The assignment idea needs to come from you. You can work in groups of max 2 people. Needs to be completed by end of March.

Total: 30 marks.

Assignment statement: 5 marks

Implementation: 10 marks

Result summary and conclusions (can be in the form of a report): 15 marks

Only one person in the group needs to submit. Write names of group members in the report.

Plagiarism penalty: 0 marks for the assignment. One grade penalty on the overall grade.

----

1. Download ChampSim: https://github.com/ChampSim/ChampSim
Trace files to run the simulation can be downloaded from:
https://www.dropbox.com/sh/hh09tt8myuz0jbp/AACAS5zMWHL7coVuS-RbpUksa?dl=0
Warning: These can be huge files (each can be few MBs to GBs)- so download only a few of them

Trace files are nothing but files that contain memory addresses of benchmark programs. These can be used as inputs to the cache/memory system.

2. Follow the Readme file in the Champsim directory

First build Champsim and then run Champsim with the built executable

```
$ ./build_champsim.sh bimodal no no lru 1
```

After building, it will show:

```
ChampSim is successfully built
Branch Predictor: bimodal
L1D Prefetcher: no
L2C Prefetcher: no
LLC Replacement: lru
Cores: 1
Binary: bin/bimodal-no-no-lru-1core
```

This last line has the name of the executable: bimodal-no-no-lru-1core

3. Now Copy `scripts/run_champsim.sh` to the ChampSim root directory
Go through the run_champsim.sh file → edit the first line in this file to enter the path where the trace files are available.
For eg- If your trace file is astar_23B.trace.gz, it should be available at that link mentioned in the run_champsim.sh file
Run the tool using the command:

```
$ ./run_champsim.sh bimodal-no-no-lru-1core 1 4 astar_23B
```

Description of 1 and 4 are available in the Readme

4. After the run, results are available in results_4M folder.

If you entered ./run_champsim.sh bimodal-no-no-lru-1core 1 6 bzip2_183B, it will run this bzip benchmark and dump the results in results_6M folder

---

5. Running 4 core workloads:
./build_champsim.sh bimodal no no lru 4 --> rebuild champsim for 4 cores

Copy this script run_4core.sh to the ChampSim root directory. The first line in this file has path for the sim_list folder and trace files from 4coreworkloads.txt
(sim_list/4core_workloads.txt)

4coreworkloads.txt contains same workload name repeated 4 times so that it runs the same workload 4 times on the 4 cores

6. How to run this?

```
./run_4core.sh bimodal-no-no-lru-4core 1 4 2
```

2 means line number in the 4coreworkloads.txt. The 4 benchmarks in that line are run by the four cores

Results are available in results_4core folder

7. Similarly you can run the 8 core system with the same workloads

---

8. You can modify the following files to change the number of sets, add your own replacement policy etc

cache.h --> add a new replacement policy called “xyz”
basereplacement.cc --> add the above function “xyz” here

cache.h - change the number of sets and ways

Few assignment/project ideas:
https://cseweb.ucsd.edu/classes/fa11/cse240A-a/project1.html
https://safari.ethz.ch/architecture/fall2019/lib/exe/fetch.php?media=comparchfs19_lab4_prefetching_fix.pdf

http://pages.cs.wisc.edu/~alaa/CRC/CRC.html

https://www.sigarch.org/call-contributions/the-2nd-cache-replacement-championship/

Some studies that can be done:

1. Miss rate versus block size. (See Chapter 5 of Computer organisation and design- H&P)
2. Miss rate vs number of sets, or cache size
3. Impact of different replacement policies on multi-core systems
4. Impact of prefetching
5. Impact of different branch predictors
6. Try to include a predictor or replacement policy which does not exist in this tool