

简介

本文主要通过裁剪现有Linux系统，打造一个属于自己的Linux小系统，让其能够装载网卡驱动，并配置IP地址，实现网络功能。

主要步骤：

1. 添加硬盘
2. 对硬盘进行分区，并创建文件系统
3. 创建挂载点 `/mnt/boot` 和 `/mnt/root` 并挂载
4. 安装 `grub`
5. 复制虚根和内核
6. 编写 `grub.conf` 配置文件
7. 创建一级子目录，如 `/bin`，`/lib`等
8. 复制 `bash` 和相关库文件及其他命令
9. 添加网络功能，并实现开机自动加载网卡
10. 测试是否能够开机并引导系统启动
11. 错误排查

1. 添加硬盘

硬盘的大小 `20G` 已经足够用了。

添加硬件向导

指定磁盘容量
磁盘大小为多少?

最大磁盘大小(GB)(S):

针对 CentOS 64 位 的建议大小: 20 GB

☐ 立即分配所有磁盘空间(A)。
分配所有容量可以提高性能，但要求所有物理磁盘空间立即可用。如果不立即分配所有空间，虚拟磁盘的空间最初很小，会随着您向其中添加数据而不断变大。

☒ 将虚拟磁盘存储为单个文件(O)
☐ 将虚拟磁盘拆分成多个文件(M)
拆分磁盘后，可以更轻松地在计算机之间移动虚拟机，但可能会降低大容量磁盘的性能。

< 上一步(B) 下一步(N) > 取消

2. 对硬盘进行分区

```

1 [root@centos6 ~]# echo "- - -" > /sys/class/scsi_host/host2/scan <==让系
2 [root@centos6 ~]# lsblk <==新增加的硬盘已经被系统识别
3 NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
4 sr0 11:0 1 3.7G 0 rom
5 sda 8:0 0 200G 0 disk
6 |sda1 8:1 0 4G 0 part /boot
7 |sda2 8:2 0 50G 0 part /
8 sdb 8:16 0 20G 0 disk
9 [root@centos6 ~]# fdisk -l /dev/sdb <==对/dev/sdb进行分区，/dev/sdb1
10
11 Disk /dev/sdb: 21.5 GB, 21474836480 bytes
12 255 heads, 63 sectors/track, 2610 cylinders
13 Units = cylinders of 16065 * 512 = 8225280 bytes
14 Sector size (logical/physical): 512 bytes / 512 bytes
15 I/O size (minimum/optimal): 512 bytes / 512 bytes
16 Disk identifier: 0x8c1b09e8
17
18 Device Boot Start End Blocks Id System
19 /dev/sdb1 1 132 1060258+ 83 Linux
20 /dev/sdb2 133 2610 19904535 83 Linux
21 [root@centos6 ~]# mkfs.ext4 /dev/sdb1
22 ... 命令结果省略
23 [root@centos6 ~]# mkfs.ext4 /dev/sdb2

```

人 人 人 人 人 人

```
24 | ... 命令结束省略
25 | [root@centos6 ~]# blkid <==分区已经创建完成, 文件系统为ext4
26 | /dev/sda2: UUID="498fb39f-0cc0-4921-9d11-0f122434bf8f" TYPE="ext4"
27 | /dev/sda1: UUID="8a3beae-e8b2-40b7-8790-a7fd847b5719" TYPE="ext4"
28 | /dev/sdb1: UUID="c2628450-9450-468f-8cf0-5aa911dcefb9" TYPE="ext4"
29 | /dev/sdb2: UUID="4f13af1d-8a48-4b62-9cca-3ea3f5fa2e45" TYPE="ext4"
```

3. 创建挂载点 **/mnt/boot** 和 **/mnt/root** 并挂载

```
1 | [root@centos6 ~]# mkdir /mnt/boot <==创建/mnt/boot目录 (目录必须为boot区
2 | [root@centos6 ~]# mkdir /mnt/sysroot <==创建/mnt/sysroot目录
3 | [root@centos6 ~]# mount /dev/sdb1 /mnt/boot/ <==挂载/dev/sdb1到/mnt/boot
4 | [root@centos6 ~]# mount /dev/sdb2 /mnt/sysroot/ <==挂载/dev/sdb1到/mnt/boot
```

4. 安装 **grub**

```
1 | [root@centos6 ~]# grub-install --root-directory=/mnt /dev/sdb <==注意: 此
2 | Probing devices to guess BIOS drives. This may take a long time.
3 | Installation finished. No error reported.
4 | This is the contents of the device map /mnt/boot/grub/device.map.
5 | Check if this is correct or not. If any of the lines is incorrect,
6 | fix it and re-run the script `grub-install'.
7 |
8 | (fd0) /dev/fd0
9 | (hd0) /dev/sda
10 | (hd1) /dev/sdb <==安装完成
11 | [root@centos6 ~]# ls /mnt/boot/ <==可以看到grub目录已经生成
12 | grub lost+found
```

5. 复制虚拟文件系统和内核

```
1 | [root@centos6 ~]# cp /boot/vmlinuz-2.6.32-696.el6.x86_64 /mnt/boot/
2 | [root@centos6 ~]# cp /boot/initramfs-2.6.32-696.el6.x86_64.img /mnt/boot/
3 | [root@centos6 ~]# ls /mnt/boot/ <==可以看到已经复制成功
4 | grub initramfs-2.6.32-696.el6.x86_64.img lost+found vmlinuz-2.6.32-696.
```

6. 编写 **grub.conf** 配置文件

```

1 [root@centos6 ~]# vim /mnt/boot/grub/grub.conf
2 default=0          <==表示有多个引导菜单时，选择哪一个作为默认启动引导菜单，0代表使用第一
3 title ihaiyun      <==可以理解为grub引导的配置列表，ihaiyun代表title的名称（可以自定义）
4 timeout=5         <==表示如果5秒内，用户没有选择任何一个title，那么就使用default指定的
5 root (hd0,0)       <==表示kernel文件和initrd文件所在分区，并不是指“根分区”
6 kernel /vmlinuz-2.6.32-696.el6.x86_64 root=/dev/sda2 selinux=0 init=/bin/
7 initrd /initramfs-2.6.32-696.el6.x86_64.img      <==指定虚拟文件系统所在位置
8

```

7. 创建一级子目录

```

1 [root@centos6 ~]# mkdir /mnt/root/{usr,sbin,bin,lib,lib64,tmp,var,usr,sys,
2 [root@centos6 ~]# ls /mnt/root/          <==目录已经创建完成
3 bin dev home lib lib64 lost+found media mnt opt root sbin sys

```

8. 复制 **bash** 和相关库文件及其他命令

复制一些常用命令 **bash,ls,cat,vim,reboot,hostname,ifconfig,netstat,ps**等

```

1 注意：复制命令时，要记得把库文件和命令都要复制过去，本人是使用脚本进行复制的，脚本内容如下
2 #!/bin/bash
3 #-----
4 # File name: copy.sh
5 # Revision: 1.1
6 # Date: 2017-09-01
7 # Author: houhaiyun
8 # Email: houhaiyun18@163.com
9 # Website: http://www.iav18.cn
10 # Description:
11 # Copyright: 2017 haiyun
12 # License: GPL
13 #-----
14 dir () {
15     DIR=`dirname $SYS$line`
16     DIR2=`dirname $SYS$CPATH`
17     [ -d $DIR ] || mkdir -p $DIR
18     [ -d $DIR2 ] || mkdir -p $DIR2
19 }
20
21 cpall () {
22     cp $line $DIR
23     cp $CPATH $DIR2
24 }

```

```

25 }
26
27 SYS=/mnt/sysroot          #<==注意: 如果你创建的/mnt/不是sysroot目录需要在此处进行修
28
29 while [ "$COMD" != "quit" ] ; do
30     read -t 10 -p "please input a command(input 'quit' to exit): " COMD
31     [ $? -ne 0 ] && echo -e "\ntimeout exit..." && exit 10
32     [ "$COMD" == "quit" ] && echo "exit..." && exit 12
33     if `which --skip-alias $COMD &> /dev/null` ; then
34         CPATH=`which --skip-alias $COMD`
35     else
36         echo "command not found or command is error"
37         continue
38     fi
39     ldd $CPATH | grep -o "/.* " | while read line ; do
40         dir
41         cpall
42         echo -e "cp $CPATH\t----->\t $DIR2"
43         echo -e "cp $line\t----->\t$DIR"
44     done
45 done
46 关于脚本内容: 将会在后面博客介绍~~~~~
47 [root@centos6 ~]# ls /mnt/root/bin/          <==可以看到/bin目录下已经有命令
48 bash      cat      hostname  ls      netstat  ps
49 [root@centos6 ~]# ls /mnt/root/sbin/        <==可以看到/sbin目录下已经有命令
ifconfig ip poweroff reboot

```

9. 添加网络功能, 并实现开机自动加载网卡

```

1 [root@centos6 ~]# ethtool -i eth0          <==显示网卡的驱动信息, 可以看到驱动为e10
2 driver: e1000
3 version: 7.3.21-k8-NAPI
4 firmware-version:
5 bus-info: 0000:02:01.0
6 supports-statistics: yes
7 supports-test: yes
8 supports-eeprom-access: yes
9 supports-register-dump: yes
10 supports-priv-flags: no
11 [root@centos6 ~]# locate e1000             <==搜索e1000
12 /lib/modules/2.6.32-696.el6.x86_64/kernel/drivers/net/e1000
13 /lib/modules/2.6.32-696.el6.x86_64/kernel/drivers/net/e1000e
14 /lib/modules/2.6.32-696.el6.x86_64/kernel/drivers/net/e1000/e1000.ko      <=
15 /lib/modules/2.6.32-696.el6.x86_64/kernel/drivers/net/e1000e/e1000e.ko
16 /usr/src/kernels/2.6.32-696.el6.x86_64/drivers/net/e1000
17

```

```

18 /usr/src/kernels/2.6.32-696.el6.x86_64/drivers/net/e1000e
19 /usr/src/kernels/2.6.32-696.el6.x86_64/drivers/net/e1000/Makefile
20 /usr/src/kernels/2.6.32-696.el6.x86_64/drivers/net/e1000e/Makefile
21 /usr/src/kernels/2.6.32-696.el6.x86_64/include/config/e1000.h
22 /usr/src/kernels/2.6.32-696.el6.x86_64/include/config/e1000e.h
23 [root@centos6 ~]# cp /lib/modules/2.6.32-696.el6.x86_64/kernel/drivers/net
mnt/root/lib64/          <==复制驱动到/mnt/root/lib64
24 [root@centos6 mnt]# vim /mnt/boot/grub/grub.conf      <==修改grub.conf文件
25 default=0
26 title ihaiyun
27 timeout=5
28 root (hd0,0)
29 kernel /vmlinuz-2.6.32-696.el6.x86_64 root=/dev/sda2 selinux=0 init=/bin/i
30 initrd /initramfs-2.6.32-696.el6.x86_64.img
31
32 [root@centos6 mnt]# vim /mnt/sysroot/bin/init.sh      <==编写init.sh脚本
33 #!/bin/bash
34 insmod /lib/e1000.ko          <==加载网卡驱动
35 ifconfig eth0 192.168.8.8/24  <==配置IP
36 /bin/bash    <==注意: 必须开启一个/bin/bash, 如果写的话, 那么开机启动完成网卡后, 就卡
37 这样我们就实现了开机启动网卡, 并配置IP

```

10. 测试是否能够开机并引导系统启动

以下为启动界面:




```

sda: sda1 sda2
sd 2:0:1:0: [sda] Cache data unavailable
sd 2:0:1:0: [sda] Assuming drive cache: write through
sd 2:0:1:0: [sda] Attached SCSI disk
EXT4-fs (sda2): INFO: recovery required on readonly filesystem
EXT4-fs (sda2): write access will be enabled during recovery
EXT4-fs (sda2): recovery complete
EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts:
dracut: Mounted root filesystem /dev/sda2
dracut: Switching root
e1000: Intel(R) PRO/1000 Network Driver - version 7.3.21-k8-NAPI
e1000: Copyright (c) 1999-2006 Intel Corporation.
e1000 0000:02:01.0: PCI INT A -> GSI 19 (level, low) -> IRQ 19
e1000 0000:02:01.0: eth0: (PCI:66MHz:32-bit) 00:0c:29:06:e7:89
e1000 0000:02:01.0: eth0: Intel(R) PRO/1000 Network Connection
e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-4.1# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:06:E7:89
          inet addr:192.168.8.8  Bcast:192.168.8.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

bash-4.1# ping -c 3 192.168.8.129
PING 192.168.8.129 (192.168.8.129) 56(84) bytes of data.
64 bytes from 192.168.8.129: icmp_seq=1 ttl=64 time=14.4 ms
64 bytes from 192.168.8.129: icmp_seq=2 ttl=64 time=0.765 ms
64 bytes from 192.168.8.129: icmp_seq=3 ttl=64 time=0.546 ms

--- 192.168.8.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.546/5.252/14.446/6.501 ms
bash-4.1# _

```

启动界面

查看IP, 已经配置好IP

ping同一网段的IP
可以ping通http://blog.csdn.net/Hai__Yun

11. 错误排查

- 如果启动过程中, 发现 `/dev/sda2` 提示 `busy...` 且一直卡在这个界面, 重新安装 `grub` 即可。
- 当第8步进行到完成后, 需要进行 `chroot /mnt/sysroot` 进行切根, 确保 `bash` 命令复制成功。