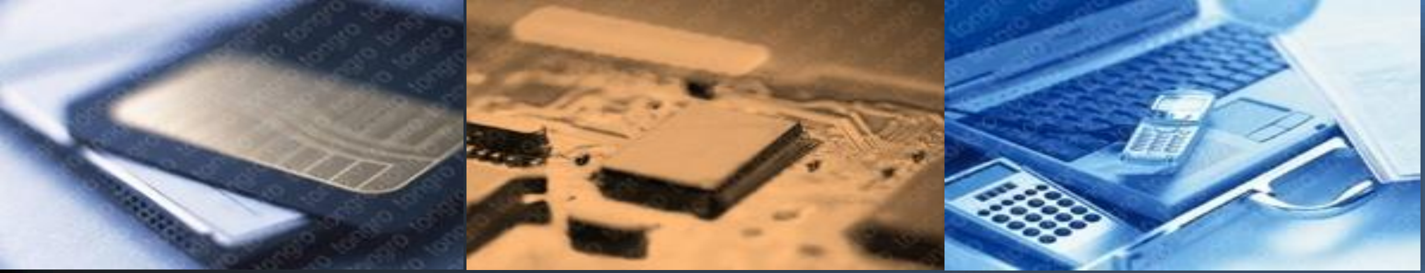
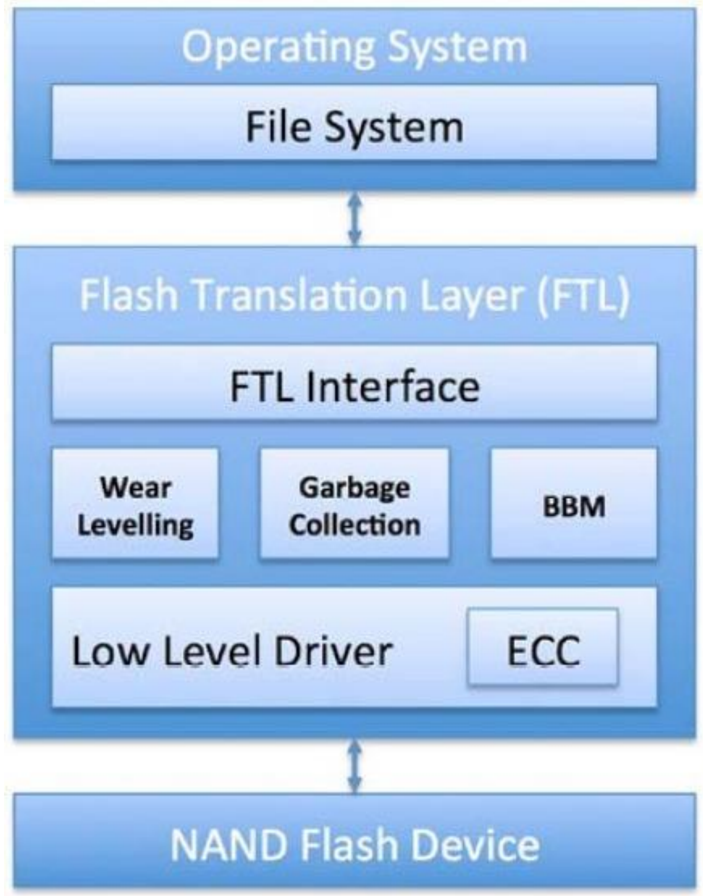


D-Tux 연구주제
- FlashSim -

Choi GunHee
choi_gunhee@dankook.ac.kr

FlashSim

- FTL (Flash Translation Layer)



NAND Flash

→ NAND 수명 존재

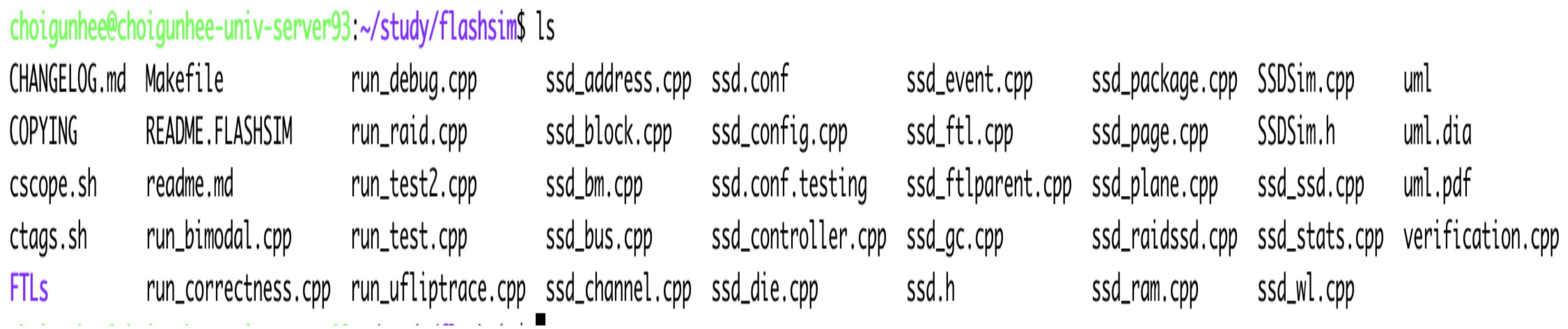
→ But, Erase before Write

→ Write을 줄이자

→ FTL 생성

FlashSim

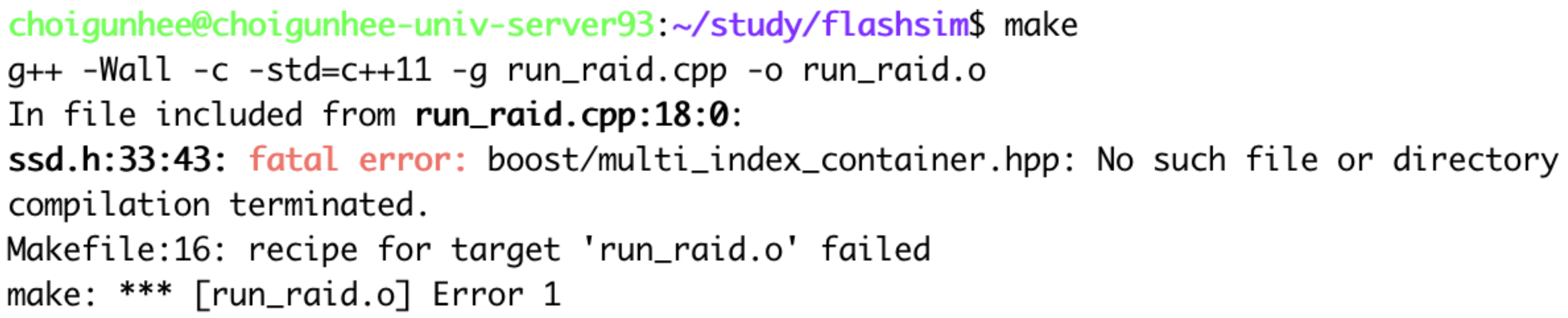
- FlashSim 설치
  - 링크 : https://github.com/MatiasBjorling/flashsim
  - git clone https://github.com/MatiasBjorling/flashsim



```
choigunhee@choigunhee-univ-server93:~/study/flashsim$ ls
CHANGELOG.md  Makefile             run_debug.cpp      ssd_address.cpp  ssd.conf            ssd_event.cpp      ssd_package.cpp  SSDSim.cpp     uml
COPYING       README.FLASHSIM      run_raid.cpp       ssd_block.cpp    ssd_config.cpp      ssd_ftl.cpp        ssd_page.cpp     SSDSim.h       uml.dia
cscope.sh     readme.md            run_test2.cpp      ssd_bm.cpp       ssd.conf.testing    ssd_ftlparent.cpp  ssd_plane.cpp    ssd_ssd.cpp    uml.pdf
ctags.sh      run_bimodal.cpp      run_test.cpp       ssd_bus.cpp      ssd_controller.cpp  ssd_gc.cpp         ssd_raidssd.cpp  ssd_stats.cpp  verification.cpp
FTLs          run_correctness.cpp  run_ufliptrace.cpp ssd_channel.cpp  ssd_die.cpp         ssd.h              ssd_ram.cpp      ssd_wl.cpp
```

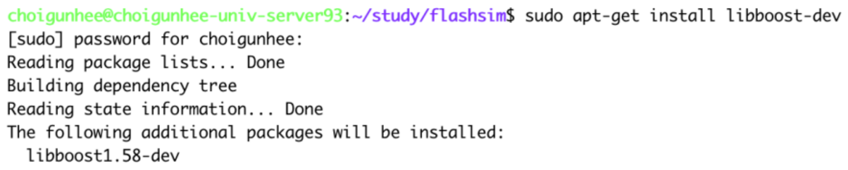
FlashSim

- FlashSim 설치
  - make

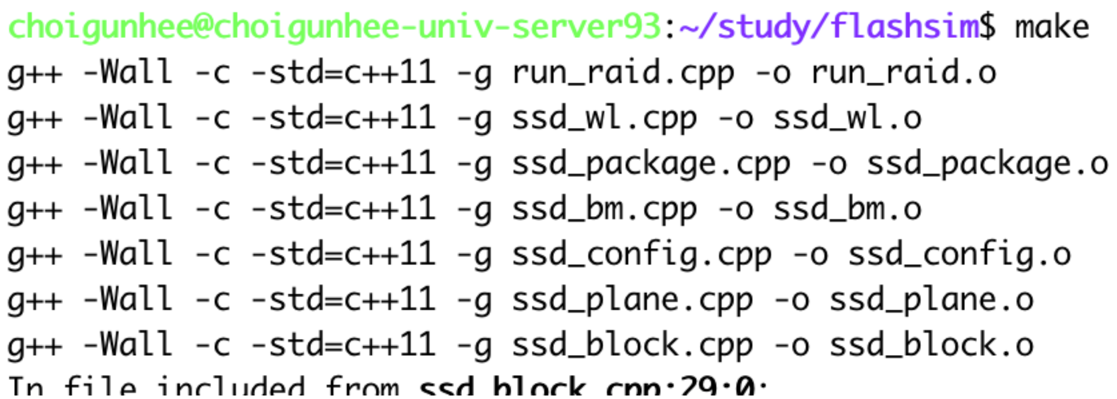


```
choigunhee@choigunhee-univ-server93:~/study/flashsim$ make
g++ -Wall -c -std=c++11 -g run_raid.cpp -o run_raid.o
In file included from run_raid.cpp:18:0:
ssd.h:33:43: fatal error: boost/multi_index_container.hpp: No such file or directory
compilation terminated.
Makefile:16: recipe for target 'run_raid.o' failed
make: *** [run_raid.o] Error 1
```

  - sudo apt-get install libboost-dev



```
choigunhee@choigunhee-univ-server93:~/study/flashsim$ sudo apt-get install libboost-dev
[sudo] password for choigunhee:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libboost1.58-dev
```



```
choigunhee@choigunhee-univ-server93:~/study/flashsim$ make
g++ -Wall -c -std=c++11 -g run_raid.cpp -o run_raid.o
g++ -Wall -c -std=c++11 -g ssd_wl.cpp -o ssd_wl.o
g++ -Wall -c -std=c++11 -g ssd_package.cpp -o ssd_package.o
g++ -Wall -c -std=c++11 -g ssd_bm.cpp -o ssd_bm.o
g++ -Wall -c -std=c++11 -g ssd_config.cpp -o ssd_config.o
g++ -Wall -c -std=c++11 -g ssd_plane.cpp -o ssd_plane.o
g++ -Wall -c -std=c++11 -g ssd_block.cpp -o ssd_block.o
In file included from ssd_block.cpp:29:0:
```

FlashSim

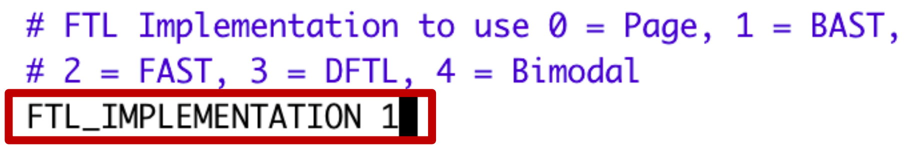
- FlashSim 실행
  - ssd.conf

```
# Ssd class:
#    number of Packages per Ssd (size)
SSD_SIZE 1

# Package class:
#    number of Dies per Package (size)
PACKAGE_SIZE 2

# Die class:
#    number of Planes per Die (size)
DIE_SIZE 2

# Plane class:
#    number of Blocks per Plane (size)
#    delay for reading from plane register
#    delay for writing to plane register
#    delay for merging is based on read, write, reg_read, reg_write
#       and does not need to be explicitly defined
PLANE_SIZE 256
PLANE_REG_READ_DELAY 0.01
PLANE_REG_WRITE_DELAY 0.01

# Block class:
#    number of Pages per Block (size)
#    number of erases in lifetime of block
#    delay for erasing block
BLOCK_SIZE 64
BLOCK_ERASES 100000
BLOCK_ERASE_DELAY 2000
```

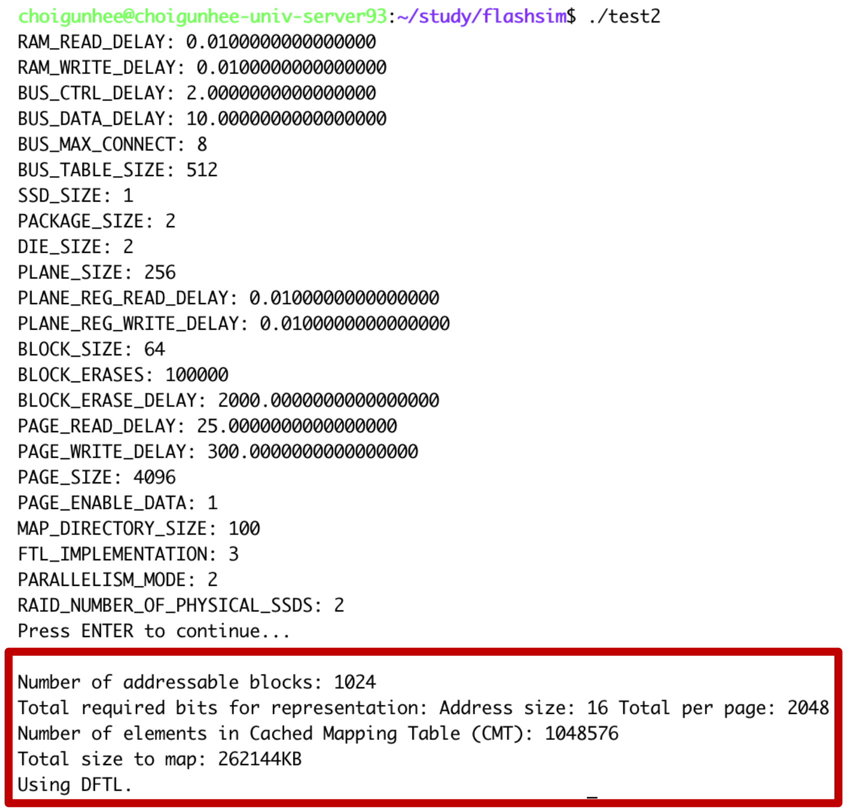
```
# Page class:
#    delay for Page reads
#    delay for Page writes
# -- A 64bit kernel is required if data pages are used. --
#        Allocate actual data for pages
#    Size of pages (in bytes)
PAGE_READ_DELAY 25
PAGE_WRITE_DELAY 300
PAGE_ENABLE_DATA 1

# MAPPING
# Specify reservation of
# blocks for mapping purposes.
MAP_DIRECTORY_SIZE 100

# FTL Implementation to use 0 = Page, 1 = BAST,
# 2 = FAST, 3 = DFTL, 4 = Bimodal
FTL_IMPLEMENTATION 3

# LOG Block limit for BAST
BAST_LOG_BLOCK_LIMIT 1024

# LOG Block limit for FAST
FAST_LOG_BLOCK_LIMIT 1024
```

FlashSim

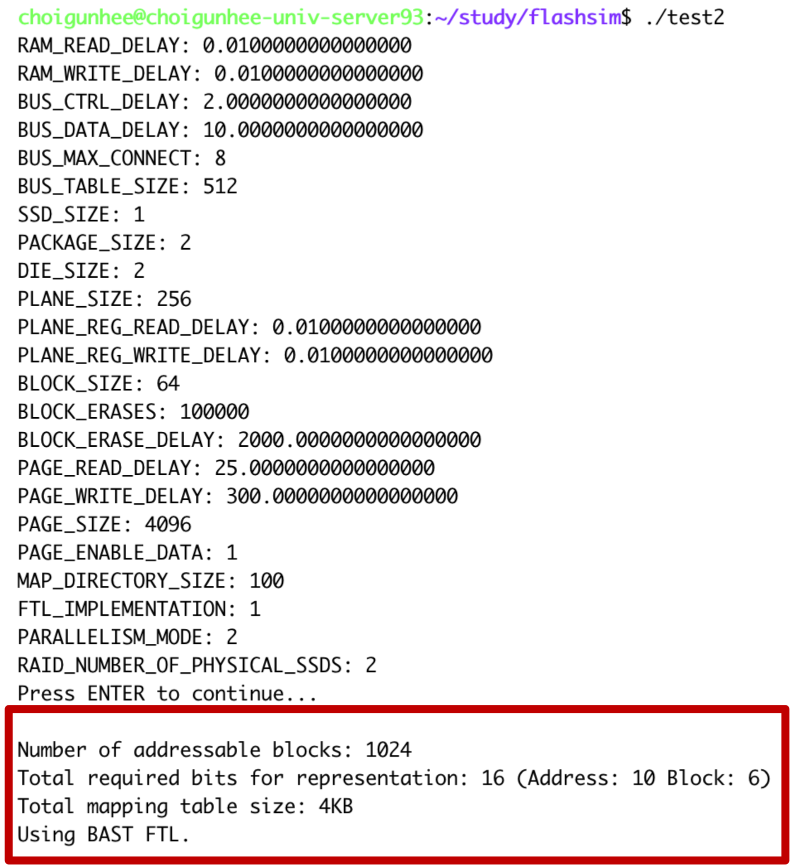
- FlashSim 실행
  - ssd.conf
  - test2 실행



```
# FTL Implementation to use 0 = Page, 1 = BAST,
# 2 = FAST, 3 = DFTL, 4 = Bimodal
FTL_IMPLEMENTATION 1
```



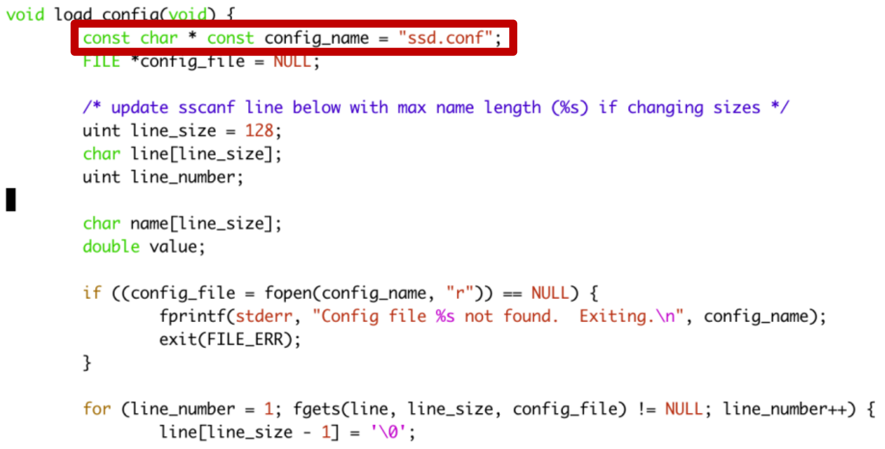
```
choigunhee@choigunhee-univ-server93:~/study/flashsim$ ./test2
RAM_READ_DELAY: 0.0100000000000000
RAM_WRITE_DELAY: 0.0100000000000000
BUS_CTRL_DELAY: 2.0000000000000000
BUS_DATA_DELAY: 10.0000000000000000
BUS_MAX_CONNECT: 8
BUS_TABLE_SIZE: 512
SSD_SIZE: 1
PACKAGE_SIZE: 2
DIE_SIZE: 2
PLANE_SIZE: 256
PLANE_REG_READ_DELAY: 0.0100000000000000
PLANE_REG_WRITE_DELAY: 0.0100000000000000
BLOCK_SIZE: 64
BLOCK_ERASES: 100000
BLOCK_ERASE_DELAY: 2000.0000000000000000
PAGE_READ_DELAY: 25.0000000000000000
PAGE_WRITE_DELAY: 300.0000000000000000
PAGE_SIZE: 4096
PAGE_ENABLE_DATA: 1
MAP_DIRECTORY_SIZE: 100
FTL_IMPLEMENTATION: 3
PARALLELISM_MODE: 2
RAID_NUMBER_OF_PHYSICAL_SSDS: 2
Press ENTER to continue...

Number of addressable blocks: 1024
Total required bits for representation: Address size: 16 Total per page: 2048
Number of elements in Cached Mapping Table (CMT): 1048576
Total size to map: 262144KB
Using DFTL.
```



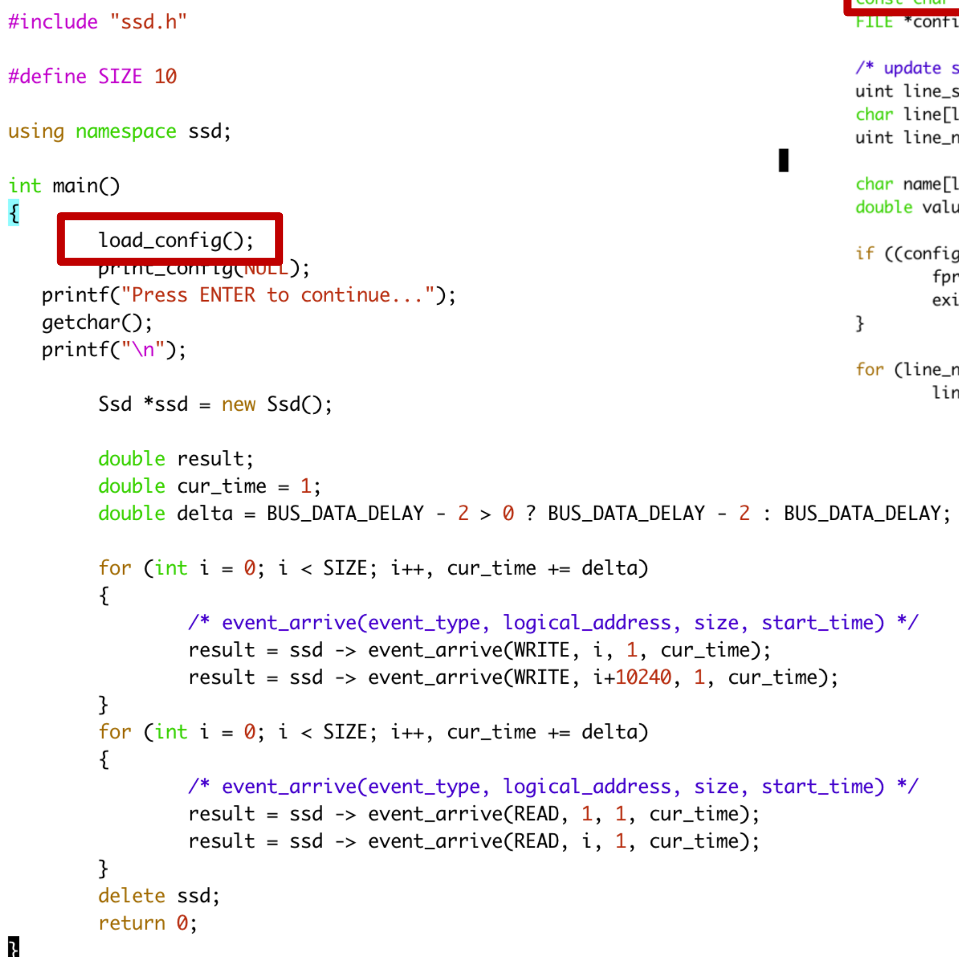
```
choigunhee@choigunhee-univ-server93:~/study/flashsim$ ./test2
RAM_READ_DELAY: 0.0100000000000000
RAM_WRITE_DELAY: 0.0100000000000000
BUS_CTRL_DELAY: 2.0000000000000000
BUS_DATA_DELAY: 10.0000000000000000
BUS_MAX_CONNECT: 8
BUS_TABLE_SIZE: 512
SSD_SIZE: 1
PACKAGE_SIZE: 2
DIE_SIZE: 2
PLANE_SIZE: 256
PLANE_REG_READ_DELAY: 0.0100000000000000
PLANE_REG_WRITE_DELAY: 0.0100000000000000
BLOCK_SIZE: 64
BLOCK_ERASES: 100000
BLOCK_ERASE_DELAY: 2000.0000000000000000
PAGE_READ_DELAY: 25.0000000000000000
PAGE_WRITE_DELAY: 300.0000000000000000
PAGE_SIZE: 4096
PAGE_ENABLE_DATA: 1
MAP_DIRECTORY_SIZE: 100
FTL_IMPLEMENTATION: 1
PARALLELISM_MODE: 2
RAID_NUMBER_OF_PHYSICAL_SSDS: 2
Press ENTER to continue...

Number of addressable blocks: 1024
Total required bits for representation: 16 (Address: 10 Block: 6)
Total mapping table size: 4KB
Using BAST FTL.
```

FlashSim

- FlashSim 구조
  - test2.cpp, ssd_config.cpp



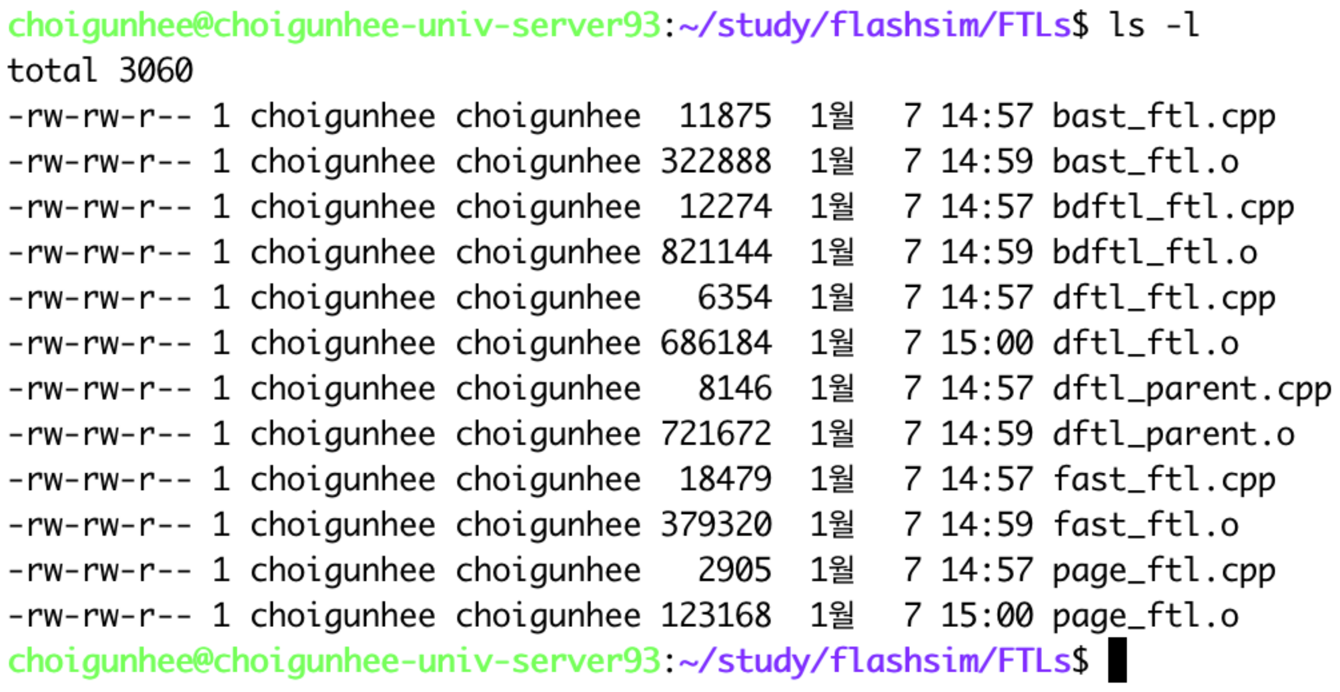
```
void load_config(void) {
        const char * const config_name = "ssd.conf";
        FILE *config_file = NULL;

        /* update sscanf line below with max name length (%s) if changing sizes */
        uint line_size = 128;
        char line[line_size];
        uint line_number;

        char name[line_size];
        double value;

        if ((config_file = fopen(config_name, "r")) == NULL) {
                fprintf(stderr, "Config file %s not found.  Exiting.\n", config_name);
                exit(FILE_ERR);
        }

        for (line_number = 1; fgets(line, line_size, config_file) != NULL; line_number++) {
                line[line_size - 1] = '\0';
```



```
#include "ssd.h"

#define SIZE 10

using namespace ssd;

int main()
{
        load_config();
        print_config(NULL);
   printf("Press ENTER to continue...");
   getchar();
   printf("\n");

        Ssd *ssd = new Ssd();

        double result;
        double cur_time = 1;
        double delta = BUS_DATA_DELAY - 2 > 0 ? BUS_DATA_DELAY - 2 : BUS_DATA_DELAY;

        for (int i = 0; i < SIZE; i++, cur_time += delta)
        {
                /* event_arrive(event_type, logical_address, size, start_time) */
                result = ssd -> event_arrive(WRITE, i, 1, cur_time);
                result = ssd -> event_arrive(WRITE, i+10240, 1, cur_time);
        }
        for (int i = 0; i < SIZE; i++, cur_time += delta)
        {
                /* event_arrive(event_type, logical_address, size, start_time) */
                result = ssd -> event_arrive(READ, 1, 1, cur_time);
                result = ssd -> event_arrive(READ, i, 1, cur_time);
        }
        delete ssd;
        return 0;
}
```

FlashSim

- FlashSim 구조
  - FTLs



```
choigunhee@choigunhee-univ-server93:~/study/flashsim/FTLs$ ls -l
total 3060
-rw-rw-r-- 1 choigunhee choigunhee  11875  1월   7 14:57 bast_ftl.cpp
-rw-rw-r-- 1 choigunhee choigunhee 322888  1월   7 14:59 bast_ftl.o
-rw-rw-r-- 1 choigunhee choigunhee  12274  1월   7 14:57 bdftl_ftl.cpp
-rw-rw-r-- 1 choigunhee choigunhee 821144  1월   7 14:59 bdftl_ftl.o
-rw-rw-r-- 1 choigunhee choigunhee   6354  1월   7 14:57 dftl_ftl.cpp
-rw-rw-r-- 1 choigunhee choigunhee 686184  1월   7 15:00 dftl_ftl.o
-rw-rw-r-- 1 choigunhee choigunhee   8146  1월   7 14:57 dftl_parent.cpp
-rw-rw-r-- 1 choigunhee choigunhee 721672  1월   7 14:59 dftl_parent.o
-rw-rw-r-- 1 choigunhee choigunhee  18479  1월   7 14:57 fast_ftl.cpp
-rw-rw-r-- 1 choigunhee choigunhee 379320  1월   7 14:59 fast_ftl.o
-rw-rw-r-- 1 choigunhee choigunhee   2905  1월   7 14:57 page_ftl.cpp
-rw-rw-r-- 1 choigunhee choigunhee 123168  1월   7 15:00 page_ftl.o
choigunhee@choigunhee-univ-server93:~/study/flashsim/FTLs$ 
```

FlashSim

- FlashSim 구조
  - FTL

```
#include "ssd.h"

using namespace ssd;

// Initialization of the block layer.
Block_manager *Block_manager::inst = NULL;

FtlParent::FtlParent(Controller &controller) : controller(controller)
{
        Block_manager::instance_initialize(this);

        printf("Number of addressable blocks: %u\n", NUMBER_OF_ADDRESSABLE_BLOCKS);

}
```

```
FtlImpl_Bast::FtlImpl_Bast(Controller &controller):
        FtlParent(controller)
{

        // Detect required number of bits for logical address size
        addressSize = log(NUMBER_OF_ADDRESSABLE_BLOCKS)/log(2);
        addressShift = log(BLOCK_SIZE)/log(2);

        // Find required number of bits for block size
        printf("Total required bits for representation: %i (Address: %i Block: %i) \n", addressSize + addressShift, addressSize, addressShift);

        // Initialise block mapping table.
        data_list = new long[NUMBER_OF_ADDRESSABLE_BLOCKS];

        for (uint i=0;i<NUMBER_OF_ADDRESSABLE_BLOCKS;i++)
                data_list[i] = -1;

        printf("Total mapping table size: %luKB\n", NUMBER_OF_ADDRESSABLE_BLOCKS * sizeof(uint) / 1024);
        printf("Using BAST FTL.\n");
}
```

FlashSim

- FlashSim 관련자료
  - FlashSim 논문 : https://ieeexplore.ieee.org/document/5283998/authors#authors
  - BAST FTL : https://ieeexplore.ieee.org/document/1010143
  - DFTL : http://www.cse.psu.edu/~buu1/papers/ps/dftl-asplos09.pdf
  - Fast FTL : http://csl.skku.edu/uploads/ICE3028S11/fast-tecs07.pdf