

# 在 **Raspberry Pi** 上运行公共区块链 平台 **Ethereum**: 加密钱包和冗余存 储的完整节点

Phodal Huang

October 24, 2017

## 目录

步骤 1: 材料清单 . . . . .	3
步骤 2: 安装 Archlinux ARM . . . . .	4
在 Linux 系统上, 使用 SD 卡安装 Arch Linux . . . . .	4
在 Windows 系统上使用 SD 卡安装 ArchLinux . . . . .	5
步骤 3: 配置 Archlinux . . . . .	7
步骤 4: 设置 RAID . . . . .	7
步骤 5: RAID /home 和用户设置 . . . . .	8
geth & 启动服务 . . . . .	9
步骤 6: 结论 . . . . .	11

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/install-ethereum-node-on-rasberri-pi-3>

在这个玩法中, 我将向您展示如何使用 **Raspberry Pi 3** 构建一个完整的以太坊 (**Ethereum**) 节点和钱包。为什么? 因为运行一个以太坊 (**Ethereum**) 节点, 我们可以: 贡献给以太坊 (**Ethereum**) 网络; 将我们的钱包存放在私有安全的位置; 并且在实验某种加密算法交易时也可以使用 (您可以从代码生成新地址, 确保同步并存储在私人位置, 进行事务处理等)。

## 步骤 1: 材料清单

首先, 我们需要以下的材料:

- **Raspberry Pi 3**
- **Raspberry Pi 3** 的盒子、电源和散热器套件
- **UPS**
- **2** 个外围的硬盘 **1TB**
- **USB** 外接电源开关
- **32GB** 的 **Micro SD** 卡

简要地解释一下, 为什么我们需要这些东西:

1. **Raspberry Pi 3** 是最受支持的 **ARM** 设备之一。它有 **1 GB RAM**, **4** 个 **USB 2.0** 端口和 **1** 个以太网端口。可以在其上运行 **Linux**; 实际上有很多个可用的 **Linux** 分发版本。
2. 为了让 **Raspberry Pi** 全天候 (**24/7**) 工作, 我们必须对它进行散热处理。它能保证我们的设备寿命长, 上述的套件中我们还发现了一种很好的有机玻璃, 可以保护设备免受外界的破坏。
3. 停电是除温度外, 导致硬件故障的第二个主要原因 (比较好的情况是, 文件系统损坏和随后的数据丢失)。为了防止严重损坏, 需要 **UPS**。此外, **Raspberry Pi** 的功耗并不是那么高, 因此 **UPS** 可以保持一段合理的时间 (嘿, 读者, 你是一个 **Raspberry Pi** 黑客, 你知道如何使用 **GPIO** 来监控墙上插座和触发器。当 **UPS** 处于 **UP** 状态, 且墙壁电源为 **DOWN** 时, 进行安全关机程序? 也许您也知道如何启动电源回来, 并且在没有 **UPS** 的任何信息 (无 **USB** 可用) 的情况下进行所有操作)? 请与我联系!)
4. 区块链 (**blockchain**) 的大小逐日增加。一个简单的 **SD** 卡不足以存储整个区块链 (**blockchain**)。此外, **SD** 卡是一个单一的故障点, 我们应该防止这种故障使我们失去了区块链 (**blockchain**)。最重要的是: 钱包!

5. Raspberry Pi 不能为 USB 端口提供足够的电源。因此，由于几乎每个外部硬盘都不是由外部供电的，所以我们必须使用来 USB 开关提供足够的电力。
6. Raspberry Pi 没有带存储介绍，安装操作系统需要 SD 卡。

好了，现在让我们开始吧。

## 步骤 2: 安装 Archlinux ARM

我选择 Archlinux，而不是最常见的 Raspbian，因为我喜欢 Archlinux。此外，它很容易安装和使用。更重要的是，`geth`（用于运行完整的区块链（blockchain）节点的命令行界面）已经在社区存储库中打包并可用，因此安装只需要 `pacman -Syu geth`。

安装 Archlinux ARM 非常简单。关于如何创建 SD 卡的安装指南是非常清楚的。我将展示创建 SD 卡的两种方法。第一个在 Linux 上，第二个在 Windows 上。

### 在 Linux 系统上，使用 SD 卡安装 Arch Linux

将 SD 卡插入 SD 读卡器。这时会出现设备 `/dev/mmcblk0`。然后，按顺序执行以下的命令：

```
1 # Become root:
2 su
3 # Create a new partition table
4 fdisk /dev/mmcblk0
5 # From the guide:
6 # At the fdisk prompt, delete old partitions and create a new one:
7 # Type o. This will clear out any partitions on the drive.
8 # Type p to list partitions. There should be no partitions left.
9 # Type n, then p for primary, 1 for the first partition on the drive, press
   ENTER to accept the default first sector, then type +100M for the last
   sector.
10 # Type t, then c to set the first partition to type W95 FAT32 (LBA).
11 # Type n, then p for primary, 2 for the second partition on the drive, and
   then press ENTER twice to accept the default first and last sector.
12 # Write the partition table and exit by typing w.
13
14 # Create a new work directory and move in
15 mkdir /tmp/wd
```

```
16 cd /tmp/wd
17 # Create and mount on /boot the FAT filesystemmkfs.vfat /dev/mmcblk0p1
18 mkdir boot
19 mount /dev/mmcblk0p1 boot
20
21 # Create and mount on / the ext4 filesystem
22 mkfs.ext4 /dev/mmcblk0p2
23 mkdir root
24 mount /dev/mmcblk0p2 root
25
26 # Download and extract the root filesystem
27 wget http://os.archlinuxarm.org/os/ArchLinuxARM-rpi-2-latest.tar.gz
28
29 bsdtar -xpf ArchLinuxARM-rpi-2-latest.tar.gz -C root
30 sync
31
32 # Move boot files to the first partition:
33 mv root/boot/* boot
34
35 # Umount the two partitions
36 umount boot root
```

现在您可以将 SD 卡插入到 Raspberry Pi 中。将 UPS 连接到墙壁插座，将 Raspberry Pi 和 USB 开关连接到 UPS，将硬盘连接到 USB 集线器并切换到 Raspberry Pi。连接以太网电缆，然后打开电源。完成！您可以跳转到 Archlinux 配置了。

### 在 Windows 系统上使用 SD 卡安装 ArchLinux

如果您是 Windows 用户，您可以：

1. 从 <https://sourceforge.net/projects/win32diskimager/> 下载 **Win32 Disk Imager**
2. 从 <https://sourceforge.net/projects/archlinux-rpi2/> 下载 **Archlinux ARM SD 镜像**
3. 运行 **Win32 Disk Imager**。选择正确的设备、下载的 .img 文件，并写入系统

现在，你的 SD 卡几乎准备好了，因为我们必须调整根分区的文件系统的大小，否

则我们只有 2GB 可用, 而不是我们的 SD 卡允许的 32 GB。现在, 您可以将 SD 卡插入 Raspberry Pi。将 UPS 连接到墙壁插座, 将 Raspberry Pi 和 USB 开关连接到 UPS, 将硬盘连接到交换机并切换到覆盆子。连接以太网电缆, 打开电源。

下载 `putty` 并连接到分配给 Raspberry Pi 的路由器的 IP。用户名 `alarm`, 密码 `alarm`。

现在我们必须要在系统运行的时候, 重新调整 `root` 分区的文件系统大小。为了做到这一点, 我们必须删除分区 (不要担心, 数据将被保留), 并创建一个占用所有可用空间的新分区。

```
1 # Become root
2 su # password root
3
4 # Use fdisk to change the partition table
5 fdisk /dev/mmcblk0
6 # At the fdisk prompt, delete the partition number 2
7 # Type d then 2.
8 # Create a new primary partition (id 2)
9 # Type n then p.
10 # At the prompt for the first sector and last sector just press enter.
11 # This will automatically start from the old first sector and will set the
    last
12 # sector to the end of the available space.
13 # Write the partition table and exit by typing w.
14
15 # Now reboot
16 reboot
```

现在, 再次登录到 Raspberry Pi。

```
1 # become root
2 su # password root
3 # Now we use resize2fs to let the kernel aware that the filesystem has been
    enlarged
4 resize2fs /dev/mmcblk0p2
5 # logout
6 exit
```

完成。现在, 您可以配置全新的 Archlinux 设置。

### 步骤 3: 配置 Archlinux

我们的目标是设置一个完整的以太坊(Ethereum)节点,其数据(区块链(blockchain)和钱包)存储在外部硬盘驱动器上。在此之前,我们必须为外置硬盘配置为 RAID 1。

注意: Archlinux 映像包含一些过时的软件。这并不是件好事。更新软件能给我们更好的安全性。但是,我们将在升级系统之前,利用这一事实。

安装的 ssh 服务器是如此的旧,作为默认选项,仍然允许 root 用户的远程登录。我们将利用此功能,以便在 /home 上创建并挂载 RAID 1 分区。这样,我们可以完全删除 alarm 用户的主目录(以及任何其他具有主目录的用户),而没有任何不便。

以 root 用户身份远程登录(在 Windows 下使用 Putty,用户 root 和密码 root,在 Linux 下只有用户 ssh 具有相同的凭据)

### 步骤 4: 设置 RAID

我们的 2 个外部硬盘位于 /dev/sda 和 /dev/sdb 之下。首先,我们将在这些设备上创建一个分区表,并为我们的数据创建一个分区。

由于,我们将在这些硬盘上存储大量数据,因此建议您创建一个 GPT 分区表。为此,我们需要 gptfdisk。

```
1 # Install gdisk
2 pacman -Sy gptfdisk
3
4 # Use it to partition both hard drives. Default values are OK since
5 # we want to create a single partition as big as the whole hard drives
6
7 gdisk /dev/sda
8 # Now create a single partition as big as the drive
9 # Type n then p.
10 # Accepts the defaults for every following question.
11 # Type w to write and exit
12
13 gdisk /dev/sdb
14 # Do the same exact procedure of above
15
16 # We created /dev/sda1 and /dev/sdb1 partitions
17 # Create an ext4 filesystem on these partitions
```

```
18 mkfs.ext4 /dev/sda1
19 mkfs.ext4 /dev/sdb1
```

好的，我们分区完了我们的硬盘。现在，我们可以创建一个逻辑 RAID 1 卷。RAID 1 是最直接的 RAID 级别：直接镜像（straight mirroring）。这意味着如果我们一个驱动器发生故障，RAID 阵列提供的块设备将继续正常工作。我们现在要使用 mdadm 创建新的块设备 /dev/md0：

```
1 mdadm --create --verbose --level=1 --metadata=1.2 --raid-devices=2 /dev/md0
   /dev/sda1 /dev/sdb1
```

最后一件事是格式化驱动器：

```
1 mkfs.ext4 /dev/md0
```

完成！我们的驱动器已经可以使用了。（有关更全面的指南，请参阅官方文档：<https://wiki.archlinux.org/index.php/RAID>）

## 步骤 5: RAID /home 和用户设置

由于，我们仍然以 root 身份登录，因此我们可以删除当前的默认用户 alarm 及其主目录。然后，我们将把 RAID 设备安装到 /home，以便为我们的数据提供冗余。接下来，我们创建一个新用户来使用 geth。

```
1 userdel -r alarm
2 mount /dev/md0 /home
3 # Add the user `geth`. Automatically creates /home/geth
4 useradd geth
5 # Set a new password for user geth
6 passwd geth
```

现在，我们需要将 /dev/md0 挂载在 /home 上，重新启动。更改 /etc/fstab 文件，添加行（使用 nano 或 vim）：

```
1 /dev/md0          /home    ext4    defaults,nofail 0        0
```

保存修改，并退出。

现在我们有：一个新的用户 geth；设置新密码；安装在 /home 上的 RAID 设备；我们准备升级整个系统。

```
1 # First change the root password
```

```
2 passwd
3 # Now upgrade everything
4 pacman -Syu# reboot
5 reboot
```

升级后, 我们无法通过 ssh 作为 root 登录 (这是一件好事)。我们以 geth 登录。

我们现在要安装 geth, 将其配置为在启动时启动, 并修复使用外置硬盘可能导致我们的 RAID-1 设置的一些小问题。

### geth & 启动服务

在安装 geth 之前, 我们必须禁用我们的硬盘中的所有省电选项。我们已经将外置硬盘, 连接到外部供电的 USB, 以便给他们足够的电力工作。当不使用磁盘或者关闭磁盘时, 外部硬盘驱动器会减慢速度。这可能是我们的 RAID 配置的一个问题, 因为一旦驱动器关闭, 它将不可用作存储, 因此我们的 RAID 设置变得无用。

我们必须禁用两个硬盘的省电设置, 每次打开我们的 Raspberry Pi, 我们都必须这样做。一个简单的系统服务文件是我们需要的。

```
1 su # become root
2 # install hdparm
3 pacman -S hdparm
```

现在, 以 root 身份使用您首选的文本编辑器, 并使用此内容创建文件 /etc/systemd/system/hdparm.service :

```
1 [Unit]
2 Description=Disable HDDs power saving
3
4 [Service]
5 Type=oneshot
6 ExecStart=/usr/bin/hdparm -B 255 /dev/sda
7 ExecStart=/usr/bin/hdparm -B 255 /dev/sdb
8 ExecStart=/usr/bin/hdparm -S 0 /dev/sda
9 ExecStart=/usr/bin/hdparm -S 0 /dev/sdb
10
11 [Install]
12 WantedBy=multi-user.target
```

此服务文件将禁用高级电源管理 (Advanced Power Management) 功能和两个硬盘的旋转。现在, 我们启动服务并在每次启动时启用它。

```
1 systemctl start hdparm.service
2 systemctl enable hdparm.service
```

好了。我们终于准备好安装 **geth**, 并在每次启动时运行。

```
1 # Install geth
2 pacman -S geth
```

创建具有以下内容的服务文件 `/etc/systemd/system/geth@.service` :

```
1 [Unit]
2 Description=geth Ethereum daemon
3 Requires=network.target
4
5 [Service]
6 Type=simple
7 User=%I
8 ExecStart=/usr/bin/geth --syncmode fast --cache 64 --maxpeers 12
9 Restart=on-failure
10
11 [Install]
12 WantedBy=multi-user.target
```

您可以使用您想要的同伴 (**peer**) 数量更改 **maxpeers** 标志。最高的就是你的网络速度, 这个数字最高 (默认值为 **25**)。

让我们启动这个进程, 并使其从 `geth` 用户运行。

```
1 systemctl enable geth@geth.service
2 systemctl start geth@geth.service
3 # back to geth user
4 exit
```

完成! 现在, 这个节点正在与 **elhere** 网络同步, 您可以使用 **geth** 的所有功能 (这里是官方文档: <https://github.com/ethereum/go-ethereum/wiki/geth>)。

您可以使用 `systemctl status geth@geth.service` 监视 `geth` 的输出, 您可以使用 `geth attach` (从 `geth` 用户) 启动附加到运行节点的控制台。

一个小的注意事项:

区块链 (**blockchain**) 同步可能需要很长时间。此外, **geth** 在同步时使用大量的 **RAM**。因此, 内核会杀死进程, 然后系统重新启动它是不常见的。然而, 一旦整个区块链已经同步, 并且在一个小时接收的块的数量很少, 具有 **1GB RAM** 的 **Raspberry Pi** 可以处理同步而不会崩溃。

理想情况下, 如果您已经同步了一个 **geth** 节点, 建议使用 **scp** 或任何其他远程复制工具, 而不是使用 **geth** 来复制块链。

## 步骤 6: 结论

在这篇文章中, 我描述了如何使用 **Archlinux ARM** 在 **Raspberry Pi 3** 上设置一个以太坊 (**Ethereum**) 节点。在初始同步后, 我们可以使用它发送/接收 **ETH**, 由于 **UPS** 而处理电源故障, 并具有处理硬盘 (**HDD**) 故障的冗余数据。

请注意, 为了增加节点的安全性, 需要做很多事情。我留在这里几点提示:

1. 不要将节点暴露在局域网之外
2. 带有 **deny all** 的 **iptables** 策略是一个好朋友
3. 在路由器中配置防火墙
4. 使用强密码
5. 不要安装 **sudo**。如果你安装它, 请添加默认值 **rootpw**, 并为 **geth** 和 **root** 使用不同的密码。
6. 将节点放置在安全位置
7. 评估如果 **RAID** 设备的加密是一个选项 (尽管 **geth** 强制我们用密码来保护我们的钱包, 这对我需要是足够的)。

如果你发现这篇文章有帮助, 让我知道! 只需在下面的表格中留下评论。

因为我们在这里尝试 **ETH**, 如果你想给我一个啤酒 (或其他任何东西), 你可以发送给我 **ETH** 到: `0xa1d283e77f8308559f62909526ccb2d9444d96fc`。

原文链接: <https://pgaleone.eu/raspberry/ethereum/archlinux/2017/09/06/ethereum-node-rasberri-pi-3/>

原文链接: <https://www.wandianshenme.com/play/install-ethereum-node-on-rasberri-pi-3>