

# FreeBSD 操作系统在无远程控制台下的 远程安装

## 摘要

本文介绍了当远程控制台不可用的情况下 FreeBSD 操作系统的远程安装。文章背后的主要灵感归功于和 [Martin Matuska](#) 有由 [Paweł Jakub Dawidek](#) 提供的宝贵输入合作的结果。

## 目录

1. 背景.....	1
2. 简介.....	1
3. 准备工作 - mfsBSD.....	2
4. FreeBSD 操作系统的安装.....	4
5. ZFS.....	7

## 1. 背景

世界上有很多的服务器主机供应商，但是他中只有很少的一部分正式支持 FreeBSD，他通常对他提供的服务器上安装 Linux® 发行版提供支持。

在某些情况下，如果请求些公司他会安装一个首选的 Linux® 发行版。有了这个，我将安装 FreeBSD。在其他情况下，他可能提供一个急救系统用于紧急情况。使用这个可能将有利于我的目的更好的。

本文涵盖了引用一个包含 RAID-1 及 ZFS 性能的 FreeBSD 系统的远程安装的基本安装配置所必需的。

## 2. 简介

这一节会摘要本文的目的以及更好描述里所概括的东西。本文中的这些指令将有益于那些使用不支持 FreeBSD 的托管设施提供服务器的人。

1. 如我提到背景的那一节，许多的有声望的服务器主机托管公司提供了各自的急救系统。可以从他自己的局域网 并且可以通过 SSH 访问。他通常提供技术支持目的用于帮助他客户修正坏的操作系统。如文章将明确的，我将能够通过些急救系统的帮助来安装 FreeBSD。
2. 文章的下一小节会描述如何配置，并在本地机器上构建最小限度的 FreeBSD。新版本最会从随机存储运行到远程机器上面去。他将允许我使用 Sysinstall 应用程序从一个 FTP 镜像安装一套完整的 FreeBSD 操作系统。
3. 文章的剩余内容除了描述 ZFS 文件系统的配置将描述系统本身的安装。

## 2.1. 需求

想要成功地做下去，你必须：

- 有一个可通 SSH 网络的操作系统。
- 理解 FreeBSD 的安装过程
- 熟悉 `sysinstall(8)` 实用程序
- 有 FreeBSD 安装的 ISO 映像文件或者易于使用的 CD

## 3. 准备工作 - mfsBSD

在 FreeBSD 可能安装到目标系统上之前，需要先创建一个最小化的从磁碟的 FreeBSD 操作系统映像文件。此方法中新系统必须能从网络启动，并且安装的其他过程能在没有进程到系统控制台的情况下完成。

mfsBSD 实用工具能被用来创建一个微小的 FreeBSD 映像。如 mfsBSD 名字的含义 ("mfs" 的意思是 "memory file system" 内存文件系统)，最后的映像全部从随机存取器运行。多了这个特性，磁碟的操作将不会有任何限制，因此它能被用来安装一个完整的 FreeBSD 操作系统。mfsBSD 的主页在 <http://people.freebsd.org/~mm/mfsbsd/>，包含了指向最新出的实用工具。

请注意于 mfsBSD 内幕以及它所有的实用都超出了本文的内容，感兴趣的读者可以去 mfs 的原始文档得到更多内容。

下面并解出最新的 mfsBSD 版本，并改自己的当前工作目录到存在 mfsBSD 脚本文件的目录：

```
# fetch http://people.freebsd.org/~mm/mfsbsd/mfsbsd-latest.tar.gz
# tar xvzf mfsbsd-1.0-beta1.tar.gz
# cd mfsbsd-1.0-beta1/
```

### 3.1. mfsBSD 的配置

引入 mfsBSD 之前，必须配置一些重要的配置项。最重要的是我们必须有正确地，自然地，网络配置。最合适的方法配置网络取决于我们是否事先知道我们会用到的网络接口，而且网络接口驱动程序被系我的硬件输入。我们将看到 mfsBSD 如何能在任一情况下被配置。

另外一件重要的事情是设置 `root` 的密码。我们将通过 `conf/rootpw.conf` 文件来完成。记住文件将把密码保存在文本中，所以在此我们不推荐使用真的密码。然而，它只是一个使用一次的密码，可以在随后安装好的系统中更改它。

#### 3.1.1. 配置 `conf/interfaces.conf` 的方法

如果我安装好的网络是未知类型的，我可以使用 mfsBSD 的自探测功能。mfsBSD 脚本能探测到正确的设备来使用，基于网络接口的 MAC 地址，我假设在 `conf/interfaces.conf` 文件中配置如下：

```
initconf_interfaces="ext1"
initconf_mac_ext1="00:00:00:00:00:00"
initconf_ip_ext1="192.168.0.2"
initconf_netmask_ext1="255.255.255.0"
```

别忘了添加 `defaultrouter` 信息到 `conf/rc.conf` 文件中：

```
defaultrouter="192.168.0.1"
```

### 3.1.2. 编辑 `conf/rc.conf` 的方法

当网络接口的配置是已知类型的，使用 `conf/rc.conf` 文件添加网络配置会更加方便。该文件的写法跟 FreeBSD 中标准的 `rc.conf(5)` 文件的写法相同。

例如，当你知道被使用的将是一个 `re(4)` 网络接口时，可以在 `conf/rc.conf` 文件中配置如下：

```
defaultrouter="192.168.0.1"
ifconfig_re0="inet 192.168.0.2 netmask 255.255.255.0"
```

## 3.2. 构建一个 mfsBSD 映像

构建一个 mfsBSD 映像文件的流程是非常明了的。

第一步是挂载 FreeBSD 的安装 CD，或者挂载安装 ISO 文件到 `/cdrom`。因本例子的缘故，在文章中我将假定你挂载的是 FreeBSD 7.0-RELEASE ISO 文件。使用 `mdconfig(8)` 程序挂载 ISO 映像文件到 `/cdrom` 目录非常：

```
# mdconfig -a -t vnode -u 10 -f 7.0-RELEASE-amd64-disc1.iso
# mount_cd9660 /dev/md10 /cdrom
```

接着，构建可启动的 mfsBSD 映像：

```
# make BASE=/cdrom/7.0-RELEASE
```



上面的 `make` 命令必须在 mfsBSD 目录的最高一行，也就是：`~/mfsbsd-1.0-beta1/`。

## 3.3. 启动 mfsBSD

在 mfsBSD 映像已准备好后，必须把它上到远程的一个正在运行的急救系统上或者一个安装了 Linux® 发行版的系统上。最适合做这个工作的工具是 `scp`：

```
# scp disk.img root@192.168.0.2:.
```

想要正确的引导 mfsBSD 映像，必须把它安放在机器的第一块（可写）磁盘上。可能会和使用的例子我假定的不同，第一块可写磁盘是 sda：

```
# dd if=/root/disk.img of=/dev/sda bs=1m
```

如果一切正常，映像应该在磁盘存在于第一磁道的 MBR（主引导区）而机器也应该能被启动了。使用工具 [ping\(8\)](#) 来看机器是否被正确启动。一旦它回到在线状态，就应该能使用 `root` 用户和配置好的密码通过 [ssh\(1\)](#) 来访问它了。

## 4. FreeBSD 操作系统的安装

mfsBSD 成功被引导后它就能通过 [ssh\(1\)](#) 登录了。这一节会描述如何创建 slices 并给 slices 的 label， RAID-1 配置 [gmirror](#)， 以及如何使用 [Sysinstall](#) 来安装一个最小的 FreeBSD 操作系统版本。

### 4.1. 准备磁盘

首要的任务是准备 FreeBSD 分配磁盘空间，也就是，创建 slices 和 partitions。很自然，当前运行的系统是全部被写入到系统内存中的因此操作磁盘将没有任何问题。要完成这个任务，可以使用 [Sysinstall](#) 或者 [fdisk\(8\)](#) 中的二者任一并组合工具 [bsdlabeled\(8\)](#)。

在开始，将所有磁盘都清空，在每个磁盘上重复如下命令：

```
# dd if=/dev/zero of=/dev/ad0 count=2
```

下面，使用喜欢的工具创建 slices 并给磁盘 label。比较常用的方法是使用 [Sysinstall](#)，它大也可能几乎没有漏洞方法是使用标准的基于文本的 UNIX® 工具，类似于 [fdisk\(8\)](#) 和 [bsdlabeled\(8\)](#) 这些工具的使用也会在下一节中包括。前者已被包括在 FreeBSD 手册的 [安装FreeBSD](#) 一章中了。如本节中提到的，这篇文章会展示如何配置一个具有 RAID-1 和 ZFS 性能的系统。我的配置由一个小工具 [gmirror\(8\)](#) 映像 / (root), /usr 和 /var 文件系统，并把剩余的磁盘空间被分配给 [zpool\(8\)](#) 映像出的 ZFS 文件系统。注意，ZFS 文件系统将在 FreeBSD 操作系统成功安装并启动后才会被配置。

下面的例子会描述如何去创建 slices 和 labels，在每个 partition 上初始化 [gmirror\(8\)](#) 并如何在每个被映像的 partition 上创建 UFS2 文件系统：

```
# fdisk -BI /dev/ad0 ①
# fdisk -BI /dev/ad1
# bsdlablel -wB /dev/ad0s1 ②
# bsdlablel -wB /dev/ad1s1
# bsdlablel -e /dev/ad0s1 ③
# bsdlablel /dev/ad0s1 > /tmp/bsdlablel.txt && bsdlablel -R /dev/ad1s1 /tmp/bsdlablel.txt
④
# gmirror label root /dev/ad[01]s1a ⑤
# gmirror label var /dev/ad[01]s1d
# gmirror label usr /dev/ad[01]s1e
# gmirror label -F swap /dev/ad[01]s1b ⑥
# newfs /dev/mirror/root ⑦
# newfs /dev/mirror/var
# newfs /dev/mirror/usr
```

- ① 在整个磁碟上建立一个 slice 并初始化包含在磁碟第一个扇区代号。重新在系统上全部的磁碟上执行此命令。
- ② 将磁碟写入一个包括代号的内容的基准 label。
- ③ 现在，手动去磁碟的 label。可以参考 [bsdlablel\(8\)](#) 的联机手册来得到如何建立 partitions 的方法。建立如下 partitions, a 为 (/root) 文件系统, b 为 swap 交换空间, d 为 /usr 有最后 f 被用于 ZFS。
- ④ 引入刚才建立的 label 到第二磁碟, 所以磁碟会使用同样的 label。
- ⑤ 在这个 partition 上初始化 [gmirror\(8\)](#)。
- ⑥ 注意 -F 被用在 swap 交换分区的 partition。 [gmirror\(8\)](#) 这个指令依赖于可写的状态除非源系统故障。
- ⑦ 在这个被镜像的分区上建立一个 UFS2 的文件系统。

## 4.2. 系统安装

这是最重要的一部分。此节将描述如何在我上一小节已准备好的磁碟上安装一个最小的 FreeBSD 版本。要完成这个目的, 所有的文件系统需要被挂载乃至 Sysinstall 可以把 FreeBSD 系统的内容写到磁碟上:

```
# mount /dev/mirror/root /mnt
# mkdir /mnt/var /mnt/usr
# mount /dev/mirror/var /mnt/var
# mount /dev/mirror/usr /mnt/usr
```

当做完这些, 打 [sysinstall\(8\)](#)。从主菜单自定义 Custom 安装。其中 Options 然后按回。使用方向键取帮助, 移动鼠标指针到 Install Root, 按 **空格** 更改 /mnt。按 **回** 提交更改并使用 **q** 退出 Options ( ) 菜单。



注意——非常重要, 如果被跳过了, Sysinstall 将不能安装 FreeBSD。

到 Distributions (行版) 菜单, 使用方向键移动鼠标指针到 Minimal (最小化), 并使用 **空格** 选中。本文使用了最小版本来保存网络通信信息, 因为系统本身会通过 ftp 来安装。使用 **Exit** (退出) 退出这个菜单。



Partition 和 Label 菜单将被跳过, 有些没有多少价了。

Media (媒介) 菜单, 选择 **FTP** 选项。 选择一个距离最近的镜像站点并提交 Sysinstall 假定网络已配置完好。 将再回到 Custom (自定义) 菜单。

最后, 输入最后的来自发行版的安装程序, Commit, 当安装完成后退出 Sysinstall 即可。

## 4.3. 后期安装

FreeBSD 操作系统在安装完了; 通常情况下, 安装程序没有结束。 需要进行一些安装后期的操作使得容 Freebsd 在将来并能登入系统。

必须在 `chroot(8)` 到安装的全新的系统中来完成安装。 使用如下命令:

```
# chroot /mnt
```

要到达我们的目的, 进行如下操作:

- 拷入 **GENERIC** (通用) 内核到 `/boot/kernel` 目录:

```
# cp -Rp /boot/GENERIC/* /boot/kernel
```

- 创建 `/etc/rc.conf`, `/etc/resolv.conf` 和有 `/etc/fstab` 文件。 不要忘记正确地配置网络信息并在 `/etc/rc.conf` 文件中启用 `sshd`。 `/etc/fstab` 文件内容类似于下面的内容:

# Device	Mountpoint	FStype	Options	Dump	Pass#
<code>/dev/mirror/swap</code>	<code>none</code>	<code>swap</code>	<code>sw</code>	<code>0</code>	<code>0</code>
<code>/dev/mirror/root</code>	<code>/</code>	<code>ufs</code>	<code>rw</code>	<code>1</code>	<code>1</code>
<code>/dev/mirror/usr</code>	<code>/usr</code>	<code>ufs</code>	<code>rw</code>	<code>2</code>	<code>2</code>
<code>/dev/mirror/var</code>	<code>/var</code>	<code>ufs</code>	<code>rw</code>	<code>2</code>	<code>2</code>
<code>/dev/cd0</code>	<code>/cdrom</code>	<code>cd9660</code>	<code>ro,noauto</code>	<code>0</code>	<code>0</code>

- 创建 `/boot/loader.conf` 文件, 并写入如下内容:

```
geom_mirror_load="YES"
zfs_load="YES"
```

- 运行下面的命令, 使得 ZFS 在下次启动后可用:

```
# echo 'zfs_enable="YES"' >> /etc/rc.conf
```

- 可以用 `adduser(8)` 工具来添加额外的用户。 不要忘记添加一个用户到 `wheel`, 用户可以在重新启动后获得 root 权限。
- 反方向的配置是否正确。

在您的系统在下次启动后可用了。 使用 `reboot(8)` 命令重新启动您的系统。

## 5. ZFS

如果你的系统重新安装后完好，你在安装后能登录了。欢迎来到新的 FreeBSD 安装，进行程序的不使用程序控制台的安装。

最后剩下的工作是配置 [zpool\(8\)](#) 并创建一些 [zfs\(8\)](#) 文件系统。建立并管理 ZFS 非常有趣。首先，创建一个镜像的 pool：

```
# zpool create tank mirror /dev/ad[01]s1f
```

再接着，创建一些文件系统：

```
# zfs create tank/ports
# zfs create tank/src
# zfs set compression=gzip tank/ports
# zfs set compression=on tank/src
# zfs set mountpoint=/usr/ports tank/ports
# zfs set mountpoint=/usr/src tank/src
```

这就是全部了。如果你对 FreeBSD 上的 ZFS 感兴趣，看看 FreeBSD WIKI 中的 [ZFS](#) 一节。