FreeBSD NUT Server on Raspberry Pi

Discussion

I originally planned my Pi to use Raspian as the OS, even though I detest SystemD. However, I had a serious bug with NUT v2.8.0 and my CyberPower RMCARD206. I needed to either downgrade to NUT v2.7.4, or upgrade to v2.8.2, or build it myself.

I prefer FreeBSD for most servers, and went looking for something I could use on my Pi. I found https://wiki.freebsd.org/arm/Raspberry%20Pi which was exactly what I was looking for. It had the latest version of FreeBSD (v14.2 as of this article) in a Pi image.

Preparation

1. Download the Pi image from the BSD site. For my Raspberry Pi 3B, I downloaded the **FreeBSD-14.2-RELEASE-arm64-aarch64-RPI.img.xz**
2. Decompress it

```
xz -d FreeBSD-14.2-RELEASE-arm64-aarch64-RPI.img.xz
```

3. Burn it onto the microSD (assume it is sdb, if not, change that)

```
sudo dd if=FreeBSD-14.2-RELEASE-arm64-aarch64-RPI.img of=/dev/sdb bs=1M status=progress
```

4. Insert into your Pi and boot up.

Just leave it alone, booting. It will resize the file system, do some other stuff, then eventually get to the login screen. You can do this headless as the image has ssh turned on and a user already defined.

Initial Setup

The system has two pre-defined users. **root**, with a password of **root**, and be accessed from a keyboard/monitor, or **freebsd**, with a password of **freebsd**, and be accessed via ssh. I chose to access via ssh, then did an **su -** to become root.

First thing is to create a new user. The **adduser** command is a nice, curses based menuing option that allows you to do this quickly. When it asks for **Additional Groups**, type **wheel** to allow sudo in our next step. While you're at it, change root's password. I'd recommend logging out and logging back in as the new user.

As root, issue the command **visudo** to edit the sudo users. If you're not familiar with vi, I'll explain the steps

1. **/** - puts you in Find mode

2. **wheel** - search for *wheel*. Press the Enter key
3. Using your arrow keys move down to the entry which does **not** have NOPASSWD: in it
4. Move your cursor (using arrow keys) to the % sign in front of the word wheel
5. press **d**, then the **back arrow** to delete the blank space left of the % sign
6. Do it again to delete the pound sign
7. **:** - put you in command mode
8. **wq** - does a write and quit (save and exit the file)
9. **Note**: if you make a mistake, press the ESC key, then :, then q!. The q! says “exit without saving”
10. Exit back to the login screen/exit ssh session, log back in, and use the command

    ```
    sudo su -
    ```

    to become root (using your password, not roots)

At this point, you have a user who can do everything necessary, so issue the command

```
pw userdel freebsd -r
```

to remove the freebsd user.

Setup

I'm doing the rest in just a line of commands. I have shell comments for explanations of what is going on, so you can simply copy/paste

```
# Fix some stuff in the base configuration
# rename your server
sysrc hostname='yourservername'
# use ntp to set your time and keep it current
sysrc ntpd_enable='YES'
sysrc ntpd_sync_on_start='YES'
# set your network card to do dhcp (overrides the default)
sysrc ifconfig_DEFAULT='DHCP'

# set your timezone with a curses based menu
tzsetup

# set your system clock; the Pi has no internal clock
ntpdate time.nist.gov

# install NUT. I also install the editor joe as I like it
# I tend to use the package manager instead of compiling everything fro the
source tree.
pkg install nut joe
```

Configure NUT Server

Now, configure the nut server. When you have the server configured, do the following.

```
service nut onestart # test if it works
# verify you can access the UPS's (in this case, ups1 on localhost)
# suggest doing for each UPS you defined
upsc ups1 ups.status # should return OL, for Online
# turn on nut as a service
sysrc nut_enable="YES"
```

Configure NUT client

Configure the nut client (aka nut monitor), then

```
service nut_upsmon onestart # test if it works
# turn on nut_upsmon as a service
sysrc nut_upsmon_enable="YES"
```

You should now have a working NUT server and local monitor.

Tuning

The main tuning is to minimize disk writes. MicroSD's have a much more limited number of writes it can have before corrupting, but with a few simple steps, I've had Pi based cameras last up to 5 years.

atime

Look in /etc/fstab and ensure the primary disk has a noatime flag. Mine did not for the root fs, so I added it. It already had it on the EFI partition. Mine looks like this, after editing.

fstab

```
# Custom /etc/fstab for FreeBSD embedded images
/dev/ufs/rootfs         /               ufs     rw,noatime
1       1
/dev/msdosfs/EFI                /boot/efi       msdosfs rw,noatime
0       0
/dev/label/growfs_swap  none            swap    sw              0
0
```

Use tmpfs file system

/tmp, /var/log and /var/tmp can all be placed in tmpfs file systems (aka RAM disks). It means you lose your logs when you reboot, and it means some long term temp files (like vi recovery files) will not be remembered if you're booting. But, for this system, that is not necessary, and this will greatly cut down on disk writes.

```
# add /tmp and /var/log as a tmpfs file system (memory) into fstab
echo 'tmpfs                    /tmp            tmpfs   rw,mode=1777    0
0' >> /etc/fstab
echo 'tmpfs                    /var/log        tmpfs
rw,mode=755,uid=0,gid=0 0       0' >> /etc/fstab

# point /var/tmp to /tmp (clean out /var/tmp first)
rm -fR /var/tmp && ln -s /tmp /var/tmp && chmod -h 777 /var/tmp

# clean out /var/log and turn it into a tmpfs also
# shutdown syslog
service syslogd stop
# remove everything in /var/log
find /var/log -mindepth 1 -delete
# mount a tmpfs file system on /var/log
mount /var/log
# start syslog back up
service syslogd start

# clean up and remount /tmp as a tmpfs
find /tmp -mindepth 1 -delete
mount /tmp
```

Ok, you've messed with a bunch of stuff, so **reboot** your system and see what it looks like now.

Links

- https://wiki.freebsd.org/TMPFS
- https://forums.freebsd.org/threads/minimizing-writes.11590/
- https://forums.freebsd.org/threads/tmpfs-vs-md.80522/
- https://github.com/networkupstools/nut/issues/1275
- https://forums.freebsd.org/threads/fetch-fails-with-ssl-certificate-error-on-https-download-freebsd-org.78082/

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/software/nut/pifreebsdserver

Last update: **2025/04/23 07:50**