## UBIFS bring up:

UBIFS may be considered as the next generation of the JFFS2 file-system.

UBIFS works on top of UBI volumes and cannot operate on top of MTD devices. In other words, there are 3 subsystems involved:

* MTD subsystem, which provides uniform interface to access flash chips. MTD provides an notion of MTD devices (e.g., /dev/mtd0) which basically represents raw flash;
* UBI subsystem, which is a wear-leveling and volume management system for flash devices; UBI works on top of MTD devices and provides a notion of UBI volumes; UBI volumes are higher level entities than MTD devices and they are devoid of many unpleasant issues MTD devices have (e.g., wearing and bad blocks); see here for more information;
* UBIFS file system, which works on top of UBI volumes.

For more information on UBI, refer <<http://www.linux-mtd.infradead.org/doc/ubi.html>>

For more information on UBIFS, refer <<http://www.linux-mtd.infradead.org/doc/ubifs.html>>

##  Linux kernel configuration:

###  Step1:

 

### Step 2:



### Step 3:



### Step 4:



## User Space Tools:

For more information on MTD, refer <<http://www.linux-mtd.infradead.org/doc/general.html>>

For more information on UBI, refer <<http://www.linux-mtd.infradead.org/doc/ubi.html>>

For more information on UBIFS, refer <<http://www.linux-mtd.infradead.org/doc/ubifs.html>>

UBI user-space tools, as well as other MTD user-space tools, are available from the the following git repository: git://git.infradead.org/mtd-utils.git

The repository contains the following UBI tools:

 ubinfo - provides information about UBI devices and volumes found in the system;

 ubiattach - attaches MTD devices (which describe raw flash) to UBI and creates corresponding UBI devices;

 ubidetach - detaches MTD devices from UBI devices (the opposite to what ubiattach does);

 ubimkvol - creates UBI volumes on UBI devices;

 ubirmvol - removes UBI volumes from UBI devices;

 ubiupdatevol - updates UBI volumes; this tool uses the UBI volume update feature which leaves the volume in

 "corrupted" state if the update was interrupted; additionally, this tool may be used to wipe

 out UBI volumes;

 ubicrc32 - calculates CRC-32 checksum of a file with the same initial seed as UBI would use;

 ubinize - generates UBI images;

 ubiformat - formats empty flash, erases flash and preserves erase counters, flashes UBI images to MTD devices;

 mtdinfo - reports information about MTD devices found in the system.

All UBI tools support "-h" option and print sufficient usage information

### Compiling UBIFS Tools

The MTD and UBI user-space tools are available from the following git repository:

git://git.infradead.org/mtd-utils.git

Dependencies

The '**mtd-utility'** requires zlib, lzo and uuid (from e2fsprogs) libraries. The former two are used for compressing the data, and the latter one is used for generating universally unique ID number for the file-system.

1. zlib
2. lzo
3. e2fsprogs

### Sources

**zlib**
Download zlib from <http://zlib.net/>

**lzo**
Download from <http://www.oberhumer.com/opensource/lzo/download/> .

**e2fsprogs**
Download e2fsprogs from <http://e2fsprogs.sourceforge.net/>

### zlib

tar -xzf zlib-1.2.7.tar.gz

cd zlib-1.2.7/

./*configure --prefix*=$INSTALLDIR/install

host$ vi Makefile

 CC=arm-xilinx-linux-gnueabi-gcc

 LDSHARED=arm-xilinx-linux-gnueabi-gcc -shared -Wl,-soname,libz.so.1,--version-script,zlib.map

 CPP=arm-xilinx-linux-gnueabi-gcc -E

 AR=arm-xilinx-linux-gnueabi-ar

 RANLIB=arm-xilinx-linux-gnueabi-ranlib

Make

Make install

### Lzo

tar -xzf lzo-2.06.tar.gz

cd lzo-2.06

export CC=arm-xilinx-linux-gnueabi-gcc

root@ lzo-2.06# ./*configure --prefix*=$INSTALLDIR/install --host=arm-xilinx-linux-gnueabi

make

make install

### e2fsprogs

tar -xzf e2fsprogs-1.42.tar.gz

cd e2fsprogs-1.42

export CC=arm-xilinx-linux-gnueabi-gcc

root@ /lzo-2.06# ./configure --prefix=$INSTALLDIR/install --host=arm-xilinx-linux-gnueabi

make

make install

cd lib/uuid/

make install

### mtd-utils

tar -xvf mtd-utils-1.4.9.tar.bz2

MTD-Utils don't have a configure script, so we have to edit Makefile again. Depending on the version of MTD Utils, make sure head of top level Makefile has:

cd mtd-utils-1.4.9

 root@/mtd-utils-1.4.9$ vi Makefile

PREFIX = $INSTALLDIR/install

CROSS=arm-none-linux-gnueabi-

ZLIBCPPFLAGS = -I$(PREFIX)/include

ZLIBLDFLAGS = -L$(PREFIX)/lib

LZOCPPFLAGS = -I$(PREFIX)/include

LZOLDFLAGS = -L$(PREFIX)/lib

LDFLAGS += $(ZLIBLDFLAGS) $(LZOLDFLAGS)

CFLAGS ?= -O2 -g $(ZLIBCPPFLAGS) $(LZOCPPFLAGS)

Edit the common.mk file and comment out the PREFIX=/usr line

 root@/mtd-utils-1.4.9$ vi common.mk

 #PREFIX=/usr

 root@/mtd-utils-1.4.9$ WITHOUT\_XATTR=1 make

 root@/mtd-utils-1.4.9$ make install DESTDIR=$INSTALLDIR/install

## How to Run:

**Erase your flash partition while preserving your erase counters:**

ubiformat /dev/mtdx

here ‘x’ is the mtd partition number

**Attach UBI to one (of several) of the MTD partitions:**

ubiattach /dev/ubi\_ctrl ­m x

**Volume creation with ubimkvol:**

ubimkvol /dev/ubi0 ­N test ­s 116MiB

(or)

ubimkvol /dev/ubi0 ­N test ­m (max available size)

**When a UBI volume is created, creating an empty UBIFS:**

***filesystem is just a matter of mounting it***

mount ­t ubifs ubi0:test /mnt/flash

### Benchmark Tools

1. Bonnie++
2. Iozone