Goal

Your main goal is to develop a console-based simulation for a simplified memory hierarchy—at a minimum, a page table (PT)—and how a virtual data reference is translated and handled by the memory hierarchy. You will read a trace of memory references from the standard input, simulate the effects of those memory references on the memory hierarchy, and produce statistics about the trace to the standard output.

This is a ***group project***—students completing this project should organize into teams of 3-5 students.

Each team of students will present their work to the class in a 10-15 minute presentation.

Background and Requirements

Trace: a memory reference has the following format:

```
<accesstype>:<hexaddress>
```

`<accesstype>` is one of the two following characters:

`R` – a read access
`W` – a write access

`<hexaddress>` is the starting address of the reference (expressed as a hexadecimal number) with a reference size ranging from 8 bits to 32 bits. See `trace.dat`, `real_tr.dat`, and the example section for examples of memory references.

Minimum Viable Product

Each team's Memory Hierarchy Simulator implementation shall:

- Display its configuration,
- Read a series of memory traces from the standard input,
- Parse read memory traces,
- Simulate the effects of each memory trace in sequence on the memory hierarchy,
- Display information about each reference,
- Display summary statistics regarding a given trace, e.g.,
    - Total number of hits,
    - Total number of misses,
    - Hit ratio (hits / misses),
    - Number of read accesses,
    - Number of write accesses,
    - Read/write ratio, and
    - Total number of memory references.

A complete implementation of the minimum viable product (MVP) meets my expectations for this project. Students may optionally extend these specifications (see below) to exceed expectations.

Simulation Algorithms

At a minimum, each team's simulation shall implement:

- The optimal (offline) greedy algorithm,
- The optimal approximation (online) algorithm: Least Recently Used (LRU), and
- The online First In, First Out (FIFO) algorithm.

Displaying References

The following C printf format specifier shows how to display each memory reference:

```
"%08x %6x %4x %4s %4x"
```

Each field corresponds to the following information:

- Virtual address
- Virtual page number
- Page offset
- PT result
- Physical page number

Deliverables

Student teams shall prepare a 10–15-minute presentation that:

- Introduces the student team,
- Briefly describes what work on the MHS simulation the team completed,
- Demonstrates the MHS application,
    - At a minimum: show the results you calculated from my provided traces
- Performs a retrospective on the project's development process,
    - Who did what,
    - What went well,
    - What did not go well, and
    - What would the team do differently in future work.

Exceeding Expectations

Think through ways to extend the MVP: adding additional algorithms and/or adding additional elements to the memory hierarchy (e.g., data cache layer(s), a translation lookaside buffer (TLB), cache/paging organization schemes, and/or configurable simulation parameters) are the most interesting extensions to me that give evidence for exceeding expectations.

I will also consider UI/UX extensions (e.g., a GUI), but these types of extensions are less impressive than those that directly relate to course learning outcomes.

Deliverables

Submit your memory hierarchy simulation source codes and presentation materials to the Dropbox by posted due date. In addition, present your simulation during the week of classes specified on the course schedule.

Grading

Project submissions that provide a functional memory hierarchy simulation that meets the minimum viable product specifications, but does not exceed those, along with a sufficient presentation, meets my expectations for the project. Project submissions that exceed the specified application requirements and/or excellent presentations will exceed my expectations, depending on the quality of the extensions (for the application) and/or excellence of the presentation (e.g., high-quality visual aids, smooth delivery, and engagement with the audience).

Examples

This is an example output from one of my own implementations of this project—achieving this level of simulation within the memory hierarchy (e.g., configurable, multi-level memory hierarchy simulation) that meets the remaining minimum viable product requirements will strongly exceed expectations.

`trace.config`

```
Data TLB configuration
Number of sets: 2
Set size: 1

Page Table configuration
Number of virtual pages: 64
Number of physical pages: 4
Page size: 256

Data Cache configuration
Number of sets: 4
Set size: 1
Line size: 16
Write through/no write allocate: n

L2 Cache configuration
Number of sets: 16
Set size: 4
Line size: 16
Write through/no write allocate: n

Virtual addresses: y
TLB: y
L2 cache: y
```

`trace.dat`

```
R:c84
R:81c
R:14c
R:c84
R:400
R:148
R:144
R:c80
R:008
```

Output

```
Data TLB contains 2 sets.
Each set contains 1 entries.
Number of bits used for the index is 1.


Number of virtual pages is 64.
Number of physical pages is 4.
Each page contains 256 bytes.
Number of bits used for the page table index is 6.
Number of bits used for the page offset is 8.
```

```
D-cache contains 4 sets.
Each set contains 1 entries.
Each line is 16 bytes.
The cache uses a write-allocate and write-back policy.
Number of bits used for the index is 2.
Number of bits used for the offset is 4.

L2-cache contains 16 sets.
Each set contains 4 entries.
Each line is 16 bytes.
The cache uses a write-allocate and write-back policy.
Number of bits used for the index is 4.
Number of bits used for the offset is 4.

The addresses read in are virtual addresses.

Virtual  Virt.  Page TLB    TLB TLB  PT   Phys         DC  DC          L2  L2
Address  Page # Off  Tag    Ind Res. Res. Pg # DC Tag Ind Res. L2 Tag Ind Res.
-------- ------ ---- ------ --- ---- ---- ---- ------ --- ---- ------ --- ----
00000c84      c   84      6   0 miss miss    0      2   0 miss      0   8 miss
0000081c      8   1c      4   0 miss miss    1      4   1 miss      1   1 miss
0000014c      1   4c      0   1 miss miss    2      9   0 miss      2   4 miss
00000c84      c   84      6   0 miss  hit    0      2   0 miss      0   8  hit
00000400      4    0      2   0 miss miss    3      c   0 miss      3   0 miss
00000148      1   48      0   1  hit         2      9   0 miss      2   4  hit
00000144      1   44      0   1  hit         2      9   0  hit
00000c80      c   80      6   0 miss  hit    0      2   0 miss      0   8  hit
00000008      0    8      0   0 miss miss    1      4   0 miss      1   0 miss

Simulation statistics

DTLB hits : 2
DTLB misses : 7
DTLB hit ratio : 0.222222

PT hits : 2
PT faults : 5
PT hit ratio : 0.285714

DC hits : 1
DC misses : 8
DC hit ratio : 0.111111

L2 hits : 3
L2 misses : 5
L2 hit ratio : 0.375000

Total reads : 9
Total writes : 0
Ratio of reads : 1.000000

Main memory refs : 5
Page table refs : 7
Disk refs : 5
```