

A step-by-step guide to installing FreeBSD 5. It assumes moderate experience with linux and leaves you with a fully updated FreeBSD system.

-----

-----

//////////\*1000+ HACKING TRICKS & TUTORIALS - ebook By Mukesh Bhardwaj Blogger - Paid Version - only @ TekGyd | itechhacks | Mukeshtricks4u\*////////

-----

-----

## FreeBSD Installation

### A. 5.x vs 4.x

The first thing to understand about FreeBSD is that there are two lines of development. The -STABLE branch is marked with a 4.x version number and the most recent version is 4.10. It is well tested and very solid, but does not include the most recent technology. The -CURRENT branch, marked with a 5.x version, is the “unstable” branch. However, it is nicely stable at the moment and is coming along quite well. Most users should go with 5.x and these instructions are only valid for that tree.

NOTE: DragonFlyBSD ( [www.dragonflybsd.org](http://www.dragonflybsd.org) ) is a continuation of the 4.x line. It uses lock-less (no mutexes) SMP support and a Light Weight Kernel Threading system. It has a lot of promise and is developing at a breakneck pace, but it should still be thought of as “R/D.”

### B. Getting Media

I am not going to say much about this. There are links to various ftp mirrors at [www.freebsd.org](http://www.freebsd.org) and the directory structure is fairly self-explanatory. There are however several choices for ISO. You should choose the miniinst ISO. It is small and will include everything you need for the base system.

### C. Starting Installation

After, the CD boots up you will enter the...ahh.....majestic sysinstall. You can safely ignore most of the options and just choose a standard installation. Most of the install process is pretty easy and anyone who has some experience with linux or unix will be able to handle it without much stress. However, creating partitions and dealing with drives will seem very odd to your standard linux user.

### D. Hard Disc Management in FreeBSD

Ok, the first thing to get used to here is that IDE drives are not hda, hdb...etc. They are ad0, ad1 and so forth. SCSI discs are da0, da1...and so on. There is one other thing that is going to freak some people out. You create slices, not partitions, on the disc and then create partitions within those slices. For example, the first partition in the first slice on the first IDE disc would be ad0s1a. Just accept it.

### E. Partitioning

Sysinstall will lead you through the partitioning and its fairly easy to understand. The first part will ask you to choose a disc or discs to partition and then will show you a "slice editor." This is where you will create your slice. I advise you to only make one. While multiple slices are easy to deal with, it just adds complication. If this is not your first installation of a BSD type OS, then you can ignore me and why are you reading this again?

After creating your slice, you will be prompted to choose the drive(s) to install an MBR on. The FreeBSD boot loader is nothing to write home about, but it tends towards working. After this step, you can create partitions. There is not much to say here. At the top of the screen it will show your slice(s) and when one of them is selected you can use the controls to create a partition on it. You will need at least a root and swap partition. On the non-swap partitions it is usually a good idea to enable soft-updates.

#### F. Distribution Sets

This is a simple section, select minimal.

#### G. Continue Installation

The rest of this is pretty simple. Make sure you install from the CD and not the Internet. The bulk of the install is now done. After it copies files to your disc, it will start the configuration process. This is all pretty self-evident, but there are a couple things you should know.

#### Network Configuration

Don't be scared by the names, in \*BSD devices are named after their drivers. There is also a short description after the name, so you should be able to choose the right one. The rest of the network config is easy, just follow the prompts.

#### System Console Configuration

You can pretty much ignore this, you may want to look around for your own knowledge and of course you could pick a nice screen saver here, but other than that I would leave it alone for now.

#### Time Zone

All I have to say here is that if you live in the US, after you choose "America – North and South" hit the end key. The US is at the bottom of the list and hitting end is the quickest way there.

Say no, we will do this later and with an updated linux\_base.

## Mouse

OK, welcome to the wonderful and amazing world of moused. Answer the first question truthfully, and then you can tweak the settings in the "Please configure your mouse" dialog. Whatever you do, be sure to enable the daemon. Also, for most users that is all you will have to do. You can safely ignore the other options.

## Package Installation

At this point, the installer will ask you to install binary packages. Say no. These binaries are out of date and not included on our CD.

## Of users and roots

This isn't the most thrilling section, add a user when it asks you to and set a root password after that. The only thing I have to say about this process is when the new user dialog comes up leave the "Group:" box unchanged and add "wheel" to the Member groups. You also might want to set your shell to /bin/tcsh. As for setting the root password, if you can't handle that we have bigger problems.

## Rebooting

Ok, next it will ask if you would like to visit the general config area. Select no and you will be brought back to the main menu. Exit the install, reboot without the CD in and enjoy the boot messages.

## H. The Real Post-Install

At this point, I am going to assume that you are now looking at a login prompt and thinking "my my...FreeBSD boots quite quickly doesn't it." Well, our task is not done yet...there is a reason we did a minimal install. We are going to do most of it ourselves. First, lets upgrade to -CURRENT. This isn't a practice I would usually recommend, but 5.x is close to being tagged stable and -CURRENT is rather solid at the moment. First, I need to explain how things are done in the BSD world.

## Sup?

CVS up; you up; we all CVSUP. Cvsup is a very interesting program that I am not going to explain in detail here. All you really need to know is that it updates source trees. You see, that is the thing. You may be used to /usr/src not doing much. In BSD it has a job, it holds the source for the entire base system. However, we did a minimal install and no source is there. It wouldn't be up2date anyway. So, lets fix that. Login as root

and type the following: `pkg_add -r cvsup-without-gui`

`pkg_add` is the binary installer for FreeBSD and the `-r` argument tells it to fetch binaries from the net. It will also fetch any deps that you might need. Switch to another console while this is happening and login as root. Do the following:

```
cd /etc
```

```
cp /usr/share/examples/etc/make.conf /etc/make.conf
```

```
cp /usr/share/examples/cvsup/standard-supfile /usr
```

```
cp /usr/share/examples/cvsup/ports-supfile /usr
```

```
chmod u+w /etc/make.conf /usr/standard-supfile /usr/ports-supfile
```

What was that? Well here is the rundown. `Make.conf` is the file that controls the building of programs from source on FreeBSD and the supfiles tell `cvsup` where to get the source for the base system and the ports system, also where to put said source. They come out of `/usr/share/examples` without the write bit set and that gets annoying. So we set that. Now switch back to the first console and type `rehash`. This tells `tcsh` to check its path for new programs. Now, edit the standard supfile that is in `/usr`. You can either use `ee` or `vi`. I like `vi`. Scroll down to the line that looks a bit odd. It will be something like:

```
*default host=CHANGE_THIS.FreeBSD.org
```

The “CHANGE\_THIS” is where you put what `cvsup` server to use. Choose a number between 1 and 9, like 4, and put `cvsup4` where `CHANGE_THIS` is. So it would end up being:

```
*default host=cvsup4.FreeBSD.org
```

Now exit your editor and run `cvsup /usr/standard-supfile`

If everything goes correctly, you will see a lot of text scrolling on the screen. If it says something about a bad connection, try another number.

## Make.conf

Now 'tis the time for all good men (and women) to edit their make.conf . This is not difficult, in fact have a look around the file. It may be long, but it is pretty simple. Now, uncomment the CPU setting and the CFLAGS setting. Set the CPU to your CPU (there will be a list in the comments above the setting) and set the CFLAGS to -O2. (NOTE: If the base system fails to build, downgrade your CFLAGS back to -O).

## Build Your World

When CVSUP finishes (it will be awhile...go get some coffee), cd to /usr/src and run make buildworld. That command will do exactly what it sounds like. It builds your world, or base system. While its doing that, lets get you a kernel. First, cd to /usr/src/sys/i386/conf then cp GENERIC to some file of your choosing. Any name you want. However, be aware that this is going to show up in a uname -a. Now, crack open your new file and take a look at what a FreeBSD kernel config is like. There are many things you can do here that will improve performance and subtract in size. However, lets keep it simple. Near the top of the file will be the name GENERIC. Change that to your new name. You can now look through the file and you will find several sections that are just for debugging. These will indeed add size and slow down the kernel a bit, but I would leave them for now. Go to the end of the file and make a new line. Add the following:

```
device pcm
```

This will add sound support to the kernel. Be aware that you don't need to do this, you can load binary modules at boot or after boot, but this way is easy and sound is used often. Save the file and exit. Go back to your buildworld console and when its done execute make buildkernel KERNCONF=YOUR\_KERNEL\_CONFIG\_NAME

NOTE: NOT THE PATH OF THE KERNEL CONFIG..it knows to look in /usr/src/sys/i386/conf

## Installing Your World

This isn't that difficult. First run make installkernel KERNCONF= your config name. Now here is the interesting part. Run mergemaster -p, this program looks through your etc and updates it to match the new /etc in /usr/src. It will display changed files to you, press q and it will give you options. Something like i (install), m (merge) etc. Pick merge and it will open a nice little screen that shows you one file on the right and one on the left. It will go section by section, showing the areas that have changed. Press r or l to choose which section to keep. Its pretty easy to see which section has new stuff and which does not. After the merge, it will prompt you with options for the newly merged file. One of these will be install and this is the one you want. In the latest current, most of what you will be showed is user and group files. Make sure you do select the sections with the new users and groups. After this is done, it will ask you a couple questions that you can say yes to. Now that your /etc is updated run make installworld then reboot.

NOTE: You usually don't need to run mergemaster. However, 5.2.1 is a pretty old release and -CURRENT has come a long way.

## I. Ports

Welcome to running current. The rest is easy. cd back to /usr and edit the ports supfile the same way you did the standard one. Run cvsup on it and wait. After its done, you will have a full ports tree. There is not much left to say. You now have a working system and a fully updated one too. To install software from ports cd to /usr/ports/category/softwarename/ and run make install clean . If you want linux binary support, install the linux\_base port. To find where it is cd to /usr/ports and run make search name=linux\_base | less . Enable loading the kernel modules for linux binary support by editing /etc/rc.conf. Just add the line linux\_enable="YES" to the file and your set. If this is a desktop system, I would recommend installing /usr/ports/x11/xorg and your choice of /usr/ports/x11/gnome2 or kde3. Have fun .

Get it from:

<http://www.madpenguin.org/cms/?m=show&id=1853>