

Extroot配置

如何使用存储设备（usb或sata或sdcard或其他）扩展您的LEDE设备的根文件系统空间，以便自由安装所需的所有软件包。

一些背景信息

在大多数支持的设备中，LEDE固件将内部存储器分为两个分区，即高度压缩的只读分区，它是实际的根文件系统（/），第二个分区是可写的，称为“覆盖”（/overlay）。

覆盖分区与根文件系统使用linux内核

(<https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/filesystems/overlayfs.txt>)
的overlayfs功能

(<https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/filesystems/overlayfs.txt>)
合并，向应用程序

(<https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/filesystems/overlayfs.txt>)
显示单个“整体”读写文件系统。

这样，LEDE甚至适合微型内部存储（低至4 MiB），但仍允许写入设置并在可写分区中安装一些软件包，而无需更改所有使用的Linux程序。

Extroot通过在外部存储设备中设置另一个覆盖分区来工作，并且在引导期间，新的覆盖分区将被安装在内部存储的覆盖分区上。这种方法允许在外部存储设备被删除的情况下容易回退，因为您的LEDE设备仍然具有自己的覆盖分区，因此将从那里加载所有配置。

这意味着它将在您设置extroot之前的行为完全相同。

具有4 MiB内部空间的设备的第一阶段

在默认的LEDE固件映像中，没有任何工具可以进行外接，因为构建系统目前只能制作准系统映像。要使用这些设备的唯一方法是使用Image Builder重新构建具有正确软件包的固件映像。

图像生成器只能运行在64位Linux操作系统中，所以如果您手头上没有linux系统，请查看教程，在VirtualBox中安装Ubuntu 64bit。

然后进入相同的下载页面，您可以在其中下载设备的固件，向下滚动，直到找到以“**lede-imagebuilder**”开头的文件。

将其下载并解压缩到Linux系统中的文件夹中。

打开该文件夹中的终端，并写：

做信息

这将在屏幕上写入该Image Builder支持的设备的所有可能的配置文件名称，因此我们可以为正确的设备构建映像。每个条目将如下所示：

TL-wr1043nd-V1:

TP-LINK TL-WR1043N / ND v1

包装: kmod-usb-core kmod-usb2 kmod-ledtrig-usbdev

第一行是配置文件名称，第二行是设备的完整描述性名称，第三行是该设备的默认包的列表，并应列出有关USB或Sata或任何其他存储设备的一些包。

在我的情况下，我有一个TP-LINK TL-WR1043N / ND v1，所以我的设备的配置文件名称是**tl-wr1043nd-v1** 现在你需要写命令来开始构建映像（注意**PROFILE** =是我的设备的个人资料名称，请使用您的个人资料名称）：

```
make image PROFILE = tl-wr1043nd-v1 PACKAGES =“block-mount kmod-fs-f2fs kmod-usb-storage mkf2fs f2fsck kmod-usb-ohci kmod-usb-uhci”
```

这将构建一个能够使用f2fs文件系统创建和检查/修复分区的固件映像。

F2fs设计用于闪存驱动器（USB闪存驱动器或SSD），它也适用于硬盘驱动器，但不会那么好。

令人遗憾的是，**e2fsprogs**与**ext4**文件系统（更适合硬盘驱动器）的工具太大，无法适应4个MiB设备。如果有人可以从其中拆分**mke2fs**和文件系统检查工具，我们可以调整本教程以使用**ext4**。

警告：来自**f2fstools 1.7.0**的**fsck.f2fs**在**ar71xx**（**mips32 big endian**）平台上损坏。**fsck.f2fs**会破坏有效的f2fs分区。LEDE 17.01.1（4月中旬发布）中的当前**f2fstools**版本是**1.8.0**版，没有此问题，检查版本并立即升级。

然后打开Image Builder文件夹中的文件夹**bin**，然后打开目标文件夹，然后打开目录文件夹，然后打开一个名为**generic**的文件夹，然后打开一个名为**generic**的文件夹，图片。选择正确的图像（工厂或sysupgrade）并进行安装。

具有8 MiB或更多内部空间的设备的第一阶段

这些设备应该有足够的空间来安装我们需要的软件包。删除所有安装的软件包以添加功能，因为它们现在只浪费空间。在你做了**extroot**之后，你将拥有所需的空间。

从命令行界面写（单行）：

```
opkg update && opkg install block-mount kmod-fs-f2fs kmod-usb-storage mkf2fs f2fsck kmod-usb-ohci kmod-usb-uhci fdisk
```

在TL-1043ND v2上，由于F2FS **extroot**发生错误，路由器开始冻结。从**f2fstools**运行**fsck.f2fs 1.7.0 on ar71xx**会破坏有效的f2fs分区。对于具有8MiB或更高闪存的**ar71xx**，请使用**ext4**作为**extroot**。

```
opkg update && opkg install block-mount kmod-fs-ext4 kmod-usb-storage
e2fsprogs kmod-usb-ohci kmod-usb-uhci fdisk
```

共同的第二阶段

将ssh连接到设备。

看看你有什么分区：

```
root @ lede: /# block信息
/ dev / mtdblock2: UUID =“9fd43c61-c3f2c38f-13440ce7-53f0d42d”VERSION =“4.0”MOUNT
=“/ rom”TYPE =“squashfs”
/ dev / mtdblock3: MOUNT =“/ overlay”TYPE =“jffs2”
/ dev / sda1: UUID =“fdacc9f1-0e0e-45ab-acee-9cb9cc8d7d49”VERSION =“1.4”TYPE =“f2f
s”
```

这里我们看到**mtdblock**设备（内部闪存中的分区）以及**usb**闪存驱动器上的**/ dev / sda1**上的一个分区（在该示例中它已格式化为**f2fs**）

我们现在首先将外部驱动器格式化为**f2fs**或**ext4**。

对于**f2fs**：

```
root @ lede: /# mkfs.f2fs / dev / sda1
```

对于**ext4**：

```
root @ lede: /# mkfs.ext4 / dev / sda1
```

然后我们将当前叠加层的内容传输到外部驱动器中

```
root @ lede: /# mount / dev / sda1 / mnt; tar -C / overlay -cvf - . | tar -C / mnt
-xf-; umount / mnt
```

自动化fstab生成

现在我们自动创建**fstab uci**子系统，并使用正确的配置填充**/ dev / sda1**作为新的叠加层

```
root @ lede: /# block detect> / etc / config / fstab; \
sed -is / option $'\ t'enabled $'\ t'\ '0 \'/ option $'\ t'enabled $'\ t'\ '1 \'/
/ etc / config / fstab; \
sed -is#/ mnt / sda1#/ overlay#/ etc / config / fstab; \
cat / etc / config / fstab;
```

如果您有交换分区，它也将被自动识别和添加。

手动fstab生成

如果要手动执行，该命令将自动创建一个通用的fstab uci子系统

```
root @ lede: /#block detect> / etc / config / fstab
```

现在打开它

```
root @ lede: /#vi / etc / config / fstab
```

这是一个配置为自动挂载/覆盖/数据和交换分区的fstab的示例。

```
config'global'  
  选项anon_swap'0'  
  选项anon_mount'0'  
  选项auto_swap'1'  
  选项auto_mount'1'  
  选项delay_root'5'  
  选项check_fs'0'  
  
config'mount'  
  选项目标'/ overlay'  
  选项uuid'c91232a0-c50a-4eae-adb9-14b4d3ce3de1'  
  选项fstype'ext4'  
  选项启用'1'  
  
配置“交换”  
  选项uuid'08b4f0a3-f7ab-4ee1-bde9-55fc2481f355'  
  选项启用'1'  
  
config'mount'  
  选项目标'/ data'  
  选项uuid'c1068d91-863b-42e2-bcb2-b35a241b0fe2'  
  选项启用'1'
```

如你所见，大多数选择是自我解释的。

检查覆盖是否可以安装

让我们尝试手动安装以查看一切是否正常

```
root @ lede: /#mount / dev / sda1 / overlay
```

现在我们看到安装点大小：

```
root @ lede: /# df
```

这是一个示例输出：

```
文件系统1K块使用可用使用%安装在  
rootfs 896 244 652 27%/  
/ dev / root 2048 2048 0 100%/ rom  
tmpfs 14708 64 14644 0%/ tmp  
/ dev / mtblock6 7759872 477328 7221104 6%/ overlay  
overlayfs: / overlay 896 244 652 27%/  
tmpfs 512 0 512 0%/ dev  
/ dev / sda1 7759872 477328 7221104 6%/覆盖
```

注意，只有 /overlay 增长但不是 /

最后步骤


1. 重新启动路由器
2. 验证分区是否正确安装：
 - 通过LuCI
 - 系统 - 软件应显示覆盖分区的可用空间
 - 系统 - 安装点应显示USB分区 overlay
 - 通过CLI
 - mount 应该显示USB分区 /overlay

```
root @ lede: ~# mount  
/ dev / root on / rom类型squashfs (ro, relatime)  
proc on / proc类型proc (rw, noatime)  
sysfs on / sys type sysfs (rw, noatime)  
tmpfs on / tmp type tmpfs (rw, nosuid, nodev, noatime)  
/ dev / mtblock6 on / overlay类型jffs2 (rw, noatime)  
overlayfs: / overlay on / type overlay (rw, noatime, lowerdir = /, upperdir = / over  
lay / upper, workdir = / overlay / work)  
tmpfs on / dev type tmpfs (rw, relatime, size = 512k, mode = 755)  
devpts on / dev / pts type devpts (rw, relatime, mode = 600)  
/ dev / sda1 on / overlay类型ext4 (rw, relatime, data = ordered)  
/ dev / sda3 on / data type ext4 (rw, relatime, data = ordered)  
debugfs on / sys / kernel / debug type debugfs (rw, noatime)
```

- df 应该显示您 /overlay 和分区上的可用空间 / ，所有安装的存储器 /overlay 和 / (rootfs 在第一位) 应该看起来相同增加的容量：

```
root @ lede: ~# df
文件系统1K块使用可用使用%安装在
rootfs 7759872 477328 7221104 6%/
/ dev / root 2048 2048 0 100%/ rom
tmpfs 14708 64 14644 0%/ tmp
/ dev / mtdblock6 7759872 477328 7221104 6%/ overlay
overlayfs: / overlay 7759872 477328 7221104 6%/
tmpfs 512 0 512 0%/ dev
/ dev / sda1 7759872 477328 7221104 6%/覆盖
/ dev / sda3 242846048 163864 230323224 0%/ data
```

故障排除

-  **Fix Me!**: 可能已过时添加选项， `force_space` 在 `/etc/opkg.conf` 允许安装包比你大的 `/rom` 分区的可用空间:

```
echo选项force_space >> /etc/opkg.conf
```

- 不要使用 `vfat` (FAT / FAT32); 这是行不通的。如果您使用FAT预格式化的USB驱动器，则无法在不重新格式化的情况下将其用于 `extroot`。使用例如 `ext4` (安装 `e2fsprogs`, 然后 `mkfs.ext4 /dev/sda1` 按照示例格式化FAT格式的USB驱动器)。
- 如果包含您的 `extroot` 的分区在引导期间未挂载，但是您可以从 `shell` 挂载它而不会出现问题，您应该尝试增加 `config 'global' / option delay_root`。在我的系统上，我不得不将其设置为15秒才能获得 `extroot` 的工作。另一个暗示，这是罪魁祸首是在启动后安装了一个工作的交换或其他分区，但不是您的 `extroot`。
- 考虑并尝试另一种可能性是在包括 `/etc/rc.local` 命令: `export PREINIT = 1;` `mount_root`, 如在14946 (<https://dev.openwrt.org/ticket/14946>)票，这在康全电讯AR-5387un运行混沌较平静r44266的情况下，一直是让我实现 `extroot` 的唯一的事情。
- 如果成功的 `extroot` 安装后，交换未启用，并且其他分区未挂载，请检查您 `/etc/config/fstab` 的重叠分区。如果您 `/etc/config/fstab` 在覆盖分区中只包含一个 `global` 部分，则可能需要添加以下示例:

```
配置“交换”
    选项uuid'08b4f0a3-f7ab-4ee1-bde9-55fc2481f355'
    选项启用'1'

config'mount'
    选项目标'/ data'
    选项uuid'c1068d91-863b-42e2-bcb2-b35a241b0fe2'
    选项启用'1'
```

- 如果将 `extroot` 放在诸如 `mmc` 卡的非USB设备上，则所有模块都需要接收设备，应该在 `/etc/modules-boot.d` 中。例如在 `mt7688 / mt7628` 设备上使用 `sdhci` 卡 `/etc/modules-boot.d/mmc` 需要添加两行:

```
mmc_core
mmc_block
sdhci
mtk_sd
```

笔记

外插到USB加密狗插槽中的卡

将 `usb-modeswitch` 工具包含在图像中是一个好主意。

有一个警告：如果 `/overlay` 指向位于加密狗插槽中的存储卡 - 否则工作的 `pivot overlay` 设置将在操作系统 (**Operating System**) 启动的后期阶段中断。这是因为 `usb-modeswitch`（禁用CDROM并启用调制解调器）也会间歇性地影响加密狗中的读卡器，从而伤害文件系统。

为了避免这种情况，您需要一个可以预先配置为在上电时启用其“调制解调器”或“网络适配器（以及读卡器）”的加密狗，而无需 `usb-modeswitch` 在路由器上进行此操作。

将您的加密狗插入桌面，并使用终端发送必要的AT命令。

检查您的加密狗的初始配置：

```
在^ setport?  
^ SETPORT: A1, A2; 1,3,2, A1, A2  
好
```

上述报告的含义可以用以下命令来理解：

```
在^ setport =?  
^ SETPORT: A1: CDROM  
^ SETPORT: A2: SD  
^ SETPORT: A: 蓝牙  
^ SETPORT: B: 指纹打印  
^ SETPORT: D: 彩信  
^ SETPORT: E: PC VOICE  
^ SETPORT: 1: MODEM  
^ SETPORT: 2: PCUI  
^ SETPORT: 3: DIAG  
^ SETPORT: 4: PCSC  
^ SETPORT: 5: GPS  
^ SETPORT: 6: GPS控制  
^ SETPORT: 16: NCM  
好
```

所以，在上面的例子中，我们有一个带有CDROM和读卡器的加密狗，可以在第一个配置（字符左侧；）和调制解调器，控制和诊断接口以及读卡器在其他配置中可用。在这些配置之间 `usb-modeswitch` 切换路由器上的加密狗。

您的目标是在启用读卡器（以上）时禁用CDROM并启用调制解调器（1 上述）或网络适配器（上述）。注意：不要禁用PCUI（以上） - 这将锁定您的加密狗！一些加密狗接受“禁用所有”操作数（如下）。根据上面的加密狗答案，将您的加密狗上所需的所有功能列表默认放置在字符的右侧。 16 A2

2

FF

;

```
在^ setport = "FF; 1,2,3, A2"  
好
```

```
在^复位  
好
```

```
在^ setport?  
^ SETPORT: ; 1,2,3, A2  
好
```

该序列已禁用CDROM，并使调制解调器，控制和诊断接口和读卡器默认可用 - 无需任何 `usb-modeswitch` 交互。因此，只有一个配置现在在加密狗 - 看到的 ; 字符，现在没有什么左边它。

已知的Dongles支持预配置

- 华为E3131s-2 f / w v21.158.47.00.1094

远程文件系统

- 论坛: `Fstab`在启动时或通过CLI挂载cifs (<https://forum.openwrt.org/viewtopic.php?id=32812>)

系统升级

本节适用于LEDE中继线，但不适用于LEDE版本，因为版本中的内核相关软件包（及其需要的软件包）将仅接收修补程序和安全修补程序。

由extroot维护者推荐

我建议你不要尝试使用升级 `opkg upgrade` 。你可能最终会出现一种不一致的状态和软布路由器：

- 主要原因是uClibc ABI（应用程序二进制接口）不稳定，从修订版本更改为修订版本，因此一个版本的uClibc的二进制文件可能与其他版本不兼容。
- 可能出现的另一个问题是，如果您尝试升级内核软件包，然后重新启动，然后重新启动，但您的操作以任何方式中断，那么您将有一个内核和模块不匹配，可能是一个砖。
- 最后，如果您升级所有软件包，但内核和内核模块，某些软件包 `iptables` 将被破坏。


访问原始根

有时您可能需要访问原始根覆盖层，例如更改您的extroot设置。一个方便的方法是在extroot分区 `/etc/config/fstab` 上进行配置，将原始的根覆盖层安装在另一个目录中，如下所示：

```
配置挂载  
  选项目标/覆盖启动  
  选项设备/ dev / mtblock3  
  选项fstype jffs2  
  选项选项rw, 同步  
  选项启用1  
  选项enabled_fsck 0
```

这假设原始的内部覆盖层是在 `/dev/mtblock3` - 检查你的路由器的页面在这个维基，并查看闪存映射，以确认是什么MTD阻止它是为您。或者运行 `$ cat /proc/mtd` 并搜索名为的分区， `rootfs_data` 以了解您的叠加位置。

如果然后 `/overlay-boot` 在Extroot分区上创建，则该目录将包含原始根覆盖层，用作主根覆盖，直到外接程序启动并运行。所以您可以通过编辑 `/overlay-boot/etc/config/fstab` 来改变你的extroot配置（或暂时禁用它），如果你需要。

 最后修改：2017/04/24 11:34 通过bobafetthotmail

除非另有说明，本维基的内容将根据以下许可证获得许可：CC Attribution-Share Alike 4.0 International (<http://creativecommons.org/licenses/by-sa/4.0/>)