**Computer Architecture Project Report**

Project 2: Cache Simulator

2015117629 김지수, 2015113163 박효상

1. **Brief description of code**

Program simulates cache, and shows results of simulation with given trace file(memtrace.trc). result page includes cache/block size, associativity, replacement policy, cache access, hit, miss and miss rate.

Program reads cache settings from command-line arguments, and initiates cache with read value (init\_cache). Cache simulates memory accesses read from trace file, with given setting and replacement policy of cache.

Each replacement policy is implemented in insert\_to\_cache(). rand policy was first implemented by one rand() function, but changed to perform one more rand() function when same value keeps generating, for better result.

While simulating, Program counts number of total accesses and hit. After reading all memory accesses in memtrace file, Program prints result page.

Execution enrivonment of the program is windows 7, Visual studio 2017.

1. **Graphs of miss rates with different parameters**
2. **Cache miss rates with different block size.**

Fixed parameter : cache size(32768B), associativity(8), replacement policy(LRU).

Trace file : memtrace.trc (original trace file)

Miss rate stops decreasing at block size of 128B, at miss rate of 5.5455%. Further increase of block size causes slight increase of miss rate, due to the decrease of total number of blocks.

1. **Cache miss rates with different cache size.**

Fixed parameter : block size(32B), associativity(8), replacement policy(LRU).

Trace file : memtrace.trc (original trace file)

We simulated until miss rate stops decreasing at cache size of 4096kb, at miss rate of 1.4462%

1. **Cache miss rates with different associativity**

Fixed parameter : cache size(32KB), block size(32B), replacement policy(LRU).

Trace file : memtrace.trc (original trace file)

We could not get meaningful value using original memtrace.trc, so we simulated again with test\_assoc.trc.

Fixed parameter : cache size(32KB), block size(32B), replacement policy(LRU).

Trace file : test\_assoc.trc (new trace file)

With new trace file, miss rate decreases as associativity increases. miss rate drastically decreases when associativity increases from 16 to 32, and stops decreasing when associativity is 64, at miss rate of 0.033%.

Simulated result is also similar to the provided expected result.

1. **Cache miss rates with different replacement policy**

Fixed parameter : cache size(32KB), block size(32B), associativity(8).

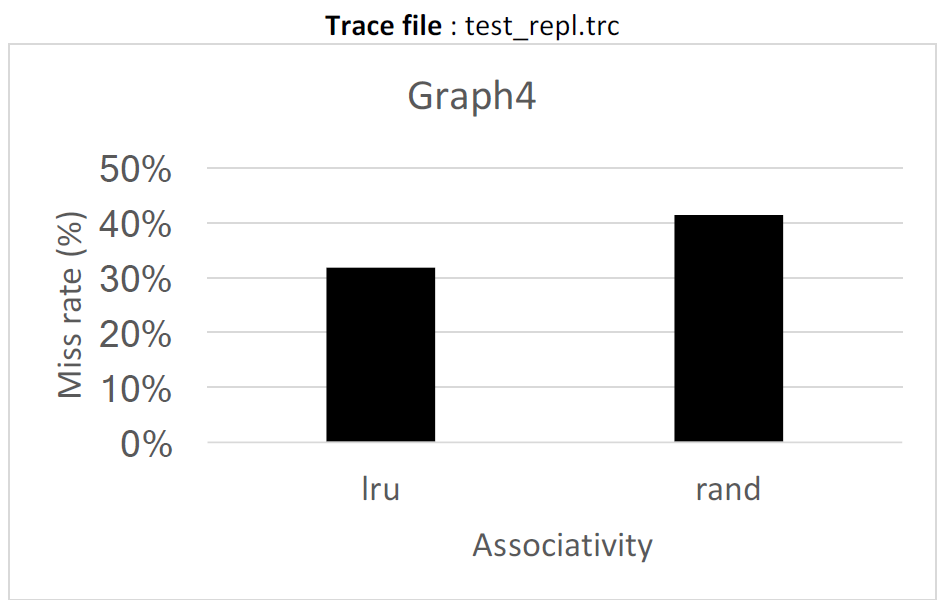
Trace file : memtrace.trc (original trace file)

LRU had slightly lower miss rate (5.8551%) than simulations with random policy. To check our program’s random policy is well-implemented or not, we simulated again with test\_repl.trc file, and compared our result with expected results.

Fixed parameter : cache size(32KB), block size(32B), associativity(8).

Trace file : test-repl.trc (new trace file)

When we simulated with test-repl.trc file, we got result similar to the provided expected result below.



1. **Contribution of each member**

Both of team member contributed well on the project (5:5).