FIXMYNIX

How to setup OpenWrt extroot, rootfs on USB storage

Last Updated on March 29, 2017 by Arnab Satapathi — 77 Comments

Here I'm going to share my experience with *OpenWrt extroot* on a TP-Link MR-3220 v2.4 router.

Though I did this on a MR-3220, this guide also applies to other routers like TP-Link MR703N, MR3020, D-Link DIR-505 etc etc. Just make sure you have downloaded the correct firmware for your router.

Probably you know all about it's hardware specifications, here is a short recap again

- AR9331-AL3A 400MHz SoC
- 32 MB DDR RAM chip
- 4 MB SPI NOR flash memory
- One 10/100M ethernet WAN port
- Four 10/100M ethernet switches
- 150Mbps b/g/n Wi-Fi
- One USB 2.0 port

Little about MR 3220 v2.4 hardware quality, I teared it apart within 1 hour of receiving, the hardware quality is average with average quality components.

Enough hardware introduction, lets start the main topic, installing OpneWrt and configuring extroot on a USB pendrive. OpwnWrt firmware installation is very easy through the web interface and **openwrt extroot** part is same for almost any other router too.

Which OpenWrt firmware version should I choose ?

This is one of the most important question, choosing the correct OpenWrt version is absolutely crucial. This router works perfectly with the latest OpenWrt **trunk** (Designated Driver), stable **15.05**, **14.07** or **12.09** .

But OpenWrt is bloating day by day while limited NOR flash storage is not, the stable firmware is also bloated with LUCI web interface. The free JFFS2 space after installing OpenWrt is decreasing too. So you can't install more than a few package with this limited space, that's when putting extra files on a external drive is necessary.

After about 12-14 hour experiment with various OpenWrt versions, I figured out that **only OpenWrt 14.07** (Barrier Breaker) or earlier versions have the right amount of free JFFS2 space to install required bare minimal minimal packages to setup an **extroot**.

Note: There's another little difficult way to run latest OpenWrt on your router, then you've to **build OpenWrt custom firmware**.

Which filesystem should I use for extroot ?

Another very important question, the official OpenWrt extroot guide says that you could any UNIX-like filesystem of your choice, forget about FAT32 or NTFS. But the reality is extroot is **only and only** possible with **ext4/ext3** or **ext2** filesystem. I tested with almost every slim (I mean which equips less NOR flash space) filesystem

1. **minix**, not supported
2. **HFS**, not supported
3. **HFS+**, not supported
4. **JFS**, not supported
5. **XFS**, not sopported
6. **reiserfs**, not supported

There are other UNIX-like filesystems like BTRFS, LVM2 PV , UFS etc. but none of them will fit in the 288 KB free JFFS2 space, so I didn't tested them.

So the final choice is for a router with 4MB flash, choose only **OpenWrt 14.07** or earlier and **ext4** as extroot filesystem.

Request to OpenWrt pro users, if you had ever setup chroot with any other filesystem except ext4/ext3/ext2, please mention.

Installing OpenWrt on TP-Link MR3220

Assuming you have a PC running a Linux distribution, this is required for flashing the firmware and

Download the OpenWrt 14.07 firmware **from here** .

```
wget -c https://downloads.openwrt.org/barrier_breaker/14.07/ar71xx/generic/openwrt-ar71x
```

WARNING: Do it at your own risk, event with great precautions there is a chance of bricking the router. If possible connect the router to an UPS before flashing it.

1. Power on the router and connect a PC through a ethernet straight cable (patch cord) to one of four the LAN ports(the yellow ones).
2. Open up a browser and go to this address; **192.168.0.1** to access the web interface and login. **Username:** admin **Password:** admin .
3. Go to the **System Tools** > **Firmware Upgrade** choose the previously downloaded OpenWrt firmware with the **Browse** button and Hit the **Upgrade** button.
4. Wait for few minutes to let the flashing complete and automatic reboot.

Installation is complete now your PC will automatically receive a different IP via DHCP.

Initial OpenWrt setup

For first time we have to use telnet to login into OpenWrt,

```
telnet 192.168.1.1
```

or if you don't have telnet try this

```
busybox telnet 192.168.1.1
```

after login, change the password to enable ssh

```
passwd root
```

now exit from the telnet session with **exit** command and re-login with SSH,

```
ssh root@192.168.1.1
```

Installing necessary packages

At this step the router must be connected to the internet, you could use your existing DSL/cable internet for this purpose or share an existing connection from your PC. I personally did it by sharing my 3G connection with **Network-Manager** from my laptop.

Make sure that the router could connect to the internet

```
ping -c 4 8.8.8.8
```

Update `opkg` package repository

```
opkg update
```

Install necessary packages to support USB storage device

```
opkg install kmod-usb-storage kmod-scsi-core block-mount
```

Install ext4 filesystem kernel module

```
opkg install kmod-fs-ext4
```

That's all, check free flash space with `df -h` , do not install anything more ! if you try to do so, any packages larger than 10KB will return no space left on device error.

```
root@OpenWrt:~# df
Filesystem           1K-blocks      Used Available Use% Mounted
rootfs                     640       564        76  88% /
/dev/root                 2304      2304         0 100% /rom
tmpfs                    14428        60     14368   0% /tmp
/dev/mtdblock3             640       564        76  88% /overlay
overlayfs:/overlay         640       564        76  88% /
tmpfs                      512         0       512   0% /dev
```

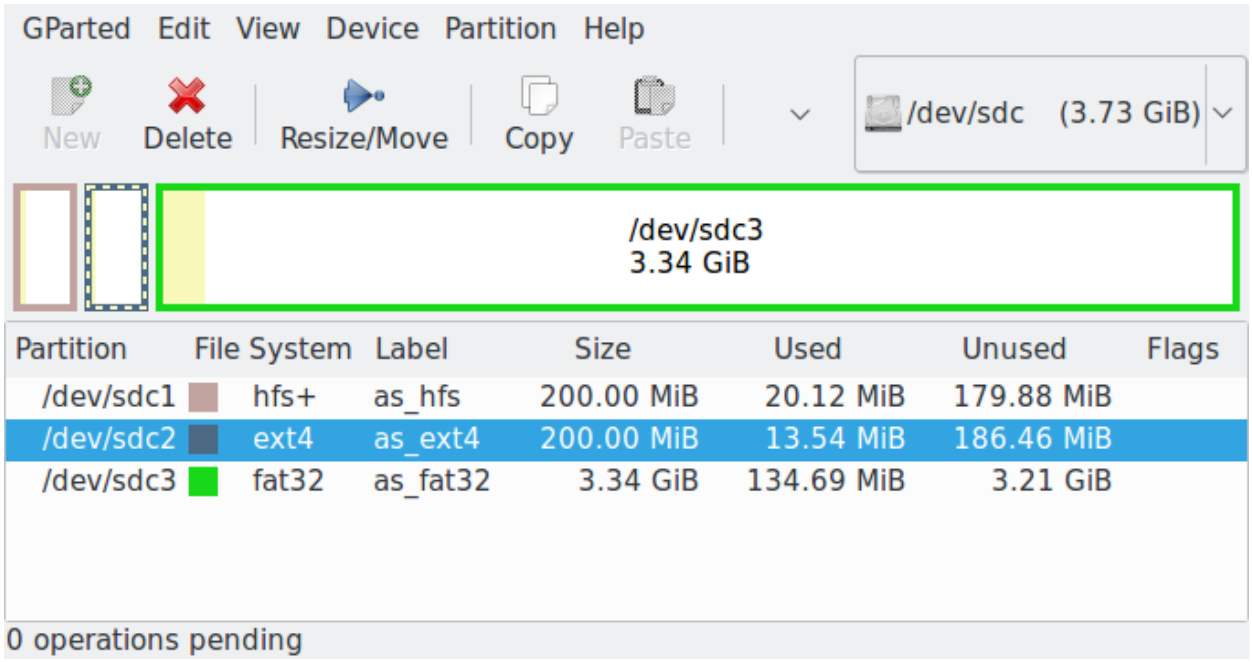
Now load all necessary kernel modules with modprobe command

```
modprobe sd_mod
modprobe usb-storage
modprobe ext4
```

Or simply reboot the device with `reboot` command.

Partitioning the USB drive

Lets partition the USB drive, another question, how much space should I use for extroot partition ? Just 100-200 MB is more than enough, I allocated a 200 MB ext4 partition for extroot, you could use more if you wish. I used Gparted for partitioning a 4 GB USB drive.



Now safely remove the USB drive and plug it in the routers USB port.

Configuring the OpenWrt extroot

After plugging the USB drive, it should show up as a storage device under the **/dev** directory as /dev/sda1 , /dev/sda2 etc. Simply go through the commands bellow

Mount the USB drive, change /dev/sda2 according to yours

```
mount /dev/sda2 /mnt
```

Copy data from /overlay partition to the USB drive

```
tar -C /overlay/ -c . -f - | tar -C /mnt/ -xf -
```

Un-mount the USB drive

```
sync && umount /dev/sda2
```

Configure `/etc/config/fstab` to mount the USB drive as `/overlay` partition

```
block detect > /etc/config/fstab
```

Now edit the `/etc/config/fstab` with vi to mount the **etxt4** partition, `/dev/sda2` at `/overlay` partition at startup,

```
vi /etc/config/fstab
```

have a look on a sample fstab configuration,

```
config 'global'
        option  anon_swap       '0'
        option  anon_mount      '0'
        option  auto_swap       '1'
        option  auto_mount      '1'
        option  delay_root      '5'
        option  check_fs        '0'

config 'mount'
        option  target  '/overlay'
        option  uuid    '7669178c-3f77-4fb1-b421-6ec6f61be672'
        option  enabled '1'
```

Set the target to **/overlay**, and change the **option enabled** line from **0** to **1** . Now enable the **fstab** service at startup

```
/etc/init.d/fstab enable
```

and check it

```
readlink -f /etc/rc.d/*fstab
```

reboot the router with the reboot command

```
reboot
```

Check if the extroot is working or not

After the reboot is complete login to the router with ssh again and check the free disk space with `df` and `mount` command.

```
df -h
```

```
mount
```

The `df` commnd should return a lot of free space available at rootfs, that means extroot is working fine.

```
root@OpenWrt:~# df
Filesystem           1K-blocks      Used Available Use% Mounted
rootfs                  194241      2918    176987   2% /
/dev/root                 2304      2304         0 100% /rom
tmpfs                    14428        60     14368   0% /tmp
/dev/sda2               194241      2918    176987   2% /overlay
overlayfs:/overlay      194241      2918    176987   2% /
tmpfs                      512         0       512   0% /dev
```

Now check it again by installing some big packages, example >

```
opkg install e2fsprogs
```

I hope this simple beginner friendly tutorial will help you to start your experiment with OpenWrt. If you have any question or suggestion, just leave a comment and also don't forget to share this with your friends 🙂 .

Comments

Jerry says

May 8, 2017 at 5:02 pm

Hi Arnab,

Thanks for your clearly written and informative tutorial.
Worked great for me on an older TP-Link TL-WDR4300.

Nice work!

Jerry

Reply

‹ Older Comments

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

POST COMMENT

Notify me of follow-up comments by email.

Notify me of new posts by email.

Search this site

Search this website ...

On Social Media

Facebook Twitter Google+

Fresh Posts

Setup USB tethering to OpenWrt router

Use dracut in Ubuntu to generate minimal initramfs

Easiest way to compile Linux kernel from source

5 Best LXQt themes for a better desktop experience

4 ways to disable NMI watchdog in Linux

Ubuntu nVIDIA Optimus setup – Install bumblebee and nvidia drivers

How to find Linux CPU usage with CoreFreq

Qarma – A lightweight zenity alternative in Qt

Recent Comments

ksieben on Configure NVIDIA Optimus in Debian, Kali Linux with bumblebee

Arnab Satapathi on GPU benchmarking and stress testing in Linux

paul crypto GPU on GPU benchmarking and stress testing in Linux

Arnab Satapathi on Run shell scripts from udev rules

osukaruda on How to install and configure LXQt Desktop in Debian

More Awesome Stuff