Efficiency: caching (continued); Multiprocessing: operating systems; limited direct execution; system calls

COSC 208, Introduction to Computer Systems, 2021-11-03

Announcements

- Project 2 Part B due date extended to Tues, Nov 9

Outline

- Warm-up
- Cache replacement
- OS overview
- Accessing hardware
- Limited direct execution
- System calls

Warm-up

Q1: Where are caches used in computer systems?

🛑 **STOP HERE** after completing the above question; if you have extra time take a few deep breaths to help reduce stress.

Cache replacement

- If a cache is full, then a cache entry must be removed so different data can be placed in the cache
- Cache replacement policy governs which data is removed
- *What should a good cache replacement policy do?* — maximize the number of cache hits (or minimize the number of cache misses)
    - Evaluation metric: Hit ratio = number of hits / total number of memory accesses
- *How do we determine which cache entry to replace?*
- Optimal replacement policy: policy that replaces the entry that will be accessed furthest in the future
    - Impractical because we don't know data access patterns a priori
- First-in First-out (FIFO)
    - Simple to implement
    - Doesn't consider the importance of a cache entry

- Random
    - Even simpler to implement
    - Doesn't consider the importance of a cache entry
- Least Frequently Used (LFU) and Least Recently Used (LRU)
    - Based on the principle of locality
    - LFU assumes a page that is accessed many times will be accessed many more times
    - LRU assumes a page that was accessed recently will be accessed again soon
    - Inverse is very bad replacement policy
    - Downside: lots of overhead to implement — need to store an ordered list of pages and move a page up in the list whenever it's accessed
    - Where does this go wrong? — when working-set size (i.e., number of repeatedly accessed entries) is (slightly) greater than size of the cache
- *Assume a cache can hold 3 entries and the following 15 data accesses occur: 3, 4, 4, 5, 3, 2, 3, 4, 1, 4, 4, 2, 5, 2, 4. Assuming the cache is initially empty, what is the hit ratio for each of the following algorithms? Assume a cache can hold 3 entries and the following 15 data accesses occur: 3, 4, 4, 5, 3, 2, 3, 4, 1, 4, 4, 2, 5, 2, 4. Assuming the cache is initially empty, what is the hit ratio for each of the following algorithms?*
- Q2: *LRU*

- Q3: *LFU*

- *Optimal*