Chapter

4

Backup and Restore the Root File System in SD Card

This Chapter details how to backup and restore the root file systems in SD card of MXM-6410/APC-6410.

Section include：

- Backup the root file systems in SD card
- Restore the root file systems in SD card

Chapter 4 Backup and Restore the Root File System in SD Card

This chapter gives an instruction in regarding to how to backup and restore the root file systems in SD card. First, we would like to detail how to backup the root file system in SD card and next, we would like to tell you how to restore the root file system in SD card.

4.1 Backup the root file system in SD card

After developing your program under the Ubuntu Jaunty Jackalope, users might want to backup the whole file system. In this section, we will tell users how to backup the whole root file system.

Take the SD/SDHC card off from the device and plug it into a USB SD/SDHC card reader and plug the card reader into the USB port of your Linux PC. The operating system of the Linux PC in this example is FC8 and the SDHC card storage is 4GB.

Use the # *fdisk -l* command to list your disk information and find the device descriptor of you SD USB reader.

```
[root@dns1 ~]# fdisk -l

Disk /dev/sda: 41.1 GB, 41174138880 bytes
255 heads, 63 sectors/track, 5005 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xeda4eda4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          25      200781   83  Linux
/dev/sda2              26         156     1052257+  82  Linux swap / Solaris
/dev/sda3             157        5005    38949592+  83  Linux

Disk /dev/sdb: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xff7f205c

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1        4865    39078081   83  Linux

Disk /dev/sdc: 4093 MB, 4093640704 bytes
126 heads, 62 sectors/track, 1023 cylinders
Units = cylinders of 7812 * 512 = 3999744 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1        1023     3995807   83  Linux
[root@dns1 ~]#
```

We can see the device descriptor of the USB SD card reader is in disk */dev/sdc* and there is one partition */dev/sdc1*. (Note: The device descriptor might be different in your Linux PC.)

Next, mount SD/SDHC card to */mnt* directory and change directory to the */mnt*.

```
[root@dns1 ~]# mount -t ext3 /dev/sdc1 /mnt
[root@dns1 ~]# cd /mnt
[root@dns1 mnt]#
```

You can *ls* the file structure.

```
[root@dns1 mnt]# ls
bin   build  etc   lib         media  nand  proc  sbin     srv  tmp  var
boot  dev    home  lost+found  mnt    opt   root  selinux  sys  usr
[root@dns1 mnt]#
```

Next, tar the file system into a file. (The file name in this example is rootfs_ubuntu_backup.tar.gz and the backup directory is /home/eric)

```
[root@dns1 mnt]# tar cvfz /home/eric/rootfs_ubuntu_backup.tar.gz *
```

You have backup the SD/SDHC root file systems!

4.2 Restore the root file system in SD card

Plug a SD/SDHC card into a USB card reader and plug the card reader into the USB port of your Linux PC. The operating system of the Linux PC in this example is FC8 and the SDHC card storage is 4GB. (Note: 1GB is minimal requirement for the Embedian official root file system.)

Use the # *fdisk -l* command to list your disk information and find the device descriptor of you SD USB reader.

```
[root@dns1 ~]# fdisk -l

Disk /dev/sda: 41.1 GB, 41174138880 bytes
255 heads, 63 sectors/track, 5005 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xeda4eda4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          25      200781   83  Linux
/dev/sda2              26         156     1052257+  82  Linux swap / Solaris
/dev/sda3             157        5005    38949592+  83  Linux

Disk /dev/sdb: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0xff7f205c

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1        4865    39078081   83  Linux

Disk /dev/sdc: 4093 MB, 4093640704 bytes
126 heads, 62 sectors/track, 1023 cylinders
Units = cylinders of 7812 * 512 = 3999744 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1        1023     3995807   83  Linux
[root@dns1 ~]#
```

We can see the device descriptor of the USB SD card reader is in disk */dev/sdc* and there is one partition */dev/sdc1*. (Note: The device descriptor might be different in your Linux PC.)

If there is no partition in your SD card, you have to use *fdisk* to partition it first, here we partitioned the SD card as one partition. (New SD card should have one partition already by default.)

```
[root@dns1 ~]# fdisk /dev/sdc

Command (m for help): d
Selected partition 1

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1023, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1023, default 1023):
Using default value 1023

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@dns1 ~]#
```

Next, we need to format the SD/SDHC card as ext3 file system by using # *mkfs -t ext3 /dev/sdc1* command. (In FC, you can also use # *mkfs.ext3 /dev/sdc1* command.)

```
[root@dns1 ~]# mkfs -t ext3 /dev/sdc1
mke2fs 1.40.2 (12-Jul-2007)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
499968 inodes, 998951 blocks
49947 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1023410176
31 block groups
32768 blocks per group, 32768 fragments per group
16128 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 30 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@dns1 ~]#
```

And next, mount SD/SDHC card to */mnt* directory and change directory to the */mnt*.

```
[root@dns1 ~]# mount -t ext3 /dev/sdc1 /mnt
[root@dns1 ~]# cd /mnt
[root@dns1 mnt]#
```

Next, *cp* the rootfs file into /mnt directory and extracting the root file system file into this directory.

```
[root@dns1 mnt]# ls
lost+found  ubuntu.20090701.tar.gz
[root@dns1 mnt]# tar xvfz ubuntu.20090701.tar.gz
```

You can *ls* the file structure now.

```
[root@dns1 mnt]# ls
bin    dev   lib         mnt   proc  selinux  tmp                  var
boot   etc   lost+found  nand  root  srv      ubuntu.20090701.tar.gz
build  home  media       opt   sbin  sys      usr
[root@dns1 mnt]#
```

Last, remove the tarball and leave the */mnt* directory and *umount* the device.

```
[root@dns1 mnt]# rm -f ubuntu.20090701.tar.gz
[root@dns1 mnt]# cd ../
[root@dns1 /]# umount /mnt
[root@dns1 /]#
```

Take the SD/SDHC card off from the card reader and put the SD/SDHC card back to SBC and boot. You are done!