuming 13520 bytes.
C:>dir
Volume in drive C is FREEDOS
Volume Serial Number is 0F59-14E6
Directory of C:\

FDOS          <DIR>  08/22/2012  1:13p
AUTOEXEC.BAT      711  08/22/2012  1:22p
COMMAND.COM      66,945  08/28/2006  9:40p
FDCONFIG.SYS      818  08/22/2012  1:22p
KERNEL.SYS       45,341  08/18/2006  4:58a
               4 file(s)    113,815 bytes
               1 dir(s)    98,187,264 bytes free
C:>_
```

**Result:** Thus the program for virtualization is performed successfully.

**EX.No:3**

## **COMPILING FROM SOURCE**

**DATE:**

**Aim:**

To learn about the common build systems available in Linux and to use them.

### **Introduction:**

Open source software is distributed in source code form. In case of popular software Linux distributions will often have the software packaged in their repositories. If the package is not in the repository the user has to compile the software from source. To do this the user has to understand about the build system used in the project.

The GNU build system, also known as the Autotools, is a suite of programming tools designed to assist in making source-code packages portable to many Unix-like systems. It can be difficult to make a software program portable: the C compiler differs from system to system; certain library functions are missing on some systems; header files may have different names. One way to handle this is write conditional code, with code blocks selected by means of preprocessor directives (#ifdef); but because of the wide variety of build environments this approach quickly becomes unmanageable. The GNU build system is designed to address this problem more manageably.

### **Tools included in the GNU build system**

The GNU build system comprises the GNU utility programs Autoconf, Automake, and Libtool. Other related tools frequently used with the GNU build system are GNU's make program, GNU gettext, pkg-config, and the GNU Compiler Collection, also called GCC.

#### **GNU Autoconf**

Autoconf generates a configure script based on the contents of a configure.ac file which characterizes a particular body of source code. The configure script, when run, scans the build environment and generates a subordinate config.status script which, in turn, converts other input files and most commonly Makefile.in into output files (Makefile) which are appropriate for that build environment. Finally the make program uses Makefile to generate executable programs from source code.

The complexity of the GNU build system reflects the variety of circumstances under which a body of source code may be built. If a source code file is changed then it suffices to re-run make which only re-compiles that part of the body of the source code affected by the change.

If a .in file has changed then it suffices to re-run config.status and make. If the body of source code is copied to another computer then it suffices to re-run configure (which runs config.status) and make. (For this reason source code using the GNU build system is normally distributed without the files that configure generates.)

If the body of source code is changed more fundamentally then configure.ac and the .in files need to be changed and all subsequent steps also followed.

To process files, autoconf uses the GNU implementation of the m4 macro system. Autoconf comes with several auxiliary programs such as Autoheader, which is used to help manage C header files; Autoscans, which can create an initial input file for Autoconf; and ifnames, which can list C pre-processor identifiers used in the program.

### **GNU Automake**

Automake helps to create portable Makefiles, which are in turn processed with the make utility. It takes its input as Makefile.am, and turns it into Makefile.in, which is used by the configure script to generate the file Makefile output.

### **GNU Libtool**

Libtool helps manage the creation of static and dynamic libraries on various Unix-like operating systems. Libtool accomplishes this by abstracting the library-creation process, hiding differences between various systems (e.g. GNU/Linuxsystems vs. Solaris).

### **Gnulib**

Gnulib simplifies the process of making software that uses Autoconf and Automake portable to a wide range of systems.

### **Make**

In software development, make is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to derive the target program. Make can decide where to start through topological sorting. Though integrated development environments and language-specific compiler features can also be used to manage the build process in modern systems, make remains widely used, especially in Unix.

Make is typically used to build executable programs and libraries from source code. Generally though, any process that involves transforming a dependency file to a target result (by executing some number of arbitrary commands) is applicable to make. To cite an example, make could be used to detect a change made to an image file (the dependency) and the target actions that result might be to convert the file to some specific format, copy the result into a content management system, and then send e-mail to a predefined set of users that the above actions were performed.

### **Cmake**

CMake is a unified, cross-platform, open-source build system that enables developers to build, test and package software by specifying build parameters in simple, portable text files. It works in a compiler-independent manner and the build process works in conjunction with native build environments, such as make, Apple's Xcode and Microsoft Visual Studio. It also has minimal dependencies, C++ only.

CMake is open source software. CMake can:

- Create libraries
- Generate wrappers
- Compile source code

- Build executable in arbitrary combination

### **Apache Ant**

Apache Ant is a software tool for automating software build processes. It is similar to Make but is implemented using the Java language, requires the Java platform, and is best suited to building Java projects. The most immediately noticeable difference between Ant and Make is that Ant uses XML to describe the build process and its dependencies, whereas Make uses Makefile format. By default the XML file is named build.xml. Ant is an Apache project. It is open source software, and is released under the Apache Software License.

#### **Pre-requisites:**

To ensure that all tools required are installed.

**Type the following commands in terminal and type the password for root user, when prompted.**

#### **STEP:1**

```
[fosslab@fosslab~]$ su
```

Password:(admin123)

#### **STEP 2:**

```
[root@ fosslab fosslab]# rpm -qa cmake
```

cmake-2.8.2-2.fc14.i686

#### **STEP:3**

```
fosslab fosslab]# rpm -qa ant
```

ant-1.7.1-13.fc13.i686

#### **STEP:4**

```
[root@ fosslab fosslab]# rpm -qa java-1.6.0-openjdk-devel
```

java-1.6.0-openjdk-devel-1.6.0.0-44.1.9.1.fc14.i686

#### **STEP:5**

```
[root@ fosslab fosslab]# exit
```

#### **Exercises:**

Create a directory for all the programs in the exercise.

#### **STEP:6**

```
[fosslab@fosslab ~]$ mkdir build_systems
```

#### **STEP:7**

```
[fosslab@fosslab ~]$ cd build_systems
```

### **1. Make**

We shall be using a simple program written in C and write a makefile to compile the program.

### **STEP:8**

```
[fosslab@fosslab build_systems]$ mkdir gnumake
```

### **STEP:9**

```
[fosslab@fosslab build_systems]$ cd gnumake
```

### **STEP:10**

```
[fosslab@fosslab gnumake]$ gedit squareroot.c
```

Type and save the following simple program for square root of a number

```
// A simple program that computes the square root of a number

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main (int argc, char *argv[])

{

if (argc < 2)

{

fprintf(stdout,"Usage: %s number\n",argv[0]);

return 1;

}

double inputValue = atof(argv[1]);

double outputValue = sqrt(inputValue);

fprintf(stdout,"The square root of %g is %g\n",inputValue, outputValue);
```

```
return 0;
```

```
}
```

**Close gedit and test it by compiling it once:**

**STEP:11**

```
[fosslab@fosslab gnumake]$ gcc squareroot.c -o squareroot -lm
```

**STEP:12**

```
[fosslab@fosslab gnumake]$ ./squareroot 25
```

The square root of 25 is 5

**Write a simple makefile to compile the program.**

**STEP:13**

```
[fosslab@fosslab gnumake]$ gedit Makefile
```

Type and save the following code

```
# Commands start with TAB not spaces
```

```
CC= gcc
```

```
CFLAGS= -g
```

```
LDFLAGS = -lm
```

```
all: squareroot
```

```
squareroot: squareroot.o
```

```
squareroot.o: squareroot.c
```

```
clean:
```

```
    rm -f squareroot squareroot.o
```

**Close gedit and test the Makefile**

**STEP:14**

```
[fosslab@fosslab gnumake]$ make
```

make: Nothing to be done for `all'.

**STEP:15**

[fosslab@fosslab gnumake]\$ make clean

**STEP:16**

[fosslab@fosslab gnumake]\$ make

gcc -g -c -o squareroot.o squareroot.c

gcc -lm squareroot.o -o squareroot

**STEP:17**

[fosslab@fosslab gnumake] ./squareroot 25

The square root of 25 is 5

**Close gedit and test the Makefile**

**STEP:18**

[fosslab@fosslab gnumake]\$ make

make: Nothing to be done for `all'.

**STEP:19**

[fosslab@fosslab gnumake]\$ make clean

**STEP:20**

[fosslab@fosslab gnumake]\$ make

gcc -g -c -o squareroot.o squareroot.c

gcc -lm squareroot.o -o squareroot

**STEP:21**

[fosslab@fosslab gnumake] ./squareroot 25

The square root of 25 is 5

**2. Cmake**

**Write a simple script for CMake to compile the previously written program.**

**Create a new directory and copy the source code to it.**

**STEP:22**

```
[fosslab@fosslab gnumake] mkdir cmake
```

**STEP:23**

```
[fosslab@fosslab gnumake] cp squareroot.c /home/fosslab/build_systems/gnumake/cmake/
```

**Create configuration files for Cmake.**

**STEP:24**

```
[fosslab@fosslab gnumake] cd cmake
```

**STEP:25**

```
[fosslab@fosslab cmake]$ gedit CmakeLists.txt
```

**Type and save the following code**

```
cmake_minimum_required(VERSION 2.6)
project(squareroot)
add_executable(squareroot squareroot.c)
TARGET_LINK_LIBRARIES(squareroot m)
```

CMake is commonly used with out of source builds ie, we build the program in a directory separate from the source. We use the generated makefile to compile the program.

**STEP:26**

```
[fosslab@fosslab cmake]$ mkdir build
```

**STEP:27**

```
[fosslab@fosslab cmake]$ cd build
```

**STEP:28**

```
[fosslab@fosslab build]$ cmake ..
```

**STEP:29**

```
[fosslab@fosslab build]$ make
```

**STEP:30**

```
[fosslab@fosslab build]$ ./squareroot 25
```

The square root of 25 is 5

**3. Apache Ant**

Create a new directory for the ant exercise. (open a new terminal )

**STEP:31**

```
[fosslab@fosslab ~]$ cd build_systems
```

**STEP:32**

```
[fosslab@fosslab build_systems]$ mkdir ant
```

**STEP:33**

```
[fosslab@fosslab build_systems]$ cd ant
```

**STEP:34**

```
[fosslab@fosslab ant]$ mkdir -p src/hello
```

**STEP:35**

```
[fosslab@fosslab ant]$ gedit src/hello/HelloWorld.java
```

**Type and save the following code**

```
package hello;  
  
public class HelloWorld  
{  
  
    public static void main(String[] args)  
    {  
  
        System.out.println("Hello World");  
  
    }  
}
```

}

**STEP:36**

[fosslab@fosslab ant]\$ mkdir -p build/classes

**STEP:37**

[fosslab@fosslab ant]\$ javac -sourcepath src -d build/classes/ src/hello/HelloWorld.java

**STEP:38**

[fosslab@fosslab ant]\$ java -cp build/classes hello.HelloWorld

Hello World

**STEP:39**

[fosslab@fosslab ant]\$

**Write the ant build script.**

[fosslab@fosslab ant]\$ gedit build.xml

**Type the following code and save**

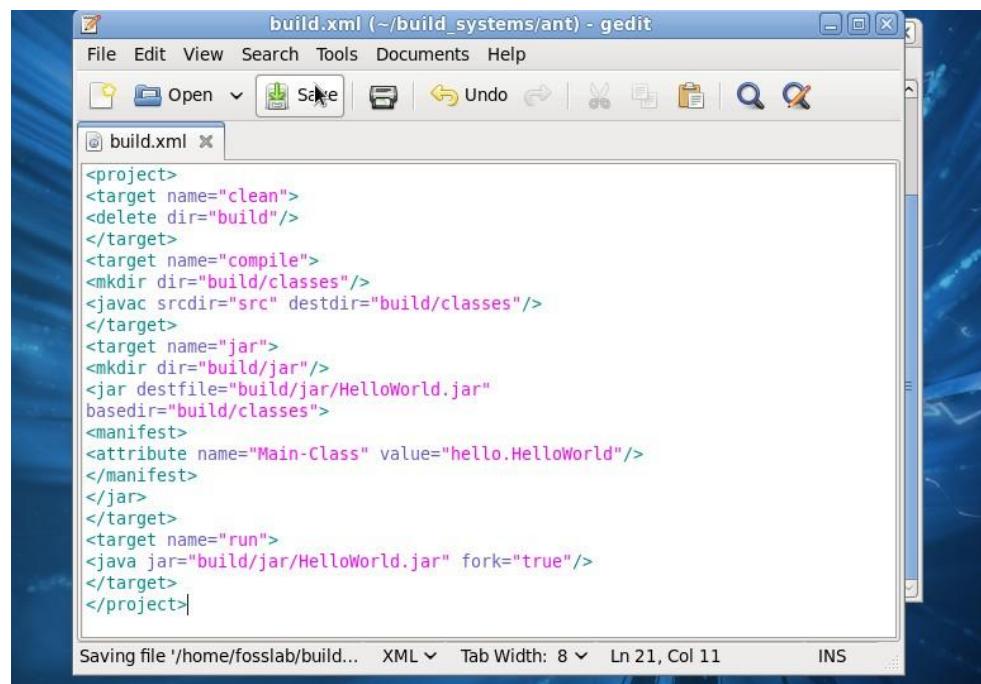
```
<project>

    <target name="clean">
        <delete dir="build"/>
    </target>

    <target name="compile">
        <mkdir dir="build/classes"/>
        <javac srcdir="src" destdir="build/classes"/>
    </target>

    <target name="jar">
        <mkdir dir="build/jar"/>
        <jar destfile="build/jar/HelloWorld.jar"
            basedir="build/classes">
            <manifest>
```

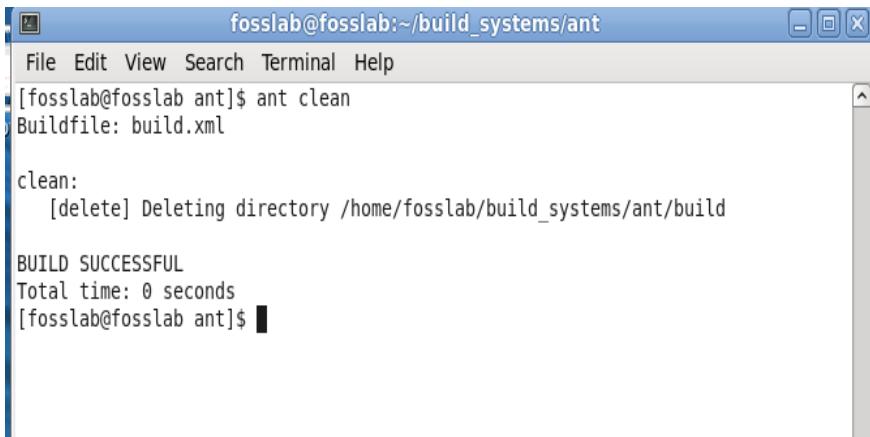
```
<attribute name="Main-Class" value="hello.HelloWorld"/>  
</manifest>  
</jar>  
</target>  
<target name="run">  
<java jar="build/jar/HelloWorld.jar" fork="true"/>  
</target>  
</project>
```



Now the project can be compile and run using ant.

**STEP:40**

```
[fosslab@fosslab ant]$ ant clean
```



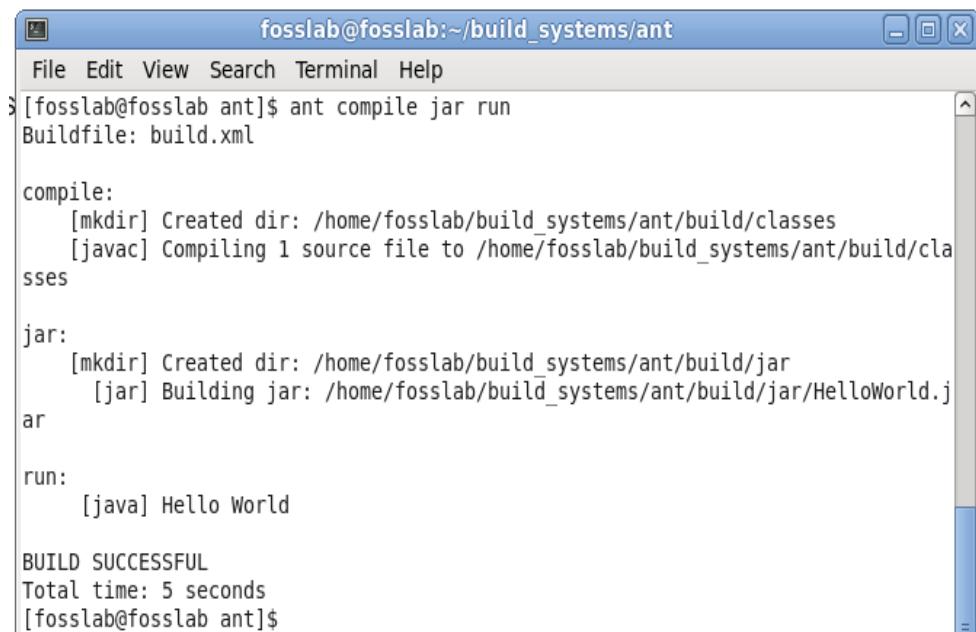
```
fosslab@fosslab:~/build_systems/ant
File Edit View Search Terminal Help
[fosslab@fosslab ant]$ ant clean
Buildfile: build.xml

clean:
[delete] Deleting directory /home/fosslab/build_systems/ant/build

BUILD SUCCESSFUL
Total time: 0 seconds
[fosslab@fosslab ant]$
```

**STEP:41**

[fosslab@fosslab ant]\$ ant compile jar run



```
fosslab@fosslab:~/build_systems/ant
File Edit View Search Terminal Help
> [fosslab@fosslab ant]$ ant compile jar run
Buildfile: build.xml

compile:
[mkdir] Created dir: /home/fosslab/build_systems/ant/build/classes
[javac] Compiling 1 source file to /home/fosslab/build_systems/ant/build/classes

jar:
[mkdir] Created dir: /home/fosslab/build_systems/ant/build/jar
[jar] Building jar: /home/fosslab/build_systems/ant/build/jar/HelloWorld.jar

run:
[java] Hello World

BUILD SUCCESSFUL
Total time: 5 seconds
[fosslab@fosslab ant]$
```

**GNU Autotools**

Copy the file hello-2.7.tar.gz to the buildsystems project directory and uncompress it

**STEP:42**

[fosslab@fosslab ~]\$ cd Downloads

**STEP:43**

[fosslab@fosslab Downloads]\$ mv hello-2.7.tar.gz /home/fosslab/build\_systems

**STEP:44**

[fosslab@fosslab Downloads]\$ cd /home/fosslab/build\_systems

**STEP:45**

```
[fosslab@fosslab build_systems]$ tar -xzf hello-2.7.tar.gz
```

**STEP:46**

```
[fosslab@fosslab build_systems]$ cd hello-2.7
```

**STEP:47**

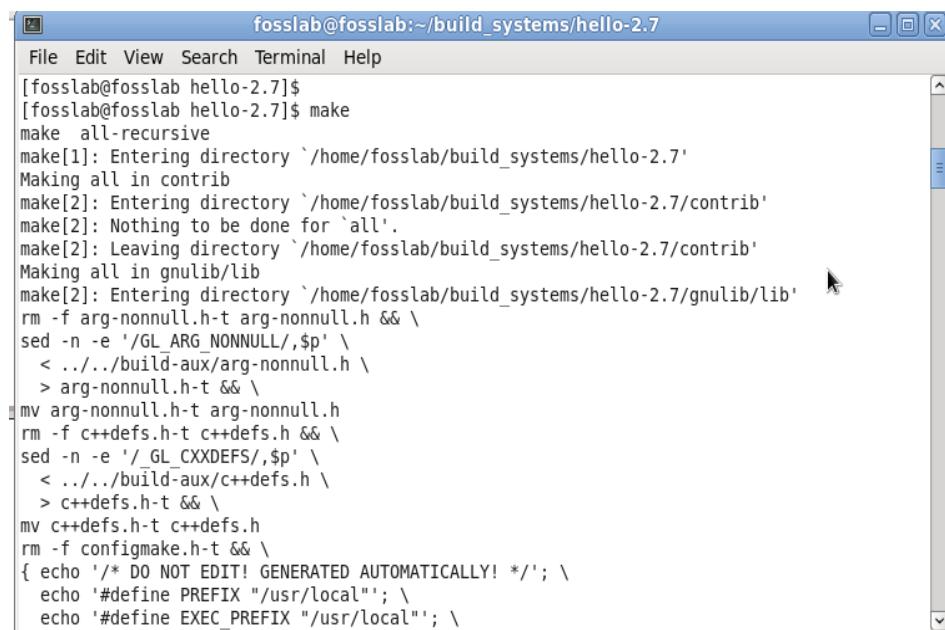
```
[fosslab@fosslab hello-2.7]$
```

**STEP:48**

```
[fosslab@fosslab hello-2.7]$ ./configure
```

**STEP:49**

```
[fosslab@fosslab hello-2.7]$ make
```

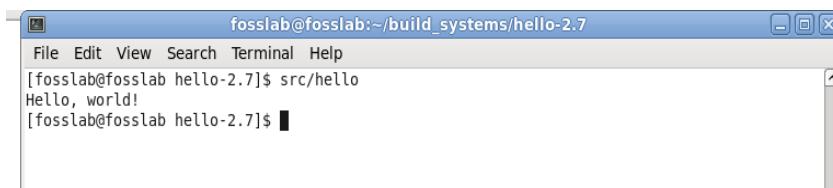


A screenshot of a terminal window titled "fosslab@fosslab:~/build\_systems/hello-2.7". The window shows the command "make" being run, which triggers a recursive build process. The output includes several "make[1]" and "make[2]" messages indicating directory entries and file manipulations like "rm", "mv", and "sed" operations on files such as "arg-nonnull.h", "c++defs.h", and "configmake.h".

```
fosslab@fosslab:~/build_systems/hello-2.7
File Edit View Search Terminal Help
[fosslab@fosslab hello-2.7]$
[fosslab@fosslab hello-2.7]$ make
make all-recursive
make[1]: Entering directory `/home/fosslab/build_systems/hello-2.7'
Making all in contrib
make[2]: Entering directory `/home/fosslab/build_systems/hello-2.7/contrib'
make[2]: Nothing to be done for `all'.
make[2]: Leaving directory `/home/fosslab/build_systems/hello-2.7/contrib'
Making all in gnulib/lib
make[2]: Entering directory `/home/fosslab/build_systems/hello-2.7/gnulib/lib'
rm -f arg-nonnull.h-t arg-nonnull.h && \
sed -n -e '/GL_ARG_NONNULL/,${p}' \
< ../../build-aux/arg-nonnull.h \
> arg-nonnull.h-t && \
mv arg-nonnull.h-t arg-nonnull.h
rm -f c++defs.h-t c++defs.h && \
sed -n -e '/GL_CXXDEFS/,${p}' \
< ../../build-aux/c++defs.h \
> c++defs.h-t && \
mv c++defs.h-t c++defs.h
rm -f configmake.h-t && \
{ echo /* DO NOT EDIT! GENERATED AUTOMATICALLY! */; \
echo '#define PREFIX "/usr/local"'; \
echo '#define EXEC_PREFIX "/usr/local"'; \
```

**STEP:49**

```
[fosslab@fosslab hello-2.7]$ src/hello
```



A screenshot of a terminal window titled "fosslab@fosslab:~/build\_systems/hello-2.7". The window shows the command "src/hello" being run, which outputs "Hello, world!" to the screen.

```
fosslab@fosslab:~/build_systems/hello-2.7
File Edit View Search Terminal Help
[fosslab@fosslab hello-2.7]$
[fosslab@fosslab hello-2.7]$ src/hello
Hello, world!
[fosslab@fosslab hello-2.7]$
```

The program will now reside in the src directory. To install the program log in as root.

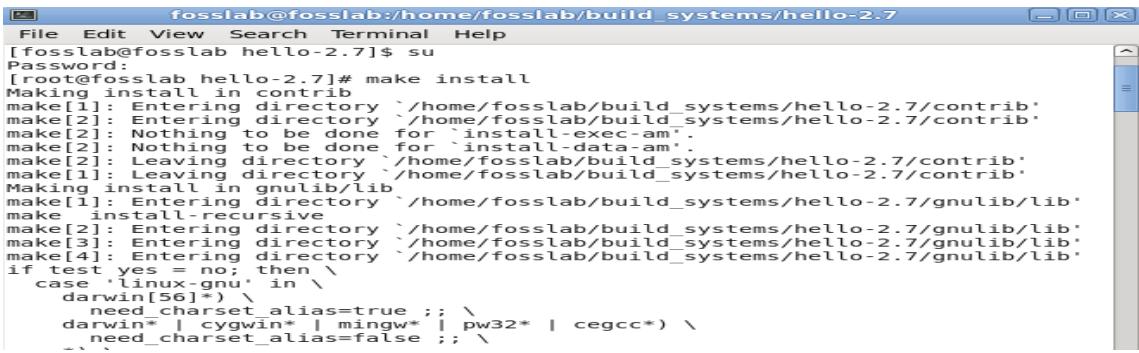
**STEP:50**

```
[fosslab@fosslab hello-2.7]$ su
```

Password:(admin123)

**STEP:51**

```
[root@fosslab hello-2.7]# make install
```



```
fosslab@fosslab:/home/fosslab/build_systems/hello-2.7
File Edit View Search Terminal Help
[fosslab@fosslab hello-2.7]$ su
Password:
[root@fosslab hello-2.7]# make install
Making install in contrib
make[1]: Entering directory `/home/fosslab/build_systems/hello-2.7/contrib'
make[2]: Entering directory `/home/fosslab/build_systems/hello-2.7/contrib'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/fosslab/build_systems/hello-2.7/contrib'
make[1]: Leaving directory `/home/fosslab/build_systems/hello-2.7/contrib'
Making install in gnulib/lib
make[1]: Entering directory `/home/fosslab/build_systems/hello-2.7/gnulib/lib'
make[2]: Entering directory `/home/fosslab/build_systems/hello-2.7/gnulib/lib'
make[3]: Entering directory `/home/fosslab/build_systems/hello-2.7/gnulib/lib'
make[4]: Entering directory `/home/fosslab/build_systems/hello-2.7/gnulib/lib'
if test yes = no; then \
  case 'linux-gnu' in \
    darwin[56]* ) \
      need_charset_alias=true ;; \
    darwin* | cygwin* | mingw* | pw32* | cegcc*) \
      need_charset_alias=false ;; \
    * \
  esac
fi
[root@fosslab hello-2.7]#
```

### STEP:52

```
[root@fosslab hello-2.7]# exit
```

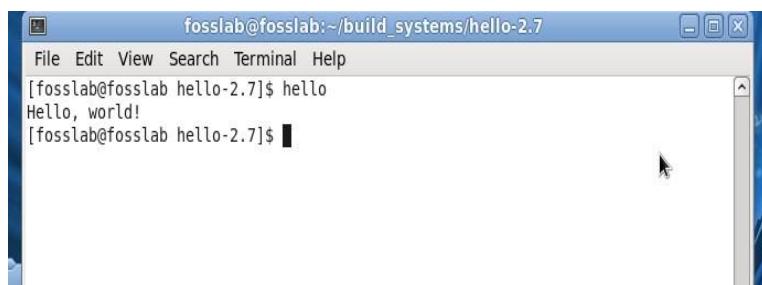
```
exit
```

Now the program can be run from anywhere.

### STEP:53

```
[fosslab@fosslab hello-2.7]$ hello
```

Hello, world!



```
fosslab@fosslab:~/build_systems/hello-2.7
File Edit View Search Terminal Help
[fosslab@fosslab hello-2.7]$ hello
Hello, world!
[fosslab@fosslab hello-2.7]$
```

### Result:

Thus the program for compiling from source is performed successfully.

**EX:NO: 4**

**DATE :**

## **INTRODUCTION TO PACKET MANAGEMENT SYSTEM**

**Aim :**

**Given a set of RPM or DEB, to build and maintain, serve packages over http or ftp. and also to configure client systems to access the package repository.**

**Procedure:**

**Step1:** Open Firefox and type the following in url

<http://192.168.105.254/cs2406/Software> Tools Required for Lab Exercises (CS-2406,CS-77)Qmail Packages (ex-4)/

Now Click qmail-packages.zip and save it

**Step2:** Now goto places menu --> Downloads and choose qmail-packages.zip and Right Click Mouse and press Extract here

**Step3:** Now open Command line terminal

[fosslab@fosslab ~]\$ **pwd**

/home/fosslab

**Step4:** [fosslab@fosslab ~]\$ **cd Downloads/**

Step5: [fosslab@fosslab Downloads]\$ **mv qmail-packages qmail**

**Step6:**

[fosslab@fosslab Downloads]\$ **ls -l**

total 156032

-rw-rw-r-- 1 fosslab fosslab 344011 Jun 28 01:50 04524284.pdf

-rw-rw-r-- 1 fosslab fosslab 251225 Jun 28 01:36 91-US-31-1\_Cloud\_Computing.pdf

-rw-rw-r-- 1 fosslab fosslab 556032 Jun 28 01:41 ABSTRACT and pro\_vidhya.doc

-rw-rw-r-- 1 fosslab fosslab 429466 Jun 28 01:37 computing-whitepaper.pdf

-rw-rw-r-- 1 fosslab fosslab 99019 Aug 1 01:03 Criterion-8.docx

-rw-rw-r-- 1 fosslab fosslab 110194 Aug 1 01:03 Criterion 9(2).docx

-rw-rw-r-- 1 fosslab fosslab 110194 Aug 1 01:03 Criterion 9.docx

-rw-rw-r-- 1 fosslab fosslab 512000 Aug 1 01:03 criter XP12-chandran.doc

-rw-rw-r-- 1 fosslab fosslab 422400 Aug 1 01:03 criter XP12\_Mech.doc

-rwxrwxr-x 1 fosslab fosslab 493564 Jun 7 2011 Downloads.exe

```
-rw-rw-r-- 1 fosslab fosslab 634100 Jun 28 00:13 EJSR_64_2_05.pdf
-rw-rw-r-- 1 fosslab fosslab 237418 Jun 28 00:13 EJSR_64_2_14.pdf
-rw-rw-r-- 1 fosslab fosslab 54227 Jun 28 01:39 EJSR_74_3_04.pdf
-rw-rw-r-- 1 fosslab fosslab 527523 Jun 28 01:44 EJSR_77_1_06.pdf
-rw-rw-r-- 1 fosslab fosslab 147175 Jul 9 01:55 foss-lab-manual-p1-1.0-rc1.pdf
-rw-rw-r-- 1 fosslab fosslab 20228 Jun 28 01:50 Gartner Data Mining Addtl.pdf
-rw-rw-r-- 1 fosslab fosslab 12253 Aug 11 14:02 HP-LaserJet-laserjet.ppd
-rw-rw-r-- 1 fosslab fosslab 219237 Jun 28 00:10 kdd98_elder_abbott_nopics_bw.pdf
drwxrwxr-x 24 fosslab fosslab 4096 Aug 17 11:48 linux-2.6.35.7
-rw-rw-r-- 1 fosslab fosslab 88323744 Aug 16 14:27 linux-2.6.35.7.tar.gz
-rw-rw-r-- 1 fosslab fosslab 60416 Jul 14 04:56 newFOC LP.doc
-rw-rw-r-- 1 fosslab fosslab 43520 Jun 28 01:40 ProjectTitles.doc
drwxr-xr-x 2 fosslab fosslab 4096 Aug 22 10:04 qmail
-rw-rw-r-- 1 fosslab fosslab 64195115 Aug 22 09:57 qmail-packages.zip
-rw-rw-r-- 1 fosslab fosslab 180964 Jun 28 01:57 sensor-route-security.pdf
-rw-rw-r-- 1 fosslab fosslab 56320 Jul 18 01:55 ssLABabet outcome.doc
-rw-rw-r-- 1 fosslab fosslab 142336 Jul 18 01:53 ss Lesson-plan.doc
-rw-rw-r-- 1 fosslab fosslab 14540 Jul 30 04:55 st_newmark3(2).jsp
-rw-rw-r-- 1 fosslab fosslab 14702 Jul 30 04:45 st_newmark3.jsp
-rw-rw-r-- 1 fosslab fosslab 1517376 Jul 28 02:36 wrar420.exe
```

**Step 7:** [fosslab@fosslab Downloads]\$ **cd qmail**

Step8: [fosslab@fosslab qmail]\$ pwd

/home/fosslab/Downloads/qmail

//This directory has 25 rpm packages

**Step9:** [fosslab@fosslab qmail]\$ **ls**

autorespond-toaster-2.0.4-1.3.6.i686.rpm  
clamav-toaster-0.96.1-1.3.36.i686.rpm  
control-panel-toaster-0.5-1.3.7.noarch.rpm  
courier-authlib-toaster-0.59.2-1.3.10.i686.rpm  
courier-imap-toaster-4.1.2-1.3.10.i686.rpm  
daemontools-toaster-0.76-1.3.6.i686.rpm  
ezmlm-cgi-toaster-0.53.324-1.3.6.i686.rpm  
ezmlm-toaster-0.53.324-1.3.6.i686.rpm  
isoqllog-toaster-2.1-1.3.7.i686.rpm  
libdomainkeys-toaster-0.68-1.3.6.i686.rpm  
libsrs2-toaster-1.0.18-1.3.6.i686.rpm  
maildrop-toaster-2.0.3-1.3.8.i686.rpm  
maildrop-toaster-devel-2.0.3-1.3.8.i686.rpm  
qmailadmin-toaster-1.2.12-1.3.8.i686.rpm  
qmailmrtg-toaster-4.2-1.3.6.i686.rpm  
qmail-pop3d-toaster-1.03-1.3.20.i686.rpm  
qmail-toaster-1.03-1.3.20.i686.rpm  
ripmime-toaster-1.4.0.6-1.3.6.i686.rpm  
send-emails-toaster-0.5-1.3.7.noarch.rpm  
simscan-toaster-1.4.0-1.3.8.i686.rpm  
spamassassin-toaster-3.2.5-1.3.17.i686.rpm  
squirrelmail-toaster-1.4.20-1.3.17.noarch.rpm  
ucspi-tcp-toaster-0.88-1.3.9.i686.rpm  
vpopmail-toaster-5.4.17-1.3.7.i686.rpm

vqadmin-toaster-2.3.4-1.3.6.i686.rpm

**Step10:** [fosslab@fosslab qmail]\$ **ls \*.rpm |wc**

25 25 1003

**Step11:** [fosslab@fosslab qmail]\$ **cd ..**

[fosslab@fosslab Downloads]\$

**Step12:** [fosslab@fosslab Downloads]\$

**pwd**

[fosslab@fosslab Downloads]\$

**ls -l**

**Step13:** [fosslab@fosslab Downloads]\$ **createrepo qmail/**

25/25 - vpopmail-toaster-5.4.17-1.3.7.i686.rpm

Saving Primary metadata

Saving file lists metadata

Saving other metadata

**Step14:** [fosslab@fosslab Downloads]\$ cd qmail

**Step15:** [fosslab@fosslab qmail]\$ ls

autorespond-toaster-2.0.4-1.3.6.i686.rpm

clamav-toaster-0.96.1-1.3.36.i686.rpm

control-panel-toaster-0.5-1.3.7.noarch.rpm

courier-authlib-toaster-0.59.2-1.3.10.i686.rpm

courier-imap-toaster-4.1.2-1.3.10.i686.rpm

daemontools-toaster-0.76-1.3.6.i686.rpm

ezmlm-cgi-toaster-0.53.324-1.3.6.i686.rpm

ezmlm-toaster-0.53.324-1.3.6.i686.rpm

isoqllog-toaster-2.1-1.3.7.i686.rpm

libdomainkeys-toaster-0.68-1.3.6.i686.rpm

libsrs2-toaster-1.0.18-1.3.6.i686.rpm  
maildrop-toaster-2.0.3-1.3.8.i686.rpm  
maildrop-toaster-devel-2.0.3-1.3.8.i686.rpm  
qmailadmin-toaster-1.2.12-1.3.8.i686.rpm  
qmailmrtg-toaster-4.2-1.3.6.i686.rpm  
qmail-pop3d-toaster-1.03-1.3.20.i686.rpm  
qmail-toaster-1.03-1.3.20.i686.rpm

**repodata**

ripmime-toaster-1.4.0.6-1.3.6.i686.rpm  
send-emails-toaster-0.5-1.3.7.noarch.rpm  
simscan-toaster-1.4.0-1.3.8.i686.rpm  
spamassassin-toaster-3.2.5-1.3.17.i686.rpm  
squirrelmail-toaster-1.4.20-1.3.17.noarch.rpm  
ucspi-tcp-toaster-0.88-1.3.9.i686.rpm  
vpopmail-toaster-5.4.17-1.3.7.i686.rpm  
vqadmin-toaster-2.3.4-1.3.6.i686.rpm

**Step16:** [fosslab@fosslab qmail]\$ cd repodata/

**Step17:** [fosslab@fosslab repodata]\$ ls -l

total 44  
-rw-r--r-- 1 fosslab fosslab 14992 Aug 22 10:12 filelists.xml.gz  
-rw-r--r-- 1 fosslab fosslab 5519 Aug 22 10:12 other.xml.gz  
-rw-r--r-- 1 fosslab fosslab 15855 Aug 22 10:12 primary.xml.gz  
**-rw-r--r-- 1 fosslab fosslab 1362 Aug 22 10:12 repomd.xml**

This directory contains XML metadata information about the qmail packages

**Step18:**[fosslab@fosslab repodata]\$ cd ..

**Step19:**[fosslab@fosslab qmail]\$ cd ..

[fosslab@fosslab Downloads]\$

**Step20:** [fosslab@fosslab qmail]\$ cd ..

[fosslab@fosslab Downloads]\$

**Step21 :**Have to check whether Apache Web Server is Running in local machine

**Step22:** Open terminal be in super user

[fosslab@fosslab ~]\$ su

Password:

(admin123)

[root@fosslab fosslab]#

**Step23:** [root@fosslab fosslab]# service httpd status

httpd (pid 3658) is running...

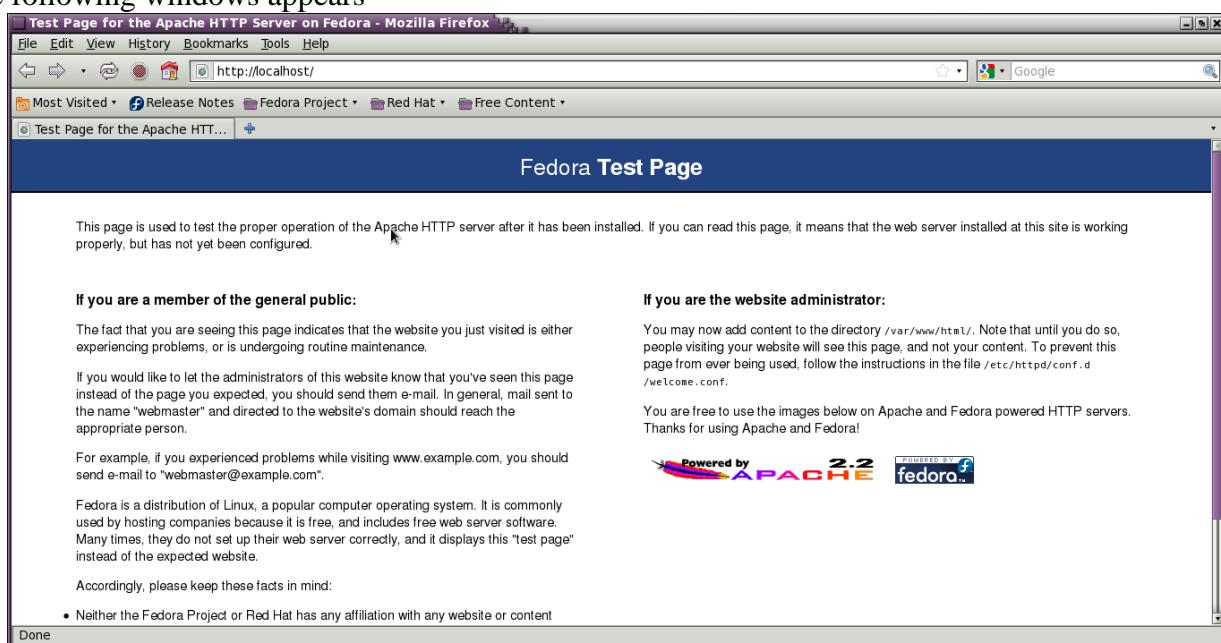
(if Stoppes can start by using the command service httpd start)

**Step24:**Now have to check whether Apache Web Server is working properly

Open Firefox and type following in URL

<http://localhost>

The following windows appears



**Step25:** If Fedora Test page appears the Apache Web Server is working

Now have to move the qmail packages from /home/fosslab/Downloads/ directory to /var/www/html directory which can be served by Apache Web Server

**Step26:**[root@fosslab fosslab]# pwd

/home/fosslab

[root@fosslab fosslab]# mv /home/fosslab/Downloads/qmail/ /var/www/html

[root@fosslab fosslab]# cd /var/www/html/

**Step27:** [root@fosslab html]# ls -l

total 356

```
-rwxrwxrwx 1 root root 131 Jul 31 03:49 arr.php
-rwxrwxrwx 1 root root 804 Jul 30 05:09 checklogin.php
-rwxrwxrwx 1 root root 396 Aug 2 02:48 connect11.php
-rwxrwxrwx 1 root root 405 Aug 6 02:36 connect1.php
-rwxrwxrwx 1 root root 368 Aug 1 01:46 connect2.php
-rwxrwxrwx 1 root root 175 Jul 30 05:38 connectdb.php
-rwxrwxrwx 1 root root 487 Jul 31 02:36 connection.php
-rwxrwxrwx 1 root root 357 Jul 30 05:39 cretable.php
-rwxrwxrwx 1 root root 68 Jul 31 03:46 date1.php
-rwxrwxrwx 1 root root 186 Aug 1 00:02 elseif.html
-rwxrwxrwx 1 root root 186 Aug 1 00:01 elseif.php
-rwxrwxrwx 1 root root 146 Jul 31 04:58 form100.html
-rwxrwxrwx 1 root root 160 Jul 31 05:02 form100.php
-rwxrwxrwx 1 root root 299 Aug 6 02:36 form1.html
-rwxrwxrwx 1 root root 307 Jul 30 02:13 form1.php
-rwxrwxrwx 1 root root 349 Jul 31 02:20 form2.html
-rwxrwxrwx 1 root root 184 Aug 1 01:35 form3.html
```

```
-rwxrwxrwx 1 root root 421 Jul 31 02:27 form5.html  
-rw-r--r-- 1 root root 377 Aug 2 02:47 formhandling.html  
-rwxrwxrwx 1 root root 350 Jul 31 02:18 form.html  
-rwxrwxrwx 1 root root 357 Jul 31 02:18 form.php  
-rw-r--r-- 1 root root 0 Aug 1 02:46 from.html  
-rwxrwxrwx 1 root root 62 Jul 31 23:48 helloworld.html  
-rwxrwxrwx 1 root root 62 Jul 31 23:59 helloworld.php  
-rwxrwxrwx 1 root root 689 Jul 30 05:03 homepage.html  
-rwxrwxrwx 1 root root 135 Aug 1 00:02 ifelse.html  
-rwxrwxrwx 1 root root 135 Jul 31 23:57 ifelse.php  
-rwxrwxrwx 1 root root 97 Jul 31 23:49 if.html  
-rwxrwxrwx 1 root root 97 Aug 1 00:00 if.php  
-rwxrwxrwx. 1 root root 224786 Oct 1 2009 mimetex.html  
drwxrwxrwx. 4 munin munin 4096 May 29 04:15 munin
```

**drwxr-xr-x 3 fosslab fosslab 4096 Aug 22 10:12 qmail**

```
-rwxrwxrwx 1 root root 460 Aug 1 00:15 retrieve.php  
-rwxrwxrwx 1 root root 462 Jul 30 05:49 retrive.php  
-rwxrwxrwx 1 root root 118 Aug 1 01:33 welcome1.php  
-rwxrwxrwx 1 root root 124 Aug 1 01:37 welcome.php
```

**Step28:** Open FireFox and type the following URL

<http://localhost/qmail/>

The following Appears

**Index of /qmail**

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>			

<a href="#">autorespond-toaster-2.0.4-1.3.6.i686.rpm</a>	27-Jun-2010 05:49	14K
<a href="#">clamav-toaster-0.96.1-1.3.36.i686.rpm</a>	27-Jun-2010 05:49	55M
<a href="#">control-panel-toaster-0.5-1.3.7.noarch.rpm</a>	27-Jun-2010 05:49	25K
<a href="#">courier-authlib-toaster-0.59.2-1.3.10.i686.rpm</a>	27-Jun-2010 05:49	234K
<a href="#">courier-imap-toaster-4.1.2-1.3.10.i686.rpm</a>	27-Jun-2010 05:49	679K
<a href="#">daemontools-toaster-0.76-1.3.6.i686.rpm</a>	27-Jun-2010 05:49	67K
<a href="#">gmail-pop3d-toaster-1.03-1.3.20.i686.rpm</a>	27-Jun-2010 05:49	31K
<a href="#">gmail-toaster-1.03-1.3.20.i686.rpm</a>	27-Jun-2010 05:49	494K
<a href="#">gmailadmin-toaster-1.2.12-1.3.8.i686.rpm</a>	27-Jan-2010 18:16	2.2M
<a href="#">gmailmrtg-toaster-4.2-1.3.6.i686.rpm</a>	27-Jun-2010 05:49	16K
<a href="#">repodata/</a>	22-Aug-2012 10:12	-
<a href="#">ripmime-toaster-1.4.0.6-1.3.6.i686.rpm</a>	27-Jun-2010 05:49	77K
<a href="#">send-emails-toaster-0.5-1.3.7.noarch.rpm</a>	27-Jun-2010 05:49	8.4K

Press repodata Folder and check and then close the Firefox

**Step29:** Now have to create the repository configuration File that the client can use the file for installing the qmail packages from server

**Step30:** Open terminal and be in Super user  
[fosslab@fosslab ~]\$ su -

Password: (admin123)

[root@fosslab ~]#

**Step31:** Peform the following

[root@fosslab ~]# cd /etc/yum.repos.d/

[root@fosslab yum.repos.d]# pwd

/etc/yum.repos.d

[root@fosslab yum.repos.d]# ls -l

total 12

-rw-r--r--. 1 root root 1186 May 29 02:56 fedora.repo

-rw-r--r--. 1 root root 1264 May 29 02:56 rpmfusion-free.repo

-rw-r--r--. 1 root root 1312 May 29 02:56 rpmfusion-nonfree.repo

[root@fosslab yum.repos.d]#

**Step32:** [root@fosslab yum.repos.d]#

vi qmail.repo

Add the following lines

[qmail]

name=Q-Mail packages

baseurl=http://localhost/qmail

enabled=1

gpgcheck =0

and press Esc :wq (For Saving)

**Step33:** [root@fosslab yum.repos.d]# cat qmail.repo

[qmail]

name=Q-Mail packages

baseurl=http://localhost/qmail

enabled=1

gpgcheck =0

**Step34:** Now have to rebuild the Yum cache so that yum package-manager will include qmail packge information

**Step35:** [root@fosslab yum.repos.d]# yum clean all

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,

: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,

: langpacks, list-data, local, merge-conf, post-transaction-

: actions, presto, priorities, protectbase, ps, refresh-

: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-

: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Cleaning repos: fedora qmail rpmfusion-free rpmfusion-nonfree

Cleaning up Everything

Cleaning up list of fastest mirrors

0 delta-package files removed, by presto

**Step36:**

[root@fosslab yum.repos.d]# **yum makecache**

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,

: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,  
: langpacks, list-data, local, merge-conf, post-transaction-  
: actions, presto, priorities, protectbase, ps, refresh-  
: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-  
: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Determining fastest mirrors

fedora	4.3 kB 00:00
fedora/filelists_db	17 MB 00:01
fedora/prestodelta	92 kB 00:00
fedora/primary_db	11 MB 00:01
fedora/other_db	6.3 MB 00:00
fedora/group_gz	407 kB 00:00

qmail	1.3 kB 00:00
qmail/filelists	15 kB 00:00
qmail/primary	15 kB 00:00
qmail/other	5.4 kB 00:00
rpmfusion-free	2.7 kB 00:00
rpmfusion-free/filelists_db	182 kB 00:00
rpmfusion-free/primary_db	266 kB 00:00
rpmfusion-free/other_db	95 kB 00:00
rpmfusion-nonfree	2.7 kB 00:00
rpmfusion-nonfree/filelists_db	84 kB 00:00
rpmfusion-nonfree/primary_db	114 kB 00:00
rpmfusion-nonfree/other_db	48 kB 00:00
qmail	25/25
qmail	25/25
qmail	25/25

Metadata Cache Created

**Step37:** [root@fosslab yum.repos.d]# yum search qmail

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,

: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,  
: langpacks, list-data, local, merge-conf, post-transaction-  
: actions, presto, priorities, protectbase, ps, refresh-  
: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-  
: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Loading mirror speeds from cached hostfile

0 packages excluded due to repository protections

```
=====
Matched: qmail
=====
```

**qmail-pop3d-toaster.i686 : POP3 daemon for qmail**

**qmail-toaster.i686 : qmail Mail Transfer Agent**

**qmailadmin-toaster.i686 : Web Administration for qmail-toaster**

**qmailmrtg-toaster.i686 : Mrtg for qmail-toaster**

control-panel-toaster.noarch : Toaster Control Panel.

ezmlm-toaster.i686 : Qmail Easy Mailing List Manager + IDX patches with mysql

: database support.

simscan-toaster.i686 : Simscan for qmail-toaster

vpopmail-toaster.i686 : Vpopmail for qmail-toaster

vqadmin-toaster.i686 : Web Administration for qmail-toaster

autorespond-toaster.i686 : Simple autoresponder for qmail

clamav-toaster.i686 : ClamAV for qmail-toaster

courier-authlib-toaster.i686 : courier-authlib for qmail-toaster

courier-imap-toaster.i686 : Courier-IMAP is an IMAP server that uses Maildirs

dspam.i686 : A library and Mail Delivery Agent for Bayesian SPAM filtering

gnubiff.i686 : A mail notification program

isoqllog-toaster.i686 : Isoqllog is an MTA log analysis program written in C.

libdomainkeys-toaster.i686 : ripMIME for qmail-toaster

libnss-mysql.i686 : NSS library for MySQL

libsrs2-toaster.i686 : libsrs2 for qmail-toaster

mlmmj.i686 : Mailserver-independent ezmlm-like mailing list manager

php-PHPMailer.noarch : PHP email transport class with a lot of features

ripmime-toaster.i686 : ripMIME for qmail-toaster

send-emails-toaster.noarch : control-panel-toaster module

**Step38:** can install now the qmail packages **yum install qmail-toaster.i686**

**Step39:** can install libsrs2-toaster

```
[root@fosslab yum.repos.d]# yum install libsrs2-toaster
```

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,

: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,

: langpacks, list-data, local, merge-conf, post-transaction-

: actions, presto, priorities, protectbase, ps, refresh-

: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-

: cache, security, show-leaves, tmpprepo, tsflags, upgrade-helper,

: verify, versionlock

Adding en\_US to language list

Loading mirror speeds from cached hostfile

Skipping filters plugin, no data

0 packages excluded due to repository protections

Setting up Install Process

Resolving Dependencies

Skipping filters plugin, no data

--> Running transaction check

--> Package libsrs2-toaster.i686 0:1.0.18-1.3.6 set to be installed

--> Finished Dependency Resolution

Dependencies Resolved

---

---

Package	Arch	Version	Repository	Size
---------	------	---------	------------	------

---

---

Installing:

libsrs2-toaster	i686	1.0.18-1.3.6	qmail	35 k
-----------------	------	--------------	-------	------

Transaction Summary

---

---

Install 1 Package(s)

Total download size: 35 k

Installed size: 79 k

Is this ok [y/N]: y

Downloading Packages:

Setting up and reading Presto delta metadata

Processing delta metadata

Package(s) data still to download: 35 k

libsrs2-toaster-1.0.18-1.3.6.i686.rpm	35 kB	00:00
---------------------------------------	-------	-------

Previous repo file missing:

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Warning: RPMDB altered outside of yum.

Installing : libsrs2-toaster-1.0.18-1.3.6.i686	1/1
--	-----

yum-updatesd not on the bus

**Installed:**

**libsrs2-toaster.i686 0:1.0.18-1.3.6**

Complete!

New leaves:

libsrs2-toaster.i686

**Step40 :**Open Terminal and perform the following

```
[fosslab@fosslab ~]$ su -
```

Password: admin123

```
[root@fosslab ~]# cd /etc/yum.repos.d/
```

```
[root@fosslab yum.repos.d]# ls -l
```

total 20

```
-rw-r--r--. 1 root root 1186 May 29 02:56 fedora.repo
```

```
-rw-r--r-- 1 root root 664 Aug 22 10:54 _local.repo
```

```
-rw-r--r-- 1 root root 83 Aug 22 10:44 qmail.repo
```

```
-rw-r--r--. 1 root root 1264 May 29 02:56 rpmfusion-free.repo
```

```
-rw-r--r--. 1 root root 1312 May 29 02:56 rpmfusion-nonfree.repo
```

**Step41:** vi fedora.repo

Add the line in appropriate place

baseurl=<http://192.168.105.254/f14repo>

**Step42:** [root@fosslab yum.repos.d]# vi rpmfusion-free.repo

Add the line in appropriate place

baseurl=<http://192.168.105.254/f14repo/rpmfusion/i386/free>

Make the “ enabled=0”

**Step43 :** [root@fosslab yum.repos.d]# vi rpmfusion-nonfree.repo

Add the line in appropriate place

baseurl=http://192.168.105.254/f14repo/rpmfusion/i386/nonfree

**Step44 :**

[root@fosslab yum.repos.d]# yum clean all

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,

: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,  
: langpacks, list-data, local, merge-conf, post-transaction-  
: actions, presto, priorities, protectbase, ps, refresh-  
: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-  
: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Cleaning repos: \_local fedora qmail rpmfusion-nonfree

Cleaning up Everything

Cleaning up list of fastest mirrors

0 delta-package files removed, by presto

**Step45 :** [root@fosslab yum.repos.d]# yum makecache

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,

: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,  
: langpacks, list-data, local, merge-conf, post-transaction-  
: actions, presto, priorities, protectbase, ps, refresh-  
: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-  
: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Determining fastest mirrors

_local	3.0 kB 00:00 ...
_local/filelists_db	932 B 00:00 ...
_local/primary_db	2.1 kB 00:00 ...
_local/other_db	1.1 kB 00:00 ...
fedora	4.3 kB 00:00
fedora/filelists_db	17 MB 00:01
fedora/prestodelta	92 kB 00:00
fedora/primary_db	11 MB 00:01
fedora/other_db	6.3 MB 00:00
fedora/group_gz	407 kB 00:00
qmail	1.3 kB 00:00
qmail/filelists	15 kB 00:00
qmail/primary	15 kB 00:00
qmail/other	5.4 kB 00:00
rpmfusion-nonfree	2.7 kB 00:00
rpmfusion-nonfree/filelists_db	84 kB 00:00
rpmfusion-nonfree/primary_db	114 kB 00:00
rpmfusion-nonfree/other_db	48 kB 00:00
qmail	25/25
qmail	25/25
qmail	25/25

Metadata Cache Created

**Step46:** to show the group list packages  
[root@fosslab yum.repos.d]# yum grouplist

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,  
: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,  
: langpacks, list-data, local, merge-conf, post-transaction-  
: actions, presto, priorities, protectbase, ps, refresh-  
: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-  
: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Setting up Group Process

Loading mirror speeds from cached hostfile

Installed Groups:

Administration Tools

Afrikaans Support

Albanian Support

Arabic Support

Armenian Support

Assamese Support

Asturian Support

Authoring and Publishing

Base

Basque Support

Belarusian Support

Bengali Support

Bhutanese Support

Books and Guides

Bosnian Support

Brazilian Portuguese Support

Bulgarian Support

Catalan Support

Chinese Support

Clustering

Croatian Support

Czech Support

DNS Name Server

Danish Support

Development Libraries

Development Tools

Dial-up Networking Support

Directory Server

Dutch Support

Editors

Educational Software

Electronic Lab

Engineering and Scientific

Estonian Support

Ethiopic Support

FTP Server

Fedora Eclipse

Fedora Packager

Filipino Support

Finnish Support

Font design and packaging

Fonts

French Support

GNOME Desktop Environment

GNOME Software Development

Galician Support

Games and Entertainment

Georgian Support

German Support

Graphical Internet

Graphics

Greek Support

Gujarati Support

Hardware Support

Haskell

Hebrew Support

Hindi Support

Hungarian Support

Icelandic Support

Indonesian Support

Input Methods

Inuktitut Support

Irish Support

Italian Support

Japanese Support

Java

Java Development

KDE Software Compilation

KDE Software Development

Kannada Support

Kashmiri Support

Kazakh Support

Khmer Support

Konkani Support

Korean Support

Kurdish Support

Lao Support

Latvian Support

Legacy Fonts

Legacy Network Server

Legacy Software Development

Lithuanian Support

Macedonian Support

Mail Server

Maithili Support

Malay Support

Malayalam Support

Maori Support

Marathi Support

Milkymist

MinGW cross-compiler

Mongolian Support

MySQL Database

Myanmar (Burmese) Support

Nepali Support

Network Servers

News Server

Norwegian Support

Office/Productivity

OpenOffice.org Development

Oriya Support

Perl Development

Persian Support

Polish Support

Portuguese Support

PostgreSQL Database

Printing Support

Punjabi Support

Romanian Support

Ruby

Russian Support

Samoan Support

Sanskrit Support

Serbian Support

Server Configuration Tools

Sindhi Support

Sinhala Support

Slovak Support

Slovenian Support

Somali Support

Sound and Video

Spanish Support

Swedish Support

System Tools

Tagalog Support

Tajik Support

Tamil Support

Telugu Support

Text-based Internet

Thai Support

Tibetan Support

Tonga Support

Turkish Support

Ukrainian Support

Urdu Support

Uzbek Support

Venda Support

Vietnamese Support

Virtualization

Web Development

Web Server

Welsh Support

Window Managers

Windows File Server

X Software Development

X Window System

Zulu Support

Available Groups:

Amazigh Support

Azerbaijani Support

Breton Support

Chichewa Support

Coptic Support

Dogtag Certificate System

English (UK) Support

Esperanto Support

Faroese Support

Fijian Support

Frisian Support

Friulian Support

Gaelic Support

Hiligaynon Support

Interlingua Support

Kashubian Support

Kinyarwanda Support

LXDE

Latin Support

Low Saxon Support

Luxembourgish Support

Malagasy Support

Maltese Support

Manx Support

MeeGo NetBook UX Environment

Northern Sotho Support

OCaml

Occitan Support

Sardinian Support

Southern Ndebele Support

Southern Sotho Support

Sugar Desktop Environment

Swahili Support

Swati Support

Tetum Support

Tsonga Support

Tswana Support

Turkmen Support

Upper Sorbian Support

Walloon Support

XFCE

XFCE Software Development

Xhosa Support

Done

**Step47 :**[root@fosslab yum.repos.d]# yum groupinstall "XFCE"

**Step48:** Open Firefox and type the following to check whether the packages are installed

<http://192.168.105.254/f14repo/rpmfusion/i386/nonfree/>

Step47: Select the package [rpmfusion-nonfree-release-14-0.4.noarch.rpm](#) and install it by

```
root@fosslab fosslab]# rpm -ivh http://192.168.105.254/f14repo/rpmfusion/i386/free/rpmfusion-free-remix-kickstarts-0.11.1-5.fc11.noarch.rpm
```

Retrieving http://192.168.105.254/f14repo/rpmfusion/i386/free/rpmfusion-free-remix-kickstarts-0.11.1-5.fc11.noarch.rpm

```
warning: /var/tmp/rpm-tmp.7Q2QJE: Header V3 RSA/SHA256 Signature, key ID 865cc9ea: NOKEY
```

error: Failed dependencies:

```
spin-kickstarts >= 0.11.1 is needed by rpmfusion-free-remix-kickstarts-0.11.1-5.fc11.noarch
```

```
[root@fosslab fosslab]# rpm -ivh http://192.168.105.254/f14repo/rpmfusion/i386/nonfree/rpmfusion-nonfree-release-14-0.4.noarch.rpm
```

Retrieving http://192.168.105.254/f14repo/rpmfusion/i386/nonfree/rpmfusion-nonfree-release-14-0.4.noarch.rpm

```
warning: /var/tmp/rpm-tmp.HD3zNx: Header V3 RSA/SHA256 Signature, key ID f09d8368: NOKEY
```

```
Preparing... ##### [100%]
```

```
package rpmfusion-nonfree-release-14-0.4.noarch is already installed
```

**Step49:** [root@fosslab fosslab]# cd /etc/yum.repos.d/

```
[root@fosslab yum.repos.d]#
```

**Step50** : [root@fosslab yum.repos.d]# yum search vlc

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,  
: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,  
: langpacks, list-data, local, merge-conf, post-transaction-  
: actions, presto, priorities, protectbase, ps, refresh-  
: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-  
: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Loading mirror speeds from cached hostfile

0 packages excluded due to repository protections

===== Matched: vlc  
=====

python-vlc.noarch : VLC Media Player binding for Python

vlc-core.i686 : VLC media player core

vlc-devel.i686 : Development files for vlc

vlc-nox.i686 : VLC media player without Xorg

vlc-plugin-jack.i686 : JACK audio plugin for VLC

vlc.i686 : The cross-platform open-source multimedia framework, player and

: server

gnome-applet-music.i686 : A GNOME panel applet to control various music players

**Step51** : [root@fosslab yum.repos.d]# yum install vlc\*

Loaded plugins: aliases, auto-update-debuginfo, changelog, dellsysid,  
: downloadonly, fastestmirror, filter-data, fs-snapshot, keys,

: langpacks, list-data, local, merge-conf, post-transaction-  
: actions, presto, priorities, protectbase, ps, refresh-  
: packagekit, refresh-updatesd, remove-with-leaves, rpm-warm-  
: cache, security, show-leaves, tmprepo, tsflags, upgrade-helper,  
: verify, versionlock

Adding en\_US to language list

Loading mirror speeds from cached hostfile

Skipping filters plugin, no data

0 packages excluded due to repository protections

Setting up Install Process

Nothing to do

**Result :** Thus we have successfully built and served RPM packages over Apache Web Server

[www.vidyarthiplus.com](http://www.vidyarthiplus.com)

[www.vidyarthiplus.com](http://www.vidyarthiplus.com)

**EX. NO: 5**  
**DATE:**

## **INSTALLING VARIOUS SOFTWARE PACKAGES**

**EX. NO: 5.1**

### **INSTALLATION AND FILE SHARING OF SAMBA SOFTWARE**

**Aim:**

To install Samba Software and share files between Linux and Windows operating system.

**Tasks:**

1. Installation of Samba Software.
2. Disabling security mechanisms like Firewall
3. Accessing folder and files from Linux System

#### **1. Installation of Samba:**

**Requirement:** Internet for downloading application. The installation of samba is simple.

Step 1: Login to the super user by typing,

**ssh**

Step 2: Then install the samba by typing the following command,

**yum install samba** (Not needed if it is already installed)

Step 3: Make the Samba server to run by using the following command

**service smb start**

#### **2. Testing Communication of Linux with windows:**

**Requirement:** Two systems, one is booted with **Fedora** and other is booted with **Windows**. Before going to accessing files of different environment, it is required to **turnoff the Security mechanisms** of Linux.

- a) To access Windows from Linux:

**Step1 :** Find the IP Address of Windows system (sample: 192.168.1.26)

**Step2 :** In Fedora , in places, type Ctrl +L and in the address bar type **smb://192.168.1.26**

**Step3 :** It will ask for the following details

**Pwd** required for sharing C\$ on 192.168.1.26

User Name:**administrator**

Domain :**cse**

Password:**(skrcse2011)**

For getting the Domain name use the following steps in Windows:

In command Prompt type :

C:\> hostname

**cse**

**Step4 :** the available files and folders will be visible in Fedora.

- b) To access Linux from Windows

**Step1 :** In fedora , check the status of samba using the following command

[fosslab@fosslab ~]\$ **ssh -**

[fosslab@fosslab ~]\$ service smb status

[fosslab@fosslab ~]\$ rpm -qa |grep samba

[fosslab@fosslab ~]\$ service smb restart (if not running already)

**Step2:** Now go to Windows Systems, Type the IP address of Fedora System in Run prompt as  
Run ://192.168.1.245

**Step 3:** To Map linux user into Samba user do the following

[fosslab@fosslab ~]\$useradd winuser

[fosslab@fosslab ~]\$passwd winusersmbpasswd –a winuser

Asks for password, Now type admin123 or any

**Step4:** To extend the functionality of Samba

[fosslab@fosslab ~]\$vi /etc/samba/smb.conf

**Step5:** Make the directory skrmicrosoft with machine names

[fosslab@fosslab ~]\$mkdir /software/skrmicrosoft/ {Adobe, IBM, Oracle } –p

**Step6:** To configure the samba file

[fosslab@fosslab ~]\$ vi smb.conf

Type the following in last line

[software]

Comment=Common Software Repository

Path=/software

Writable=no

Readonly=yes

Public=yes

Browsable=yes

Save the file

**Step7:** To restart the samba service

[fosslab@fosslab ~]\$service smb restart

Now check in windows the folder software will be visible with subfolders Adobe, IBM, Oracle

**Result :**

Thus the program to share the file using SAMBA is successfully completed.

**EX. NO 5.2                   COMMON UNIX PRINTING SYSTEM (CUPS)**

**DATE:**

**Aim:**

To Installing Common UNIX Printing System (CUPS) and testing the given objective.

**Procedure:**

- Step1: Open the terminal and log in.
- Step2: Type the following command  
      \$ rpm -qa | grep cups.
- Step3: Type the command  
      \$rpm -qi cups
- Step4: Then using ~~clear~~ command clear the screen.
- Step5: Login to super user .  
      \$su
- Step6: Check the status of the cups using the following command  
      #service cups status
- Step7: Start the service of ~~cups~~ using the following command  
      #service cups status
- Step8: Check the status of the cups again.
- Step9: Click Mozilla browser and type  
      <http://localhost:631/>
- Step10: Select the option **Adding printer and classes.**
- Step11: Select the option “**Add printer**”
- Step12: Give the root username and root password.
- Step13: Select the printer type and select continue option.
- Step14: Give the printer name, location and description and give continue option.
- Step15: Select the maker of ~~printer~~ and select the continue option.
- Step16: Select the model of printer and select the option “Add printer”
- Step17: Select the paper size and set the default options.
- Step18: Click on the printer name and see the jobs that are pending.

**Result :**

Thus the program Installing Common UNIX Printing System (CUPS) and testing is successfully completed

**EX.NO : 6**

## **WRITING USERSPACE DRIVERS USING FUSE**

**DATE:**

**Aim :**

To write userspace drivers using fuse

**Procedure:**

**Perform the following steps:**

[fosslab@fosslab ~]\$ pwd

/home/fosslab

[fosslab@fosslab ~]\$ ls -l

```
total 156
-rw-rw-r-- 1 fosslab fosslab 62 Jul 12 11:33 cel.py
-rw-rw-r-- 1 fosslab fosslab 477 Jul 24 14:07 create.py
drwxr-xr-x. 2 fosslab fosslab 4096 May 29 12:18 data
-rw-rw-r-- 1 fosslab fosslab 690 Jul 3 14:58 data.pl
-rw-rw-r-- 1 fosslab fosslab 177 Jul 17 14:32 defarg.py
-rw-rw-r-- 1 fosslab fosslab 14 Jul 12 12:06 default.pu
-rw-rw-r-- 1 fosslab fosslab 110 Jul 17 13:27 default.py
-rw-rw-r-- 1 fosslab fosslab 102 Jul 12 12:07 def.py
drwxr-xr-x 3 fosslab fosslab 4096 Aug 22 08:40 Desktop
drwxr-xr-x 2 fosslab fosslab 4096 Jun 26 14:00 Documents
drwxr-xr-x 4 fosslab fosslab 4096 Aug 22 09:03 Downloads
-rw-rw-r-- 1 fosslab fosslab 0 Jul 9 10:37 fibanocci.py
-rw-rw-r-- 1 fosslab fosslab 43 Jul 12 11:30 fibo.py
-rw-rw-r-- 1 fosslab fosslab 8 Jul 3 14:31 filename.out
-rw-rw-r-- 1 fosslab fosslab 597 Jul 3 14:33 file.pl
-rw-rw-r-- 1 fosslab fosslab 147 Jul 12 12:11 func.py
-rw-rw-r-- 1 fosslab fosslab 131 Jul 12 11:45 fun.py
-rw-rw-r-- 1 fosslab fosslab 289 Jul 24 14:48 insert.py
drwxrwxr-x 2 fosslab fosslab 4096 Jun 27 11:08 karthi
drwxr-xr-x 2 fosslab fosslab 4096 Jun 26 14:00 Music
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 08:40 myprojects
drwxrwxr-x 2 fosslab fosslab 4096 Jul 23 11:53 naren
-rw-rw-r-- 1 fosslab fosslab 102 Jul 12 11:37 oddeven.py
drwxr-xr-x 2 fosslab fosslab 4096 Jun 26 14:00 Pictures
drwxr-xr-x 2 fosslab fosslab 4096 Jun 26 14:00 Public
drwxr-xr-x. 3 fosslab fosslab 4096 May 29 12:18 public_html
-rw-rw-r-- 1 fosslab fosslab 129 Jul 24 13:48 sh1.py
-rw-rw-r-- 1 fosslab fosslab 239 Jul 24 13:54 sh2.py
-rw-rw-r-- 1 fosslab fosslab 673 Jul 24 15:01 sh3.pl
-rw-rw-r-- 1 fosslab fosslab 175 Jul 24 14:41 sh3.py
drwxr-xr-x. 2 fosslab fosslab 4096 May 29 12:18 share
-rw-rw-r-- 1 fosslab fosslab 470 Jul 24 14:51 shcrt.py
-rw-rw-r-- 1 fosslab fosslab 401 Jul 24 14:53 shinsrt.py
-rw-rw-r-- 1 fosslab fosslab 635 Jul 2 10:44 s.pl
-rw-rw-r-- 1 fosslab fosslab 967 Jun 26 14:54 ss.pl
```

```
-rw-rw-r-- 1 fosslab fosslab 0 Jul 2 10:18 switch  
-rw-rw-r-- 1 fosslab fosslab 775 Jul 2 10:45 switch1.pl  
-rw-rw-r-- 1 fosslab fosslab 731 Jul 3 14:40 switchcase.pl  
-rw-rw-r-- 1 fosslab fosslab 722 Jul 3 14:08 switch.pl  
drwxr-xr-x 2 fosslab fosslab 4096 Jun 26 14:00 Templates  
drwxr-xr-x 2 fosslab fosslab 4096 Jun 26 14:00 Videos
```

[fosslab@fosslab ~]\$ cd myprojects

[fosslab@fosslab myprojects]\$ echo "now we are going to see the log og all the revision-1 of the document from subversion server"

now we are going to see the log og all the revision-1 of the document from subversion server

[fosslab@fosslab myprojects]\$ echo "now we are going to see the log og all the revision svndemo"

now we are going to see the log og all the revision svndemo

[fosslab@fosslab myprojects]\$ echo "now we are going to see the log og all the revision-1 of the document from subversion server"

now we are going to see the log og all the revision-1 of the document from subversion server

[fosslab@fosslab myprojects]\$ svn co -r http://localhost:8000/repos/svndemo/

svn: Syntax error in revision argument '<http://localhost:8000/repos/svndemo/>'

[fosslab@fosslab myprojects]\$ cd ../

[fosslab@fosslab ~]\$ clear

[fosslab@fosslab ~]\$ echo "checking whether FUSE development packges has got installed"  
"checking whether FUSE development packges has got installed"

[fosslab@fosslab ~]\$ rpm -qa |grep fuse  
fuse-gmailfs-0.8.0-5.fc14.noarch  
fuse-emulator-utils-0.10.0.1-5.fc13.i686  
fuse-sshfs-2.2-6.fc14.i686  
fuse-convmvfs-0.2.6-1.fc14.i686  
ifuse-1.0.0-1.fc14.i686  
fuse-zip-0.2.11-1.fc13.i686  
fuse-2.8.4-1.fc14.i686  
fuse-emulator-roms-0.9.0-3.fc11.noarch  
afuse-0.2-5.fc12.i686  
diffuse-0.4.3-1.fc14.noarch  
fuse-emulator-0.10.0.2-3.fc12.i686  
gvfs-fuse-1.6.4-2.fc14.i686  
ceph-fuse-0.20.2-1.fc14.i686  
fuse-s3fs-0.7-4.fc12.noarch  
fuse-devel-2.8.4-1.fc14.i686  
libconfuse-2.6-3.fc12.i686  
fuseiso-20070708-9.fc13.i686

```
libconfuse-devel-2.6-3.fc12.i686
fusecompress_offline1-1.99.19-6.fc12.i686
fuse-libs-2.8.4-1.fc14.i686
fuse-python-0.2.1-2.fc14.i686
fuse-encfs-1.7.2-1.fc14.i686
zfs-fuse-0.6.9-7.20100709git.fc14.i686
fuse-afp-0.8.1-6.fc12.i686
fuse-smb-0.8.7-8.fc13.i686
fusecompress-2.6-8.20100223git754bc0de.fc14.i686
```

**sslab@fosslab ~]\$ rpm -qi fuse**

```
Name      : fuse           Relocations: (not relocatable)
Version   : 2.8.4          Vendor: Fedora Project
Release   : 1.fc14         Build Date: Tue 08 Jun 2010 10:42:44 AM IST
Install Date: Tue 29 May 2012 11:20:46 AM IST  Build Host: x86-04.phx2.fedoraproject.org
Group     : System Environment/Base  Source RPM: fuse-2.8.4-1.fc14.src.rpm
Size      : 187240          License: GPL+
Signature : RSA/SHA256, Sun 25 Jul 2010 03:27:02 AM IST, Key ID 421caddb97a1071f
Packager  : Fedora Project
URL       : http://fuse.sf.net
Summary   : File System in Userspace (FUSE) utilities
Description : With FUSE it is possible to implement a fully functional filesystem in a userspace program. This package contains the FUSE userspace tools to mount a FUSE filesystem.
```

**fosssslab@fosslab ~]\$ echo "checked for FUSE development packages"**  
**checked for FUSE development packages**

**open the browser type the following url:**

http:192.168.105.254/cs2406/sample code for cs2406,cs77/  
select the fuse folder->click that folder->display the fuse tutorials.tgz->save the file in to the downloads directory and the extract the fuse tutorials from downloads directory

**[fosslab@fosslab ~]\$ pwd**  
**/home/fosslab**

**[fosslab@fosslab ~]\$ cd Downloads**

**[fosslab@fosslab Downloads]\$ ls -l**

total 72832
-rw-rw-r-- 1 fosslab fosslab 74072485 Aug 22 09:02 CollabNetSubversionEdge-1.3.2_linux-x86.tar.gz
drwxrwxr-x 11 fosslab fosslab 4096 Mar 2 2011 csvn
-rw-rw-r-- 1 fosslab fosslab 30720 Aug 22 09:27 fuse
drwxr-xr-x 2 fosslab fosslab 4096 Jun 21 2011 fuse (2)
-rw-rw-r-- 1 fosslab fosslab 21878 Aug 22 09:27 fuse.gz

```
drwxr-xr-x 4 fosslab fosslab 4096 Jan 13 2011 fuse-tutorial  
-rw-rw-r-- 1 fosslab fosslab 21656 Aug 22 09:35 fuse-tutorial(2).tgz  
-rw-rw-r-- 1 fosslab fosslab 21656 Aug 22 09:30 fuse-tutorial.tgz  
drwxr-xr-x 5 fosslab fosslab 4096 Jan 25 2010 webattery-1.2  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:06 webattery-1.2-6.fc14.src  
-rw-rw-r-- 1 fosslab fosslab 185694 Aug 22 09:06 webattery-1.2-6.fc14.src.rpm  
-rw-rw-r-- 1 fosslab fosslab 181238 Aug 22 08:51 webattery-1.2.tar.gz
```

[fosslab@fosslab Downloads]\$ cd fuse-tutorial/

[fosslab@fosslab fuse-tutorial]\$ ls -l

```
total 60  
-rw-r--r-- 1 fosslab fosslab 6426 Jan 13 2011 callbacks.html  
drwxr-xr-x 4 fosslab fosslab 4096 Jan 13 2011 example  
-rw-r--r-- 1 fosslab fosslab 2899 Jan 13 2011 files.html  
-rw-r--r-- 1 fosslab fosslab 5074 Jan 13 2011 index.html  
-rw-r--r-- 1 fosslab fosslab 3721 Jan 13 2011 init.html  
-rw-r--r-- 1 fosslab fosslab 2142 Jan 13 2011 private.html  
-rw-r--r-- 1 fosslab fosslab 3576 Jan 13 2011 running.html  
-rw-r--r-- 1 fosslab fosslab 6069 Jan 13 2011 security.html  
drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 src  
-rw-r--r-- 1 fosslab fosslab 1505 Jan 13 2011 thanks.html  
-rw-r--r-- 1 fosslab fosslab 1505 Jan 13 2011 thanks.html
```

[fosslab@fosslab fuse-tutorial]\$ pwd

/home/fosslab/Downloads/fuse-tutorial

[fosslab@fosslab fuse-tutorial]\$ clear

[fosslab@fosslab fuse-tutorial]\$ ls -l

```
total 60  
-rw-r--r-- 1 fosslab fosslab 6426 Jan 13 2011 callbacks.html  
drwxr-xr-x 4 fosslab fosslab 4096 Jan 13 2011 example  
-rw-r--r-- 1 fosslab fosslab 2899 Jan 13 2011 files.html  
-rw-r--r-- 1 fosslab fosslab 5074 Jan 13 2011 index.html  
-rw-r--r-- 1 fosslab fosslab 3721 Jan 13 2011 init.html  
-rw-r--r-- 1 fosslab fosslab 2142 Jan 13 2011 private.html  
-rw-r--r-- 1 fosslab fosslab 3576 Jan 13 2011 running.html  
-rw-r--r-- 1 fosslab fosslab 6069 Jan 13 2011 security.html  
drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 src  
-rw-r--r-- 1 fosslab fosslab 1505 Jan 13 2011 thanks.html  
-rw-r--r-- 1 fosslab fosslab 6269 Jan 13 2011 unclear.html
```

[fosslab@fosslab fuse-tutorial]\$ cd src

[fosslab@fosslab src]\$ ls -l

```
total 48  
-rw-r--r-- 1 fosslab fosslab 27060 Jan 13 2011 bbfs.c  
-rw-r--r-- 1 fosslab fosslab 4994 Jan 13 2011 log.c  
-rw-r--r-- 1 fosslab fosslab 420 Jan 13 2011 log.h  
-rw-r--r-- 1 fosslab fosslab 649 Jan 13 2011 Makefile  
-rw-r--r-- 1 fosslab fosslab 687 Jan 13 2011 params.h
```

[fosslab@fosslab fuse-tutorial]\$ cd src

```
[fosslab@fosslab src]$ ls -l
total 48
-rw-r--r-- 1 fosslab fosslab 27060 Jan 13 2011 bbfs.c
-rw-r--r-- 1 fosslab fosslab 4994 Jan 13 2011 log.c
-rw-r--r-- 1 fosslab fosslab 420 Jan 13 2011 log.h
-rw-r--r-- 1 fosslab fosslab 649 Jan 13 2011 Makefile
-rw-r--r-- 1 fosslab fosslab 687 Jan 13 2011 params.h
```

[fosslab@fosslab src]\$ vi bbfs.c

[fosslab@fosslab src]\$ vi log.c

```
[fosslab@fosslab src]$ pkg-config fuse --cflags
-D_FILE_OFFSET_BITS=64 -I/usr/include/fuse
[fosslab@fosslab src]$ pkg-config fuse --libs
-pthread -L/lib -lfuse -lrt -ldl
```

[fosslab@fosslab src]\$ ls -l

```
total 48
-rw-r--r-- 1 fosslab fosslab 27060 Aug 22 09:42 bbfs.c
-rw-r--r-- 1 fosslab fosslab 4994 Aug 22 09:42 log.c
-rw-r--r-- 1 fosslab fosslab 420 Jan 13 2011 log.h
-rw-r--r-- 1 fosslab fosslab 649 Jan 13 2011 Makefile
-rw-r--r-- 1 fosslab fosslab 687 Jan 13 2011 params.h
```

[fosslab@fosslab src]\$ make

```
gcc -g -Wall `pkg-config fuse --cflags` -c bbfs.c
gcc -g -Wall `pkg-config fuse --cflags` -c log.c
gcc -g `pkg-config fuse --libs` -o bbfs bbfs.o log.o
```

[fosslab@fosslab src]\$ ls -l

```
total 132
-rwxrwxr-x 1 fosslab fosslab 37502 Aug 22 09:45 bbfs
-rw-r--r-- 1 fosslab fosslab 27060 Aug 22 09:42 bbfs.c
-rw-rw-r-- 1 fosslab fosslab 31260 Aug 22 09:45 bbfs.o
-rw-r--r-- 1 fosslab fosslab 4994 Aug 22 09:42 log.c
-rw-r--r-- 1 fosslab fosslab 420 Jan 13 2011 log.h
-rw-rw-r-- 1 fosslab fosslab 10064 Aug 22 09:45 log.o
-rw-r--r-- 1 fosslab fosslab 649 Jan 13 2011 Makefile
-rw-r--r-- 1 fosslab fosslab 687 Jan 13 2011 params.h
```

[fosslab@fosslab src]\$ pwd

/home/fosslab/Downloads/fuse-tutorial/src

[fosslab@fosslab src]\$ ls -l

```
total 132
-rwxrwxr-x 1 fosslab fosslab 37502 Aug 22 09:45 bbfs
-rw-r--r-- 1 fosslab fosslab 27060 Aug 22 09:42 bbfs.c
-rw-rw-r-- 1 fosslab fosslab 31260 Aug 22 09:45 bbfs.o
-rw-r--r-- 1 fosslab fosslab 4994 Aug 22 09:42 log.c
-rw-r--r-- 1 fosslab fosslab 420 Jan 13 2011 log.h
-rw-rw-r-- 1 fosslab fosslab 10064 Aug 22 09:45 log.o
-rw-r--r-- 1 fosslab fosslab 649 Jan 13 2011 Makefile
-rw-r--r-- 1 fosslab fosslab 687 Jan 13 2011 params.h
```

[fosslab@fosslab src]\$ cd ..//example/

[fosslab@fosslab example]\$ ls -l  
total 8  
drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 mountdir  
drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 rootdir

[fosslab@fosslab example]\$ ls -al mountdir

```
total 8
drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 .
drwxr-xr-x 4 fosslab fosslab 4096 Jan 13 2011 ..
-rw-r--r-- 1 fosslab fosslab 13 Jan 13 2011 bogus.txt
```

[fosslab@fosslab example]\$ ls -al rootdir

```
total 12
drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 .
drwxr-xr-x 4 fosslab fosslab 4096 Jan 13 2011 ..
-rw-r--r-- 1 fosslab fosslab 13 Jan 13 2011 bogus.txt
```

[fosslab@fosslab example]\$ cd rootdir/

[fosslab@fosslab rootdir]\$ pwd  
/home/fosslab/Downloads/fuse-tutorial/example/rootdir

[fosslab@fosslab rootdir]\$ touch myfile

[fosslab@fosslab rootdir]\$ mkdir mydir

[fosslab@fosslab rootdir]\$ mkdir fusetest

[fosslab@fosslab rootdir]\$ mkdir fusefile

[fosslab@fosslab rootdir]\$ ls -l  
total 16  
-rw-r--r-- 1 fosslab fosslab 13 Jan 13 2011 bogus.txt  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusefile  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusetest  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 mydir  
-rw-rw-r-- 1 fosslab fosslab 0 Aug 22 09:49 myfile

[fosslab@fosslab rootdir]\$ cd ..

[fosslab@fosslab example]\$ pwd

/home/fosslab/Downloads/fuse-tutorial/example

[fosslab@fosslab example]\$ ls -l

total 8

drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 mountdir  
drwxr-xr-x 5 fosslab fosslab 4096 Aug 22 09:49 rootdir

[fosslab@fosslab example]\$ ls -l mountdir/

total 0

[fosslab@fosslab example]\$ ls -l rootdir/

total 16

-rw-r--r-- 1 fosslab fosslab 13 Jan 13 2011 bogus.txt  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusefile  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusetest  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 mydir  
-rw-rw-r-- 1 fosslab fosslab 0 Aug 22 09:49 myfile

[fosslab@fosslab example]\$ ../src/bbfs rootdir/ mountdir/

about to call fuse\_main

[fosslab@fosslab example]\$ ls

bbfs.log mountdir rootdir

[fosslab@fosslab example]\$ ls -l mountdir/

total 16

-rw-r--r-- 1 fosslab fosslab 13 Jan 13 2011 bogus.txt  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusefile  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusetest  
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 mydir  
-rw-rw-r-- 1 fosslab fosslab 0 Aug 22 09:49 myfile

[fosslab@fosslab example]\$ mount

/dev/sda3 on / type ext4 (rw)

proc on /proc type proc (rw)

sysfs on /sys type sysfs (rw)

devpts on /dev/pts type devpts (rw,gid=5,mode=620)

tmpfs on /dev/shm type tmpfs (rw)

none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)

/tmp on /tmp type none (rw,bind)

/var/tmp on /var/tmp type none (rw,bind)

/home on /home type none (rw,bind)

capifs on /dev/capi type capifs (rw,mode=0666)

sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)

nfsd on /proc/fs/nfsd type nfsd (rw)

fusectl on /sys/fs/fuse/connections type fusectl (rw)

gvfs-fuse-daemon on /home/fosslab/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=fosslab)

bbfs on /home/fosslab/Downloads/fuse-tutorial/example/mountdir type fuse.bbfs  
(rw,nosuid,nodev,user=fosslab)

```
[fosslab@fosslab example]$ ls -l mountdir/
total 16
-rw-r--r-- 1 fosslab fosslab 13 Jan 13 2011 bogus.txt
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusefile
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusetest
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 mydir
-rw-rw-r-- 1 fosslab fosslab 0 Aug 22 09:49 myfile
```

```
[fosslab@fosslab example]$ fusermount -u mountdir
```

```
[fosslab@fosslab example]$ ls -l mountdir/
total 0
```

```
[fosslab@fosslab example]$ mount
/dev/sda3 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/tmp on /tmp type none (rw,bind)
/var/tmp on /var/tmp type none (rw,bind)
/home on /home type none (rw,bind)
capifs on /dev/capi type capifs (rw,mode=0666)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
gvfs-fuse-daemon on /home/fosslab/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=fosslab)
```

```
[fosslab@fosslab example]$ ls -l
total 56
-rw-rw-r-- 1 fosslab fosslab 46549 Aug 22 10:05 bbfs.log
drwxr-xr-x 2 fosslab fosslab 4096 Jan 13 2011 mountdir
drwxr-xr-x 5 fosslab fosslab 4096 Aug 22 09:49 rootdir
[fosslab@fosslab example]$ ls -l rootdir/total 16
-rw-r--r-- 1 fosslab fosslab 13 Jan 13 2011 bogus.txt
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusefile
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 fusetest
drwxrwxr-x 2 fosslab fosslab 4096 Aug 22 09:49 mydir
-rw-rw-r-- 1 fosslab fosslab 0 Aug 22 09:49 myfile
```

```
[fosslab@fosslab example]$ ls -l mountdir/
total 0
```

```
[fosslab@fosslab example]$ tail -f bbfs.log
```

```
bb_releasedir(path="/", fi=0xb62ff1e0)
  flags = 0x00000000
  fh_old = 0xb5900638
  writepage = 0
  direct_io = 0
  keep_cache = 0
  fh = 0xffffffffb5900638
  lock_owner = 0x0000000000000000
```

```
[fosslab@fosslab example]$ pwd
/home/fosslab/Downloads/fuse-tutorial/example
```

**Result:** Thus writing userspace drivers using fuse is successfully completed

**EX:NO:7**

## **GUI PROGRAMMING**

**DATE:**

**Aim:**

To write Python and Perl programs to illustrate GUI concepts.

### **Introduction:**

1. In GUI programming, a top -level root windowing object contains all of the little windowing objects that will be part of your complete GUI application.
2. These can be text Labels, Buttons, List Boxes, etc.
3. These Individual GUI components are known as **Widgets**
4. A **widget** (or **control**) is an element of a graphical user interface (GUI) that displays an information arrangement changeable by the user, such as a window or a text box.
5. The *Tkinter* module ("Tk interface") is the standard Python interface to the Tk GUI toolkit from Scriptics (formerly developed by Sun Labs).
6. Both Tk and Tkinter are available on most Unix platforms, as well as on Windows and Macintosh systems. Starting with the 8.0 release, Tk offers native look and feel on all platforms.
7. Keeping Python window "on top" of others
8. *Tkinter.mainloop()* - "main loop" is the heart of our programs, the repeating chunk of code that carries out the task at hand. *But* Perl/Tk programs are *event driven*, so even if we write what we believe is our program's main loop, it must coexist with a higher order main loop that's a fundamental part of Tk.
9. The Tk main loop is typically referred to as the *event loop*, and its job is to invoke callbacks in response to events such as button presses or timer expirations. Callbacks are Perl subroutines associated with *Tk* events. In Perl/T

### **Execution Steps:**

1.`python filename.py`

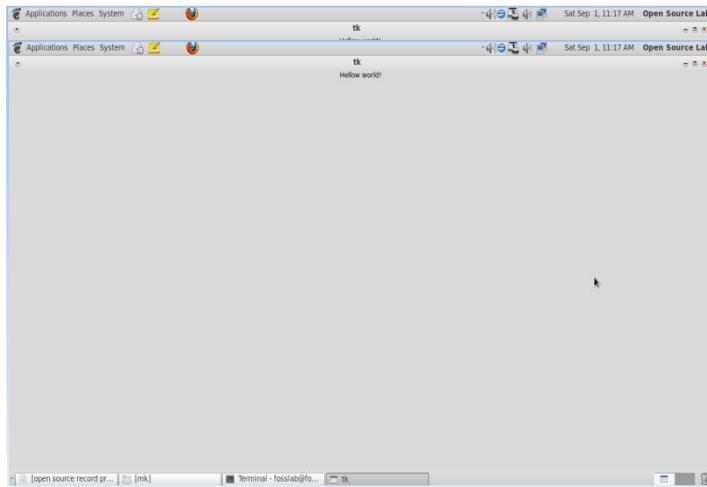
2.`perl filename.pl`

## **GUI PROGRAMMING PROGRAMS**

### **Example 1:GUI with Python Program for Printing Welcome Message**

```
#!/usr/bin/env python
import Tkinter
top = Tkinter.Tk()
label = Tkinter.Label(top, text = 'Hellow world!')
label.pack()
Tkinter.mainloop()
```

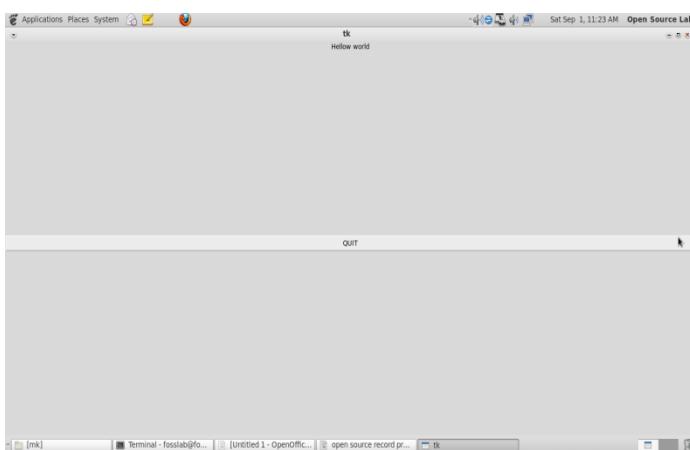
**Output:**



**Example 2.GUI with Python Program to Print Window and Command Button**

```
#!/usr/bin/env python
import Tkinter
top = Tkinter.Tk()
hello = Tkinter.Label(top,text='Hellow world')
hello.pack()
quit = Tkinter.Button(top, text='QUIT', command=top.quit, bg='red', fg='white')
quit.pack(fill=Tkinter.X, expand=1)
Tkinter.mainloop()
```

**Output:**

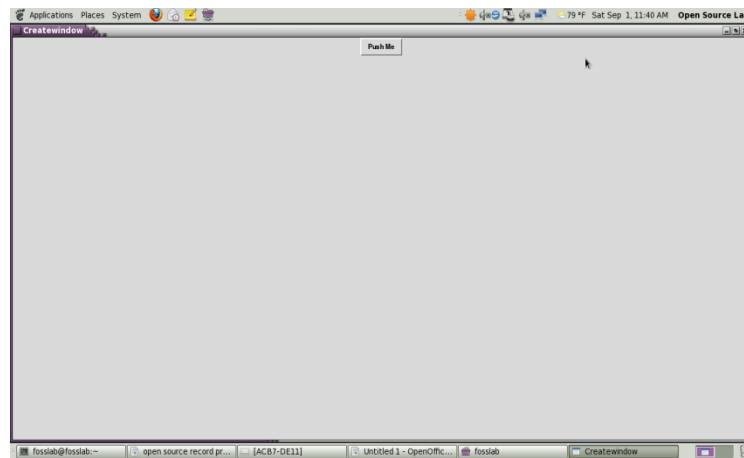


**Example 3.GUI with PERL Program to for creating Window.**

```
#!/usr/local/bin/perl
use Tk;
# Main Window
```

```
my $mw = new MainWindow;
my $but = $mw -> Button(-text => "Push Me",
                           -command =>\&push_button);
$but -> pack();
MainLoop;
#This is executed when the button is pressed
sub push_button { whatever }
```

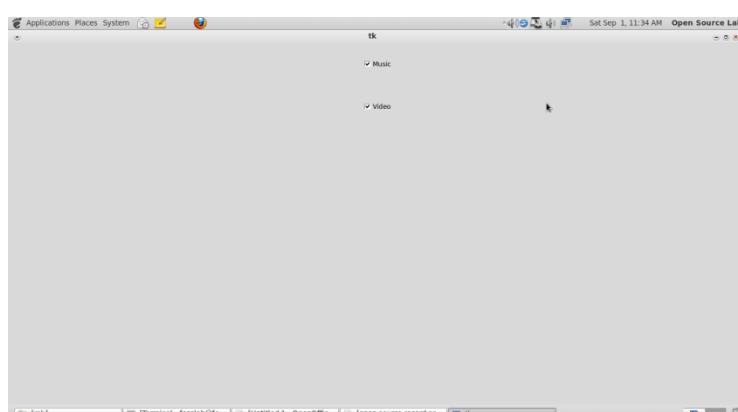
**Output:**



**Example 4.GUI with PYTHON Program to for Check Button**

```
from Tkinter import *
import tkMessageBox
import Tkinter
top = Tkinter.Tk()
CheckVar1 = IntVar()
CheckVar2 = IntVar()
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, onvalue = 1, offvalue = 0, height=5, width = 20)
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, onvalue = 1, offvalue = 0, height=5, width = 20)
C1.pack()
C2.pack()
top.mainloop()
```

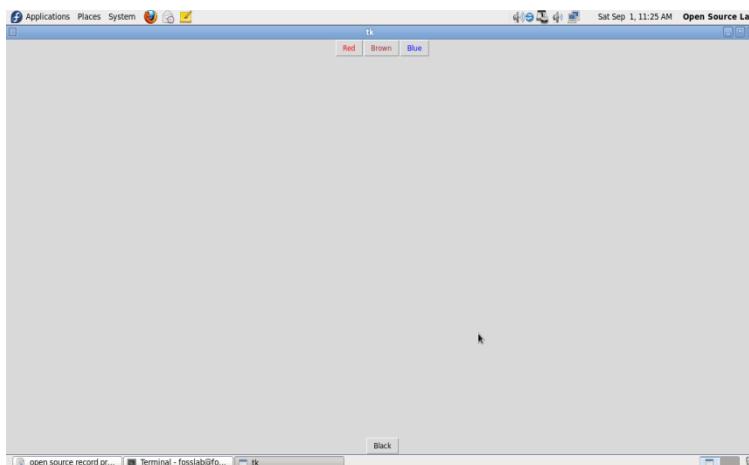
**Output:**



### **Example 5.GUI with PYTHON Program to for Frame**

```
from Tkinter import *
root = Tk()
frame = Frame(root)
frame.pack()
bottomframe = Frame(root)
bottomframe.pack( side = BOTTOM )
redbutton = Button(frame, text="Red", fg="red")
redbutton.pack( side = LEFT )
greenbutton = Button(frame, text="Brown", fg="brown")
greenbutton.pack( side = LEFT )
bluebutton = Button(frame, text="Blue", fg="blue")
bluebutton.pack( side = LEFT )
blackbutton = Button(bottomframe, text="Black", fg="black")
blackbutton.pack( side = BOTTOM )
root.mainloop()
```

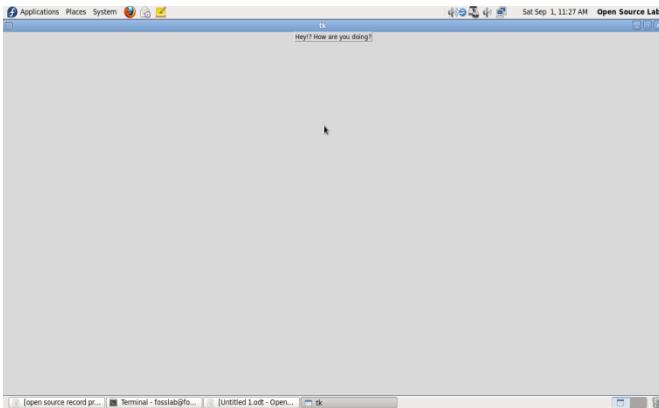
#### **Output:**



### **Example 6. GUI with PYTHON Program to for Label**

```
from Tkinter import *
root = Tk()
var = StringVar()
label = Label( root, textvariable=var, relief=RAISED )
var.set("Hey!? How are you doing?")
label.pack()
root.mainloop()
```

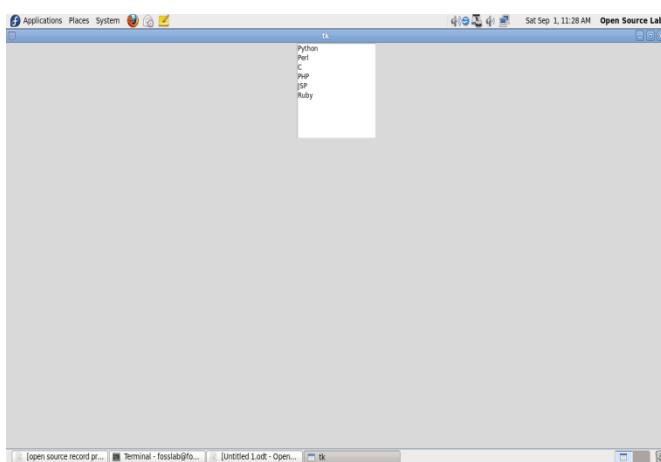
#### **Output:**



### Example 7.GUI with PYTHON Program to for ListBox

```
from Tkinter import *
import tkMessageBox
import Tkinter
top = Tk()
Lb1 = Listbox(top)
Lb1.insert(1, "Python")
Lb1.insert(2, "Perl")
Lb1.insert(3, "C")
Lb1.insert(4, "PHP")
Lb1.insert(5, "JSP")
Lb1.insert(6, "Ruby")
Lb1.pack()
top.mainloop()
```

#### Output:

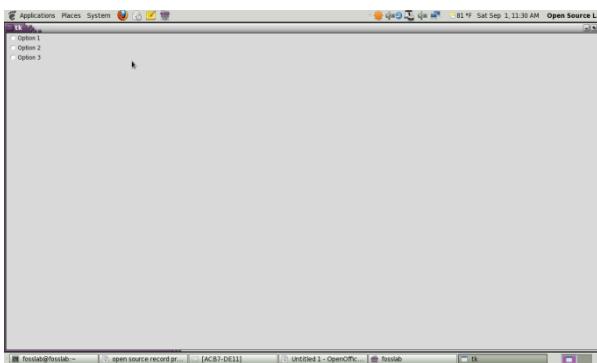


### Example 8.GUI with PYTHON Program to for Radio Button

```
from Tkinter import *
def sel():
    selection = "You selected the option " + str(var.get())
    label.config(text = selection)
root = Tk()
```

```
var = IntVar()
R1 = Radiobutton(root, text="Option 1", variable=var, value=1, command=sel)
R1.pack( anchor = W )
R2 = Radiobutton(root, text="Option 2", variable=var, value=2,command=sel)
R2.pack( anchor = W )
R3 = Radiobutton(root, text="Option 3", variable=var, value=3,command=sel)
R3.pack( anchor = W )
label = Label(root)
label.pack()
root.mainloop()
```

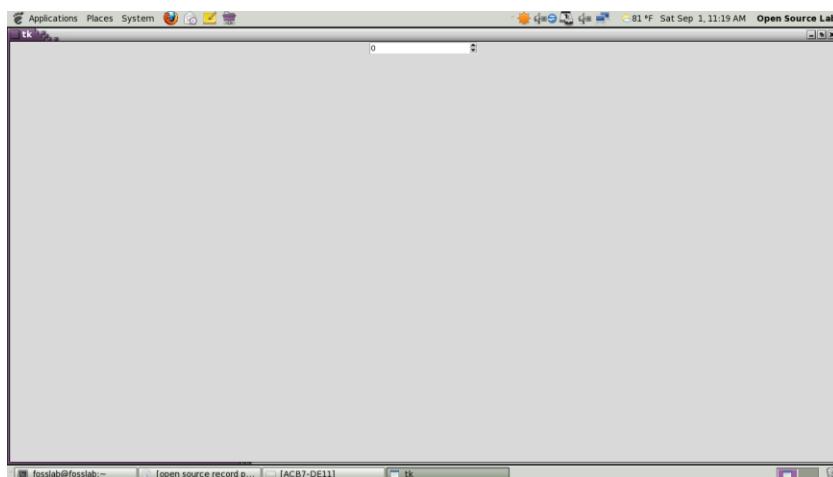
### **Output:**



### **Example 9.GUI with PYTHON Program to for SpinBox**

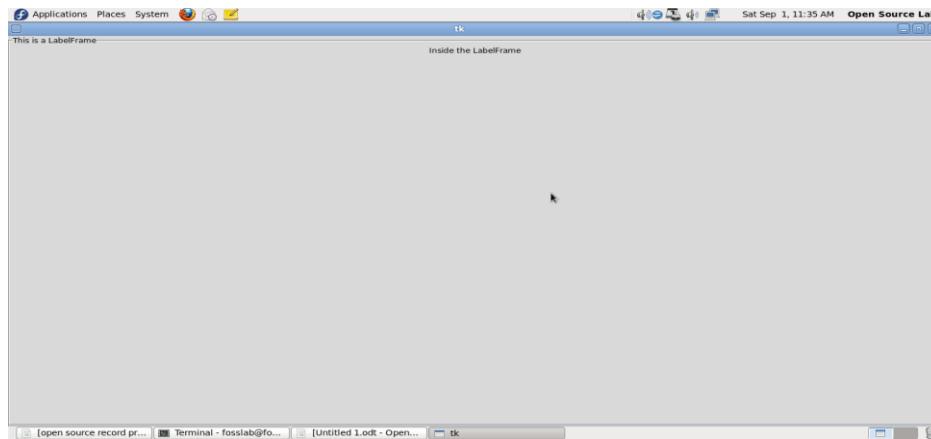
```
from Tkinter import *
master = Tk()
w = Spinbox(master, from_=0, to=10)
w.pack()
mainloop()
```

### **Output:**



### **Example 10.GUI with PYTHON Program to for Lable Frame**

```
from Tkinter import *
root = Tk()
labelframe = LabelFrame(root, text="This is a LabelFrame")
labelframe.pack(fill="both", expand="yes")
left = Label(labelframe, text="Inside the LabelFrame")
left.pack()
root.mainloop()
```

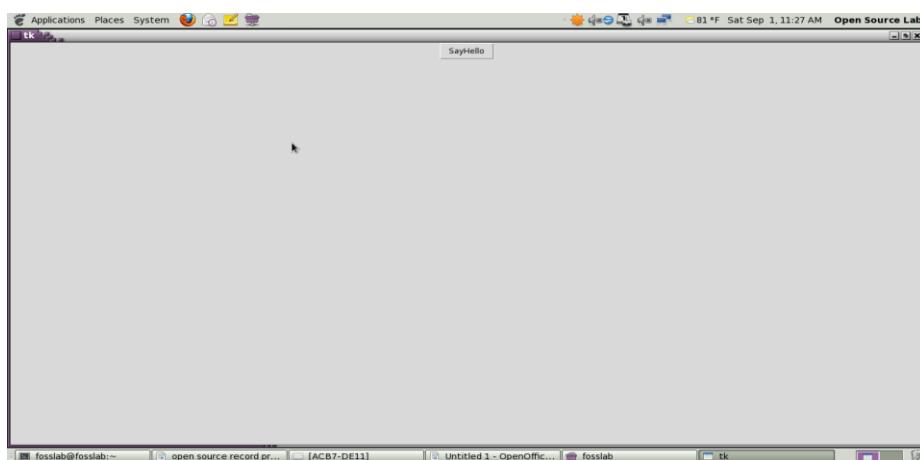


## Output:

### **Example 11.GUI with PYTHON Program to for TkmassageBox**

```
import Tkinter
import tkMessageBox
top = Tkinter.Tk()
def hello():
    tkMessageBox.showinfo("Say Hello", "Hello World")
B1 = Tkinter.Button(top, text = "Say Hello", command = hello)
B1.pack()
top.mainloop()
```

## Output:



## **RESULT :**

Thus the program for gui programing using perl and python is executed and verified

**EX.NO:8.**

## **VERSION CONTROL SYSTEM SETUP AND USAGE**

**DATE:**

### **Aim**

To setup a version control system for managing files using svn. The system should be able to track changes to the files/folders and a period of time and revert if necessary.

### **INTRODUCTION OF THE CONCEPT/TOOL**

- Version control is the concept of managing all changes to a repository. The repository could be containing any type of file.
- In computer software engineering, revision control is any practice that tracks and provides control over changes to source code. Software developers sometimes use revision control software to maintain documentation and configuration files as well as source code.
- As teams design, develop and deploy software, it is common for multiple versions of the same software to be deployed in different sites and for the software's developers to be working simultaneously on updates .Bugs or features of the software are often only present in certain versions (because of the fixing of some problems and the introduction of others as the program develops). Therefore, for the purposes of locating and fixing bugs, it is vitally important to be able to retrieve and run different versions of the software to determine in which version(s) the problem occurs. It may also be necessary to develop two versions of the software concurrently (for instance, where one version has bugs fixed, but no new features (branch), while the other version is where new features are worked on (trunk ).
- At the simplest level, developers could simply retain multiple copies of the different versions of the program, and label them appropriately. This simple approach has been used on many large software projects. While this method can work, it is inefficient as many near-identical copies of the program have to be maintained. This requires a lot of self-discipline on the part of developers, and often leads to mistakes. Consequently, systems to automate some or all of the revision control process have been developed.
- Moreover, in software development, legal and business practice and other environments, it has become increasingly common for a single document or snippet of code to be edited by a team, the members of which may be geographically dispersed and may pursue different and even contrary interests. Sophisticated revision control that tracks and accounts for ownership of changes to documents and code may be extremely helpful or even necessary in such situations.
- Revision control may also track changes to configuration files, such as those typically stored in /etc or /usr/local/etc on Unix systems. This gives system administrators another way to easily track changes made and a way to roll back to earlier versions should the need arise.

Subversion is a free/open-source version control system. That is, Subversion manages files and directories over time. A tree of files is placed into a central *repository*. The repository is much like an ordinary file server, except that it remembers every change ever made to your files and directories. This

allows you to recover older versions of your data, or examine the history of how your data changed. In this regard, many people think of a version control system as a sort of “time machine”.

Subversion can access its repository across networks, which allows it to be used by people on different computers. At some level, the ability for various people to modify and manage the same set of data from their respective locations fosters collaboration. Progress can occur more quickly without a single conduit through which all modifications must occur.

### [1] Description of the exercise

- In this exercise, we would be setting up version control using svn. We would create a repository and then access that repository over http.
- Briefly, we would be doing the following
- Install/setup svn repository
- Configure apache for mod\_svn access. We would also password-protect svn access
- Check in code/files using svn commands
- Check out codes to another location
- Modify/commit codes
- Revert to an older version of the code

### Pre-requisite

- Fedora 12 installed with root access (root access for installation/configuration purpose
  - LAN connection
  - Apache Webserver
  - selinux disabled - “ setenforce 0 “ can be given at command line to disable it runtime and
  - Reboot the machine
  - FOSS Lab server or internet connectivity for downloading packages from internet using yum

### Detailed steps for the exercise-

Note:In the description (steps), all greyed out lines are actual commands/configurations done on the server

#### Step 1 - Installation of packages

The first thing to do is to install the packages required for the setup of svn. We also need to install/configure packages required for letting svn command be accessed over http

**yum install mod\_dav\_svn subversion.**

Its recommended to install packages using yum as it will pull in any dependencies required for the installation

#### Step 2 - Configuration of the repository

4. The following commands create the necessary folder structure for the repository

```
mkdir /svn  
mkdir /svn/repos  
cd /svn/repos  
chown -R apache.apache ..repos.
```

Now we need to initialize the svn structure and create a repository

```
# svnadmin create svn-test
```

With this, the repository is configured

### **Step 3 - Creation of a local area for code**

The purpose of this step is to have a staging area for code from where you would be doing some development. This code can then be committed to the svn

- a. Create the local folder structure  
**# mkdir /home/fosslab svn-test**  
**# cd /home/fosslab svn-test**

- b. Create local directories as per requirement - This can be customized to suit your needs  
**# mkdir configurations options main**

- c. Edit any file as required  
**vim configurations/test.cfg**

Edit and save any other file required.

### **Step 4 : Import of the code area onto svn**

#### **a. Import the initial structure onto svn**

**# svn import /home/fosslab svn-test/ file:///svn/repos/svn-test/ -m "Initial Import"**

Adding /root/svn-test/main

Adding /root/svn-test/configurations

Adding /root/svn-test/configurations/test.cfgA

dding /root/svn-test/options

Committed revision 1.

This imports the files/folders onto the repo with a log message of “Initial Import”

**# chown -R apache.apache /svn/repos/**

for fixing the permissions of the newly committed repo to be accessible by apache

### **Step 5 : Configuring Apache. Edit subversion.conf**

at /etc/httpd/conf.d/subversion.conf and add the following lines

**# gedit /etc/httpd/conf.d/subversion.conf <Location /repos>DAV svn SVNParentPath /svn/repos AuthType BasicAuthName "Subversion repos" AuthUserFile /etc/svn-auth-conf Require valid-user</Location>** The above line allow for a repo on /svn/repos to be accessible with a password contained in /etc/svn-auth-conf

- b. Restart apache to apply the changes

**# /etc/init.d/httpd restart**

- c. configure password for authentication

**htpasswd -cm /etc/svn-auth-conf <username>for example:**

**# htpasswd -cm /etc/svn-auth-conf fedora**

Please enter the password twice to create the user

### **Step 6 - Checkout the repo**

- a. We can check out the repo into a third party area by doing the following steps

```
#cd /tmp/  
# svn --username=fedora co http://localhost:80/repos/svn-test/  
(or) svn checkout file:///svn/repos/svn-test/  
AAAAsvn-test/mainsvn-test/configurations
```

svn-test/configurations/test.cfg svn-test/options  
Checked out revision 1.

### Step 7 : Editing and Committing

- a. Editing

```
# cd /tmp/svn-test  
#ls  
configurations main options  
# vim configurations/test.cfg
```

Edit the file and make any changes/save them

- b. Committing the changes

```
# svn commit -m "Added a line toSendin configurations/test.cfg Transmitting file data .
```

(or) svn commit -m "Added a line to testconf1.cfg."

Committed revision 2.

- c. Adding/Deleting Items

```
# ls  
configurations main options  
# cd configurations/  
# cp /etc/yum.conf .  
# svn add yum.conf  
# yum.conf  
#svn commit -m "Added yum conf"
```

Adding configurations/yum.conf

Transmitting file data .Committed revision 3.

- d. Reverting Items

- i. Listing out all revisions.

This step will give you all the checkpoints

```
# svn log http://localhost:80/repos/svn-test/ (or) #svn log file:///svn/repos/svn-test/  
Added the fedora repo-----r2 | fedora | 2011-07-21  
20:30:22 +0530 (Thu, 21 Jul 2011) | 1 line Added a line to testconf1.cfg-----  
-----r1 | fedora | 2011-07-21 20:24:40 +0530 (Thu, 21 Jul 2011) | 1 line Initial Import-----
```

- ii) reverting/checkout to revision 1 -

```
# svn co -r 1 http://localhost:80/repos/svn-test/ (or) #svn co -r 1 file:///svn/repos/svn-test/  
Asvn-test/main
```

A svn-test/configurations  
A svn-test/configurations/test.cfg

A svn-test/optionsChecked out revision 1.

**Result:**

Thus the version control system setup for managing files using svn was successfully verified

**EX:NO: 9**

## **PERL & MYSQL CONNECTIVITY**

**DATE:**

**Aim:**

To demonstrate the Perl programs and connectivity with MySQL database.

### **Introduction:**

Perl was initially an efficient interpreted language optimized for processing text files. Perl stands for Practical Extraction and Report *Language*. Perl is a high-level, general-purpose, interpreted, dynamic programming language. Perl was originally developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier.

Perl borrows features from other programming languages including C, shell scripting(sh) AWK, and sed. The language provides powerful text processing facilities without the arbitrary data length limits of many contemporary Unix tools, facilitating easy manipulation of text files. Though originally developed for text manipulation, Perl is used for a wide range of tasks including system administration, web development, network programming, games, bioinformatics, and GUI development.

The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). Its major features include support for multiple programming paradigms (procedural, object-oriented, and functional styles), reference counting memory management , built-in support for text processing, and a large collection of third-party modules.

### **Uses:**

- Replace excessively complex shell/awk/sed scripts in Unix.
- Web applications
- System administration scripting.
- Data Mining
- Data Format Conversion
- "Middle-tier" ware. (Socket-based servers that also talk to databases.)

### **Steps to execute perl prgrams :**

1. Open the vi editor and write the program inside the following tag <?php ?>
2. Save the program with .pl as extension
3. Execute the program using perl filename.pl

### **Steps for MySQL**

**1. Go to terminal and perform the following :**

```
[fosslab@fosslab ~]$ su -
```

Password: (admin123)

**2. To enter into mysql**

```
[root@fosslab ~]# mysql
```

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 7

Server version: 5.1.51 Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

This software comes with ABSOLUTELY NO WARRANTY. This is free software,

and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

**3. To display the available databases:**

```
mysql> show databases;
```

```
+-----+
```

```
| Database      |
```

```
+-----+
```

```
| information_schema |
```

```
| TESTDB      |
```

```
| college     |
```

```
| employee    |
```

```
| mysql       |
```

```
| sample      |
```

```
| steel       |
```

```
+-----+
```

7 rows in set (0.00 sec)

**4. To create the new Database :**

```
mysql> create database Student;
```

```
mysql> show databases;
```

```
+-----+
```

```
| Database      |
```

```
+-----+
| information_schema |
| TESTDB      |
| college     |
| employee    |
| mysql       |
| sample      |
| steel       |
| student     |
+-----+
```

8 rows in set (0.00 sec)

## 5. To use the student Database

mysql> **use student;**

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> show tables;

```
+-----+
| Tables_in_student |
+-----+
| EMPLOYEE      |
| courses       |
| courses1      |
| veni          |
| veni1         |
+-----+
```

5 rows in set (0.00 sec)

## 6. To create table details in the student Database

mysql> create table **details** (regno int, name varchar(20),age int);

mysql> **desc details;**

```
regno |      name |          age  
int     varchar(20)      int
```

## PERL PROGRAMS

### Program 1:

```
[fosslab@fosslab nm]$ vi peal.pl  
#!/usr/bin/perl  
#Define a string to replace  
$mystring="Hello, PERL!";  
print "Before Replacement :$mystring\n";  
substr($mystring,6)="WORLD!";  
print "After Replacement:$mystring\n";
```

### Output:

```
[fosslab@fosslab ~]$ perl peal.pl  
before replacement:Hello,PERL!  
After Replacemant>Hello,WORLD!
```

### Program 2:

```
[fosslab@fosslab nm]$ vi perl1.pl  
#!/usr/bin/perl  
#Define an array  
@coins=(“Quarter”,”Dine”,”Nickel”);  
print “First statement:@coins”;  
print “\n”;  
#Add one element at the end of array  
push(@coins,”penny”);  
print “Second statement : @coins”;  
print “\n”;  
#Add one element at the beginning of the array  
unshift (@coins,”Dollar”);  
print “Third statement:@coins”;  
print “\n”;  
#Remove one element from the last of the array  
pop(@coins);  
print “Fourth statement:@coins”;  
print “\n”;  
#Remove one element from the beginning of the array  
shift(@coins);  
print “Fifth statement:@coins”;  
print “@coins”;
```

### Output:

```
[fosslab@fosslab ~]$ perl perl1.pl  
First statement:Quarter Dine Nickel  
Second statement : Quarter Dine Nickel penny  
third statement :Doller Quarter Dine Nickel penny
```

fourth statement:Doller Quarter Dine Nickel

Fifth statement :Quarter Dime Nickel

**Program 3:**

```
[fosslab@fosslab nm]$ vi one.pl
#/usr/bin/perl
print"Hello sample perl program";
print"That's counts to Ten\n\n";
for(my$i=1;$i<10;$i++)
{
    print"$i\n";
}
print"\n Thanks for running me\n";
```

**Output:**

```
[fosslab@fosslab nm]$ perl one.pl
Hello sample perl programThat's counts to Ten
1 2 3 4 5 6 7 8 9Thanks for running me
```

**Program 4:**

```
#!/usr/bin/perl
print"Enter the value";
my$value=<STDIN>;
if($value>=5){    print"value>5";    }
else    {    print"value<5";    }
```

**Output :**

```
[fosslab@fosslab nm]$ perl if.pl
Enter the value 9
value>5[fosslab@fosslab nm]$ perl if.pl
Enter the value 3
```

**Program 5:**

```
#!/usr/bin/perl
use warnings;
use strict;
use Switch;
print"1.chocolate\n2.pepsi\n3.sweets\n4.fruits";
print"enter ur choice";
chomp(my$value=<STDIN>);
switch($value)
{
    case(1) {    print"i like chocolate";    }
    case(2) {    print"i like pepsi";    }
    case(3) {    print"i like sweets";    }
    case(4) {    print"i like fruits";    }
}
```

**Output:**

```
[fosslab@fosslab dilan]$ vi caseswitch.pl
```

[fosslab@fosslab dilan]\$ perl caseswitch.pl

Switch will be removed from the Perl core distribution in the next major release. Please install it from CPAN. It is being used at caseswitch.pl, line 22.

1.chocolate

2.pepsi

3.sweets

4.fruitcenter

ur choice2

i like pepsi

**Execution Steps:**

python filename.py

perl filename.pl

## PERL &SQL CONNECTIVITY PROCEDURES

### Steps for MySQL

**Go to terminal and perform the following :**

[fosslab@fosslab ~]\$ su -

**Password: (admin123)**

**To enter into mysql**

[root@fosslab ~]# mysql

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 7

Server version: 5.1.51 Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

This software comes with ABSOLUTELY NO WARRANTY. This is free software,

and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

**To display the available databases:**

mysql> show databases;

Database
information_schema
TESTDB
college
employee
mysql
sample
steel

7 rows in set (0.00 sec)

**To create the new Database :**

```
mysql> create database Student;
```

```
mysql> show databases;
```

```
+-----+
| Database      |
+-----+
| information_schema |
| TESTDB        |
| college       |
| employee      |
| mysql          |
| sample         |
| steel          |
| student        |
+-----+
8 rows in set (0.00 sec)
```

**To use the student Database**

```
mysql> use student;
```

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

```
mysql> show tables;
```

```
+-----+
| Tables_in_student |
+-----+
| EMPLOYEE        |
| courses         |
| courses1        |
| veni            |
| veni1           |
+-----+
```

e. rows in set (0.00 sec)

**To create table details in the student Database**

```
mysql> create table details (regno int, name varchar(20),age int);
```

```
mysql> desc details;
```

regno	name	age
int	varchar(20)	int

**RESULT:**

Thus the program for perl is executed and verified successfully.

## PERL MySQL CONNECTIVITY PROGRAMS

**Program 6. Create the table in Perl that will be automatically updated in Database using mysql**

```
# !usr/bin/perl
use DBI;
my $dbh=DBI->connect("dbi:mysql:stud","root","");
If(!$dbh)
{
die("error:$!");
}
$sth=$dbh->prepare("Create table student(rollno int,sname varchar(20))");
$sth->execute();
$dbh->disconnect;
```

**Output:**

**Run the Perl script**

```
[root@fosslab ~]# perl connect.pl
now see the tables in database ("student")
```

```
mysql> show tables;
+-----+
| Tables_in_student |
+-----+
| student |
+-----+
1 row in set (0.00 sec)
```

**Program 7. Insert the values in Perl that will be automatically updated in database using mysql as follows**

```
#!/usr/bin/perl
use DBI; #to use the build in package we use "Use", DBI is the build in
package in perl
my $dbh=DBI->connect("dbi:mysql:student","root","");
#connect to
database
if(!$dbh)
{
die("error:$!");
}
```

```
$sth=$dbh->prepare("insert into students values(100,'thamarai')"); # create  
the table  
$sth->execute();  
$dbh->disconnect;
```

**Compile the perl:**

```
[root@fosslab ~]# perl dbinsert.pl
```

**Output:**

```
mysql> select * from students;
```

rollno	sname
100	thamarai

1 row in set (0.00 sec)

**Program 8. Insert the values in Perl using execute statement**

```
#!/usr/bin/perl  
use DBI; #to use the build in package we use "Use", DBI is the build in  
package in perl  
$rollno=200;  
$sname="selvi";  
my $dbh=DBI->connect("dbi:mysql:student","root","");
#connect to  
database  
if(!$dbh)  
{  
die("error:$!");  
}  
$sth=$dbh->prepare("insert into students values(?,?)");
#create the table  
$sth->execute($rollno,$sname);  
$dbh->disconnect;
```

**Compile:**

```
[root@fosslab ~]# perl dbinsert1.pl
```

**Output:**

```
mysql> select * from students;
```

rollno	sname
100	thamarai
200	selvi

2 rows in set (0.08 sec)

**RESULT:**

Thus the program for perl using mysql connectivity is executed and verified successfully.

**EX:NO: 10**

## **PHP & MYSQL CONNECTIVITY**

**DATE:**

**Aim:**

To create a login form and test the connectivity with MYSQL

### **Introduction:**

PHP stands for **PHP: Hypertext Preprocessor**. PHP is a general-purpose scripting language originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source

document and interpreted by a web server with a PHP processor module, which generates the webpage document. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP was originally created by Rasmus Lerdorf in 1995. The main implementation of PHP is now produced by The PHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.

PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)

### **Uses of PHP :**

- a) PHP runs on different platforms (Windows, Linux, Unix, etc.)
- b) PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- c) PHP is FREE to download from the official PHP resource: [www.php.net](http://www.php.net)
- d) PHP is easy to learn and runs efficiently on the server side

**Steps to execute php programs :**

- a. Open the vi editor and write the program
- b. Save it using .php extension
- c. Execute using the command php filename.php

**Steps to execute php programs involving html forms :**

1. Open the vi editor and write the program
2. Save it using .php extension with the html forms using .html extension in the directory /var/www/html
3. Open the browser and type the URL as : http://localhost/formname.html

## **PHP PROGRAMS**

```
[fosslab@fosslab ~]$ su -  
Password: (admin123)  
[root@fosslab ~]# cd /var/www/html/  
[root@ fosslab html]# gedit form1.html (or) vi form1.html
```

**Example 1:**

**Program:form1.html**

```
<html>  
<body>  
<form action="welcome.php" method="post">  
Name: <input type="text" name="fname" />  
Age: <input type="text" name="age" />  
<input type="submit" />  
</form>  
</body>  
</html>
```

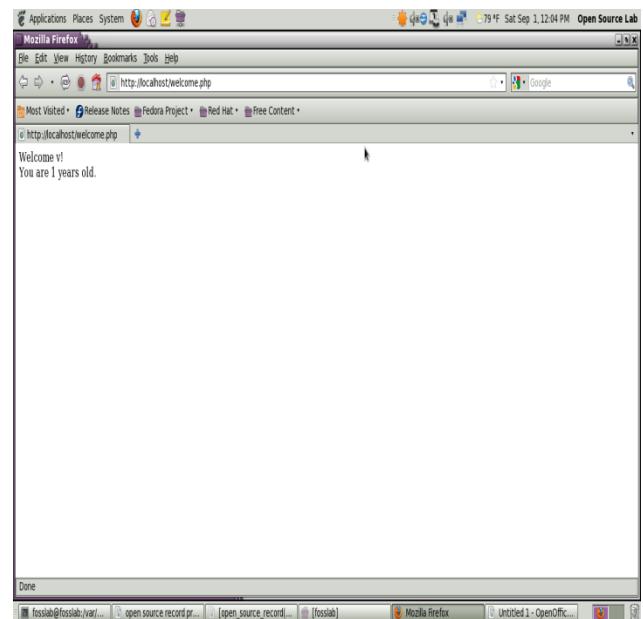
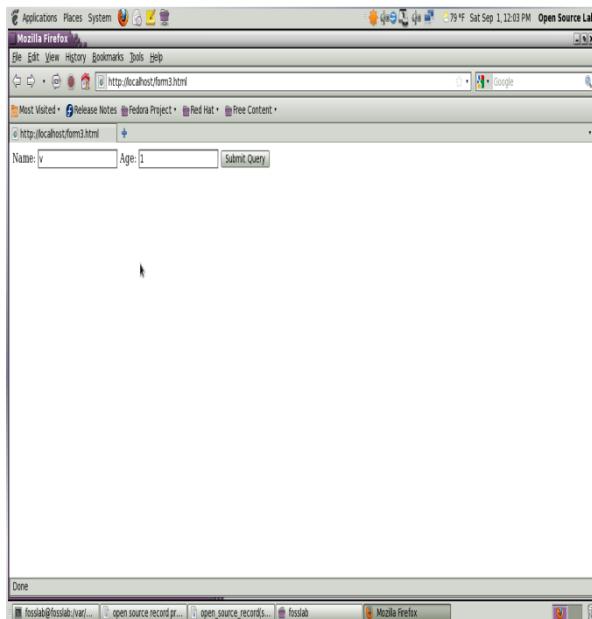
**Welcome.php**

```
<html>  
<body>  
Welcome <?php echo $_POST["fname"]; ?>!<br />  
You are <?php echo $_POST["age"]; ?> years old.  
</body>  
</html>
```

**Output:**

OPEN THE BROWSER AND TYPE THE FOLLOWING PATH

<http://localhost/form1.html>



## Example 2:

### Program:form2.html

```
<html>
<head>
<title>LOGIN</title></head>
<body>
<form action="connect.php" method="post">
<p>"Enter course no"<input type="text" name="cid"></p>
<p>"enter the coursename"<input type="text" name="cname"></p>
<p>"click here to submit"<input type="submit" name="submit"></p>
</form>
</body>
Connect.php:
<?php
$cid=$_POST['cid'];
$cname=$_POST['cname'];
$con=@mysql_connect("localhost","root","");
or die (mysql_error());
echo "connected to database";
$db=@mysql_select_db("student",$con);
or die(mysql_error());
echo "selected database";
$str="insert into courses1 values($cid,$cname)";
$res=@mysql_query($str) or die(mysql_error());
if($res>0)
{
echo "Record created";
}
```

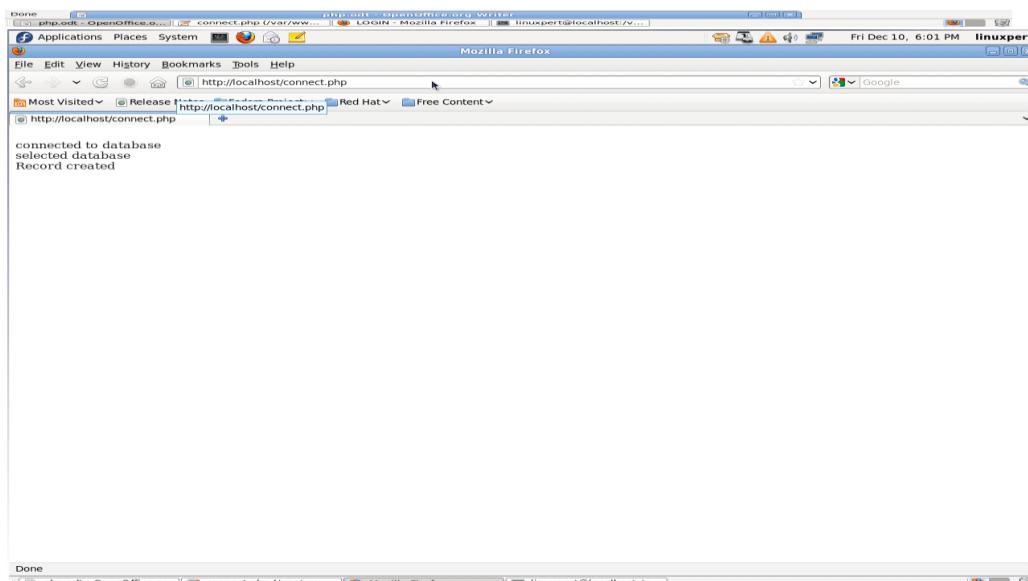
?>

### **Output:**

TYPE THE FOLLOWING IN THE BROWSER

<http://localhost/form2.html> When u press the submit button

A screenshot of a Mozilla Firefox browser window. The title bar says "LOGIN - Mozilla Firefox". The address bar shows the URL "http://localhost/form2.html". The page content contains a form with two input fields: one for "Enter course no" containing "1" and another for "enter the coursename" containing "java". Below the form is a button labeled "click here to submit" with the text "Submit Query".



### **RESULT:**

Thus the program for php is executed and verified successfully.

## **PHP MYSQL CONNECTIVITY PROGRAMS**

## PROCEDURE FOR PHP MySQL CONNECTIVITY

### Step 1:Open the Terminal

#### Go to terminal:

```
[fosslab@fosslab ~]$ su -  
Password: (admin123)  
[root@fosslab ~]# mysql  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 7  
Server version: 5.1.51 Source distribution
```

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.  
This software comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to modify and redistribute it under the GPL v2 license  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

### Step 2:Show the databases

```
mysql> show databases;
```

```
+-----+  
| Database      |  
+-----+  
| information_schema |  
| TESTDB        |  
| college       |  
| employee      |  
| mysql         |  
| sample        |  
| steel         |  
| student       |  
| student1      |  
| test          |  
+-----+  
10 rows in set (0.00 sec)
```

```
mysql> use student;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
Database changed
```

### Step 3:Show the Tables

```
mysql> show tables;
```

```
+-----+  
| Tables_in_student |  
+-----+  
| EMPLOYEE        |  
| courses         |  
| courses1        |  
| veni            |  
+-----+
```

```
| veni1      |
+-----+
5 rows in set (0.00 sec)
```

### **Example 3. Create login page**

#### **Create Login Page (HomePage.html):**

```
<table width="300" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#CCCCCC">
<tr>
<form name="form1" method="post" action="checklogin.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1"
bgcolor="#FFFFFF">
<tr>
<td colspan="3"><strong>Member Login </strong></td>
</tr>
<tr>
<td width="78">Username</td>
<td width="6">:</td>
<td width="294"><input name="myusername" type="text"
id="myusername"></td>
</tr>
<tr>
<td>Password</td>
<td>:</td>
<td><input name="mypassword" type="text" id="mypassword"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td><input type="submit" name="Submit" value="Login"></td>
</tr>
</table>
</td>
</form>
</tr>
</table>
```

#### **Create Validation Page (checklogin.php):**

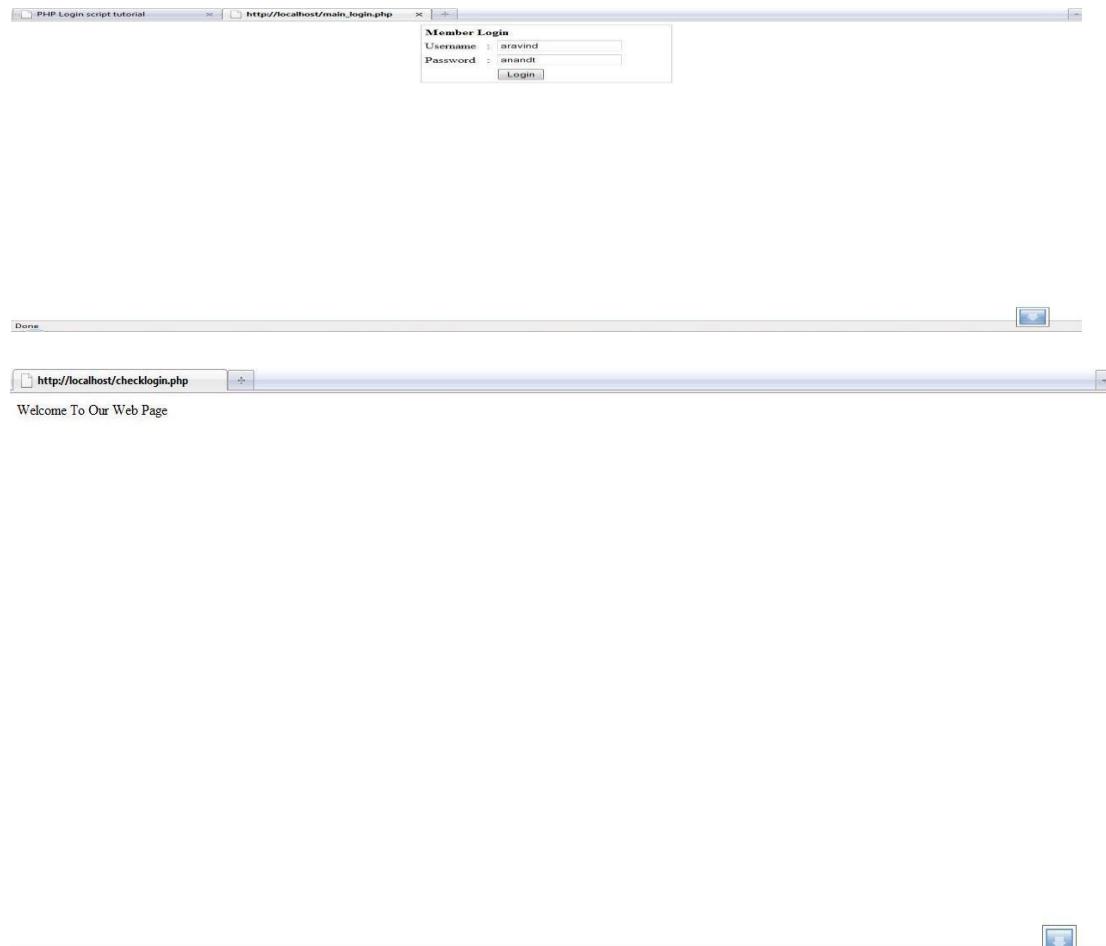
```
<?php
$host="localhost"; // Host name
$username="root"; // Mysql default username
// $password="" ; // Mysql No password
$db_name="student"; // Database name
$tbl_name="members"; // Table name
// Connect to server and select database.
mysql_connect("$host", "$username")or die("cannot connect");
```

```
mysql_select_db("$db_name")or die("cannot select DB");
// username and password sent from form
$username=$_POST['myusername'];
$password=$_POST['mypassword'];
$sql="SELECT * FROM $tbl_name WHERE
username='".$username. "' and password='".$password."'";
$result=mysql_query($sql);
// Mysql_num_row is counting table row
$count=mysql_num_rows($result);
// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1)
echo "Welcome To Our Web Page";
else
echo "Wrong Username or Password";
?>
```

### **Output:**

TYPE THE FOLLOWING IN THE BROWSER

<http://localhost/HomePage.html>



**Example 4.php program for create database**

```
<?php  
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
else
{ echo "ok data base"; }
mysql_close($con);
?>
```

**Output:**

```
[root@fosslab html]# php connectdb.php
ok data base
```

**Program 5:php program to create a table in the database**

```
[root@fosslab html]# vi cretable.php
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
else
{ echo "ok data base"; }
mysql_select_db("student", $con);
$sql = "CREATE TABLE Per
(
FirstName varchar(15),
LastName varchar(15),
Age int
)";
echo "table create";
// Execute query
mysql_query($sql,$con);
mysql_close($con);
?>
```

**Output:**

```
[root@fosslab html]# php cretable.php
ok data basetable create
```

**Open the another terminal**

Type the following commands on the another terminal

**mysql> show databases;**

```
+-----+
| Database      |
+-----+
| information_schema |
| TESTDB        |
```

```
| college      |
| employee     |
| mysql        |
| sample       |
| steel        |
| student      |
| student1    |
| test         |
+-----+
10 rows in set (0.00 sec)
```

**mysql> use student;**

Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A

Database changed

**mysql> show tables;**

```
+-----+
| Tables_in_student |
+-----+
| EMPLOYEE      |
| Per           |
| courses       |
| courses1     |
| veni          |
| veni1         |
+-----+
6 rows in set (0.00 sec)
```

**RESULT:**

Thus the program for php using mysql database is executed and verified successfully.

**EX:NO:11**

### **PYTHON & MYSQL CONNECTIVITY**

**DATE:**

**Aim:**

To demonstrate the Python programs and Python program connectivity with MYSQL database.

#### **Introduction**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Python is a high-level programming language. It is an easy to learn, powerful programming language. It has efficient, high-level data structures and a simple yet effective approach to object oriented programming.

Python's elegant syntax and dynamic typing, together with its interpreted nature, makes it an ideal language for scripting and rapid application development in many areas and on most platforms. The Creator of this Language is Guido van Rossum. He named this language after the BBC show "Monty Python's Flying Circus". But he doesn't particularly like snakes that kill animals for food by winding their long bodies around them and crushing them.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved open source license.

**Steps to execute python programs:**

1. Open the `vi` editor and write the program
2. Save the program `with .py` as extension
3. Execute the program using `python filename.py`

## PYTHON PROGRAMS

**Program 1:**

```
#!/usr/bin/python
#program to select odd number from the list
a=[11,12,13,14,15,16,17,18,19,20,21,31,44,45,10];
print("List is:",a);
n=len(a);
print("length:",n);
i=0;
print("Odd number");
for i in range(len(a)):
    if(a[i]%2==1):
        print(a[i]);
```

**Output:**

```
[root@fosslab html]# python odd.py
('List is:', [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 31, 44, 45, 10])
('length:', 15)
Odd number
11 13 15 17 19 21 31 45
```

**Program 2: Program to display the celsius value\**

```
#!/usr/bin/python
a=input("enter the celsius value:")
f=(a*1.8)
b=f+32;
print b
```

**Output:**

```
[fosslab@fosslab ~]$ python celsius.py
```

```
enter the celsius value:23
```

```
73.4
```

**Program 3: Program to display the fibonacciseries values**

```
#!/usr/bin/python
a, b = 0, 1
while b < 200:
    print b,
    a, b = b, a+b
```

**Output:**

```
[fosslab@fosslab ~]$ python fibanoo.py
```

```
1 1 2 3 5 8 13 21 34 55 89 144
```

**Program 4:**

```
#!/usr/bin/python
a = int(raw_input("Please enter an integer: "));
if (a % 2 == 1):
    print 'a' + ' is odd.'
else:
    print 'a' + ' is even.'
```

**Output:**

```
[fosslab@fosslab ~]$ python odd.py
```

```
Please enter an integer: 23
```

```
a is odd.
```

**RESULT:** Thus the program for python is executed and verified successfully.

## PYTHON MYSQL CONNECTIVITY PROGRAMS

**Program 5: CREATE TABLE PROGRAM USING PYTHON**

```
#!/usr/bin/python
```

```
import MySQLdb

# Open database connection
db = MySQLdb.connect("localhost","root","","student" )

# prepare a cursor object using cursor() method
cursor = db.cursor()

# Drop table if it already exist using execute() method.
cursor.execute("DROP TABLE IF EXISTS student ")

# Create table as per requirement
sql = """CREATE TABLE student ( FIRST_NAME CHAR(20) NOT NULL, LAST_NAME
CHAR(20), AGE INT, SEX CHAR(1),
cursor.execute(sql)

# disconnect from server
db.close()
```

### **Output:**

**Go to terminal:**

```
[fosslab@fosslab ~]$ su -
Password: (admin123)
[root@fosslab ~]# mysql -u root
```

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 26

Server version: 5.1.51 Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

This software comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

**mysql> show databases;**

```
+-----+
```

```
| Database      |
+-----+
| information_schema |
| TESTDB        |
| college       |
| employee      |
| mysql          |
| sample         |
| steel          |
| student        |
| test           |
+-----+
```

9 rows in set (0.00 sec)

**mysql> use student;**

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

**mysql> connect;**

Connection id: 27

Current database: student

**mysql> show tables;**

```
+-----+
| Tables_in_student |
+-----+
| EMPLOYEE        |
| courses         |
| student         |
+-----+
```

3 rows in set (0.00 sec)

**Program 6: Insert the values in Python that will be automatically updated in database using mysql as follows**

```
#!/usr/bin/python
```

```
import MySQLdb  
# Open database connection  
db = MySQLdb.connect("localhost","root","","","student" )  
# prepare a cursor object using cursor() methods  
cursor = db.cursor()  
# Prepare SQL query to INSERT a record into the database.  
sql = """INSERT INTO student(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)  
VALUES ('Mac', 'Mohan', 20, 'M', 2000)"""  
try:  
    # Execute the SQL command  
    cursor.execute(sql)  
    # Commit your changes in the database  
    db.commit()  
except:  
    # Rollback in case there is any error  
    db.rollback()  
# disconnect from server  
    db.close()
```

### **Output:**

```
mysql> show databases;  
+-----+  
| Database      |  
+-----+  
| information_schema |  
| TESTDB        |  
| college        |  
| employee       |  
| mysql          |  
| sample         |  
| steel          |  
| student        |  
| test           |
```

```
+-----+
9 rows in set (0.00 sec)

mysql> use student;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed

mysql> connect;
Connection id:  27
Current database: student

mysql> show tables;
+-----+
| Tables_in_student |
+-----+
| EMPLOYEE          |
| courses           |
| student           |
+-----+
3 rows in set (0.00 sec)

mysql> select * from student;
+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+
| Mac       | Mohan     | 20 | M   | 2000 |
| Malar    | vizhi     | 24 | F   | 15000 |
| Rahul    | kannan    | 23 | M   | 12000 |
| Raksha   | kumar     | 24 | F   | 16500 |
+-----+-----+-----+-----+
4row in set (0.02 sec)
```

### **Program 7: Fetch some useful information from the database**

```
#!/usr/bin/python
import MySQLdb
# Open database connection
```

```
db = MySQLdb.connect("localhost","root","","student" )  
# prepare a cursor object using cursor() method  
cursor = db.cursor()  
# Prepare SQL query to INSERT a record into the database.  
sql = "SELECT * FROM STUDENT \WHERE INCOME > '%d'" % (1000)  
try:  
    # Execute the SQL command  
    cursor.execute(sql)  
    # Fetch all the rows in a list of lists.  
    results = cursor.fetchall()  
    for row in results:  
        fname = row[0]  
        lname = row[1]  
        age = row[2]  
        sex = row[3]  
        income = row[4]  
        # Now print fetched result  
        print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" % \  
              (fname, lname, age, sex, income )  
except:  
    print "Error: unable to fetch data"  
# disconnect from server  
db.close()
```

### **Output:**

```
mysql> show databases;  
+-----+  
| Database      |  
+-----+  
| information_schema |  
| TESTDB         |  
| college        |  
| employee       |
```

```
| mysql      |
| sample     |
| steel      |
| student    |
| test       |
+-----+
```

9 rows in set (0.00 sec)

**mysql> use student;**

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

**mysql> connect;**

Connection id: 27

Current database: student

**mysql> show tables;**

```
+-----+
| Tables_in_student |
+-----+
| EMPLOYEE      |
| courses       |
| student       |
+-----+
```

3 rows in set (0.00 sec)

**mysql> select \* from student;**

```
+-----+-----+-----+-----+
| FIRST_NAME | LAST_NAME | AGE | SEX | INCOME |
+-----+-----+-----+-----+
| Mac      | Mohan    | 20 | M   | 2000 |
| Malar    | vizhi    | 24 | F   | 15000 |
| Rahul    | kannan   | 23 | M   | 12000 |
| Raksha   | kumar    | 24 | F   | 16500 |
+-----+-----+-----+-----+
```

4row in set (0.02 sec)

**Program 8: Updating the values in a database**

```
#!/usr/bin/python
import MySQLdb
# Open database connection
db = MySQLdb.connect("", "root", "", "student" )
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to UPDATE required records
sql = "UPDATE student SET AGE = AGE + 1 WHERE SEX = '%c'" % ('M')
try:
    # Execute the SQL command
    cursor.execute(sql)
    # Commit your changes in the database
    db.commit()
except:
    # Rollback in case there is any error
    db.rollback()
# disconnect from server
db.close()
```

**Program 9: Delete some values from database**

```
#!/usr/bin/python
import MySQLdb
# Open database connection
db = MySQLdb.connect("localhost", "testuser", "test123", "TESTDB" )
# prepare a cursor object using cursor() method
cursor = db.cursor()
# Prepare SQL query to DELETE required records
sql = "DELETE FROM student WHERE AGE > '%d'" % (25)
try:
    # Execute the SQL command
    cursor.execute(sql)
```

```
# Commit your changes in the database
db.commit()

except:
    # Rollback in case there is any error
    db.rollback()

# disconnect from server
db.close()
```

#### **RESULT:**

Thus the program for python and mysqlconnectivity is executed and verified successfully.

**Ex.No: 12**

## NETWORK INTERFACE CONFIGURATION

**DATE:**

**Aim :** To Set up the complete network interface using ifconfig command like setting gateway, DNS, IP tables, etc.,

**Procedure :**

Step1 : Enter into the Super User

```
[fosslab@fosslab ~]$ su -
```

Password: (admin123)

step2: To display the interfaces available in our system

```
[root@fosslab ~]# ifconfig
```

step3: To display available interface (if ethernet1) interface details

```
[fosslab@fosslab ~]$ ifconfig eth1
```

step4: To display ethernet1 broadcast address

```
[root@fosslab ~]# ifconfig eth1 broadcast 192.168.255.255
```

step5: To Reset the ethernet1 broadcast addressss

```
[root@fosslab ~]# ifconfig eth1 broadcast 192.168.255.250
```

step6: To display the ethernet1 interface

```
[root@fosslab ~]# ifconfig eth1
```

step 7: To drop the ethernet1 address

```
[root@fosslab ~]# ifconfig eth1 down
```

step:8 To update the ethernet1 interface

```
[root@fosslab ~]# ifconfig eth1 up
```

step:9 To set the address for ethernet and netmask

```
[root@fosslab ~]# ifconfig eth0 192.168.1.25 netmask 255.255.0.0
```

step:10 to display the ethernet 0 interface

```
[root@fosslab ~]# ifconfig eth0
```

Step11: to display the router table

```
[root@fosslab ~]# route
```

step12: to display the format for ip address

```
[root@fosslab ~]# route -n
```

Step13: To set the default gateway address

```
[root@fosslab ~]# route add default gw 192.168.7.1
```

step14: To display the new route table

```
[root@fosslab ~]# route
```

step15: to display the physical connectivity

```
[root@fosslab ~]# mii-tool
```

step 16: To connect the network status

```
[root@fosslab ~]# service network status
```

step 17: To stop the network status

```
[root@fosslab ~]# service network stop
```

step 18: To restart the network status

```
[root@fosslab ~]# service network restart
```

Step19: To display the sysconfig for ethernet1

```
[root@fosslab ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

step20: To display the sysconfig for ethernet1

```
[root@fosslab ~]# vi /etc/sysconfig/network
```

**Result:** Thus Setting up the complete network interface using ifconfig command can be successfully completed.