mkfs.jffs2使用方法

实例：mkfs.jffs2 -r rootfs -o rootfs.jffs2 -e 128KiB --pad=0x800000 -s 0x200 -n －b

mkfs.jffs2: Usage: mkfs.jffs2 [OPTIONS]

Make a JFFS2 file system image from an existing directory tree

Options:

-p, --pad[=SIZE] 用16進制表示所要輸出檔案的大小，也就是root.jffs2的size。很重要的 是,為了不浪費flash空間,這個值最好符合flash driver的區塊大小。

如果不足则使用0xff来填充补满。

-r, -d, --root=DIR 指定要做成image的源資料夾.(默认：当前文件夹)

-s, --pagesize=SIZE 节点页大小(默认: 4KiB)

-e, --eraseblock=SIZE 设定擦除块的大小为(默认: 64KiB)

-c, --cleanmarker=SIZE Size of cleanmarker (default 12)

-m, --compr-mode=MODE Select compression mode (default: priortiry)

-x, --disable-compressor=COMPRESSOR\_NAME Disable a compressor

-X, --enable-compressor=COMPRESSOR\_NAME Enable a compressor

-y, --compressor-priority=PRIORITY:COMPRESSOR\_NAME

Set the priority of a compressor

-L, --list-compressors Show the list of the avaiable compressors

-t, --test-compression Call decompress and compare with the original (for test)

-n, --no-cleanmarkers 指明不添加清楚标记（nand flash 有自己的校检块，存放相关的信息。）

如果挂载后会出现类似：CLEANMARKER node found at 0x0042c000 has totlen 0xc != normal 0x0 的警告，则加上-n 就会消失。

-o, --output=FILE 指定輸出image檔案的文件名.(default: stdout)

-l, --little-endian 指定使用小端格式

-b, --big-endian 指定使用大端格式

-D, --devtable=FILE Use the named FILE as a device table file

-f, --faketime Change all file times to '0' for regression testing

-q, --squash Squash permissions and owners making all files be owned byroot

-U, --squash-uids 将所有文件的拥有者设为root用户

-P, --squash-perms Squash permissions on all files

--with-xattr stuff all xattr entries into image

--with-selinux stuff only SELinux Labels into jffs2 image

--with-posix-acl stuff only POSIX ACL entries into jffs2 image

-h, --help 显示这些文字

-v, --verbose Verbose operation

-V, --version 显示版本

-i, --incremental=FILE Parse FILE and generate appendage output for it

**用mkfs.jffs2 命令制作jffs2镜像文件**

下载：

<ftp://sources.redhat.com/pub/jffs2/mkfs.jffs2>

或者使用命令：

apt-get install mtd-utils

命令使用实例：

mkfs.jffs2  -s 0x1000 -e 0x40000 -p 0x500000 -d  rootfs/ -o  jffs2.img

说明：

页大小0x1000   4k  
块大小0x40000  256k  
jffs2分区总空间0x500000即5M  
当然生成的jffs2.img并没有一下子分配5M,还是实际大小0xc0000

mkfs.jffs2使用方法

选项含义(man a mkfs.jffs2)

mkfs.jffs2: Usage: mkfs.jffs2 [OPTIONS]  
Make a JFFS2 file system image from an existing directory tree  
Options:  
-p, --pad[=SIZE]       用16進制來表示所要輸出檔案的大小，也就是root.jffs2的size。  
                                       很重要的是, 為了不浪費flash空間, 這個值最好符合flash driver所規劃的區塊大小。  
                                       如果不足则使用0xff来填充补满。   
-r, -d, --root=DIR      指定要做成image的源資料夾.(默认：当前文件夹)  
-s, --pagesize=SIZE     节点页大小(默认: 4KiB)  
-e, --eraseblock=SIZE   设定擦除块的大小为(默认: 64KiB)  
-c, --cleanmarker=SIZE Size of cleanmarker (default 12)  
-m, --compr-mode=MODE   Select compression mode (default: priortiry)  
-x, --disable-compressor=COMPRESSOR\_NAME  
                          Disable a compressor  
-X, --enable-compressor=COMPRESSOR\_NAME  
                          Enable a compressor  
-y, --compressor-priority=PRIORITY:COMPRESSOR\_NAME  
                          Set the priority of a compressor  
-L, --list-compressors Show the list of the avaiable compressors  
-t, --test-compression Call decompress and compare with the original (for test)  
-n, --no-cleanmarkers   指明不添加清楚标记（nand flash 有自己的校检块，存放相关的信息。）  
                          如果挂载后会出现类似：  
                          CLEANMARKER node found at 0x0042c000 has totlen 0xc != normal 0x0  
                          的警告，则加上-n 就会消失。  
-o, --output=FILE       指定輸出image檔案的文件名.(default: stdout)  
-l, --little-endian     指定使用小端格式  
-b, --big-endian        指定使用大端格式  
-D, --devtable=FILE     Use the named FILE as a device table file  
-f, --faketime          Change all file times to '0' for regression testing  
-q, --squash            Squash permissions and owners making all files be owned by root  
-U, --squash-uids       将所有文件的拥有者设为root用户  
-P, --squash-perms      Squash permissions on all files  
      --with-xattr        stuff all xattr entries into image  
      --with-selinux      stuff only SELinux Labels into jffs2 image  
      --with-posix-acl    stuff only POSIX ACL entries into jffs2 image  
-h, --help              显示这些文字  
-v, --verbose           Verbose operation  
-V, --version           显示版本  
-i, --incremental=FILE Parse FILE and generate appendage output for it

生成的rootfs.img可通过u-boot直接烧写至nand flash：  
              nand erase 0x200000 0x3e00000//擦写  
              tftp 0x30000000 rootfs.img//下载  
              nand write.jffs2 0x200000 $(filesize)//烧入

**（二） 在linux的PC上挂载jffs2根文件系统映像**  
因为jffs2是构建于MTD设备上的文件系统，所以无法通过loop设备来挂载，但是可以通过mtdram设备来挂载。mtdram是在用RAM实现的MTD设备，可以通过mtdblock设备来访问。使用mtdram设备很简单，只要加载mtdram和mtdblock两个内核模块即可。这两个内核模块一般的linux内核发行版都有编译好的，直接用modprobe命令加载。

下面是具体步骤：

(1).加载mtdblock内核模块

modprobe mtdblock

(2).加载mtdram内核模块，将该设备的大小指定为jffs2根文件系统映像的大小，块擦除大小（即flash的块大小）指定为制作该jffs2根文件系统时“-e”参数指定的大小，缺省为64KB。下面两个参数的单位都是KB。

modprobe mtdram total\_size=5120 erase\_size=256

(3).这时将出现MTD设备/dev/mtdblock0，使用dd命令将jffs2根文件系统拷贝到/dev/mtdblock0设备中。

dd if=jffs2.img of=/dev/mtdblock0

(4).将保存了jffs2根文件系统的MTD设备挂载到指定的目录上。

mount -t jffs2 /dev/mtdblock0 /mnt/mtd

这之后就可以到/mnt/mtd目录查看、修改挂载的jffs2根文件系统了，修改后的jffs2根文件系统可以通过dd命令拷贝为一个jffs2的映像文件