

Ubifs 文件系统的制作和启动

UBI 文件系统是 JFFS2 的下一代文件系统，更适合 NAND FLASH。看到一些 AM335x 的用户在应用 UBIFS 时遇到些问题，现在做个简要介绍。

1. 在 U-boot 和 kernel 中使能对 UBIFS 的支持

1.1 U-boot 的 `\include\configs\am335x_evm.h` 中加入如下宏的定义：

```
#define CONFIG_CMD_NAND
#define CONFIG_CMD_UBI
#define CONFIG_CMD_UBIFS
#define CONFIG_RBTREE
#define CONFIG_MTD_DEVICE
#define CONFIG_MTD_PARTITIONS
#define CONFIG_CMD_MTDPARTS
#define CONFIG_LZO
```

1.2 Kernel 中的 menuconfig:

- Enabling UBI support on MTD devices.

```
Device Drivers --->
    Memory Technology Device (MTD) support --->
        Enable UBI - Unsorted block images --->
```

- Enabling UBIFS file-system support.

```
File systems --->
    Miscellaneous filesystems --->
        UBIFS file system support
```

2. 下载和编译 MTD Utilites

2.1 下载 MTD Utilites:

通过 `git://git.infradead.org/mtd-utils.git` 下载最新的 MTD-Utils 源代码。

下载 3 个编译所需的依赖库

Zlib: <http://zlib.net/>

Lzo: <http://www.oberhumer.com/opensource/lzo/download/>

e2fsprogs: <http://e2fsprogs.sourceforge.net/>

在用户目录下建立 `/mtd`, `/mtd/install` 目录，将以上 4 个 source code 包拷贝到 `/mtd` 下，安装如下步骤先编译依赖包：

zlib

```
host$ tar xvf zlib-1.2.5.tar.gz
host$ cd zlib-1.2.5/
host$ ./configure --prefix=~/.mtd/install
host$ make
host$ make install
host$ cd ..
```

lzo

```
host$ tar xvf lzo-2.06.tar.gz
host$ cd lzo-2.06/
host$ ./configure --build=i686-pc-linux --prefix=~/.mtd/install
host$ make
host$ make install
host$ cd ..
```

e2fsprogs

```
host$ tar xvf e2fsprogs-1.42.tar.gz
host$ cd e2fsprogs-1.42/
host$ ./configure --build=i686-pc-linux --prefix=~/.mtd/install
host$ make
host$ make install
host$ cd lib/uuid/
host$ make install
host$ cd ../../../../
```

2.1 编译 MTD Utilites:

在编译 mtd-utils 之前，先修改 makfile 文件：

```
host$ vi Makefile
PREFIX = ~/.mtd/install
ZLIBCPPFLAGS = -I$(PREFIX)/include
LZOCPPFLAGS = -I$(PREFIX)/include
ZLIBLDFLAGS = -L$(PREFIX)/lib
LZOLDLDFLAGS = -L$(PREFIX)/lib
LDFLAGS += $(ZLIBLDFLAGS) $(LZOLDLDFLAGS)
CFLAGS ?= -O2 -g $(ZLIBCPPFLAGS) $(LZOCPPFLAGS)
```

```
host$ vi common.mk
#PREFIX=/usr
```

然后依照如下指令编译：

```
host$ WITHOUT_XATTR=1 make
host$ make install DESTDIR=~/.mtd/install
host$ cd ..
```

制作 ubi 文件系统所用的工具在如下目录下：

```
~/mtd/install/home/<username>/mtd/install/sbin/mkfs.ubifs
~/mtd/install/home/<username>/mtd/install/sbin/ubinize
```

3. 制作 ubifs

Ubifs 的制作需要以下两个命令

①mkfs.ubifs: 制作 UBIFS image

②ubinize: 根据 UBIFS image 制作 ubi.img, 这个 ubi.img 是通过 u-boot 直接烧写在 nand flash 分区上的。

AM335x Linux SDK 里面带有制作好的文件系统, 是.tar.gz 的压缩文件, 可以解压在一个目录下做为 UBI 文件系统内容, 如/home/usr/fs。

GPEVM 板上的 NAND 型号为 MT29F2G08, page size 为 2048B, block size 为 64x2048B=131072B, block count 为 2048。如果制作针对 GPEVM 板的 ubifs, 执行如下两条命令:

```
1. $ mkfs.ubifs -F -q -r /home/usr/fs -m 2048 -e 126976 -c 2047 -o
    ubifs.img
```

参数简介:

-F: 使能"white-space-fixup", 如果是通过 u-boot 烧写需要使能此功能。

-r: 待制作的文件系统目录

-m: NAND FLASH 的最小读写单元, 一般为 page size

-e: LEB size, 对于 AM335x 的 NAND driver, 为 block size-2x(page size)

-c: 文件系统所占用的最大 block 数, 一般小于等于 block count -1

-o: 输出的 ubifs.img 文件

```
2. $ ubinize -o ubi.img -m 2048 -p 128KiB ubinize.cfg
```

参数简介:

-p: block size。

-m: NAND FLASH 的最小读写单元, 一般为 page size

-o: 输出的 ubi.img 文件

ubinize.cfg 为 ubinize 所需要的配置文件, 内容如下:

```
[ubifs]
```

```
mode=ubi
image=ubifs.img
vol_id=0
vol_size=200MiB
vol_type=dynamic
vol_name=rootfs
vol_flags=autoresize
```

4. 烧写 ubifs

可通过 u-boot 命令将生成的 ubi.img（25M）烧写到 NAND FLASH 分区上，如下示例是将 ubi.img 先存储到 SD 卡上，然后通过 u-boot 的 fatload 命令将其拷贝至内存中。

```
u-boot# mw.b 0x82000000 0xFF
u-boot# mmc rescan
u-boot# fatload mmc 0 0x82000000 ubi.img
u-boot# nand erase 0x00780000 0xF880000
u-boot# nand write 0x82000000 0x00780000 0x1E00000
```

5. Linux 启动设置

在 U-boot 下设置启动信息如下：

```
#setenv bootargs 'console=ttyO0,115200n8 noinitrd ip=off mem=256M
rootwait=1 rw ubi.mtd=7,2048 rootfstype=ubifs root=ubi0:rootfs
init=/init'
```

顺利启动后，会在终端显示如下相关信息：

```
COM1:115200baud - Tera Term VT
File Edit Setup Control Window Help
[ 1.239166] omap2-nand driver initializing
[ 1.243804] ONFI flash detected
[ 1.247222] ONFI param page 0 valid
[ 1.250885] NAND device: Manufacturer ID: 0x2c, Chip ID: 0xda (Micron MT29F2G08ABAEHAP)
[ 1.259460] Creating 8 MTD partitions on "omap2-nand.0":
[ 1.265045] 0x000000000000-0x000000002000 : "SPL"
[ 1.271362] 0x000000002000-0x000000004000 : "SPL.backup1"
[ 1.278228] 0x000000004000-0x000000006000 : "SPL.backup2"
[ 1.285156] 0x000000006000-0x000000008000 : "SPL.backup3"
[ 1.292083] 0x000000008000-0x000000026000 : "U-Boot"
[ 1.299255] 0x000000026000-0x000000028000 : "U-Boot Env"
[ 1.306152] 0x000000028000-0x000000078000 : "Kernel"
[ 1.314666] 0x000000078000-0x000010000000 : "File System"
[ 1.424774] OneNAND driver initializing
[ 1.429534] UBI: attaching mtd7 to ubi0
[ 1.433593] UBI: physical eraseblock size: 131072 bytes (128 KiB)
[ 1.440124] UBI: logical eraseblock size: 126976 bytes
[ 1.445770] UBI: smallest flash I/O unit: 2048
[ 1.450683] UBI: VID header offset: 2048 (aligned 2048)
[ 1.456939] UBI: data offset: 4096
[ 3.578552] UBI: max. sequence number: 100
[ 3.598724] UBI: attached mtd7 to ubi0
[ 3.602722] UBI: MTD device name: "File System"
[ 3.608428] UBI: MTD device size: 248 MiB
[ 3.613616] UBI: number of good PEBs: 1988
[ 3.618530] UBI: number of bad PEBs: 0
[ 3.623168] UBI: number of corrupted PEBs: 0
[ 3.627777] UBI: max. allowed volumes: 128
[ 3.632598] UBI: wear-leveling threshold: 4096
[ 3.637512] UBI: number of internal volumes: 1
[ 3.642150] UBI: number of user volumes: 1
[ 3.646789] UBI: available PEBs: 0
[ 3.651428] UBI: total number of reserved PEBs: 1988
[ 3.656585] UBI: number of PEBs reserved for bad PEB handling: 19
[ 3.662963] UBI: max/mean erase counter: 2/0
[ 3.667419] UBI: image sequence number: 396842824
[ 3.672546] UBI: background thread "ubi_bgt0d" started, PID 608
[ 3.679351] CAN device driver interface
[ 3.683380] CAN bus driver for Bosch B-CAN controller 1.0
[ 3.739654] davinci_mdio davinci_mdio.0: davinci mdio revision 1.6
[ 3.746124] davinci_mdio davinci_mdio.0: detected phy mask fffffffe
[ 3.753570] davinci_mdio.0: probed
[ 3.757141] davinci_mdio davinci_mdio.0: phy[0]: device 0:00, driver unknown
[ 3.764831] usbcore: registered new interface driver zd1201
[ 3.770874] usbcore: registered new interface driver cdc_ether
[ 3.777099] usbcore: registered new interface driver cdc_aem
[ 3.783172] usbcore: registered new interface driver dn9601
[ 3.789031] cdc_ncm: 04-Aug-2011
[ 3.792633] usbcore: registered new interface driver cdc_ncm
[ 3.798522] Initializing USB Mass Storage driver...
[ 3.803863] usbcore: registered new interface driver usb-storage
[ 3.810150] USB Mass Storage support registered.
[ 3.815399] mousedev: PS/2 mouse device common for all mice
[ 3.822174] input: matrix-keypad as /devices/platform/matrix-keypad/input/input1
[ 3.831176] input: ti-tsc as /devices/platform/omap/ti_tscadc/tsc/input/input2
[ 3.839813] omap_rtc an33xx-rtc: rtc core: registered an33xx-rtc as rtc0
```