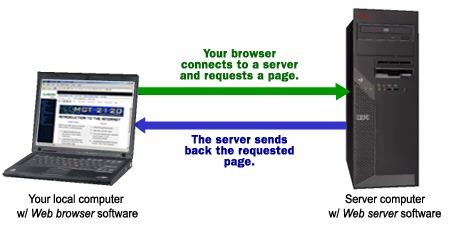
# Experiment No - 3

***Aim:*** To study installation and configuration of Web server.

***THEORY:***

A **web server** is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web. The term can refer to the entire system, or specifically to the software that accepts and supervises the HTTP requests.The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to text content.



A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary storage, but this is not necessarily the case and depends on how the web server is implemented.

While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files.

The process is an example of the client/server model. All computers that host Web sites must have Web server programs. Leading Web servers include Apache (the most widely-installed Web server), Microsoft's Internet Information Server (IIS) and nginx (pronounced *engine X*) from NGNIX. Other Web servers include Novell's NetWare server, Google Web Server (GWS) and IBM's family of Domino servers.

Web servers often come as part of a larger package of Internet- and intranet-related programs for serving email, downloading requests for File Transfer Protocol ([FTP](http://searchenterprisewan.techtarget.com/definition/File-Transfer-Protocol)) files, and building and publishing Web pages. Considerations in choosing a Web server include how well it works with the operating system and other servers, its ability to handle server-side programming, security characteristics, and the particular publishing, search engine and site building tools that come with it.

Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to generate HTML documents dynamically ("on-the-fly") as opposed to returning static documents. The former is primarily used for retrieving and/or modifying information from databases. The latter is typically much faster and more easily cached but cannot deliver dynamic content.

Web servers are not only used for serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring or administering the device in question. This usually means that no additional software has to be installed on the client computer; since only a web browser is required (which now is included with most operating systems).

Most web servers have features that allow you to do the following:

* Create one or more websites. (No I don't mean build a set of web pages. What I mean is, set up the website in the web server, so that the website can be viewed via HTTP)
* Configure log file settings, including where the log files are saved, what data to include on the log files etc. (Log files can be used to analyze traffic etc)
* Configure website/directory security. For example, which user accounts are/aren't allowed to view the website, which IP addresses are/aren't allowed to view the website etc.
* Create an FTP site. An FTP site allows users to transfer files to and from the site.
* Create virtual directories, and map them to physical directories
* Configure/nominate custom error pages. This allows you to build and display user friendly error messages on your website. For example, you can specify which page is displayed when a user tries to access a page that doesn't exist (i.e. a 404 error).
* Specify default documents. Default documents are those that are displayed when no filename is specified. For example, if you open http://localhost, which file should be displayed? This is typically index.html or similar but it doesn't need to be. You could nominate index.cfm if your website is using ColdFusion. You could also nominate a 2nd choice (in case there is no index.cfm file), and a 3rd choice, and so on.

XAMPP

**XAMPP** is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

The term XAMPP is an apparent acronym. However, there is no official acronym expansion specified on the Apache Friends website. Their homepage header reads “XAMPP Apache + MariaDB + PHP + Perl”, indicating that this abbreviation is a [recursive acronym](https://en.wikipedia.org/wiki/Recursive_acronym).

The term can be unofficially broken down as follows:

|  |  |
| --- | --- |
| **Letter** | **Meaning** |
| X | XAMPP  or an ideographic letter X, meaning cross platform |
| A | Apache  or its expanded form, Apache HTTP Server |
| M | MariaDB (formerly: MySQL) |
| P | PHP |
| P | Perl |

**FEATURES**

XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also comes with a number of other modules including OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version).

## Steps To Install XAMPP in Ubuntu

**1.**Download XAMPP Package for Ubuntu

Head over to XAMMP’s official download page to choose the one you want to install. You would observe the download section for Linux when you scroll down. Depending on your computer architecture *(32-bit or 64-bit)*, download any one of them and install XAMPP in Ubuntu. If you know the exact PHP version that you need, choose the appropriate one.

**2.** After the download completes, navigate your way to the folder where the file resides using the following command.

**Note**: *Type the commands by opening the****Terminal****(press****ctrl + alt + T****). Here, our file exists on “Downloads” directory.*

Here’s what you’ve to type:

cd Downloads

**3.**Convert the downloaded file to an executable form by typing the following command:

***Additional note:*** *Use the exact file name you have downloaded. The following example uses a 64-bit installation file with PHP 5.5.37*

chmod +x xampp-linux-x64-5.5.37-0-installer.run

**4.**Now, run the installer as the root user using s*udo* command

sudo ./xampp-linux-x64-5.5.37-0-installer.run

The installer opens after entering the mentioned command

**5.**Now that you’ve opened the installer. All you have to do now is – follow the on-screen instructions to complete the setup.

## Final Setup Process To Install XAMPP in Ubuntu

**6**. Click on “**Next**“.

**7.**Now, you’ll get to choose the components to install. Leave as it is. And, click “**Next**“.

**8.** You’ll observe the installation directory as opt/lampp. There’s nothing to be done here. Just don’t miss knowing the directory where it gets installed. Now, click “**Next**“.

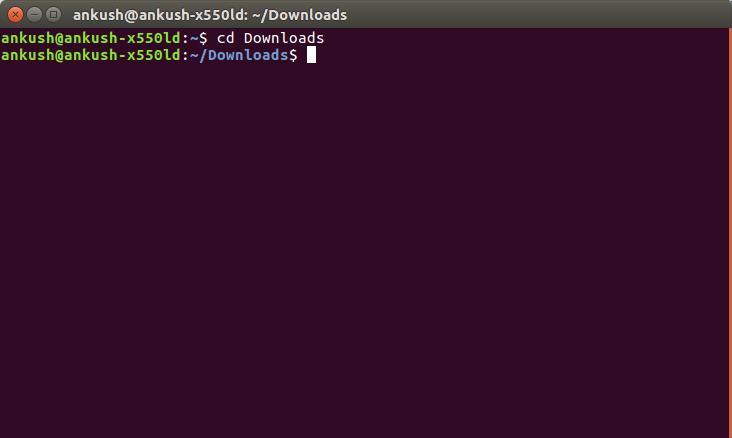
**9.** *Bitnami* would be installed by default to enable access to some modules (as you can see in the image below) that you can utilize. Finally, click “**Next**” to start the installation.

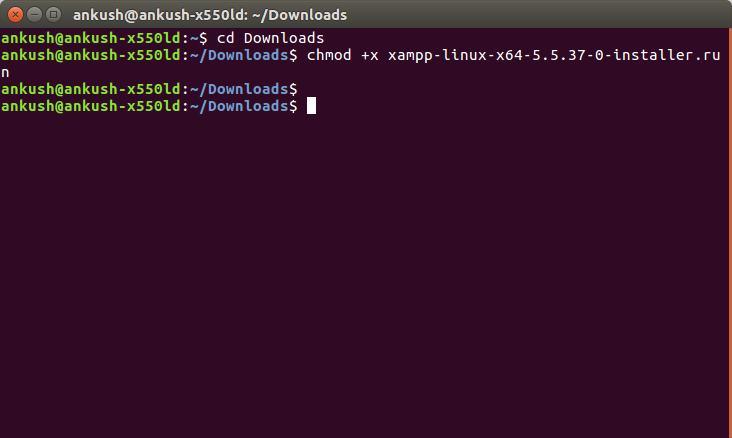
**10.** You will have to wait for a few minutes (*3-5 mins*) for the installation to complete. After the installation is done, click “**Finish**“.

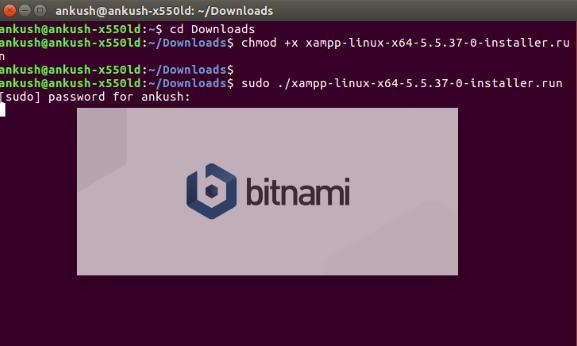
**11**. Now, you’ll be greeted with the **XAMPP** control center dialogue box.

**12.**Click on “**Go To Application**” to get started.

***Output:***

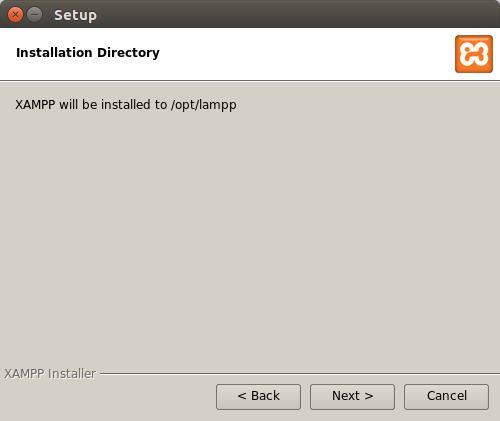




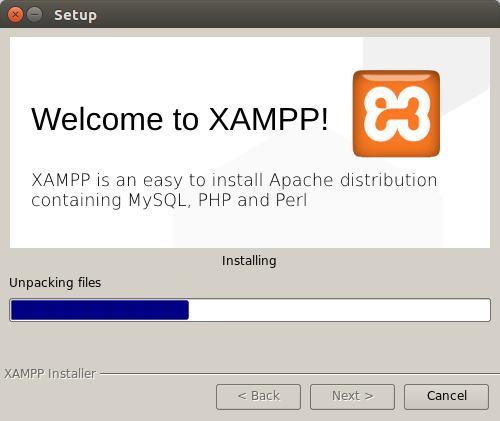














***Conclusion:*** Hence Web Server has been installed and studied.