Linux

Introduction to Open Source Software

Building a Web Server

ALUMNI BY MTHREE CONSULTING

What is Open Source Software?

Open Source Software (OSS) is a type of computer software whose source code is under a license that allows anyone to study, change and distribute the software to anyone for any purpose.

Linux is based on OSS as are many popular tools.

Most OSS written with Linux in mind follows a similar build design.

OSS Web Tools

Two most popular Linux web servers: Apache and Nginix

Apache

Solid, but large
Very flexible – 60+ modules
.htaccess file flexibility
Built around “1 thread/request” model
Better Windows performance

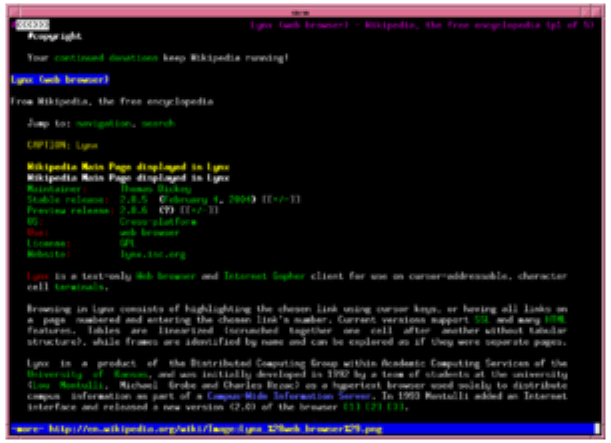
Nginx

2.5x faster for **static content**
Lighter weight
Uses an event driven model
Not as feature rich as Apache

OSS Web Tools

Simple command line web browser: Lynx

One of the earliest browsers (1992)
No graphics.
No JavaScript.
Just text HTML.
Good for simple command-line work.
Available on most platforms.



```
[illegible]                       Lynx (web browser) - Wikipedia, the free encyclopedia (p1 of 5)
   #copyright

   Your continued donations keep Wikipedia running!

Lynx (web browser)

From Wikipedia, the free encyclopedia

   Jump to: navigation, search

   CAPTION: Lynx

   Wikipedia Main Page displayed in Lynx
   Wikipedia Main Page displayed in Lynx
   Maintainer:       Thomas Dickey
   Stable release:   2.8.5  (February 4, 2004) [[+/-]]
   Preview release:  2.8.6  (?) [[+/-]]
   OS:               Cross-platform
   Use:              web browser
   License:          GPL
   Website:          lynx.isc.org

   Lynx is a text-only Web browser and Internet Gopher client for use on cursor-addressable, character
   cell terminals.

   Browsing in Lynx consists of highlighting the chosen link using cursor keys, or having all links on
   a  page  numbered and entering the chosen link's number. Current versions support SSL and many HTML
   features.  Tables  are  linearized  (scrunched  together  one  cell  after  another without tabular
   structure), while frames are identified by name and can be explored as if they were separate pages.

   Lynx  is  a  product  of  the Distributed Computing Group within Academic Computing Services of the
   University  of  Kansas, and was initially developed in 1992 by a team of students at the university
   (Lou  Montulli,  Michael  Grobe and Charles Rezac) as a hypertext browser used solely to distribute
   campus  information as part of a Campus-Wide Information Server. In 1993 Montulli added an Internet
   interface and released a new version (2.0) of the browser [1] [2] [3].

-more- http://en.wikipedia.org/wiki/Image:Lynx_%28web_browser%29.png
```

ALUMNI
BY MTHREE CONSULTING

Before we begin…

We are going to create a lot of files, so we want to isolate them from our other coursework by creating a ‘`www`’ subdirectory.

```
[usr@lndev ~/]$ cd ~
[usr@lndev ~/]$ mkdir www
[usr@lndev ~/]$ cd www
```

When we’re done with this exercise, we will delete everything in ‘`www`’.

Getting the Apache HTTPD Source

Download the source tarball for Apache HTTPD:

That's a capital "O"

```
[usr@lndev ~/www]$ curl -O
http://mirrors.ocf.berkeley.edu/apache/httpd/httpd-2.4.38.tar.gz
```

Unpack the source distribution:

```
[usr@lndev ~/www]$ tar xzvf httpd-2.4.38.tar.gz
```

Take a look at the files in the source distribution:

```
[usr@lndev ~/www]$ cd httpd-2.4.38
[usr@lndev ~/httpd-2.4.38]$ ls
```

Getting the Apache HTTPD Source

Take a look at the files in the source distribution:

```
[usr@lndev ~/httpd-2.4.38]$ ls
```

```
lndev:/home/bbernstein/www/httpd-2.4.38> ls -F
ABOUT_APACHE      buildconf*       httpd.dsp        libhttpd.mak    README.cmake
acinclude.m4      CHANGES          httpd.mak        LICENSE         README.platforms
Apache-apr2.dsw   CMakeLists.txt   httpd.spec       Makefile.in     ROADMAP
Apache.dsw        config.layout    include/         Makefile.win    server/
apache_probes.d   configure*       INSTALL          modules/        srclib/
ap.d              configure.in     InstallBin.dsp   NOTICE          support/
build/            docs/            LAYOUT           NWGNUmakefile   test/
BuildAll.dsp      emacs-style      libhttpd.dep     os/             VERSIONING
BuildBin.dsp      httpd.dep        libhttpd.dsp     README
```

Building the Apache HTTPD Source

We need to run the configure script, but first look at what it offers.

```
[usr@lndev ~/httpd-2.4.38]$ ./configure --help
```

- What are the common options?
- What are the Apache-specific options?
- How can the C compiler be influenced?

We can accept the default choices, but we must specify where it will install to:

```
[usr@lndev ~/httpd-2.4.38]$ ./configure --prefix=/home/bbernstein/www/httpd
```

Watch how it determines OS, build environment, compiler abilities and then generates Makefiles.

Building the Apache HTTPD Source

Once configure is done, build and install.

```
[usr@lndev ~/httpd-2.4.38]$ make install
```

When it completes, go into the install directory and look at the files:

```
[usr@lndev ~/httpd-2.4.38]$ cd ../httpd
[usr@lndev ~/httpd]$ ls
```

```
lndev:/home/bbernstein/www/httpd> ls -F
bin/     cgi-bin/  error/    icons/    logs/  manual/
build/   conf/     htdocs/   include/  man/   modules/
```

1. The ‘`htdocs`’ directory is the root of our web server.
2. Activity is recorded in the ‘`logs`’ directory.
3. Start/stop scripts are contained in ‘`bin`’.
4. However, we need to configure our server first!

ALUMNI BY MTHREE CONSULTING

Configuring the Apache HTTPD Server

We need to configure our web server.

```
[usr@lndev ~/httpd]$ vi conf/httpd.conf
```

1. '`ServerRoot`' is the base of our installation (remember `--prefix`?)
2. Set '**`Listen`**' **to a unique port** (e.g. 80xx where xx is your linux #)
3. Various '`LoadModule`' is extensions of HTTPD server.
4. Various '`IfModule`' is extension-specific configuration.
5. '`ServerAdmin`' and '`ServerName`' directives are helpful.
6. '`<Directory />`' denies everything … why?
7. '`DocumentRoot`' points to your '`htdocs`' directory.
8. What does '`IfModule dir_module`' do?
9. What does '`IfModule alias_module`' do?

When done, save file and exit back to command prompt.

ALUMNI
BY MTHREE CONSULTING

Running the Apache HTTPD Server

Before we run, look at the ‘`apachectl`’ script.

```
[usr@lndev ~/httpd]$ vi bin/apachectl
```

1. Note the configuration section. What does it do?
2. This is a shell script. What options does it take?
3. What is the point of the shell script?

When ready, run your server!

```
[usr@lndev ~/httpd]$ bin/apachectl start
```

Unfortunately, there are no bells-and-whistles when it starts up. But how do you know it is running?

- Logs
- Web browser

Building the Lynx Client

Download the Lynx source and build the same way as Apache:

```
[usr@lndev ~/]$ cd ~/www
[usr@lndev ~/www]$ curl -O https://invisible-mirror.net/archives/lynx/tarballs/lynx2.8.9rel.1.tar.gz

[usr@lndev ~/www]$ cd lynx2.8.9rel.1
[usr@lndev ~/lynx2.8.9rel.1]$ ./configure --help
[usr@lndev ~/lynx2.8.9rel.1]$ ./configure --prefix=/home/bbernstein/www/lynx
[usr@lndev ~/lynx2.8.9rel.1]$ make install
```

Compare and contrast:

- Distribution directory files.
- Configure script options.
- Build behavior.

Running Lynx and Testing Your Webserver

Lynx runs from the command line. You can supply a URL as the argument.

```
[usr@lndev ~/lynx2.8.9rel.1]$ cd ../lynx
[usr@lndev ~/lynx]$ bin/lynx http://localhost:80xx
```

Extra credit:

- Why is there a ‘`bin`’ directory?
- How can lynx be available from any directory?
- What happens when you try to go to https://google.com?
- How can you do you fix the https problem?

ALUMNI BY MTHREE CONSULTING

Nginx

Now that you’ve done this for Apache and Lynx, try building a webserver with Nginx instead.

Follow similar steps for what we did for the Apache server in our base ‘`www`’ directory.

Show that you can bring up a server, but this time use port `81xx`.

Final Thoughts

~ PLEASE CLEAN UP AFTER YOURSELF ~

1. Shut down your Apache server (use "`apachectl stop`")
2. Shut down your Nginx server (use "`nginx -s stop`")
3. Delete the 'www' directory you created (use "`rm -rf ~/www`")

Questions?

Thank You

ALUMNI
BY MTHREE CONSULTING