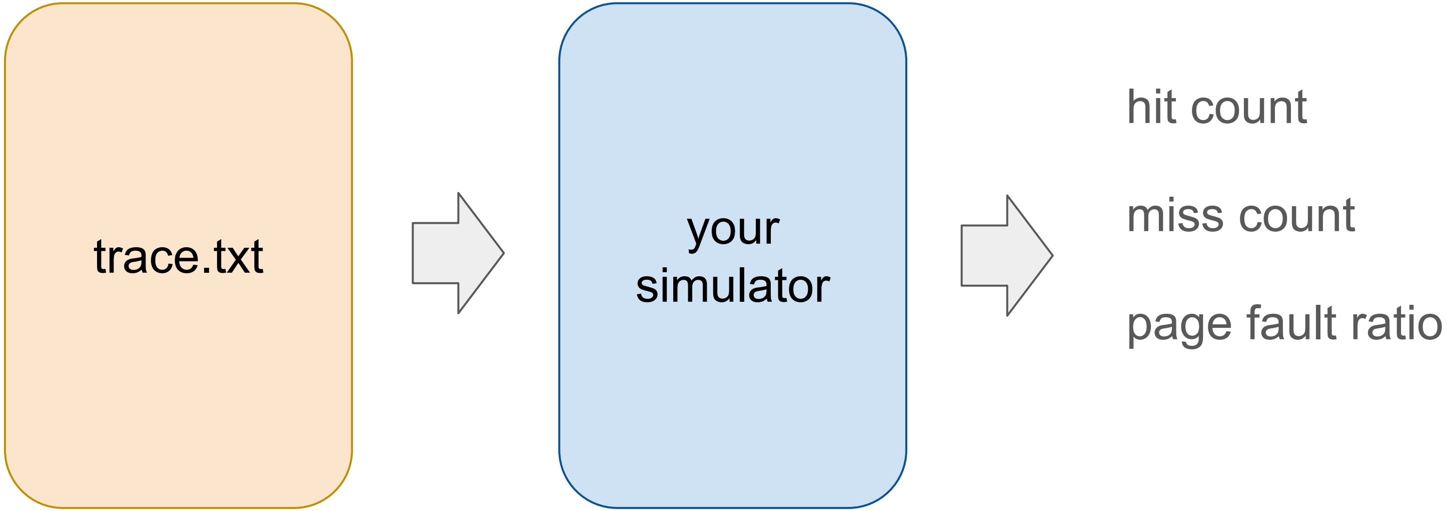
OS HW3

Page Cache Simulation: FIFO and LRU

Simulation



trace.txt

your simulator

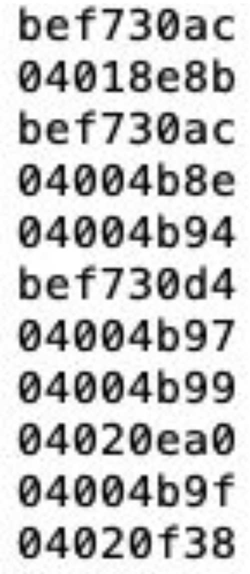
hit count

miss count

page fault ratio

Trace File Format(trace.txt)

memory access:



```
bef730ac
04018e8b
bef730ac
04004b8e
04004b94
bef730d4
04004b97
04004b99
04020ea0
04004b9f
04020f38
```

- page reference pattern

page size: 4 KB

040011a0 → 04001

040011a2 → 04001

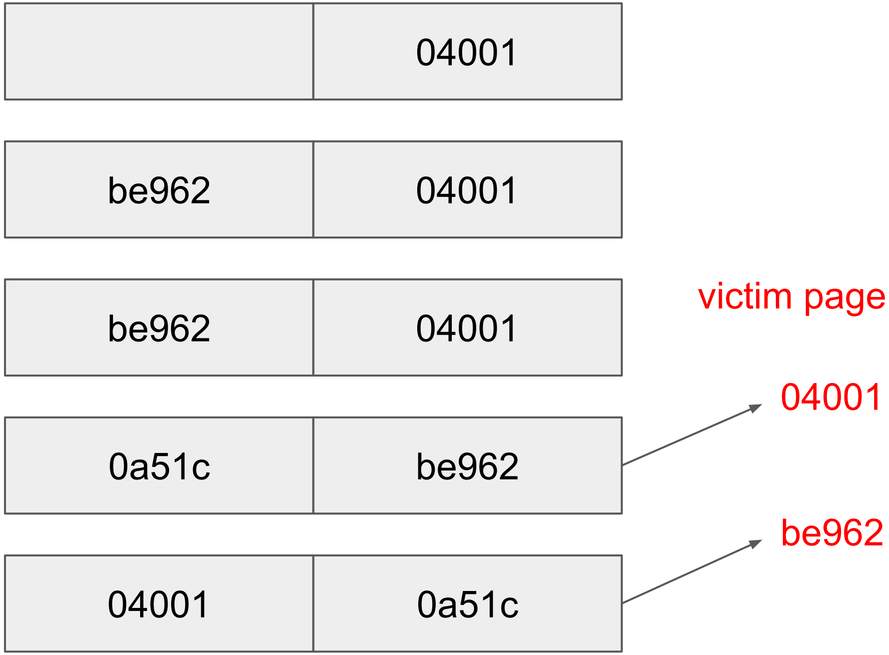
bef7311c → bef73

04004b80 → 04004

trace data was modified from http://valgrind.org

Page Replacement(FIFO)

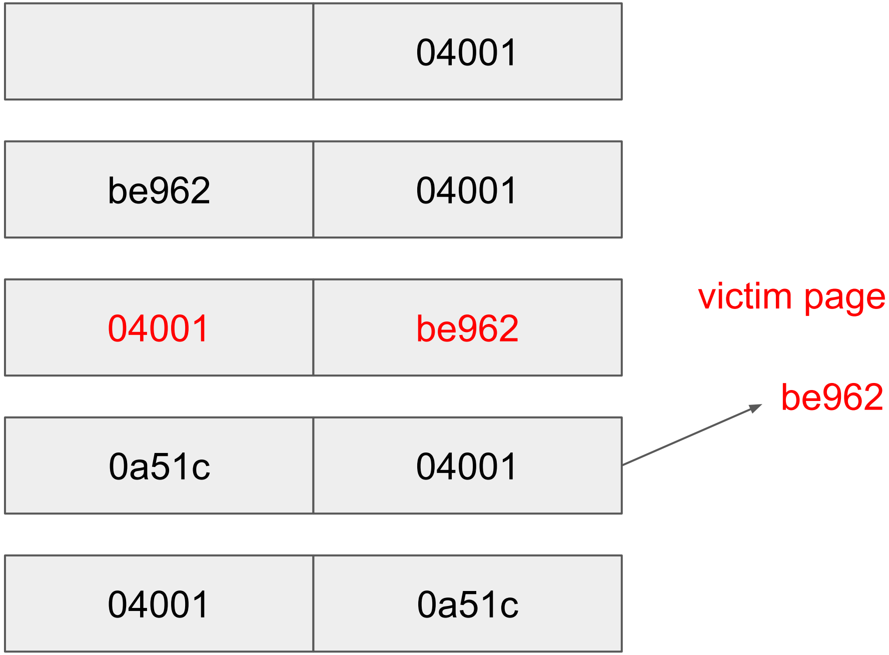
- Victim selection: the oldest page
- Example: Frame #=2



			victim page
04001 (miss)		04001	
be962 (miss)	be962	04001	
04001 (hit)	be962	04001	
0a51c (miss)	0a51c	be962	04001
04001 (miss)	04001	0a51c	be962

Page Replacement(LRU)

- Victim selection: The least recently used page
- Example: Frame #=2

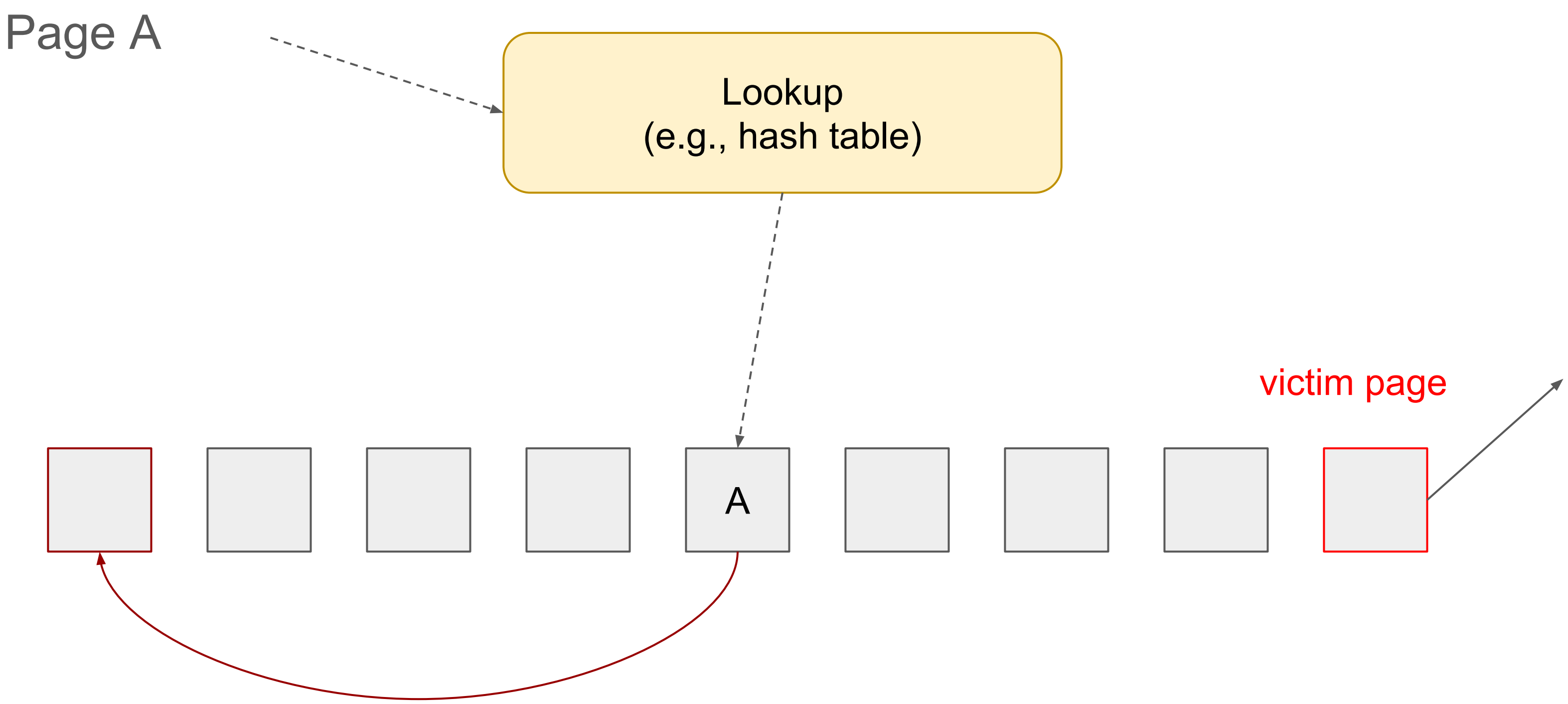


04001 (miss)		04001
be962 (miss)	be962	04001
04001 (hit)	04001	be962
0a51c (miss)	0a51c	04001
04001 (hit)	04001	0a51c

victim page

be962

Simulator Structure (LRU)



Page Cache Operations

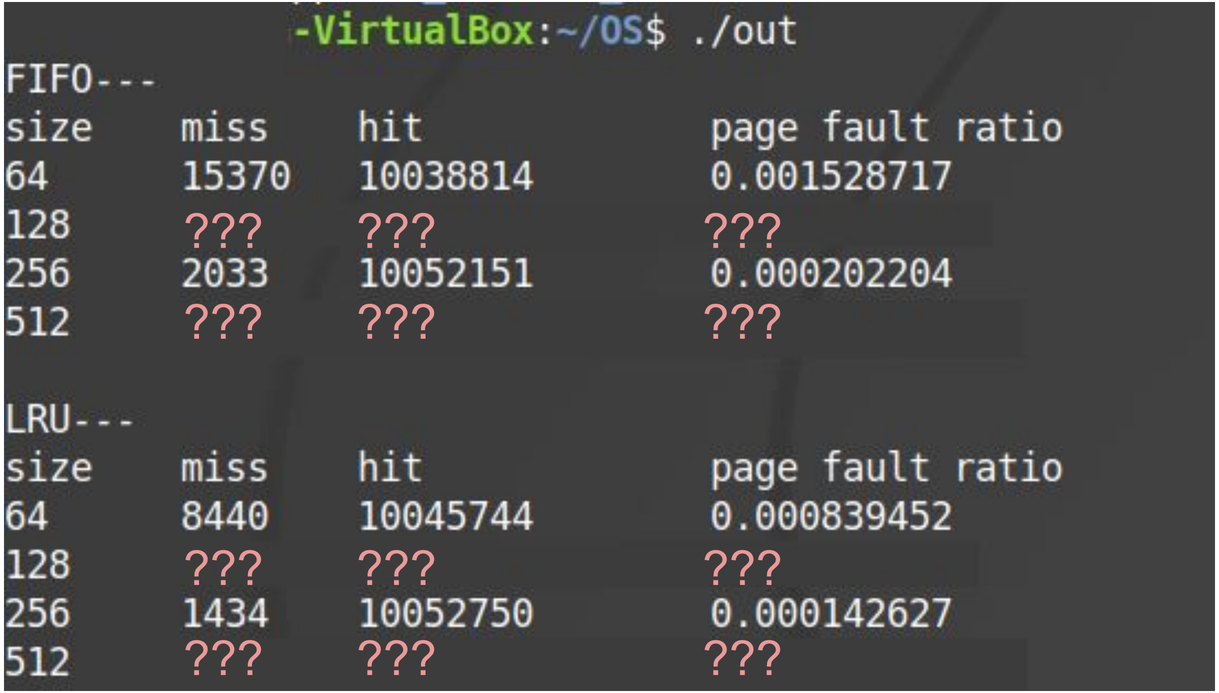
- Page lookup
  - Check whether or a new reference is a hit or a miss
  - Hash tables, binary search trees, skip lists....
- Do not use linear search!!!
  - TA will read your code, you will receive a grade penalty if you do
  - Implement your own search, or reuse any existing libraries/classes for searching
  - Duplication in this part does not count

Procedure

1. Algorithm=FIFO
2. For (Frame #=64; <=512; *=2)
   - Read the trace file “trace.txt”
   - Run simulation
   - Print out the miss count, hit count, page fault ratio
3. Algorithm=LRU
4. For (Frame #=64; <=512; *=2)
   - Read the trace file “trace.txt”
   - Run simulation
   - Print out the miss count, hit count, page fault ratio

Output Format

you need to print complete table



```
        -VirtualBox:~/OS$ ./out
FIFO---
size    miss    hit             page fault ratio
64      15370   10038814        0.001528717
128     ???     ???             ???
256     2033    10052151        0.000202204
512     ???     ???             ???

LRU---
size    miss    hit             page fault ratio
64      8440    10045744        0.000839452
128     ???     ???             ???
256     1434    10052750        0.000142627
512     ???     ???             ???
```

Requirements

- deadline : 2019/01/03 23:55
- upload you code to New e3 before deadline and named studentID_hw3.cpp
- Please write fp=fopen("trace.txt", "r"); in your code.
  - TAs will test your code only with the following commands

    ```
    g++ studentID_hw3.cpp -o out
    ./out
    ```
- the output miss count, hit count, page fault ratio must be correct
- Don't use linear search
- violating any requirement above will get score penalty

testing environment

- ubuntu 16.04
- ubuntu 14.04
- CS linux work station

your code should compile successfully in one of the above environments!