Apache

Downloading Apache

The Apache server can be downloaded online from the project website. There is a special page for downloads. The current stable release, as of the time I write this, is 2.4.2. We however will start by installing the 2.2.2 version of the Apache server. Furthermore, since all these links are likely to change in time, and to ensure a greater reliability for this process we have provided a local copy of the server software on silo. Here’s how you get your copy:

```
[tbigrant@silo ~]$ pwd
/u/tbigrant
[tbigrant@silo ~]$ ls httpd*
ls: cannot access httpd*: No such file or directory
[tbigrant@silo ~]$ ls -ld apache
ls: cannot access apache: No such file or directory
[tbigrant@silo ~]$ ls /l/www/classes/a202/software/httpd-2.2.2.tar.gz
/l/www/classes/a202/software/httpd-2.2.2.tar.gz
[tbigrant@silo ~]$ cp /l/www/classes/a202/software/httpd-2.2.2.tar.gz .
[tbigrant@silo ~]$ ls httpd*
httpd-2.2.2.tar.gz
[tbigrant@silo ~]$ ls -ld httpd-2.2.2.tar.gz
-rw------- 1 tblgrant projects 6282043 Jun 12 10:48 httpd-2.2.2.tar.gz
[tbigrant@silo ~]$ 
```

We start with `pwd` which identifies the location (the folder and the owner). We then look for files (or folders) whose name is `apache` or start with `httpd`. There aren’t any, so we look for the promised archive in the class software folder. We find it so we copy it in the current folder (which is identified in the command line with a dot—at the end of the line, as the second argument of `cp`). As we copy the compressed (.gz) archive (tar) we also make a note of its

---

1http://httpd.apache.org/
2http://httpd.apache.org/download.cgi
3http://httpd.apache.org/download.cgi#apache24
4http://httpd.apache.org/download.cgi#apache22
5You can find it in /l/www/classes/a202/software
6Notice that the prompts appear in red, the commands issued appear in blue and the shell’s responses are in black. In the snapshot session presented above my username is `tbigrant`.

size (around 6.2 megabytes). It will have to be uncompressed. We will see then if compression really makes any difference.

## Installing Apache

Installing Apache starts with compressing and unarchiving the downloaded software bundle.

```
[tblgrant@silo ~]$ pwd
/u/tblgrant
[tblgrant@silo ~]$ ls httpd*
httpd-2.2.2.tar.gz
[tblgrant@silo ~]$ ls -ld httpd-2.2.2.tar.gz
-rw------- 1 tblgrant projects 6282043 Jun 12 10:48 httpd-2.2.2.tar.gz
[tblgrant@silo ~]$ gunzip httpd-2.2.2.tar.gz
[tblgrant@silo ~]$ ls -ld htt*
-rw------- 1 tblgrant projects 30371840 Jun 12 10:48 httpd-2.2.2.tar
[tblgrant@silo ~]$ tar xvf httpd-2.2.2.tar
httpd-2.2.2/
httpd-2.2.2/os/
httpd-2.2.2/os/os2/
httpd-2.2.2/os/os2/os.h
httpd-2.2.2/os/os2/core.mk
[...]
httpd-2.2.2/include/http_core.h
httpd-2.2.2/include/httpd.h
httpd-2.2.2/include/http_main.h
httpd-2.2.2/include/ap_release.h
httpd-2.2.2/include/.indent.pro
httpd-2.2.2/include/util_cgtree.h
[tblgrant@silo ~]$
```

The net result of these commands is that the compressed archive that had just been copied is uncompressed and unarchived in my account. The unarchiving creates a folder `httpd-2.2.2` which has everything I need to get started. I am now ready to really start installing Apache.

The only thing I need before I start is a port number. The instructor will distribute the port numbers on the course website. With my identity (username) of `tblgrant` let’s assume that my assigned port number is 61200. Let’s also assume at this time Apache port numbers 61201 for Anna (annaeile) and 61202 for Ryan (rypauise).

Here’s how I install Apache: I go into the newly created folder and look around. There is a file `INSTALL` that I look at. It has very simple installation instructions. I decide to follow them exactly. They ask me to run a piece of software (a program) that the developers have provided (it’s called `configure`) with a command line argument indicating the location of my installed server. The configuration utility produces a `Makefile` with instructions for compiling the server from source and installing the resulting files. Once `configure` ends,
I run `make` to compile and `make install` to place the compiled files and their ancillaries in the right place(s). At this point I’d be ready to start the server if my port had been specified anywhere (which it hasn’t—yet).

Step by step, up to this point, it looks like this:

```
[tblgrant@silo ~]$ pwd
/u/tblgrant
[tblgrant@silo ~]$ ls -ld apache
ls: cannot access apache: No such file or directory
[tblgrant@silo ~]$ ls -ld http*
```

```
drwx------ 11 tblgrant projects 4096 Apr 21 2006 httpd-2.2.2
-rw------- 1 tblgrant projects 30371840 Jun 12 10:48 httpd-2.2.2.tar
```

```
[tblgrant@silo ~]$ rm httpd-2.2.2.tar
[tblgrant@silo ~]$ cd httpd-2.2.2/
```

```
ls
ABOUT_APACHE configure LAYOUT README
acinclude.m4 configure.in libhttpd.dsp README-platforms
Apache dsw docs LICENSE ROADMAP
apachew.mcp.zip emacs-style Makefile.in server
build httpd.dsp Makefile.win src/lib
BuildBin.dsp httpd.spec modules support
buildconf include NOTICE test
CHANGES INSTALL NWGNUmakefile VERSIONING
config.layout InstallBin.dsp os
```

```
[tblgrant@silo httpd-2.2.2]$ more INSTALL

APACHE INSTALLATION OVERVIEW

Quick Start - Unix

For complete installation documentation, see [ht]docs/manual/install.html or http://httpd.apache.org/docs-2.2/install.html
```

```
$ ./configure --prefix=PREFIX
$ make
$ make install
$ PREFIX/bin/apachectl start
```

```
NOTES: * Replace PREFIX with the filesystem path under which [...] 
```

The box belongs to me, as a manner of highlighting, but the rest is exactly as you will find it in the files. Next, I start `configure` (I run it from the current folder) with the expected prefix. Once it finishes I take a look at the file (it’s fairly big for a script, but it does a lot of things) and I check on the files created:

```
[tblgrant@silo httpd-2.2.2]$ pwd
/u/tblgrant/httpd-2.2.2
```
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu

Configuring Apache Portable Runtime library ...

I take a snapshot of the date so you can better identify what files had just been created. Next I run make followed by make install. They take a bit to run and have considerably large outputs:

	I
	make
Making all in srclib
make[1]: Entering directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib'
Making all in pcre
make[2]: Entering directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib/pcre'
make[3]: Entering directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib/pcre'

make[3]: Leaving directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib/pcre'

make[1]: Leaving directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib'

	make install
Making install in srclib
make[1]: Entering directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib'
Making install in pcre
make[2]: Entering directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib/pcre'
make[3]: Entering directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib/pcre'

make[3]: Leaving directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib/pcre'

make[1]: Leaving directory `/nfs/nfs1/home/tblgrant/httpd-2.2.2/srclib'

mkdir /u/tblgrant/apache/man/man8
mkdir /u/tblgrant/apache/manual
Once the installation ends I move out of the folder (actually at the top of my file system with cd) and I check that the ~/apache folder has been created:

```
[tblgrant@silo httpd-2.2.2]$ pwd
/u/tblgrant/httpd-2.2.2
[tblgrant@silo httpd-2.2.2]$ cd
[tblgrant@silo httpd-2.2.2]$ pwd
/u/tblgrant
[tblgrant@silo httpd-2.2.2]$ ls -ld apache
drwxr-xr-x 14 tblgrant projects 4096 Jun 12 11:34 apache
```

Let's see what we got.

**A tour of the Apache folders**

A tour of the apache folder reveals the following important sub-folders:

```
[tblgrant@silo ~]$ pwd
/u/tblgrant
[tblgrant@silo ~]$ ls apache
bin  cgi-bin  error  icons  logs  manual
    build  conf  htdocs  include  man  modules
```

For now just the conf folder is relevant.

**Configuring Apache**

We move into the conf folder and look for a file named httpd.conf which we need to modify. Open the file with pico -w (which disables automatic wrapping) and make two changes:

- on line 40 modify Listen 80 to refer to your port.
- on line 64 in User daemon replace daemon with your username.

Then exit and save. I list the actual steps I took below. You will recognize many of the commands and actions but there will also be some new. As an example consider the use of `grep`. It is a Unix utility that can search its input for a regular pattern. We look for the word `Listen` first: there are four lines the one I changed. I then decided to look for `User` followed by a space. To indicate the space on the command line I include the string to search for in double quotes. When the search is repeated for `Listen` followed by a space, the result of the search is a bit different, as you would expect.
Starting and stopping Apache

There is a utility (a script) called `apachectl` and located in `/apache/bin` that can be used to start or stop Apache.

```bash
[tblgrant@silou ~]$ pwd
/u/tblgrant
[tblgrant@silou ~]$ cd apache
[tblgrant@silou apache]$ cd conf
[tblgrant@silou conf]$ ls -ld httpd.conf
-rw-r--r-- 1 tblgrant projects 13298 Jun 12 11:33 httpd.conf
[tblgrant@silou conf]$ pico -w httpd.conf
[tblgrant@silou ~]$ pwd
/u/tblgrant
[tblgrant@silou ~]$ ls apache
bin  cgi-bin  error  icons  logs  manual
build  conf  htdocs  include  man  modules
[tblgrant@silou ~]$ cd apache
[tblgrant@silou apache]$ cd conf
[tblgrant@silou conf]$ ls -ld httpd.conf
-rw-r--r-- 1 tblgrant projects 13298 Jun 12 11:33 httpd.conf
[tblgrant@silou conf]$ pico -w httpd.conf
[tblgrant@silou conf]$ grep Listen httpd.conf
# Listen: Allows you to bind Apache to specific IP addresses and/or
# Change this to Listen on specific IP addresses as shown below to
#Listen 12.34.56.78:80
Listen 80
[tblgrant@silou conf]$ grep "User " httpd.conf
User tblgrant
# User home directories
[tblgrant@silou conf]$ grep "Listen " httpd.conf
# Change this to Listen on specific IP addresses as shown below to
#Listen 12.34.56.78:80
Listen 80
[tblgrant@silou conf]$
```

Starting and stopping Apache

There is a utility (a script) called `apachectl` and located in `/apache/bin` that can be used to start or stop Apache.

```bash
[tblgrant@silou conf]$ pwd
/u/tblgrant/apache/conf
[tblgrant@silou conf]$ cd ../bin
[tblgrant@silou bin]$ ls
ab  checkgid  envvars-std  htdigest  httxt2dbm
apache1  dbmmanage  htcacheclean  htpasswd  logresolve
apxs  envvars  htdbms  httpd  rotatelogs
[tblgrant@silou bin]$ ps -ef | grep tblgrant
.tblgrant 15718 31344 1 12:23 pts/51 00:00:00 ps -ef
tblgrant 15719 31344 0 12:23 pts/51 00:00:00 grep tblgrant
dgerman 31340 31272 0 10:41 pts/30 00:00:00 ssh silo -l tblgrant
root 31341 23297 0 10:41 ? 00:00:00 sshd: tblgrant [priv]
tblgrant 31343 31341 0 10:41 ? 00:00:00 sshd: tblgrant@pts/51
```
Notice the use of the `ps` utility: it is called with the `-ef` command line switches, and its output is piped (using the Unix pipe `|` symbol) into `grep`. We look for lines that have `tblgrant` (or in your case, your username) on them. There is no indication at this time that your server is running. By comparison it looks like `dgerman` has an Apache server up and running. Also it looks like he is logged into `silo` as `tblgrant`. Next we start the server and check to see the difference.

```
[tblgrant@silo bin]$ ./apachectl start
[tblgrant@silo bin]$ ps -ef | grep tblgrant
```

We stop it, check, start it again. Notice that we can also check activity on a certain port with `netstat` too:

```
[tblgrant@silo bin]$ ~/apache/bin/apachectl stop
[tblgrant@silo bin]$ ps -ef | grep tblgrant
```
tblgrant 15752 31344 0 12:23 pts/51 00:00:00 grep tblgrant
dgerman 31340 31272 0 10:41 pts/30 00:00:00 ssh silo -l tblgrant
root 31341 23297 0 10:41 ? 00:00:00 sshd: tblgrant [priv]
tblgrant 31343 31341 0 10:41 ? 00:00:00 sshd: tblgrant@pts/51
tblgrant 31344 31343 0 10:41 pts/51 00:00:00 -bash
[tblgrant@silo bin]$ ~/apache/bin/apachectl start
[tblgrant@silo bin]$ netstat -a | grep 61200
   tcp  0    0  *:61200 *:*      LISTEN
[tblgrant@silo bin]$ ~/apache/bin/apachectl stop
[tblgrant@silo bin]$ netstat -a | grep 61200
[tblgrant@silo bin]$ ~/apache/bin/apachectl start
[tblgrant@silo bin]$ 

When the server is running a process is listed listening on that port. If the server is stopped, no activity is detected on that port. Before you start your installation of Apache this is also a good way to verify that your port number has not been taken already.

**Accessing your Apache server with a browser**

I point my browser to http://silo.cs.indiana.edu:61200/ and I see that my server is running. I stop it and I don’t get any reply. Do the same, just use your own port number. Port numbers will be distributed in the first day of classes.