

Challenge of Code 2010
Association of Computing Machinery
Bangalore Chapter



Open Source Idea Number : 14
LDAP GUI for Academic Institutes

Submitted by

Ashwin Tumma ¹
Swapnil Kulkarni ²
College of Engineering, Pune
Maharashtra
India

¹Email : ashwin.tumma23@gmail.com

²Email : swapnil514@gmail.com

Abstract

LDAP - Lightweight Directory Access Protocol is an application protocol for querying and modifying data of directory services implemented in Internet Protocol(IP) networks. It is a network authentication system. We have worked on it, right from Server Installation to Client Configuration and setup. The whole process is a bit tedious and since it is a complete command line tool, it is sometimes difficult for novice users to understand it.

This tool is an attempt of creating a GUI application for the same, in which novice users/administrators can also use it and learn from it. The GUI application inturn invokes shell scripts and configure the server of the users. It is an Open-Source tool with GPLv2.

The institution can thus make efficient use of its resources. We know, that there are GUI's existing in the market for the same purpose but there isn't a GUI for academic institutions and that too which understands the format of the Indian Engineering Colleges' System. Once, this tool is released, it will cater to many academic institutes in India, and will help to promote use of Linux and Free and Open Source terminologies in depth.

Acknowledgments

We take this as an opportunity to express our deep sense of gratitude to the worldwide **FOSS community** and \LaTeX for making us available such great Open-Source technologies. We shall always remain indebted to Prof. Abhijit A. M., for everything that he has taught us. Right from teaching us what Free and Open Source World is to guiding and motivating us in LDAP Administration in College of Engineering, Pune.

We also thank the Head of Department, other faculty members and all other people in Department of Computer Engineering and Information Technology, College of Engineering, Pune for their support and the faith that they have shown in us. We are grateful to everyone who are directly or indirectly involved in LDAP Administration in our college.

We are thankful to **ACM Bangalore** for hosting this event, and providing us a platform to contribute our bit to the world of Free and Open Source Software.

Ashwin Tumma & Swapnil Kulkarni
November 2010
College of Engineering, Pune

Contents

1	Introduction	1
1.1	What exactly do we mean by using LDAP in Academia?	1
1.2	What made us select this idea?	2
1.2.1	What have we done?	2
1.3	Organization of this Report	3
2	Installing Linux on x86 Machines	4
2.1	Why Linux?	4
2.2	Ubuntu	4
2.2.1	Steps for Installation	4
3	Installing Java on Ubuntu Machines	6
3.1	Java 1.6	6
3.1.1	Steps for Installation	6
4	Installing Web Server on Ubuntu Machines	8
4.1	Why Java Server Pages?	8
4.2	Apache Tomcat 6	8
4.2.1	Steps for Installation	9
5	OpenLDAP and Berkeley Database	10
5.1	OpenLDAP 2.3.30	10
5.1.1	Steps for Installation	10
5.2	Berkeley Database 4.3.29	11
5.2.1	Steps for Installation	11
6	Requirements	12
6.1	Software Requirements	12
6.2	Hardware Requirements	12
7	LDAP GUI Tool - User Manual	13
7.1	Assumptions	13
7.2	ServerSide Tool	14
7.2.1	Installing OpenLDAP and Berkeley DB	14
7.2.2	Specify Server Configurations	15
7.2.3	Create the Directory Information Tree	15
7.2.4	Start Server and add files	16
7.2.5	Create User Accounts	16
7.3	ClientSide Tool	17
7.3.1	Package - libnss-ldap	17

7.3.2	Edit the configuration files	18
7.3.2.1	Details	19
8	Future Enhancements	20

List of Figures

1.1	Rough Sketch of LDAP	2
2.1	Ubuntu Logo	4
3.1	Java Logo	6
4.1	Apache Tomcat Logo	8
4.2	First Page that is displayed on successful configuration of Apache Tomcat	9
5.1	OpenLDAP Logo	10
5.2	Oracle Berkeley DB Logo	11
7.1	Screenshot of Homepage	13
7.2	Get values of basic parameters for the Server	14
7.3	Installation of OpenLDAP and Berkeley DB	15
7.4	Slapd.conf creation status and input values of departments	16
7.5	Build the Directory Information Tree	17
7.6	Screenshot of Client Installation procedure	18

Chapter 1

Introduction

1.1 What exactly do we mean by using LDAP in Academia?

LDAP stands for Light-weight Directory Access Protocol. In this report, we won't be describing the technical definitions and descriptions of LDAP, they are available at [7, 9]. Rather we explain what are the intricacies and what exactly do we mean by using LDAP in academic institutes.

In academic institutes, various users of the computer systems are : Students, Faculty and Non-teaching staff members. The initial investment on building the computer resources is very high. But most of the times, due to inefficient usage and management of the resources, the users do not reap the benefits of the systems to the hilt. This results in lower productivity and ever increasing requirement of storage space and other resources. Also, if one machine is considered, all the users access the same account[Say for example, all students access *student* account]. Thus, there is no security between users and no provision for customization of machines as per users' convenience.

Using LDAP will solve the above mentioned problem very easily. Basically, in LDAP what we do is, configure a server machine and all the clients connect to it for their authentication. The server machine has details of all the user accounts in its backend database. When the user wants to access the computer system, he/she enters his/her login credentials and can have access to their own accounts. The authentication system of the client machine, will consult the LDAP server in its authentication process. The server will inturn use its database for authentication. Figure 1.1 shows a rough idea as to what LDAP is all about. Now, users can have their own accounts, customize it the way they want, etc. Thus, we are leveraging the benefits of the existing system in a more efficient manner.

Rough skeleton of the LDAP Installation Process : We first install the softwares for LDAP Server[OpenLDAP] and the backend database[Oracle Berkeley DB] on the server machine. A Directory Information Tree of all the users is formed which consists the hierarchy in which the users are present in the institute. For e.g.: The users can be grouped together on the bases of their associated department. User accounts and groups need to be created on the server machine. Then, all the other machines need to be configured in such a way that they can act as LDAP clients. Certain packages need to be installed on clients for their functions. Finally, the client machines need to be instructed the way in which they should authenticate their users. Currently, the whole process has to be done manually, and it becomes quite difficult for a novice user to understand all

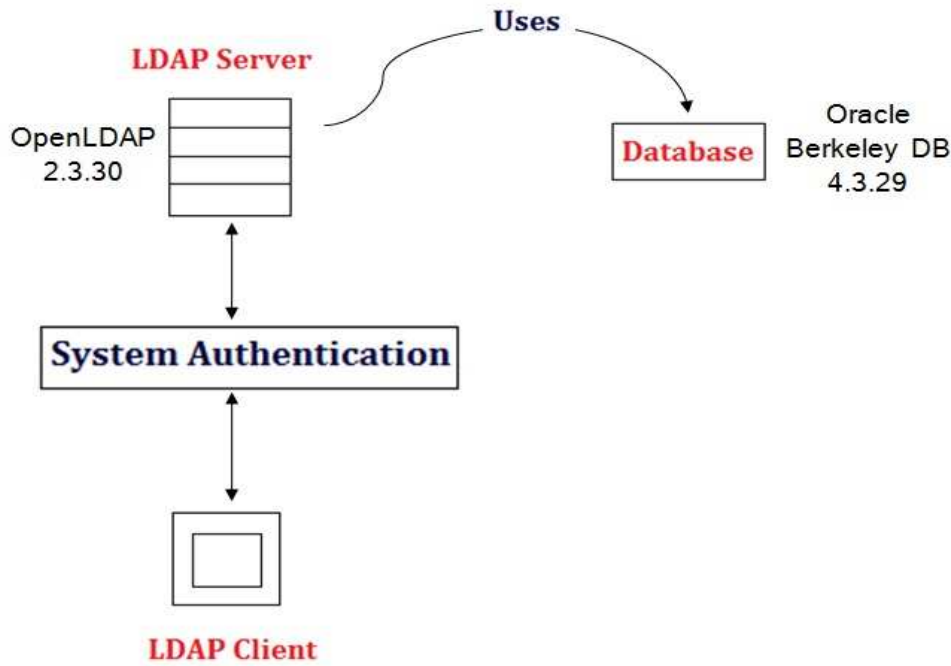


Figure 1.1: Rough Sketch of LDAP

the technical details and intricacies of the system.

1.2 What made us select this idea?

As mentioned in the previous section, LDAP Installation and configuration is a tedious process and there are no graphical user interfaces available for configuring LDAP servers specially that will cater to the academic institutes. We, at College of Engineering, Pune have worked on it, right from Server Installation to Client Configuration and setup. We found the whole process is a bit complex and since it is a complete command line tool, it is sometimes difficult for novice users to understand it. So, we thought of creating a GUI for LDAP Installation and configuration, so that beginners can start with using it and then later master the techniques and details of LDAP. This will also lead to the promotion of Open Source Technologies to the masses.

1.2.1 What have we done?

We have developed a web based application that can be accessed from the server machine which simplifies the installation and configuration procedure. Shell scripts are provided to automate the tasks of installations. The shell scripts are invoked as and when they are needed throughout the process. Almost each step is explained with respect to the inputs it takes, and the outputs it will generate. Wherever required, assumptions are mentioned. Footnotes are provided to provide external details of the associated point.

We have taken utmost care of explaining the source codes by means of comments. The user is advised to go through the comments in the shell scripts and other codes before proceeding with the installations. We have tried our best to explain the esoteric concepts in a lucid manner and hope this will assist users to gain a clear understanding of the entire installation procedure.

Certain pre-requisite softwares are also required for the execution of this tool. Complete details of such softwares are explained in the initial chapters which is followed by the **User Manual** and Future Enhancements that can be added to this tool.

1.3 Organization of this Report

- Chapter 2 describes the steps for installation of Linux Distribution (Ubuntu)[10] on x86 Machines
- Chapter 3 describes the steps for installation of Java on Ubuntu Machines [4]
- Chapter 4 describes the steps for installation of Web Server - Apache Tomcat on Ubuntu Machines [1]
- Chapter 5 gives a very brief introduction of OpenLDAP[7] and Berkeley Database[2]
- Chapter 6 highlights the *nominal* software and hardware requirements for our tool
- Chapter 7 describes the tool that we have developed. This chapter can be considered as an User Manual for the tool
- Chapter 8 discusses future enhancements that can be added to this tool.

Chapter 2

Installing Linux on x86 Machines

2.1 Why Linux?

Linux refers to the family of Unix-like computer operating systems using the Linux Kernel.

- Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and video game consoles, to mainframes and supercomputers.
- It is a leading server operating system.
- The development of Linux is one of the most prominent examples of free and open source software collaboration, typically all the underlying source code can be used, freely modified, and redistributed, both commercially and non-commercially, by anyone under licenses such as the GNU General Public License.[3]

2.2 Ubuntu

We have used the Linux distribution which is most popular in the student community, Ubuntu. Since it supports Debain packages on it, installation and configuration of any new software is comparatively easy on Ubuntu machines.



Figure 2.1: Ubuntu Logo

Assumption : We assume that the user of our tool will be working on x86 architecture machines. The following steps of installing Ubuntu are documented with this in mind. Others can refer to [10] for further details.

2.2.1 Steps for Installation

- Ubuntu can be obtained freely from Official Ubuntu website[10]. The version that we have chosen for our work is Ubuntu 10.04 - Lucid Lynx. CD/DVD can be obtained or an image of the installation disk can be downloaded.

- Insert the CD/DVD in the CD-ROM drive of the system and reboot the system.
- The machine will automatically *boot from CD* or can be made to do so by choosing appropriate options from the BIOS.
- Select the language of installation. E.g.: English
- Select the keyboard layout of your machine. E.g.: English[US]
- You can allocate the entire disk space to Ubuntu or choose to specify partitions manually.
- If you chose to specify partitions manually, then you need to allocate space of **root** directory and *swap* area. You can also allocate space for other partitions like **/boot** and **/home**.
- You be prompted for creation of an administrative account and other trivial details.
- Click on Start Installation.
- After installation procedure is complete, you need to remove the CD/DVD from the CD-ROM drive and restart your machine.
- You can now login with your administrative account. If required, create other users.

The above steps are lucidly explained at [8].

Chapter 3

Installing Java on Ubuntu Machines

3.1 Java 1.6



Figure 3.1: Java Logo

We will require Java Installation on our machine because being a web-based application our GUI application will require to run a web-server 4. Reason of why we have chosen Java as our development platform :

- One of the highly adopted language worldwide
- SDK's are readily available and are Open-Source
- High Portability across many platforms and architectures
- Web-Servers easy to manage and configure

3.1.1 Steps for Installation

Java can be obtained freely from Official Java website[4]. The version that we have chosen for our work is Java 1.6 . There are two options for installing Java on our machines.

1. Download the package from [4] and installing it manually.

2. Using *apt-get* to download the appropriate *.deb* file from the repositories. The steps that you need to follow in this approach are :

- Enter the following command on the terminal
`sudo apt-get install openjdk-6-jdk`
- This will automatically connect to the repositories, download the package for you and install the same. ¹

We suggest that the naïve users opt for the latter approach, which is comparatively easier than the former one.

¹*Assumption:* We assume that you have your `/etc/apt/sources.list` configured in such a way reflects the repository required for locating the Java package. By default, Ubuntu has taken care of this issue, and you can proceed with your installation. If any changes are done to it and the Java Installation gives errors, for details, please refer to [10]

Chapter 4

Installing Web Server on Ubuntu Machines

Being a web based application, we will have to host a web server which will act as our Graphical User Interface for our LDAP GUI tool. As mentioned in 3.1, we have chosen Apache Tomcat as our Web Server[1].

4.1 Why Java Server Pages?

Java Server Pages is chosen for the GUI creation because :

- Leverages the benefits of basic HTML
- Easy to program and configure
- Large API's are readily available
- Simpler to execute Shell Scripts from the Web Page directly

4.2 Apache Tomcat 6



Figure 4.1: Apache Tomcat Logo

Apache Tomcat (or Jakarta Tomcat or simply Tomcat) is an open source servlet container developed by the Apache Software Foundation (ASF) [1]. Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run. Tomcat should not be confused with the Apache web server, which is a C implementation of an HTTP web server; these two web servers are not bundled together.

4.2.1 Steps for Installation

Apache Tomcat can be obtained freely from Official Apache website[1]. The version that we have chosen for our work is Apache Tomcat 6.0.29 .

1. Download the package named `apache-tomcat-6.0.29.tar.gz` from [1].
2. Extract the contents of this tarball at `/usr/local/`¹
3. Start the Web Server.

```
sudo /usr/local/apache-tomcat-6.0.29/bin/startup.sh
```

If the server was properly configured, it will show the values of various environment variables that are used by the Tomcat Server. [E.g.: `CATALINA_BASE`, `CATALINA_HOME`, `CATALINA_TMPDIR`, `JRE_HOME`, `CLASSPATH`]
4. Point your browser to `http://localhost:8080/` to test your installation. The screen should look like the one shown in Figure 4.2
5. The directory where you should place your HTML and JSP files is :
`/usr/local/apache-tomcat-6.0.29/webapps/ROOT`

Trivia

1. E.g.: If we create a file named `Hello.html` then we need to place it in the `/usr/local/apache-tomcat-6.0.29/webapps/ROOT` directory and to access this file via the web server, point your browser to `http://localhost:8080/Hello.html`
2. By default, if no file name is specified, `http://localhost:8080/` then the Web Engine tries to find `index.html` file in the ROOT directory of the Web Server and runs it. 8080 is the port number on which Apache Tomcat runs.

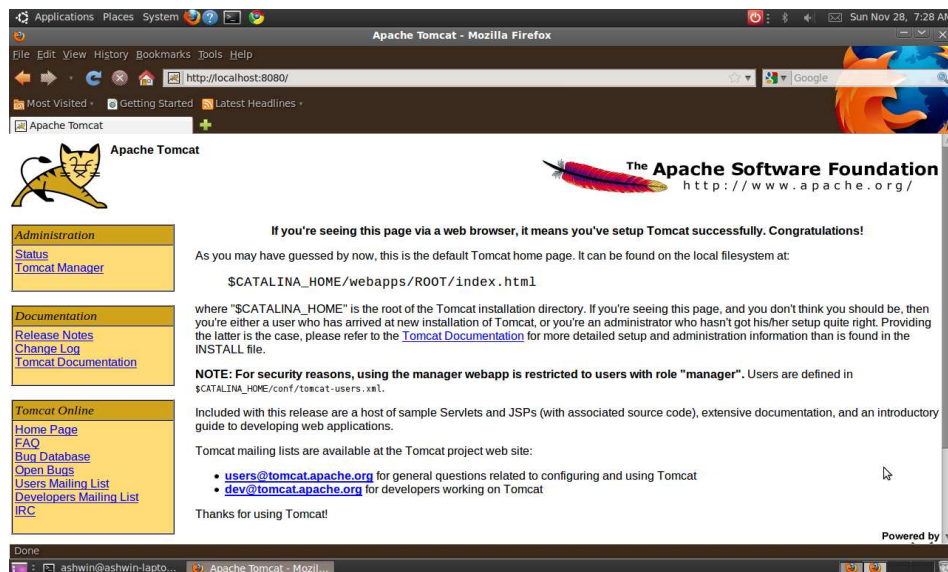


Figure 4.2: First Page that is displayed on successful configuration of Apache Tomcat

¹Note that you need to be a root or *sudo* user to "play" in this directory

Chapter 5

OpenLDAP and Berkeley Database

5.1 OpenLDAP 2.3.30



Figure 5.1: OpenLDAP Logo

OpenLDAP is an open source implementation of the Light Weight Directory Access Protocol. The suite includes:

- *slapd* - stand-alone LDAP Daemon(server)
- libraries - implementing the LDAP and
- utilities, tools and sample clients.

We use OpenLDAP for configuring the LDAP Server and Clients. OpenLDAP also provides an Administrator's Guide that covers all the *ins & outs* and *lengths & breadths* of LDAP installation and administration.

5.1.1 Steps for Installation

OpenLDAP can be obtained freely from Official OpenLDAP website[7]. The version that we have chosen for our work is OpenLDAP 2.3.30.

¹

1. Download the tarball from [7].
2. Extract the contents to /extra directory and install it by compiling the source.

¹The whole process of installation of OpenLDAP and Berkeley DB is automated in our tool using the shell script named `Install.sh`

5.2 Berkeley Database 4.3.29



Figure 5.2: Oracle Berkeley DB Logo

Berkeley DB enables the development of custom data management solutions, without the overhead traditionally associated with custom projects.

5.2.1 Steps for Installation

Berkeley DB can be obtained freely from Official Oracle Berkeley DB website[2]. The version that we have chosen for our work is Berkeley DB 4.3.29

1. Download the tarball from [2].
2. Extract the contents to /extra directory and install it by compiling the source.

Chapter 6

Requirements

Following are the pre-requisites for the tool. Note that those in **bold** are required to be present only on the machine on which we are configuring the LDAP Server.

6.1 Software Requirements

- Operating System - Linux [Ubuntu/Debian] [10]
- **Java 1.6** [4]
- **Apache Tomcat 6.0.29** [1]
- **OpenLDAP 2.3.30** [7]
- **Berkeley DB 4.3.29** [2]
- Web Browser with JavaScript Enabled. [Works best with Mozilla Firefoz 3.6 [5]]

6.2 Hardware Requirements

- x86 Machine
- RAM Requirements - *min* 1GB

Chapter 7

LDAP GUI Tool - User Manual

This chapter can be considered as the User Manual for the tool that we have developed.

7.1 Assumptions

- The user is able to run shell scripts at the terminal as a *sudo* user.
- The server has access to Internet Connection for downloading various packages.
- All the machines are connected in a network and can *access(ping)* the server.

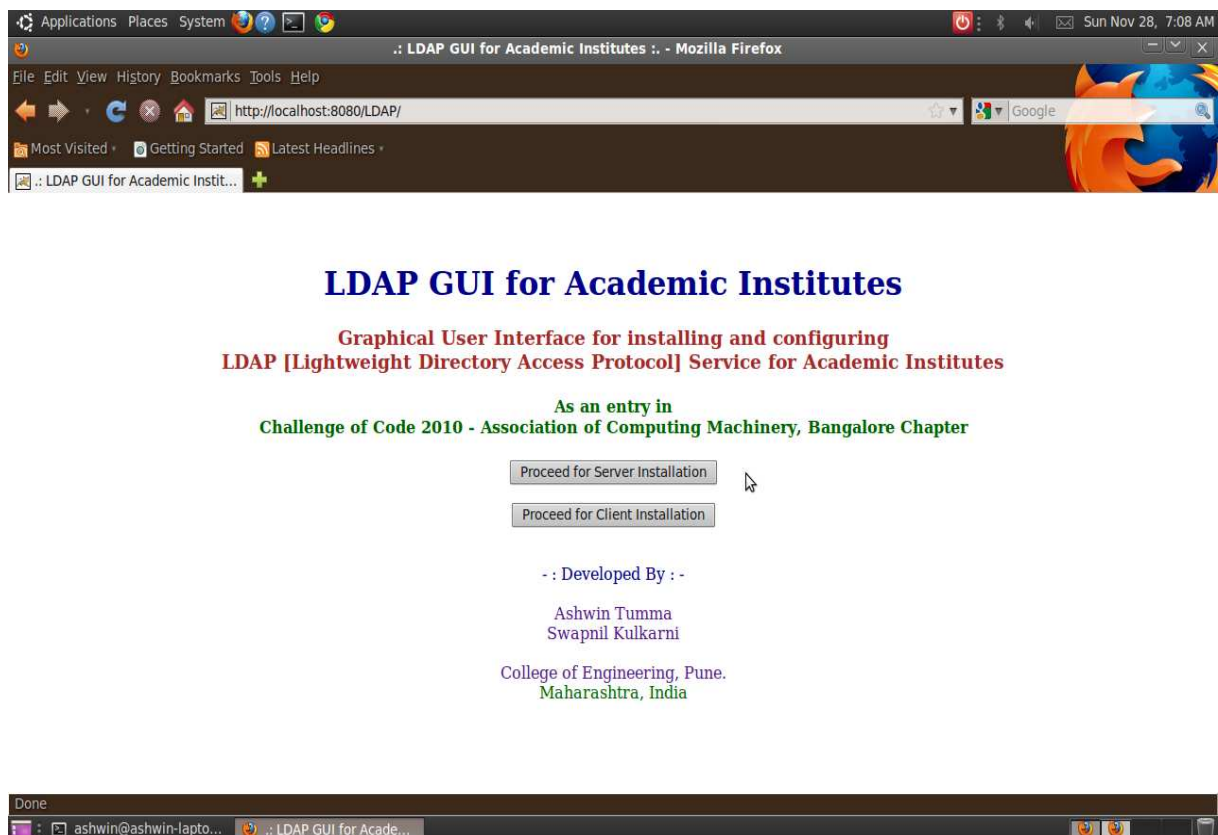


Figure 7.1: Screenshot of Homepage

7.2 ServerSide Tool

The steps to configure a machine as an LDAP Server are as follows:

1. Install OpenLDAP and Berkeley DB.
2. Specify parameters of domain name and other configuration details.
3. Create the Directory Information Tree.
4. Start the server and add the hierarchy and group files to the LDAP Server.
5. Create user accounts according to your needs.

7.2.1 Installing OpenLDAP and Berkeley DB

The steps of installing OpenLDAP and Berkeley DB are explained in 5. The Shell Script named `Install.sh` installs both the softwares.

Parameter	Value	Notes/Example
Server Details		
Name of Institute :	College of Engineering, Pune	Full Name of the Institute
Domain Name:	dc=coep,dc=com	e.g. : dc=coep, dc=com
Details for building Directory Information Tree		
Number of Departments :	2	Number of departments that will use the LDAP Server
<input type="button" value="Start Installation"/>		

Figure 7.2: Get values of basic parameters for the Server

Figure 7.2 shows the screen for getting basic parameters like *Name of Institute*, *Domain Name* and *Number of departments*. On clicking **Start Installation** button, the values you have entered are recorded in the tool and it directs you to screen as shown in figure 7.3. Here, you will see the instructions as to how to run the `Install.sh` script manually. We have deliberately not invoked this shell script from the webpage. If we have done so, the user wouldn't have got to see the actual compilation process of the software installation. On clicking **Start Configuration** button you will be redirected to screen as shown in the figure 7.4.

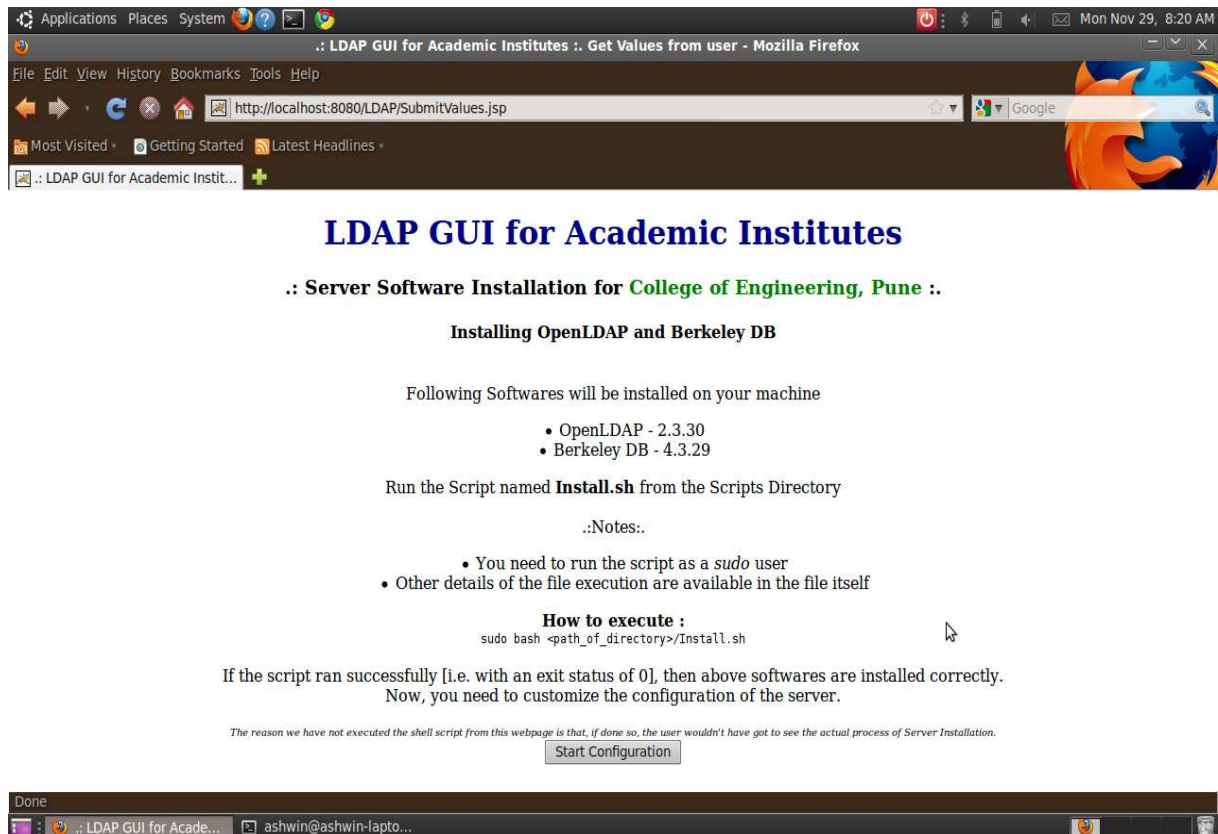


Figure 7.3: Installation of OpenLDAP and Berkeley DB

7.2.2 Specify Server Configurations

As shown in figure 7.4, the `slapd.conf` file is created from the values the user has specified for the server and its status is displayed accordingly. This file is the heart of the server. All the parameters that govern the server running are specified in this file. The shell script named `Make_slapd_conf.sh` will be invoked and a `slapd.conf` file is created as per the parameters specified by the user and is automatically placed at this proper location, *i.e.* at `/usr/local/etc/openldap/slapd.conf`. An administrative account for the server management is created in this file. We have named it as *Manager*.

7.2.3 Create the Directory Information Tree

Next, the tool will then prompt for the names of the departments for which you want to use LDAP. The number of departments value is taken from the initial pages. On clicking the **Build Directory Information Tree** button, you will be redirected to the screen as shown in the figure 7.5. Here the status of your creation of Directory Information Tree is displayed. LDAP needs files of the form `.ldif` - LDAP Directory Interchange Format. A file named `hierarchy.ldif` is created which will showcase your tree and group files are created for the various uses and departments that you have specified. The shell script `CreateLDIF.sh` will create the `hierarchy.ldif` and other group files. You can refer to the shell script for more better explanation. Now the server has to be started and the `.ldif` files just created need to be added to the server.

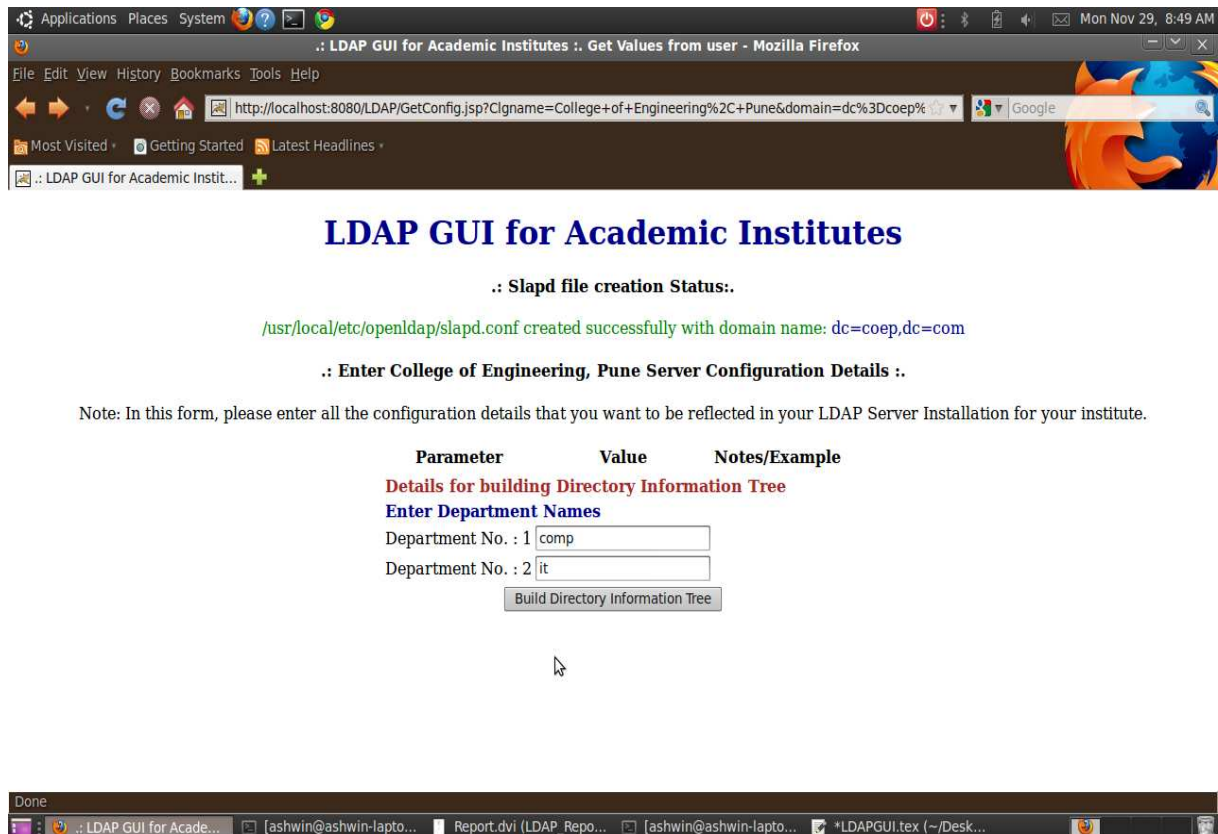


Figure 7.4: Slapd.conf creation status and input values of departments

7.2.4 Start Server and add files

As shown in figure 7.5 on clicking **Start Server** and **Add Files** button, you can start the server and the .ldif files are added to the server. The shell script named **ServerStart.sh** will start the server process by running the following command: `sudo /usr/local/libexec/slapd`. The process id on which the server is running can be obtained from the file `/usr/local/var/run/slapd`. Now, your server is functional and you are all set to create accounts for your users.

7.2.5 Create User Accounts

We now need to create accounts for various users. For this we have provided a sample 1.ldif file for one user and have explained it in the tool. A shell script named **UserAdd.sh** can be used when user accounts have to be created in bulk. The user is advised to read the comments in the shell scripts to get a better understanding of the procedure.

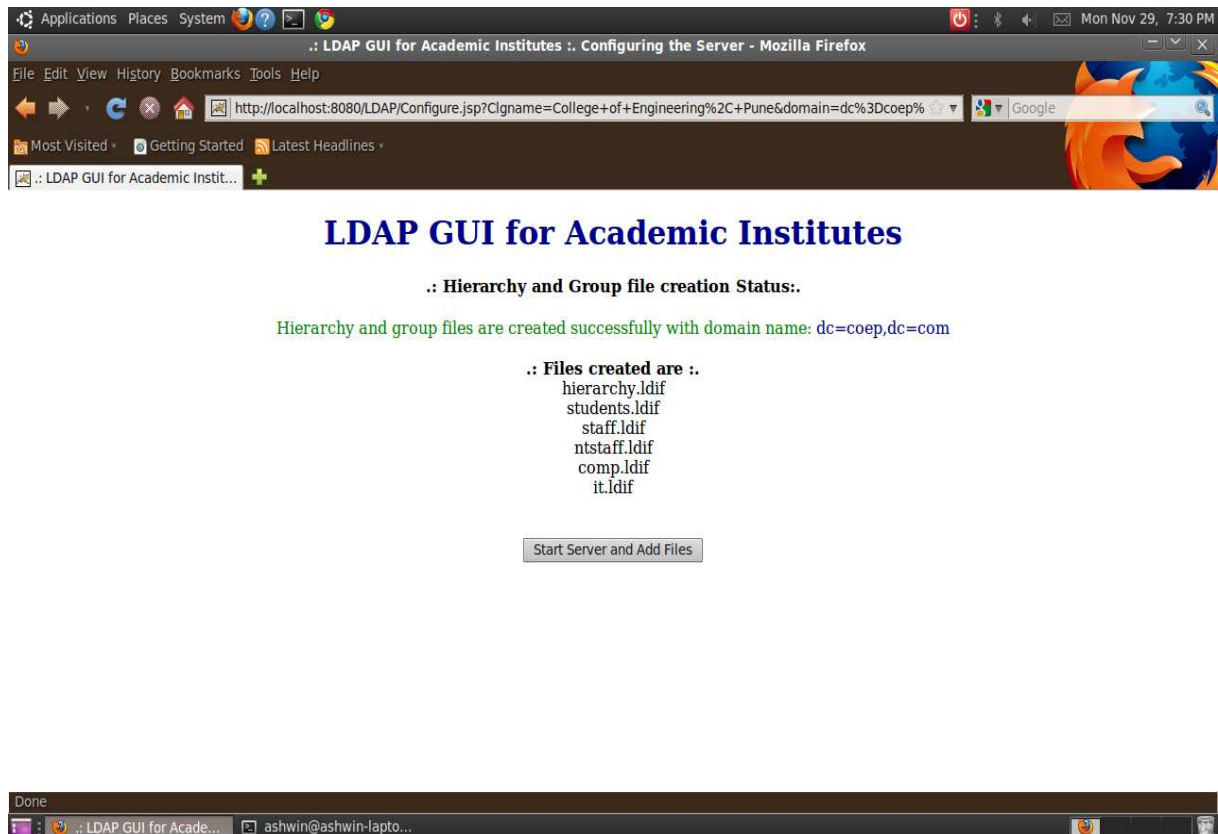


Figure 7.5: Build the Directory Information Tree

7.3 ClientSide Tool

The steps to configure a machine as an LDAP Client are as follows:

1. Download the package `libnss-ldap`
2. Edit the configuration files that Linux System uses to authenticate its users.

7.3.1 Package - `libnss-ldap`

Install the package by :

```
sudo apt-get install libnss-ldap
```

This automatically installs the following packages required for a machine to function as a LDAP Client: `libnss-ldap`, `auth-client-config`, `ldap-auth-client`, `ldap-auth-config`, `libpam-ldap`, `nscd`

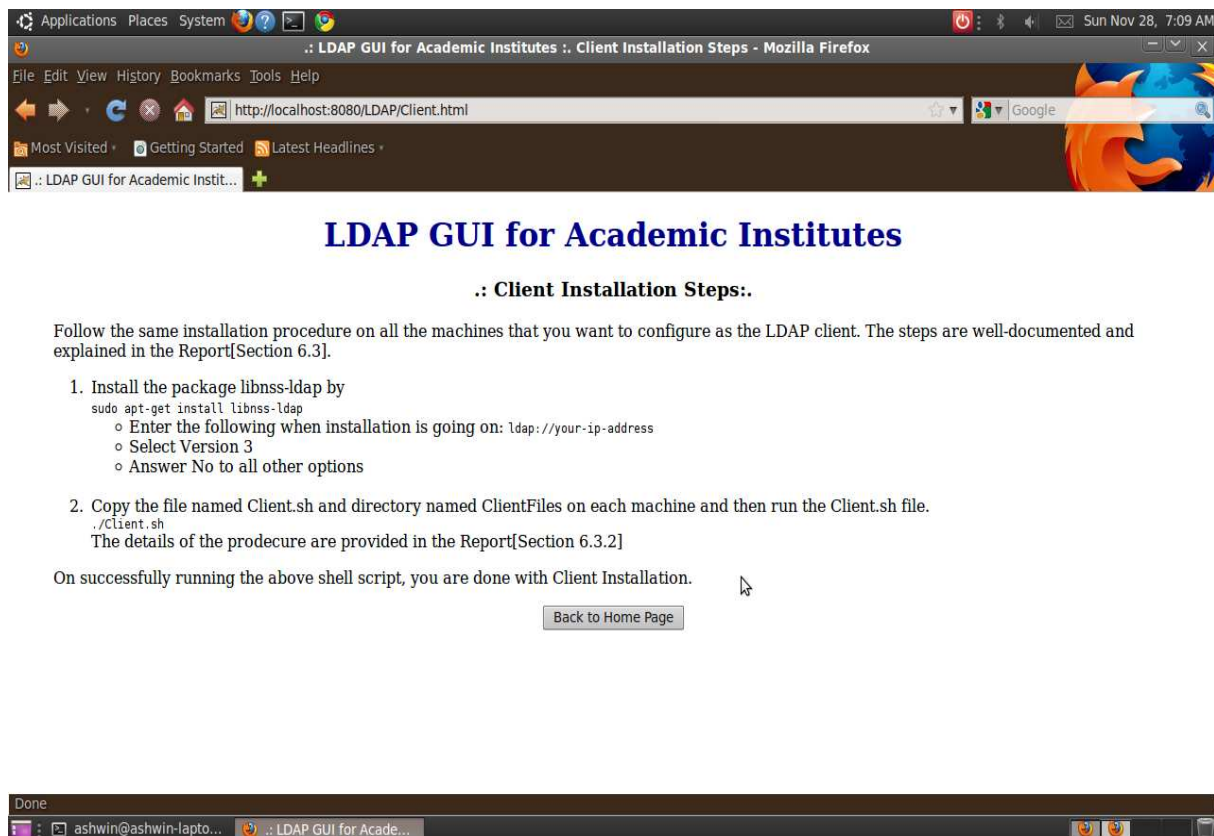


Figure 7.6: Screenshot of Client Installation procedure

Whilst installation, it will prompt for the IP address of the server, where you need to enter the IP address of your server.

- Enter the following when installation is going on:
`ldap://your-ip-address`
- Select Version 3
- Answer *No* to all other options.

7.3.2 Edit the configuration files

The shell script that we have provided needs to be executed on each client machine. Copy the file named **Client.sh** and directory named **ClientFiles** on each machine and then run the **Client.sh** file. The details of the shell script are as follows:

- Name : **Client.sh**
- Where to place : Can be placed in any directory on the client machine, only constraint is that the *current directory* of this file and the directory named **ClientFiles** should be the same.
- Supporting Files : **common-auth**, **common-account**, **common-session**, **common-password**, **nsswitch.conf**. These files are in the **ClientFiles** directory.
- How to execute : `sudo ./Client.sh`

7.3.2.1 Details

The `Client.sh` file is written to configure the machine as a LDAP Client. Basically what it does is that, it simply takes a backup of your original configuration files *viz.*

```
/etc/nsswitch.conf,  
/etc/pam.d/common-auth,  
/etc/pam.d/common-account,  
/etc/pam.d/common-session,  
/etc/pam.d/common-password
```

in the directory named `/etc/pam.d/backup_common_files/`. Then, copies our edited configuration file in their appropriate position.

Backup of the original files is taken so that, in case anything goes wrong, you can always restore the original files from the backup directory. Now, an obvious question is that why do we need to edit these files only? The `libnss-ldap` package has only provided the client machine the ability to act as a LDAP client, but to do so we also need to edit various configuration files also. Linux follows a standard procedure to authenticate its users by consulting the above mentioned files. It checks the settings in these files and moulds its procedure of authentication accordingly. We also specify in these files the library files to which the authentication process should link for using LDAP. These files are usually with the `.so` suffix.

Here's a brief explanation of it :

- `/etc/nsswitch.conf` : It is used to configure which services are to be used to determine information such as hostnames, password files, and group files. So by editing it we are instructing it to use LDAP Server Service for authenticating its users.
- `/etc/pam.d/common-auth, /etc/pam.d/common-account, /etc/pam.d/common-session, /etc/pam.d/common-password` : ¹ These files merely represent different stages in the authentication process. By editing them, we are instructing the client machine that, along with its local authentication procedure, also consider the LDAP Server.

¹PAM - Pluggable Authentication Modules. PAM Subsystem provides a layer of abstraction for applications that require authentication. Essentially, an application can authenticate against many different backends without having to know anything about the underlying protocols. A PAM-aware application simply requests authentication and then trusts PAM to retrieve it. This layer of abstraction is achieved via PAM modules

Chapter 8

Future Enhancements

This chapter gives a glimpse of some of the enhancements that can be employed in the tool in near future. They are as follows:

- Integration of the tool with Moodle. Moodle is a free, open-source PHP web application for producing modular internet-based courses that support a modern social constructionist pedagogy. Moodle has the facility of authenticating its users from the LDAP Server. This can be used for the integration purposes.[6]
- Same kind of tool can be enhanced further for installing LDAP clients with customizations.

1

Bibliography

- [1] Apache Tomcat version used : 6.0.29. <http://tomcat.apache.org/>.
- [2] Berkeley DB : 4.3.29. <http://www.ubuntu.com/server/get-ubuntu/download>.
- [3] GNU General Public License. www.gnu.org/licenses/gpl.html.
- [4] Java version used : 1.6. <http://www.oracle.com/technetwork/java/index.html>.
- [5] Mozilla Firefox : 3.6. www.mozilla.com/firefox/.
- [6] Open Source Tool for producing Internet-based courses and web sites.
<http://moodle.org/>.
- [7] Openldap : 2.3.30. <http://www.ubuntu.com/server/get-ubuntu/download>.
- [8] Steps for installing Ubuntu. <http://www.ubuntu.com/server/get-ubuntu/download>.
- [9] Technical definitions of LDAP. <http://en.wikipedia.org/wiki/LDAP>,
<http://www.webopedia.com/TERM/L/LDAP.html>.
- [10] Ubuntu Version used : 10.04 - Lucid Lynx. <http://www.ubuntu.com/>.