

# **Experiment #1 Overview of FOSS (Free Open Source Software)**

## **OBJECTIVE**

- Learning about Open Source Software and basic commands in Ubuntu

### **After completing this experiment you will be able to:**

- Student should be able to use basic syntax structure of Commands.
- Understand how to use commands in a shell
- How to get help for Commands

## **Requirements**

Tools / Apparatus: Unix/Linux Operating System, Bash Shell

## **Theoretical Foundations**

The Open Source Initiative (OSI) is a California public benefit corporation, with 501(c)3 tax-exempt status, founded in 1998. Open source software is software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition.

The Open Source Initiative (OSI) is a global non-profit that supports and promotes the open source movement. Among other things, we maintain the Open Source Definition, and a list of licenses that comply with that definition

## **The Open Source Definition**

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

### **1. Free Redistribution**

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

### **2. Source Code**

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a pre-processor or translator are not allowed.

### **3. Derived Works**

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

### **4. Integrity of the Author's Source Code**

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

### **5. No Discrimination Against Persons or Groups**

The license must not discriminate against any person or group of persons.

### **6. No Discrimination Against Fields of Endeavor**

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

### **7. Distribution of License**

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

### **8. License Must Not Be Specific to a Product**

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

### **9. License Must Not Restrict Other Software**

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

### **10. License Must Be Technology-Neutral**

No provision of the license may be predicated on any individual technology or style of interface.

## **Basic Utility Commands in Linux**

Commands can be run by themselves, or you can pass in additional arguments to make them do different things. Typical command syntax can look something like this:

command [-argument] [-argument] [--argument] [file]

Examples: ls List files in current directory

ls -l Lists files in “long” format

ls -l --color As above, with colourized output

cat filename Show contents of a file

cat -n filename Show contents of a file, with line numbers

## **Commands with its usage**

ls show files in current position

Commands	Usage
cd	change directory
cp	copy file or directory
mv	move file or directory
rm	remove file or directory
pwd	show current position
mkdir	create directory
rmdir	remove directory
cat	display file contents
less	display file contents page wise
man	display online manual
touch	to create multiple empty files
banner	to display the given text in banner

## **Exercise:**

Q1. What is a syntax structure of Commands?

Q2. How to see your login details?

Q3. How to change date and time of your Linux/Unix/Ubuntu

## **For Further Reading Student May Refer to the following Books**

1 Chapter 4 Essential Unix Commands from Linux In depth By Yashwant P Kanetkar  
(Available in Departmental and Central Library at University campus)

## **Experiment # 2:- Learning Network related and Administrative Commands**

### **OBJECTIVE**

- Learning the network related and administrative commands in Ubuntu's Shell

### **After completing this experiment you will be able to:**

- Students should be ready to work on Vim Editor.
- Student should be ready to work as a system admin or Root User
- Student should be able to design and apply Admin Commands.

### **Theoretical Foundations:**

The Linux System Administrator's Guide describes the system administration aspects of using Linux. It is intended for people who know next to nothing about system administration (those saying "what is it?"), but who have already mastered at least the basics of normal usage. This manual doesn't tell you how to install about Linux manuals. System administration covers all the things that you have to do to keep a computer system in usable order. It includes things like backing up files (and restoring them if necessary), installing new programs, creating Accounts for users (and deleting them when no longer needed), making certain that the filesystem is not corrupted, and so on. If a computer were, say, a house, system administration would be called maintenance, and would include cleaning, fixing broken windows, and other such things. The structure of this manual is such that many of the chapters should be usable independently, so if you need information about backups, for example, you can read just that chapter. However, this manual is first and foremost a tutorial and can be read sequentially or as a whole.

At times system needs to communicate with outer network and it becomes essential to know about the utilities offered by Ubuntu for such jobs.

### **Requirements**

Tools / Apparatus: Unix/Linux/Windows Operating System, Bash Shell with root login privileges. i.e. root user name and Password

### **Admin Commands**

Install Software

Commands

Sudo apt-get update install <Prod Name>

gzip compress a file

gunzip uncompress a file

tar archive or expand files

configure create Makefile

make compile & install software

Executing commands line by line:

tar xvfz mpich.tar.gz

cd mpich1.2.5.2

./configure prefix=

usr/local rsh=

ssh

make

make install

User Management Files

/etc/group, /etc/passwd, /etc/shadow

User account information.

/etc/bashrc, /etc/profile, \$HOME/.bashrc, \$HOME/.bash\_profile

Bash system wide and per user init files.

/etc/csh.cshrc

/etc/csh.login, \$HOME/.cshrc

\$HOME/.tcshrc, \$HOME/.login, tcsh system wide and per user init files

/etc/skel template files for new users.

/etc/default default for certain commands.

/etc/redhat-release, /etc/slackware-version

Redhat/Slackware version info (Linux kernel version with “uname -a”)

Commands-> adduser, useradd, userdel, usermod, create, delete, modify an new user or update,

default new user information., newusers update and create new users (batch mode). groupadd,

groupdel, groupmod, add, delete or modify group., chage. ch fn, chsh

modify account policy (password length, expire data etc.) or finger information (full

name, phone number etc.) change default login shell.

linux init=/bin/sh rw

gain root access during boot prompt without

password, can be used to fix some problems. `mount -w -n -o remount /`

## Network Configuration Files

`/etc/rc.d/rc.inet1`

`/etc/sysconfig/networkscripts/`

`ifcfg-eth0` (Redhat)

IP address, Network mask, Default gateway are in these files. May edit manually to modify network parameters. `/etc/HOSTNAME` hostname is set by `"/bin/hostname"` during `/etc/NETWORKING`

`/etc/sysconfig/network` boot and the name is read from these files.

May change manually.

`etc/resolv.conf`

specify name server, DNS domain and search order. For Example:

`search la.asu.edu`

`nameserver 129.219.17.200`

`/etc/hosts` host name to IP mapping file.

`/etc/host.conf`

host name information look up order.

Example:

`order hosts, bind`

`multi on`

Commands

`lpc, lpq, lprm`

## Network Related Command

`Ifconfig, ifup, ifdown, netstat,`

`ftp, ping, traceroute, telnet`

`ssh` that is secured shell..etc

**Exercise:**

- How to run and analyze network related command?
- How to take back up of the required data from file system?
- How to make a close look on Conf files using network and administrative commands?

**For Further Reading Student May Refer to the following Books**

1 Chapter 15 System Admin Commands from Linux In depth By Yashwant P Kanetkar (Available in Departmental and Central Library at University campus)

## **Week # 3 Learning Vi –Editor its Modes and GUI Tools in available in Ubuntu**

### **OBJECTIVE**

- Learning how to work on VI-editor and switching from different modes available in Ubuntu.
- To know about the various Graphical Tools given by Ubuntu OS.

### **Requirements**

- Tools / Apparatus: Unix/Linux/Windows Operating System, Gedit Text Editor, Bash Shell with root login credentials

### **After completing this experiment you will be able to:**

- Students should be ready to work on Vim Editor.
- Students must be able to use GUI Tools given in Ubuntu OS.

### **Theoretical Foundations**

Vim is a text editor that is upwards compatible to Vi. It Can be used to edit all kinds of plain text. It is especially useful for editing programs or shell scripts.

There is lot of enhancements above vi: multilevel undo, multi window and buffers, syntax highlighting, command line editing, filename completion. While running Vim a lot of help can be obtained from the on-line help system. Most often Vi is started to edit a single file with the command

The default editor that comes with the Linux operating system is called vi (**v**isual editor).  
[Alternate editors for these environments include pico, nano and emacs, a product of GNU.]

The Linux vi editor is a full screen editor and has two modes of operation:

1. *Command mode* commands which cause action to be taken on the file, and
2. *Insert mode* in which entered text is inserted into the file.

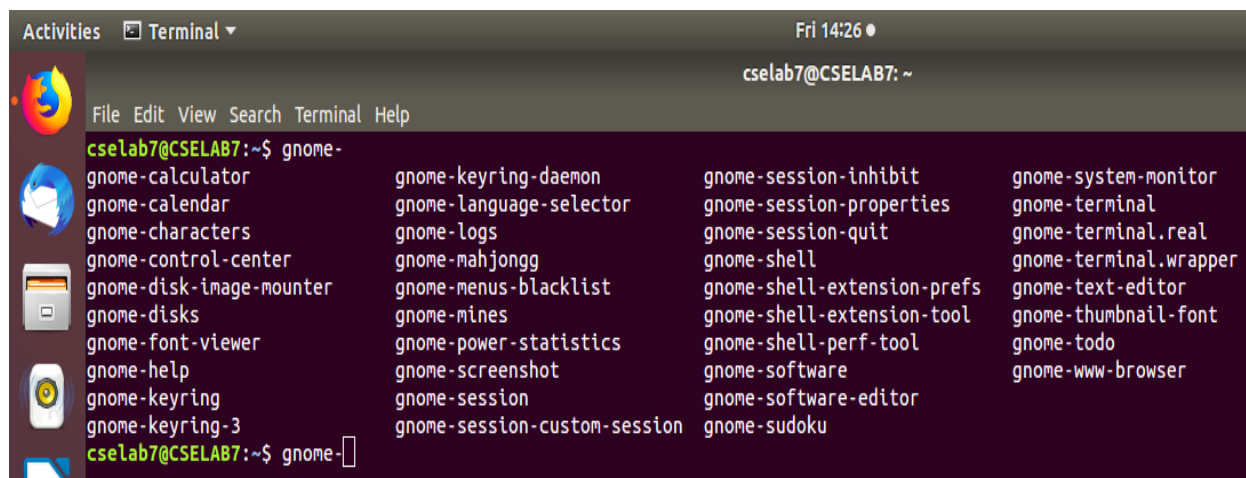
In the command mode, every character typed is a command that does something to the text file being edited; a character typed in the command mode may even cause the vi editor to enter the insert mode. In the insert mode, every character typed is added to the text in the file; pressing the <Esc> (*Escape*) key turns off the Insert mode.

While there are a number of vi commands, just a handful of these is usually sufficient for beginning vi users. To assist such users, this Web page contains a sampling of basic vi commands. The most basic and useful commands are marked with an asterisk (\* or star) in the tables below. With practice, these commands should become automatic.



**NOTE:** Both Linux and vi are **case-sensitive**. Be sure not to use a capital letter in place of a lowercase letter; the results will not be what you expect

Various GUI Tools such as given in following table



Students are required to run each and every command to know more about the GUI Tool available in OS.

## Exercise

- Student must be encouraged to use Vi Editor for more complex group functions on text

## For Further Reading Student May Refer to the following Books

1 Chapter 15 System Admin Commands from Linux In depth By Yashwant P Kanetkar (Available in Departmental and Central Library at University campus)

## **Experiment No 4: Learning Shell Scripting and demonstrate various control constructs**

### **OBJECTIVE:**

- Student should be able to create shell script to demonstrate the usage of control statements and usage of special symbols
- How to use command line argument in Shell script
- Identify the problem statement for which Shell scripting is created.

### **After completing this experiment you will be able to:**

- Student should be able to create log files with the help of shell script.
- Understand how to use command line argument to take input from user.
- Students will be ready to create customized set of commands to perform automate tasks.
- How to automate several admin level commands using `at` and `crontab` commands.

### **Theoretical Foundations**

A shell is an environment in which we can run our commands, programs, and shell scripts. There are different flavors of shells, just as there are different flavors of operating systems. Each flavor of shell has its own set of recognized commands and functions. This book works with the Bourne, Bash, and Korn shells. Shells are located in either the `/usr/bin/` directory or the `/bin/` directory, depending on the UNIX flavor and specific version. The prompt, `$`, which is called command prompt, is issued by the shell. While the prompt is displayed, you can type a command. The shell reads your input after you press Enter. It determines the command you want executed by looking at the first word of your input. A word is an unbroken set of characters. Spaces and tabs separate words. Following is a simple example of **date** command which displays current date and time:

```
$ shoewMe.sh
```

```
$ clear
```

```
$ date
```

```
$ sh ShowMe.sh
```

### **Requirements**

Tools / Apparatus: Unix/Linux/Windows Operating System, Gedit Text Editor, Bash Shell with root login credentials

### **Shell Script fragment Codes**

Shell scripting for demonstrating the use of if construct, loop construct, switch statement, working with command line argument

i        A script to check whether the given file name exists in the file system or not

```
# A Shell Script to check for file existence
```

```
FILE=$1
```

```
if [ -f $FILE ]
```

```
then
```

```
    echo "File $FILE exists"
```

```
else
```

```
    echo "File $FILE does not exist"
```

```
fi
```

ii       A script to print all the prime entries in the given list

```
echo "1"
```

```
echo "enter the limits(a-b)"
```

```
read i
```

```
read j
```

```
flag=0
```

```
tem=2
```

```
echo "1"while [ $i -ne $j ]
```

```
do
```

```
    temp=`echo $i`
```

```
    while [ $temp -ne $tem ]
```

```
    do
```

```
        temp=`expr $temp - 1`
```

```
        n=`expr $i % $temp`
```

```
        if [ $n -eq 0 -a $flag -eq 0 ]
```

```
        then
```

```
            flag=1
```

```
        fi
```

```

done

if [ $flag -eq 0 ]
then
    echo $i
else
    flag=0
fi
i=`expr $i + 1`
done
iii    # A script to execute different command to demonstrate cases in switch statement
clear
option="{1}"
case ${option} in
    -f) FILE="{2}"
        echo "File name is $FILE"
        ;;
    -d) DIR="{2}"
        echo "Dir name is $DIR"
        ;;
    *)
        echo "`basename ${0}`:usage: [-f file] | [-d directory]"
        exit 1 # Command to come out of the program with status 1
        ;;
esac

```

### Exercise:

- Write a shell script to implement Command line Argument to create conditional copy file module.
- Write a shell script to automate installation process of any given open software.
- Write a shell script to print nth column of a pattern/file without using awk, cut commands.

### For Further Reading Student May Refer to the following Books

- 1 Chapter-8 SHELL SPECIAL CHARACTER in Unix Commands from Linux In depth By Yashwant P Kanetkar (Available in Departmental and Central Library at University campus)

## **Experiment # 5 Learn how to Compile, Debug & Execute C, C++ & Java Programming Code without IDEs**

### **OBJECTIVE**

- To write program in different programming Languages. Such as C, C++, Java Programming
- Proper usage of man and gdb tool for effective usage of coding strategies in C, C plus plus, Java Programming.
- To gain interest in developing programming language products.
- Dependencies on IDEs will be removed.

**After completing this experiment you will be able to:**

- Write C , C++ and Java Programming Code on Ubuntu machines
- .Understand how to Debugging, compiling and executing tools available for different languages.
- How to get help for Commands
- Become more proficient in Coding skills on open development Environment.

### **Requirements**

Tools / Apparatus: Unix/Linux/Windows Operating System, Gedit Text Editor, Compilers and interpreters of all languages (C, C++, Java)

1. Create a sample C program for demonstrating the use of control statements then compile and execute the code

Soln

**#include<stdio.h> //let us assume file name for this code is democ.c**

**int main()**

**{**

**printf("hello world !");**

**}**

**To Compile Your Code \$cc democ.cc -o democ.exe // -o option will create object file with given file name as democ.exe**

**to Execute Your Code \$./democ.exe //use of ./is must before executing your democ.exe file**

**Note: User can use manual of cc for detailed options of compiling and debugging modes as (man cc)**

ii Create a sample C++ class based program for demonstrating the use of Object oriented Design Patterns, then compile and execute the code

**#include<stdio> //let us assume file name for this code is democplusplus.cpp**

**#include<iostream>**

**using namespace std;**

**class A**

**{**

**private:**

**int a;**

**public:**

**void getData()**

**{**

**cout<<"ENTER DATA ? "<<endl;**

**}**

**void setData()**

**{**

**a=25; //set the default value of a to 25**

**}**

**void dispData()**

**{**

**cout<<"Your Data is "<<a<<endl;**

**}**

**};**

**int main()**

**{**

**printf("hello world ! I N C + + \n\n\n" );**

**A obj;**

**obj.setData();**

**obj.getData();**

**obj.dispData();**

**}**

**To Compile Your Code \$gcc democplusplus.cpp -o democplusplus.exe // -o option will create object file with given file name as democplusplus.exe**

**to Execute Your Code \$./democplusplus.exe //use of ./is must before executing your democplusplus.exe file**

**Note: User can use manual of gcc for detailed options of compiling and debugging modes (\$ man gcc)**

iii Creating a sample java Program for demonstrating the use of control statements then compile and execute the code

**class A**

**{**

**private static int a;**

**public static void getData()**

**{**

**cout<<"ENTER DATA ? "<<endl;**

**}**

**static void setData()**

**{**

```

a=25;          //set the default value of a to 25
}
static void dispData()
{
cout<<"Your Data is "<<a<<endl;
}
};
public static void main(String as[])
{
printf("hello world ! I N  J A V A  \n\n\n" );
A obj=new A();
obj.setData();
obj.getData();
obj.dispData();
}

```

**To Compile Your Code** \$javac demojava.java // @ the time of compiling filename is given as paramter to javac (java compiler)

**to Execute Your Code** \$java A // @ the time of interpreting bytecode file class name is given

**Note:** User can use manual of gcc for detailed options of compiling and debugging modes (\$ man gcc)

### **Exercise:**

- Write C program to sort the array input will be from one file and output will be directed on other file.
- Write a C++ code to Convert an expression from Postfix to prefix.
- Write a Java Code to Print all possible combination of single word.

## **Week # 6 Learning about LAMP STACK its Installation And Configuration on Linux (Ubuntu) and Perform Post Installation Exercises**

### **OBJECTIVE:**

- How to Install and configure LAMP Stack on Ubuntu

### **After completing this experiment you will be able to:**

- Install and Configure LAMP Stack.
- Configure phpmyadmin module
- Cross Check for proper installation to initiate web development.

### **Requirements**

Tools / Apparatus: Unix/Linux/Windows Operating System, Gedit Text Editor, LAMP STACK from Internet

### **Theoretical Foundations**

**LAMP** is an acronym for a web services solution stack consisting of Linux, the Apache HTTP **Server**, the MySQL or MariaDB database engines, and the PHP or Perl, or Python programming language. All components are free and open-source software, and the combination is suitable for building dynamic web pages. The combination of the software included in a LAMP stack may vary, for example the Apache web server can be replaced by some other web server software. Although the original authors of these programs did not design them to work as a component of the LAMP stack, the development philosophy and tool sets are shared and were developed in close conjunction. The software combination has become popular because it is entirely free and open-source software, which means that each component can be adapted to the underlying hardware and customized to meet the specification as exactly as possible, without overt vendor lock-in. The complete software stack is also free of cost, maximizing the available budget for tailoring the hardware and software.

Due to the nature of free and open-source software and the ubiquity of its components, each component of the LAMP stack has been tested for performance and security. At the same time, continuing development and an abundance of experienced developers permit new features and custom modification for specific applications.

The components of the LAMP stack are present in the software repositories of most Linux distributions, providing a simple way to install, set up and operate a LAMP stack out of the box.



The website of a small company is typically not subject to high traffic and is not prone to frequent attacks, and can therefore be administered with minimal resources.

The LAMP bundle can be and often is combined with many other free and open-source software packages such as, for example:

- netsniff-ng for security testing and hardening
  - Snort, an intrusion detection (IDS) and intrusion prevention system (IPS)
  - RRDtool for diagrams
  - Nagios, Collectd or Cacti, for monitoring.
- In this guide I will show you how to install a LAMP system. LAMP stands for **L**inux, **A**pache, **M**ySQL, **P**HP. The guide is intended to help those who have very little knowledge of using Linux.

### **Install Apache**

To start off we will install Apache.

1. Open up the Terminal (Applications > Accessories > Terminal).
2. Copy/Paste the following line of code into Terminal and then press enter:  
`sudo apt-get install apache2`
3. The Terminal will then ask you for your password, type it and then press enter.

### **Testing Apache**

To make sure everything installed correctly we will now test Apache to ensure it is working properly.

1. Open up any web browser and then enter the following into the web address:

`http://localhost/`

You should see a folder entitled `apache2-default/`. Open it and you will see a message saying "It works!", congrats to you!

### **Install PHP**

In this part we will install PHP 5.

Step 1. Again open up the Terminal (Applications > Accessories > Terminal).

Step 2. Copy/Paste the following line into Terminal and press enter:

`sudo apt-get install php5 libapache2-mod-php5`

Step 3. In order for PHP to work and be compatible with Apache we must restart it. Type the following code in Terminal to do this:

`sudo /etc/init.d/apache2 restart`

### **Test PHP**

To ensure there are no issues with PHP let's give it a quick test run.

Step 1. In the terminal copy/paste the following line:

`sudo gedit /var/www/testphp.php`

This will open up a file called `phptest.php`.

Step 2. Copy/Paste this line into the `phptest` file:

`<?php phpinfo(); ?>`

Step 3. Save and close the file.

Step 4. Now open your web browser and type the following into the web address:  
`http://localhost/testphp.php`  
The page should look like this:

Congrats you have now installed both Apache and PHP!

## Install MySQL

To finish this guide up we will install MySQL. (Note - Out of Apache and PHP, MySQL is the most difficult to set up. I will provide some great resources for anyone having trouble at the end of this guide.)

Step 1. Once again open up the amazing Terminal and then copy/paste this line:

```
sudo apt-get install mysql-server
```

Step 2 (optional). In order for other computers on your network to view the server you have created, you must first edit the "Bind Address". Begin by opening up Terminal to edit the `my.cnf` file.

```
gksudo gedit /etc/mysql/my.cnf
```

Change the line

```
bind-address = 127.0.0.1
```

And change the 127.0.0.1 to your IP address.

Step 3. This is where things may start to get tricky. Begin by typing the following into Terminal:

```
mysql -u root
```

Following that copy/paste this line:

```
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('yourpassword');
```

(Make sure to change yourpassword to a password of your choice.)

**Step 4.** We are now going to install a program called phpMyAdmin which is an easy tool to edit your databases. Copy/paste the following line into Terminal:

```
sudo apt-get install libapache2-mod-auth-mysql php5-mysql phpmyadmin
```

After that is installed our next task is to get PHP to work with MySQL. To do this we will need to open a file entitled `php.ini`. To open it type the following:

```
gksudo gedit /etc/php5/apache2/php.ini
```

Now we are going to have to uncomment the following line by taking out the semicolon (;).

Change this line:

```
;extension=mysql.so
```

To look like this:

```
extension=mysql.so
```

Now just restart Apache and you are all set!

```
sudo /etc/init.d/apache2 restart
```

With This We are all set to start Our LAMP Project

Other Web Links for further Help

- <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-14-04h>
- <https://help.ubuntu.com/community/ApacheMySQLPHP>

**Exercise:**

- **Install LAMP SEVER on your machine**
- **Configure phpmyadmin Page**

**For Further Reading Student May Refer to the following Books**

- Programming PHP, 3rd Edition By O'Reilly.
- Beginning PHP 6 and MySQL By Steve Suehring, Tim Converse, Joyce Park. (Available in Departmental and Central Library at University campus)

## **Week # 7 Creating Simple Databases in MySql Server and performing queries**

### **OBJECTIVE**

- Learning about Configuration of MySql Server Instance .
- Creating DML DDL Queries in Database.
- Assessing User Roles Management In MySql.
- Managing Data backups.

### **After completing this experiment you will be able to:**

- Ready to install and configuration of MySQL server.
- Ready to analyze about the nature of queries.
- Assessing User Roles Management.
- Managing Administrative rights in MySQL.

**Requirements:-** Tools / Apparatus: Unix/Linux/Windows Operating System, Gedit Text Editor, MySql Server and Client instance

### **Theoretical Foundations**

**MySQL** officially, but also called "My Sequel" is (as of March 2014) the world's second most widely used open-source *relational database management system* (RDBMS). The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia

### **Solutions and Command**

## Mysqladmin supports the following commands:

`create db_name`

Create a new database named *db\_name*.

`Debug //` Tell the server to write debug information to the error log.

`drop db_name //` Delete the database named *db\_name* and all its tables.

`extended-status //` Display the server status variables and their values.

`flush-hosts //` Flush all information in the host cache.

`flush-logs ///` Flush all logs.

`flush-privileges //` Reload the grant tables (same as reload).

`flush-status //` Clear status variables.

`flush-tables //` Flush all tables.

`flush-threads //` Flush the thread cache. (Added in MySQL 3.23.16.)

`kill id,id,... //` Kill server threads.

`old-password new-password //` This is like the password command but stores the password using the old (pre-4.1) password-hashing format. This command was added in MySQL 4.1.0. (See Section 4.9, “Password Hashing in MySQL 4.1”.)

`password new-password //` Set a new password. This changes the password to *new-password* for the account that you use with **mysqladmin** for connecting to the server. If *new-password* contains spaces or other characters that are special to your command interpreter, you need to enclose it within quotes. On Windows, be sure to use double quotes rather than single quotes; single quotes are not stripped from the password, but rather are interpreted as part of the password. For example:

```
shell> mysqladmin password "my new password"
```

`ping`

Check whether the server is alive. The return status from **mysqladmin** is 0 if the server is running, 1 if it is not. Beginning with MySQL 4.0.22, the status is 0 even in case of an error such as Access denied, because that means the server is running but disallowed the connection, which is different from the server not running.

`processlist`

Show a list of active server threads. This is like the output of the **SHOW PROCESSLIST** statement. If the **--verbose** option is given, the output is like that of **SHOW FULL PROCESSLIST**.

reload // Reload the grant tables.  
refresh // Flush all tables and close and open log files.  
Shutdown // Stop the server.  
start-slave //Start replication on a slave server. (Added in MySQL 3.23.16.)  
status // Display a short server status message.  
stop-slave //Stop replication on a slave server. (Added in MySQL 3.23.16.)  
variables //Display the server system variables and their values.  
version // Display version information from the server.

All commands can be shortened to any unique prefix. For example:

shell> **mysqladmin proc stat**

### **Exercise:**

- Write a Program to design database in MySql
- Allot Roles and Permission to newly added user.
- How to monitor the server.

### **For Further Reading Student May Refer to the following Books**

- Chapter-3, Building SQL Server Databases from SQL Server 2012 Black Book,
- Chapter-12, Monitoring, Auditing and maintaining SQL Server from SQL Server 2012 Black Book.

\*Both books are available in Departmental and Central Library at University campus

## **Week# 8 Learning A Deep Dive in MySql Server Using PhpMyAdmin Tool for Administering and Monitoring the Database Server, Mysql Admin, Backup and restore, User Account Rights Management**

### **OBJECTIVE**

- Learning about Configuration of MySql Server Instance.
- Creating and implementing right management on User Role.
- Assessing User Roles Management In MySql.
- Managing Data backups.

**After completing this experiment you will be able to:**

- Ready to install and configuration of MySQL server.
- Ready to analyze about the nature of queries.
- Assessing User Roles Management.
- Managing Administrative rights in MySQL.

**Requirements:-** Tools / Apparatus: Unix/Linux/Windows Operating System, Gedit Text Editor, MySql Server and Client instance

### **Theoretical Foundations**

PhpMyAdmin is an open source tool where you can administer your MariaDB database(s). It is pre-installed with your web hosting subscription. In phpMyAdmin you can, import, export, optimize or drop tables. If your website is using a database, this is where all the data of your site is stored. For example, WordPress stores all your posts, comments and articles in a database

Step 1 - Log into the control panel

Step 2 - Select database

Step 3 - Administer your database

**Note:** Be very careful when you work in phpMyAdmin, it is **not possible to undo changes**. We recommend to create a backup, before you make any changes.

Step 1 - Log into the control panel

1. Log into the **One.com control panel**.
2. Click **PHP and database settings** on the **Advanced settings** tile

step 2 - Select database

Under **PhpMyAdmin** in the top right, click **Select database** and choose the database you want to access.

**Step 3 - Administer your database:** - A new window opens showing your database in phpMyAdmin. Here you can manage all the tables in your database. Check out the guides below to learn about the various actions you can perform.

**Tip:** Click on the + **sign** next to the database name, in the menu to the left, for a list of all the individual tables in this database.

### **Managing your database**

To view and manage your database, select it in the drop-down on the left-hand side. You will get an overview of your database and nine tabs for carrying out certain actions. Below is a description of the different tabs.

**Structure:** displays the different tables in the database and allows you to create a new table. A small drop-down menu, lets you perform the following on table/s you have selected; empty, drop, print view, check, optimise, repair and analyse. The Action section provides the following options: browse, search, display the structure, insert a record, empty or drop a table.

**SQL:** lets you execute SQL queries on the database.

**Search:** lets you search for words and phrases in the database.

**Query:** allows queries to be executed using the interface components and manual SQL queries to be run.

**Export:** allows you to save your database into a file format of your choice. CSV, Excel, Word, LaTeX, PDF, SQL and XML are some of the formats available.

**Import:** import an SQL file.

**Operation** allows you to create a new table, rename and copy the database.

**Privileges:** shows what users have access to the current database and their rights. You also have an edit option next to each user.

**Drop:** lets you delete the database if you need to.

### **Exercise:**

- Write a Program to design database in MySql
- Allot Roles and Permission to newly added user.
- How to monitor the server.

### **For Further Reading Student May Refer to the following Books**

- Chapter-3, Building SQL Server Databases from SQL Server 2012 Black Book,
- Chapter-12, Monitoring, Auditing and maintaining SQL Server from SQL Server 2012 Black Book.

\*Both books are available in Departmental and Central Library at University campus



## **Experiment # 9 Basics of PHP Web Programming, PHP code to demonstrate the usage of Variable, String, Array and Control Structure**

### **OBJECTIVE**

- Student should start using basics of php variable, String, Arrays
- Student should start using Control statements in Php

**After completing this experiment you will be able to:**

- Student should able to use basic syntax structure of Commands.
- Understand how to use commands in a shell
- How to get help for Commands

**Requirements:** - Tools / Apparatus: Unix/Linux/Windows Operating System, Gedit Text Editor and LAMP STACK.

### **Theoretical Foundations**

**PHP** is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers.[4] Originally created by Rasmus Lerdorf in 1994,[5] the reference implementation of PHP (the Zend Engine) is now produced by The PHP Group.[6] While PHP originally stood for Personal Home Page,[5] it now stands for PHP: Hypertext Preprocessor, which is a recursive acronym. PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page – for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications. PHP is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

Basics of PHP scripts, variables, Data Types, Operators and Expressions, Arrays, loops

```
<html>
```

```
<head><title></head>
```

```

<body><h2>
<?php
$txt="my first string type variable";
echo $txt
$x=500;
If ($x %2==0)
echo "$x is even no.";
else
echo "$x is odd number.";
?>
</h2>
</body>
</html>

```

A PHP Program to implement customized Functions

```

<html>
<body>
<?php
Function multi($x,$y)
{
$pro=$x*$y;
Return #pro;
}
echo "the value of 7*8=".multi(7,8);
</?>
</body></html>

```

Exercise:

- 1 Write a PHP code to check either a number is a prime or not.
- 2 Write a PHP code to send mail to all mail ids stored in an array
- 3 Write a login control to cross entries from database and redirect to authenticated web page else on error page.

## **For Further Reading Student May Refer to the following Books**

- Programming PHP, 3rd Edition By O'Reilly.
- Beginning PHP 6 and MySQL By Steve Suehring, Tim Converse, Joyce Park. (Available in Departmental and Central Library at University campus)

## **Experiment # 10 Some Deep Dive in PHP Programming:- A PHP Program to implement customized functions and other Form Handling Strategies**

### **OBJECTIVES**

- Learning In depth working knowledge of Php Programming
- Learning In depth working knowledge of Php Form Handling and other session level security.

**After completing this experiment you will be able to:**

- Hands on Working experience on PHP Internals
- How to send mail to any mail id
- How to create sessions during navigation from one web page to another.

**Requirements:-** Tools / Apparatus: Unix/Linux/Windows Operating System, Text Editor, MYSQL SERVER.

### **Theoretical Foundation**

The Benefits of PHP & MySQL:- At the start of this chapter, I introduced the world of Web 1.0, but it wasn't long before the rush was on to create Web 1.1, with the development of such browser enhancements as Java, JavaScript, JScript (Microsoft's slight variant of JavaScript) and ActiveX. On the server side, progress was being made on the Common Gateway Interface (CGI) using scripting languages such as Perl (an alternative to the PHP language) and server side scripting—inserting the contents of one file (or the output of a system call) into another one dynamically. Once the dust had settled, three main technologies stood head and shoulders above the others. Although Perl was still a popular scripting language with a strong following PHP's simplicity and built-in links to the MySQL database program had earned it more than double the number of users. And JavaScript, which had become an essential part of the equation for dynamically manipulating CSS (Cascading Style Sheets) now took on the even more muscular task of handling the client side of the Ajax process. Under Ajax, web pages perform data handling and send requests to web servers in the background—without the web user being aware that this is going on. No doubt the symbiotic nature of PHP and MySQL helped propel them both forward, but what attracted developers to them in the first place? The simple answer has to be the ease with which you can use them to quickly create dynamic elements on websites. MySQL is a fast and powerful yet easy-to-use database system that offers just about anything a website would need in order to find and serve

up data to browsers. When PHP allies with MySQL to store and retrieve this data, you have the fundamental parts required for the development of social networking sites and the beginnings of Web 2.0

- Php form Handling

```
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
```

- PHP Program to demonstrate the use of PHP mail function.

```
<?php
    $to = abcdef@gmail.com;
    $subject = "TEST SENDING MAIL USING PHP";
    $message = "hi there I have send you the mail kindly RSPV";
    $from = hodiou@ymail.com;
    $headers = "From: $from";
```

```
Mail($to,$subject,$message,$headers);  
Echo "Mail Sent Successfully ";  
?>
```

**Exercise:**

1. Write a PHP Code to send mail to all mail ids stored in an array.
2. Write a PHP code to upload file.
3. Write a PHP code to create session based login control without database connectivity

**For Further Reading Student May Refer to the following Books**

- Programming PHP, 3rd Edition By O'Reilly.
- Beginning PHP 6 and MySQL By Steve Suehring, Tim Converse, Joyce Park. (Available in Departmental and Central Library at University campus)

## **Experiment # 11 Learning Database Connectivity between PHP and MySql, Create a login Control for a web page to demonstrate the use of Connectivity and Basic retrieval of data from database**

### **OBJECTIVE:**

- Understanding the basic concepts of login Control used in most web based User Authentication system.
- Developing ability to analyze the problem statement related to navigating pages as per data given at run time.

### **After completing this experiment you will be able to:**

- Established connection between PHP and MySQL
- Established secure connection for database connectivity.
- Create Student Entry Form and save all data to the database.
- Create a dash board web page for individual person, entity, or group.

**Requirements :-** Tools / Apparatus: Unix/Linux/Windows Operating System, Text Editor, LAMP STACK

### **Theoretical Foundations**

In the following example we store the connection in a variable (\$con) for later use in the script:

```
• <?php
// Create connection
• $con=mysqli_connect("example.com","peter","abc123","my_db");
```

```
// Check connection
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

Close Connection

The connection will be closed automatically when the script ends. To close the connection before, use the `mysqli_close()` function:

```
<?php
$con=mysqli_connect("example.com","peter","abc123","my_db");
```

```
// Check connection
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
```

```
mysqli_close($con);  
?>
```

### Open a Connection to the MySQL Server

Before we can access data in a database, we must open a connection to the MySQL server. In PHP, this is done with the `mysqli_connect()` function.

**Syntax :-** `mysqli_connect(host,username,password,dbname);`

Parameter	Description
Host	Optional. Either a host name or an IP address
Username	Optional. The MySQL user name
Password	Optional. The password to log in with
Dbname	Optional. The default database to be used when performing queries

**Note:** There are more available parameters, but the ones listed above are the most important.

### Practical Approach:- Create a login Control for a web page to demonstrate the use of Connectivity and Basic retrieval of data from databas

Step 1:- Use following sql query to create users table

```
CREATE TABLE `users` (  
  `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 `users`
```

Here we are using two database fields , Namely username for storing user id's for logins and password for storing the passwords.

Step 2:- Create login screen After that we need to create a login screen which will ultimately autehnticate user via database !, Let's create a small page and form named login\_home.php

```
<form name="form1" method="post" action="login_check.php">  
<strong>Member Login </strong>  
<br />  
Username : <input name="myusername" type="text" id="myusername"/>  
<br />  
Password : <input name="mypassword" type="text" id="mypassword"/>  
<br />  
<input type="submit" name="Submit" value="Login"/>  
</form>
```

### Step: 3 Authenticating users

Once a user has entered the username and password on the login\_home.php , the form pass the values to login\_check.php via POST variables , This method is defined at the form attribute element on login\_home.php.

Note:- While the values are passed via `$_POST` variables, we will be querying passwords in database after encrypting them with md5 function .



```

< ?php
$db_host="localhost"; // Host name of your database server !, In 90% cases it's localhost
$db_uname=""; // Mysql username of your database
$db_passwd=""; // Mysql password of your database
$db_name="test"; // Database name
$db_table_name="users"; // Name of the table we created earlier

// Connect to server and select database.
$conn = mysql_connect("$db_host", "$db_uname", "$db_passwd") or die ("cannot connect" .
mysql_error());
$conn_db = mysql_select_db("$db_name" , $conn ) or die("cannot select DB" . mysql_error());

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

// To protect MySQL injection (more detail about MySQL injection)
$form_uname = stripslashes($form_uname);
$form_passwd = stripslashes($form_passwd);
$form_uname = mysql_real_escape_string($form_uname);
$form_passwd = mysql_real_escape_string($form_passwd);
$form_passwd = md5($form_passwd); //md5 encryption for querying the password from database

$sql="SELECT * FROM $db_table_name WHERE username='$form_uname' and
password='$form_passwd'";
$result=mysql_query($sql);

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);
// If result matched $form_uname and $form_passwd, table row must be 1 row

if($count==1){
// Register $form_uname, $form_passwd and redirect to file "login_success.php"
$logged_in = TRUE;
session_register("form_uname");
session_register("logged_in");
header("location:login_success.php");
}
else {
echo "Wrong Username or Password";
}
?>

```

Please note that this is not the best way to store passwords is to either to encrypt it or use separate algorithm to incorporate security, This code will create sessions which can use to authenticate users on the particular pages , These pages just need to check the session value of logged\_in , If logged\_in is set for true then the user is authenticated and else it will be redirected to the login page

### Exercise:

- 1 Write a PHP code to save all data given by the Student on a web form
- 2 Design a Registration Form for Patient in the Hospital
- 3 Design a query form along with details section on what to discuss, problem statement, suggested Options etc. and answer the query through database.

### **For Further Reading Student May Refer to the following Books**

- Programming PHP, 3rd Edition By O'Reilly.
- Beginning PHP 6 and MySQL By Steve Suehring, Tim Converse, Joyce Park. (Available in Departmental and Central Library at University campus)

## **Experiment # 13 A Mini Project to create a website for University Utilities Allotment of already discussed Project like E-Notice Board, Asset Management, Event Management etc**

### **OBJECTIVE**

- Students should be able to Create simple Database management Web based Solutions
- Lets Student understand the real world Web Based Software Industry Strategies.

### **After completing this experiment you will be able to:**

- Students will be ready to explore real world web entity as a web based entity.
- Student will be able to explore web development strategies
- Will become sound in all phase of software life cycle development.
- Is ready to send on Industry Oriented Training Programs.

### **Requirements: - Tools / Apparatus: Unix/Linux/Windows Operating System**

In this phase students will be given various university related web based Utilities as form of Data management systems. Some of the Examples are given below:-

- Office Assets Management System
- Students Notice Board Management System - Digital Notice Boards
- Student Grievances Management System
- Student performance management System.
- Faculty Arrangement Management System
- Rolling Duty Management System.
- Time Table Generation System
- Tutorial Website      A Complete Gate and IES Assistance Gateway
- Technical Blogging for students sharing their knowledge, tips and tricks
- Event Management System

\*NOTE :- Students are suggested to choose project from the above list or any project can be under-taken with more complexities for better understandability to the problem specifications.

### **Exercise:**

- Proper Drill for Documentation for all phases used in Web Development Strategies opted by Student or group of Students (Max can be 3).
- Well Documented Pages.
- Proper record sheet for maintaining and updating the project.