Project #1: NAND Simulator

Dong-Yun Lee(dongyun.lee@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Project 1

- Develop a NAND simulator
  - Simulate NAND flash memory using host DRAM
  - 4B for data / 4B for spare
  - Functions to implement:
    - nand_init()
    - nand_read()
    - nand_write()
    - nand_erase()
    - nand_blkdump()
  - The skeleton code is available at course homepage
nand_init()\

- nand_init(nblks, npages)
  - Description
    - Initialize your own NAND flash memory using DRAM
    - Initial state of the flash memory is ‘all-blks-erased’
    - If success, print initialized information of the flash memory
    - If not, print appropriate error message (reason for the error)
  - Argument
    - nblks : the total # of blocks (should be > 0)
    - npages : # of pages per block (should be > 0)
  - Return value
    - Return 0 on success
    - Return -1 on errors
nand_write()
nand_read()

- **nand_read**(blk, page, data, spare)
  - **Description**
    - Read ‘data’ and ‘spare’ from the memory pointed by ‘blk’ and ‘page’
    - If success, print read data and spare
    - If not, print appropriate error message (reason for the error)
  - **Argument**
    - blk, page : address of the flash memory
    - data, spare : data to load
  - **Return value**
    - Return 0 on success
    - Return -1 on errors
nand_erase()

• nand_erase(blk)
  • Description
    - Erase the NAND memory block ‘blk’
    - If success, print appropriate message with ‘blk’
    - If not, print appropriate error message (reason for the error)
  • Argument
    - blk : Address of the NAND flash memory
  • Return value
    - Return 0 on success
    - Return -1 on errors
nand_blkdump()

- **nand_blkdump**(blk)
  - **Description**
    - Dump the contents of the flash memory block ‘blk’ (for debug)
    - If success, print appropriate message with ‘blk’
    - If not, print appropriate error message (reason for the error)
  - **Argument**
    - blk : Address of the NAND flash memory
  - **Return value**
    - Return 0 on success
    - Return -1 on errors
Sample Output
Miscellaneous

▪ Recommended environment : Linux (Ubuntu is ok!)
  • You can do it in Windows, but be sure that your work also runs in Linux (I’ll score all the works only in Linux)

▪ Personal Project

▪ Submit to the e-mail (dongyun.lee@csl.skku.edu)
  • Your submission status will be noticed in course homepage

▪ Late penalty : -10 % / day
Any Questions?