DISKLESS TUTORIAL updated: 12/03/06

Introduction
The purpose of this step-by-step guide is to make life easier for people who want to setup a diskless environment. It took me a good while when I was new to *OpenBSD* to setup my first diskless server so I hope this doc will be useful to users and myself once my memory are cleared. Note there are no warranties if the steps in this document make your 'puter blow up or cause any data loss.

Who is the author?
My name is Bachman Kharazmi and I'm a Swedish student at LTH (Lund institute of technology). And no, I'm not a CS student.

Feel free to email (*bahkha AT gmail DOT com*) any suggestions or just correct me if I've made any human mistakes.

- **Before**

Make sure you've loads of time, patience and working hardware that are supported by *OpenBSD*. Please read `http://www.openbsd.org/i386.html#hardware`.

- **Hardware performance**

Most important is your diskless servers hardware. I would say that hard drive and network speed are the bottlenecks in most cases. A 10k rpm disk with at least 8mb cache and gigabit network would really speed your diskless environment.

- **Let us start...**

Start by installing *OpenBSD*. No packages needed other than what's in the default install. This guide doesn't cover the installation of the OS. Make sure to have some free space in `/home`, my definition of some is that it's a individual option depending on the number of diskless clients you want. By first reading 'man diskless' it will be easier to read this doc. The steps bellow are made on a i386 architecture with PXE, for anyone on a non-i386 platform please read the 'diskless' manual.

The clients will mount the servers `/usr` over nfs so everything you want to use on your clients must exist on the server.

- **Services**

What services will be needed ?
o **nfsd** - nfsd runs on a server machine to service NFS requests from client machines.
o **tftpd** - is a server for the IPv4 Trivial File Transfer Protocol. The TFTP protocol is extensively used to support remote booting of diskless devices. The server is normally started by **inetd**, but can also run standalone.
o **dhcpd** - server that give clients their IPs.
o **rarpd** - upon receiving a request, rarpd maps the target hardware address to an IP address via its name.
o **bootparamd** - bootparamd is a server process that provides information to diskless clients necessary for booting

(note that reading manuals related to the services above will give you a deeper understanding if you want to setup something with your own requirements)

To make it simple maintaining the diskless clients we call the first one 'term-1'.
We start by creating directories and files and then explaining how to setup the services mentioned above. Please check appendix for network design.

As root do:

```
#mkdir -p /home/{tftpboot,term-1/root/{usr,swap}}
#dd bs=128k count=1024 if=/dev/zero of=/home/term-1/swap
```

What we've done here is to create directories for our (first) diskless client and a 128Mb swap file that also will be mounted over nfs.
Now we continue by configuring the services mentioned earlier.
Use your favorite editor to create `/etc/ethers` . The file will contain the MAC address of the NICs and diskless client hostnames.

example:

```
bkw@nemo:~/ > cat /etc/ethers
00:0A:5E:46:3A:DF       term-1
```

Set up a rpc daemon by editing `/etc/bootparams` (it won't exist as default).

example:

```
term-1 root=192.168.10.1:/home/term-1/root
swap=192.168.10.1:/home/term-1/swap
```
(note that you've to replace the IP with your diskless-servers)

Edit `/etc/hosts` and add client IP and hostname which is term-1 in our case.
example:

```
bkw@nemo:~/ > cat /etc/hosts
::1 localhost.lan localhost
127.0.0.1 localhost.lan localhost
192.168.10.1 nemo.lan nemo
192.168.10.2 term-1.lan term-1
```

(note that .lan is our domain)

Now get the baseXY.tgz and etcXY.tgz from a mirror ie ftp.openbsd.org using ftp. XY is current stable release version. Don't forget p when extracting to preserve permissions.

Extract the files:

```
#tar -zxpf baseXY.tgz -C /home/term-1/root/
#tar -zxpf etcXY.tgz -C /home/term-1/root/
```

Create special devices in `/dev`.

```
#cd /home/term-1/root/dev
#./MAKEDEV all
```

Files that you'll need to edit in the term-1 dir:
`/etc/fstab`
`/etc/myname`
copy `/etc/hosts` to etc in the clients root dir.

```
#cp /etc/hosts /home/term-1/root/etc/
```

example:

```
bkw@nemo:/home/term-1/root/etc/ > cat fstab
192.168.10.1:/home/term-1/root / nfs rw 0 0
192.168.10.1:/usr /usr nfs rw 0 0
192.168.10.1:/home/term-1/swap none swap
sw,nfsmntpt=/swap

bkw@nemo:/home/term-1/root/etc/ > cat myname
term-1
```

Now it's time to build an *OpenBSD* kernel that can boot root and swap over nfs because this isn't supported by default.

```
#cd /usr
#export CVSROOT=anoncvs@anoncvs.se.openbsd.org:/cvs
```

(should be changed to the mirror nearest you)

```
#cvs -d$CVSROOT checkout -rOPENBSD_X_Y -P src/sys
```

(while you're fetching the kernel source read 'man release' from the second section, and note that X,Y should be replaced)

When the checkout has finished:

```
#cd /sys/arch/i386/conf
#cp GENERIC DISKLESS
```

now edit the DISKLESS file and replace:

```
config          bsd     swap generic
```

with

```
config          bsd     root on nfs swap on nfs
```

(save & quit)

```
#config DISKLESS
#cd ../compile/DISKLESS
#make clean depend bsd
#mv bsd /home/tftpboot/bsd && chown root:wheel
/home/tftpboot/bsd && chmod 555 /home/tftpboot/bsd
#cd /home/tftpboot && ftp
ftp://ftp.openbsd.org/pub/OpenBSD/3.7/i386/pxeboot &&
chmod 555 pxeboot
#mkdir /home/tftpboot/etc
#echo "boot" > /home/tftpboot/etc/boot.conf
```

What we've done here is building a new kernel and copied it to our tftpd working dir from where it will be transferred to the clients, pxeboot is the bootloader that the client NIC will fetch before loading bsd kernel. The last line is to disable any delay in bootloader when we are going to initialize the kernel.

Edit the servers `/etc/inetd.conf` and make sure your tftp line looks like:

```
tftp            dgram   udp     wait    root
/usr/libexec/tftpd      tftpd -s /home/tftpboot
```

Edit `/etc/dhcpd.conf`, wipe everything you don't need and add something like:

```
subnet 192.168.10.0 netmask 255.255.255.0 {
  range 192.168.10.10 192.168.10.20;
}

group {
  host term-1 {
    hardware ethernet 00:0A:5E:46:3A:DF;
    filename "/pxeboot";        # "/tftpboot/xxx"
    fixed-address term-1;        # 192.168.10.2
    option      root-path "192.168.10.1:/home/term-
1/root";
    option      swap-server 192.168.10.1;
    option      host-name "term-1";
        }
}
```

(save & quit)
note: you will have to change MAC to the clients that you also used earlier in `/etc/ethers`.
Now we're going to configure the servers nfs shares.
Edit `/etc/exports` and make it look like:

```
bkw@nemo:~/ > cat /etc/exports
# See exports(5) for more information.  Be very careful:
misconfiguration
# of this file can result in your filesystems being
readable by the world.
/usr -maproot=root term-1 term-2 term-3 term-4 term-5
/home/term-1/root -maproot=root -alldirs term-1
/home/term-1/swap -maproot=root term-1
```

We would like all the services to be started as default when the server boots so edit `/etc/rc.conf` and make sure the specific lines I write bellow look like:

```
rarpd_flags="-a"
bootparamd_flags=""
dhcpd_flags="rl1"
nfs_server=YES
portmap=YES
inetd=YES
```

(rl1 should be replaced with the iface that the clients will be connected to.)
(save & quit)

- **Reboot**

Reboot your server. After bootup login and do a 'ps -ax' check the bold, it should look something like:

```
bash-3.00# ps ax
PID TT   STAT      TIME COMMAND
1 ??  Is      0:00.00 /sbin/init
28266 ??  Is      0:00.00 dhclient: rl0 (dhclient)
3432 ??  Is      0:00.01 syslogd: [priv] (syslogd)
1940 ??  I       0:00.02 syslogd -a /var/empty/dev/log
13776 ??  Is      0:00.00 pflogd: [priv] (pflogd)
20940 ??  Is      0:00.00 portmap
27784 ??  I       0:00.03 pflogd: [running] -s 116 -f /var/log/pflog (pflogd)
26812 ??  Is      0:00.01 mountd
28090 ??  IL      0:00.00 nfsd: server (nfsd)
27724 ??  Is      0:00.00 nfsd: master (nfsd)
8764 ??  IL      0:00.00 nfsd: server (nfsd)
22136 ??  IL      0:00.00 nfsd: server (nfsd)
21516 ??  IL      0:00.00 nfsd: server (nfsd)
4262 ??  Is      0:00.00 /usr/sbin/dhcpd -c /etc/dhcpd.conf -q rl1
6570 ??  Is      0:00.00 inetd
25156 ??  Is      0:00.10 /usr/sbin/sshd
13500 ??  Is      0:00.05 sendmail: accepting connections (sendmail)
29926 ??  I       0:00.00 rarpd -a
10051 ??  Is      0:00.00 rpc.bootparamd
9737 ??  Is      0:00.01 cron
17544 ??  Is      0:00.04 sshd: bkw [priv] (sshd)
12395 ??  I       0:00.08 sshd: bkw@ttyp0 (sshd)
24690 p0  Is      0:00.02 -bash (bash)
3966 p0  I       0:00.05 bash
10951 p0  R+      0:00.00 ps -ax
23626 C0- I       0:00.00 dhclient: rl0 [priv] (dhclient)
31984 C0  Is+     0:00.00 /usr/libexec/getty Pc ttyC0
9989 C1  Is+     0:00.00 /usr/libexec/getty Pc ttyC1
959 C2  Is+     0:00.00 /usr/libexec/getty Pc ttyC2
16004 C3  Is+     0:00.00 /usr/libexec/getty Pc ttyC3
22085 C5  Is+     0:00.00 /usr/libexec/getty Pc ttyC5
```

Now to sync your account db copy your servers `/etc/master.passwd` to the clients `/etc` and type.

```
#pwd_mkdb -p -d /home/term-1/root/etc /home/term-
1/root/etc/master.passwd
```

If everything looks fine so far your client should be ready to boot. Make sure you've a NIC that supports PXEboot.

- **Securing the clients**

To avoid the risk that the clients can make any damage to each other we're going to disable sshd which is enabled by default. Disabling root account on the clients is also a good idea. Start by editing the term-1 `rc.conf` which is in the `/home/term-1/root/etc/rc.conf` and replace "`sshd_flags=YES`" with "`sshd_flags=NO`". To disable the root account boot the diskless client term-1 and then run `vipw` (as root of course) and then replace the shell path on the first line that should be the root account to `/sbin/nologin` (man nologin for more info). To keep date synced on all the clients is very important. This can be done by running ntpd on the server.
(note that the first time when you boot term-1 the root password will be the same as on the server since we have synced the password db)

- **Client Internet access using NAT with pf**

First we've to configure the server to allow packet forwarding and packet filtering with pf. Our external interface on the server is rl0 and the internal rl1 (please see figure in appendix).

Start by editing `/etc/sysctl.conf` on the server and replace:
`net.inet.ip.forwarding=0`
with
`net.inet.ip.forwarding=1`
(this allows our server to forward packets)

Edit `/etc/rc.conf`
replace
`pf=NO`
with
`pf=YES`
(it does what the comment describes, # Packet filter / NAT)

Now it's time to configure `/etc/pf.conf`, but please read 'man pf.conf' first. Then wipe everything in the file and make it look something like:

```
bash-3.00# cat /etc/pf.conf
ext_if="rl0" # External Interface
int_if="rl1" # Internal Interface
subnet="192.168.10.0/24"
set loginterface $int_if
scrub in all
#Default NAT from my subnet
nat on $ext_if from $subnet to any -> $ext_if
# default deny
block in  from any to $ext_if
block out from $ext_if to any
pass out on $ext_if inet proto tcp all flags S/SA keep
state
pass out on $ext_if inet proto udp all keep state
pass out on $ext_if inet proto icmp all keep state
pass in on $ext_if inet proto tcp from any to $ext_if
port 22 keep state
pass in on $ext_if inet proto tcp from any to $ext_if
port 53 keep state
pass in on $ext_if inet proto udp from any to $ext_if
port 53 keep state
```

(We set the NAT direction to correct iface, and block everything on the external iface by default. The last 'pass' are to allow everything out (which can be a security risk in production environment) and we also allow a few ports in, hint: read `/etc/services` if you're unsure what services use port 22 and 53)

Client setup.

```
#echo "192.168.10.1" > /home/term-1/root/etc/mygate
```

(Here we set the gateway for our client which of course is the diskless-server)
Edit `/home/term-1/root/etc/resolv.conf` and set your nameserver (it doesn't necessarily have to be the servers IP, just use your ISPs).
example:

```
bash-3.00# cat /home/term-1/root/etc/resolv.conf
search
nameserver 192.168.1.1
```

Now both server and client should be ready, reboot your server to set all rules and then just boot term-1. Login and use 'traceroute' to make sure you've access out.

- **How to run X on the clients**

We will try to give Gnome a go, please read
`http://www.openbsdsupport.org/gnome-GDM.html`.

Edit `/home/term-1/root/etc/sysctl.conf` and set "`machdep.allowaperture=2`". Reboot your client and configure X "Xorg -configure" as root, remember that this has to be done from the client itself. As option you can edit `/home/term-1/root/etc/X11/xorg.conf` by hand.

When done, try just "`startx`" from the client and see if anything happens. If that worked it's time to prepare gdm, just do as the link above described if you want gdm immediately instead of console login first.
(Keep in mind that running Gnome over the network can be slow sometimes. Make sure you have a very quick hard drive on the server, at least 100Mbit network and a lot of RAM on the server. If you want something more lightweight, try window managers like 'Fluxbox' and 'Xfce'. They do both exist as packages on mirrors. Before starting Gnome run 'top' from term-1 and make sure you've the free swap space, or just run 'mount'. If your swap isn't mounted it might fill your clients ram and then freeze the system).

- **UPDATE/CHANGES**

I will put some document changes directly here, please read these.

- Do not place the diskless root directories in /home. Instead use a separate fs mounted at `/export/diskless` . The best would of course be if you had a separate very quick disk mounted as `/export/diskless`.
  Then place every terminal directory in that path, ie `/export/diskless/term-1`
  You can move tftpboot to `/export` also. All these changes are made because `/home` isn't the proper path to keep homes for diskless clients. Especially if you later on want to use shared accounts it can cause problems with quotas.
  If you do any of the steps mentioned in this update, don't forget to update paths in your config files.
- Aviod running Gnome on diskless clients, it can and probably will give you lots of headache

- **Useful links**

`http://marc.theaimsgroup.com/?l=openbsd-misc&m=112518083924398&w=2`

Next versions of this documentation might include.

- Setup more than a single client environment
- Distributed useraccounts

Thanks to:
The *OpenBSD* team for writing useful manuals and for the best OS I ever used.
The *Openoffice* team.
NicM on freenode #openbsd

- **Appendix**

network design