**Installing a Web Server**

Here is an outline how to install and configure the Apache Web server, the most popular web server on the Internet since 1996.

**Download the source code**

1. Visit <http://www.apache.org/>
2. Find the “Apache HTTPD Server” Project
3. Download the latest stable release of the source code for Unix
4. Check theMD5 checksum to verify your download is not corrupted
5. Upload the file you downloaded into your home directory on Sciris

**Unpacking the Source Code**

Most Unix source code projects are distributed as “compressed tar balls”. You need to uncompress and “untar” it first. Note that many modern Linux distributions actually allow you to download precompiled software via a very convenient installation tool. However, we will do everything by hand to create a customized installation optimized for our hardware.

1. Create a directory “httpd” in your home directory
2. Move the downloaded file into that dir. Note the file size.
3. Use the “gunzip” program to decompress the tar ball. Note the file size now.
4. Use the “tar” program to unpack the tar’d file (check the man pages for details on tar)

**Familiarize yourself with the Source Code**

1. List directories and files to get an overview
2. Read the files README, LICENSE, and INSTALL for an overview of the installation procedure and the legal smallprint

**Configure, Compile, and Install**

The next step is to create the binary and executable files and to move all parts into their respective directories.

1. Run “configure” from the source directory. This will configure compile options for your system to optimally compile the source. Depending on your PATH you might have to run it by typing “./configure” – explain.
2. Run “make” from the source dir. This will compile all source code and build your binaries.
3. Try to run “make install”. This is supposed to move all parts to their respective directories.

This last step will not work, though, since you do not have access rights to the default directories where the web server wants to live. So, we have to do the whole process again, beginning with a clean slate.

* First we clean up our project: “make clean”
* Then we re-run the configuration program with a directory location where our server will end up: “./configure –prefix=”/home/username/httpd”
* Then we compile all: “make”
* Finally we install: “make install”

Finally we can start the web server: “/home/username/httpd/bin/apachectl start”. However, that will fail, something about a ‘socket error’ and ‘port 80’ – explain why.

**Configure the Web Server**

Before we can start the web server, we need to do at least some minimal configuration. So:

1. Edit the file “httpd.html” in the “conf” dir of your server installation
2. Configure your server to (i) run on port 777#, as assigned in class (ii) run as you (your user id and group), and (iii) enter your email address as administrator. Then start your server again, if possible.

Verify that your server is working properly by requesting a web page via the address http://sciris,shu.edu:777#/

**Link in some Content**

1. Personalize your root-level index file by editing the BASE/htdocs/index.html directory

**Check the Sample Scripts**

There is a BASE/cgi-bin directory, which includes so-called server-side scripts. That dir is outside the document root. Why is it accessible? Explain. Run the “test-cgi” scripts. Note: it should not run, so fix the problem by marking the file as “executable”. You should use <http://sciris.shu.edu:777#/cgi-bin/test-cgi> to run the script

**Homework**

1. Go to your web server. Then check the log file(s) of your web server. Do you see your request?
2. Use a web browser to request an invalid web page. Does the request get logged as access or error?
3. What is the format of your access log (look into the configuration file).
4. What is the process ID of your web server? How many copies of your httpd process are running?
5. The web server comes with its documentation in SRC\_BASE/docs/manual (check). Without moving or copying the directory, make its content available to web server. *Hint:* you can do that by adding an “alias” to the configuration file httpd.conf file. Note that whenever you modify your configuration file, the server needs to be restarted via “apachectl restart” before the changes will take effect. If all works, you might see a “strange” default page, but you should see *something*.
6. The CGI directory also contains a “printenv” script, written in “Perl”. That script, too, does not work out of the box – try it. Then try to fix it,
7. Modify the “test-cgi” script so that it includes today’s date and current time. Note that the command for getting the date/time is “date”.