Kernel-Based FTL

Jinyong Ha (jinyongha@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Contents

- uart_printf()
- Project #1 Test Setting
- Kernel-based FTL
- Q & A
## Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/11 (Mon)</td>
<td>Intro. to the Jasmine OpenSSD Platform</td>
</tr>
<tr>
<td>3/18 (Mon)</td>
<td>Dummy FTL</td>
</tr>
<tr>
<td>3/25 (Mon)</td>
<td>Tutorial FTL</td>
</tr>
<tr>
<td>4/1 (Mon)</td>
<td>Greedy FTL</td>
</tr>
<tr>
<td>4/8 (Mon)</td>
<td>Reliability Issues</td>
</tr>
<tr>
<td>4/15 (Mon)</td>
<td>Project #1</td>
</tr>
<tr>
<td>4/29 (Mon)</td>
<td>Project #1 Q&amp;A, Kernel-Based FTL</td>
</tr>
<tr>
<td>5/5 (Mon)</td>
<td>Project #2 Suggestions</td>
</tr>
<tr>
<td>5/27 (Mon)</td>
<td>Project #2 Progress Report</td>
</tr>
<tr>
<td>6/10 (Mon)</td>
<td>Project #2 Presentation</td>
</tr>
</tbody>
</table>
uart_printf()

- include/jasmine.h
uart_printf()

- Similar usage with printf()

```c
static void format(void)
{
    UINT32 bank, vblock, vcount_val;
    ASSERT(NUM_MISC_META_SECT > 0);
    ASSERT(NUM_VCOUNT_SECT > 0);

    uart_printf("Total FTL DRAM metadata size: %d KB", DRAM_BYTES_OTHER / 1024);
    uart_printf("VBLKS_PER_BANK: %d", VBLKS_PER_BANK);
    uart_printf("LBLKS_PER_BANK: %d", NUM_LPAGES / PAGES_PER_BLK / NUM_BANKS);
    uart_printf("META_BLKS_PER_BANK: %d", META_BLKS_PER_BANK);

    //-----------------------------
    // initialize DRAM metadata
    //-----------------------------
    mem_set_dram(PAGE_MAP_ADDR, NULL, PAGE_MAP_BYTES);
    mem_set_dram(VCOUNT_ADDR, NULL, VCOUNT_BYTES);
}
uart_printf()

- Hyper terminal setting
Project #1 Test Setting

- Download test file and unzip

- Locate ftl_test.c to ./locglock_ftl
Project #1 Test Setting

- Makefile

```bash
MKDIR = looblock
PREFIX = arm-none-eabi-
CC = $(PREFIX)gcc
AS = $(PREFIX)as
LD = $(PREFIX)ld
OBJCOPY = $(PREFIX)objcopy
RM = \del

INCLUDES = -I../include -I../rtl -I../sata -I../target_sow
CFLAGS = -Werror -Wall -Iinc -Iinc -Iabs -Iinc -Iinc -Iinc -Iinc -Iinc -Iinc
ASFLAGS = -Iinc
LDFLAGS = -Iinc -Iinc -Iinc -Iinc
LIBS = -lgcc
YPATH = ../rtl
SNODE = my_test.c filesystem.c merge.c
INCLUDED = filesystem/initial.c
TOTAL = $(SNODE) $(INCLUDED)
TARGET = test
TARGETLF = test.out
TARGETBIN = test.out

$(TARGETBIN): $(TARGETLF)
	$(CC) $(CFLAGS) $(CFILES) $(OBJS) $(LIBS) $(OUT)

OUCH = $(CC) $(CFLAGS) $(CFILES) $(OBJS) $(LIBS) $(OUT)

.PHONY: all clean
all: $(TARGETBIN)

clean:
	$(RM) $(OUT) $(TARGETLF) $(TARGETBIN)

include $(OBJS)
```
Project #1 Test Setting

- Include/jasmine.h

```c
#ifndef JASMINE_H
#define JASMINE_H

#define FLASH_TYPE       K9LCG08U1M
#define DRAM_SIZE        65075200
#define BANK_BMP         0x00330033
#define CLOCK_SPEED      17500000

#define OPTION_2_PLANE   1 // 1 = 2-plane mode, 0 = 1-plane mode
#define OPTION_ENABLE_ASSERT 0 // 1 = enable ASSERT() for debugging, 0 = disable ASSERT()
#define OPTION_FTL_TEST   1 // 1 = FTL test without SATA communication, 0 = normal
#define OPTION_UART_DEBUG 1 // 1 = enable UART message output, 0 = disable
#define OPTION_SLOW_SATA  1 // 1 = SATA 1.5Gbps, 0 = 3Gbps
#define OPTION_SUPPORT_NQ 0 // 1 = support SATA NCQ (=FPDMA) for AHCI hosts, 0 = support only DMA mode
#define OPTION_REduced_CAPACITY 0 // reduce the number of blocks per bank for testing purpose

#define CHN_WIDTH         2 // 2 = 16bit IO
#define NUM_CHNLs_MAX     4
#define BANKS_PER_CHN_MAX 8
#define NUM_BANKs_MAX     32
```
Project #1 Test Setting

- **Logblock_ftl/ftl.h**
  - Assign DRAM space for testing
  - Require 4MB

```c
#define FTL_TEST_ADDR (YOUR_LAST_DRAM_ADDR + YOUR_LAST_DRAM_BYTES)
#define FTL_TEST_BYTES (4 * 1024 * 1024)
```

- Set number of log blocks per bank as 3
  - As specification, your FTL can modify it easily
Project #1 Test Setting

- Target of test
  - Correct operation of log block FTL
  - Except bonus work (power off recovery, runtime bad block remapping)
Kernel-Based FTL

- FTL attached on Linux kernel
Kernel-Based FTL

• Advantage of kernel-based FTL
  - Use rich resource of host
  - Support inferior resource of jasmine board
  - Powerful processor (Compress, deduplication ...)
  - Large memory space (big-sized metadata)

• Useful tool for progressing Project #2
  - More detail is explained in provided documents
Any Questions?