

Quiz 5 Solutions

1. Consider the following code:

```
#include <iostream>
#include <thread>

const unsigned int n = 20000;
const unsigned int num_threads = 4;

double a[n], b[n], c[n];

int main(int argc, const char * argv[]) {
    unsigned int i;
    std::thread * worker[num_threads];

    for(i = 0; i < num_threads; i++) {
        worker[i] = new std::thread([i] {
            std::cout << "I am thread " << i << " " <<
            std::this_thread::get_id() << std::endl;
        });
    }

    for(i = 0; i < num_threads; i++) {
        worker[i]->join();
    }

    std::cout << "We are done" << std::endl;
    return 0;
}
```

Identify the segments that each variable represented by a symbol resides in.

`n`, `num_threads`, `a`, `b` and `c` are all in the data segment.
`argc`, `argv`, `i`, `worker` are all in the stack segment
`main`, `get_id`, and `join` are in code segment (or `get_id` and `join` could be in library segment).

2. You have been tasked to evaluate two designs for the data prefetch unit in an out of order processor. The first choice places the prefetch unit in the cache controller. A cache line is 64-byte long. When a data item is read at address “a”, then a+64, a+128, a+192 and a+256 are read by the prefetcher. In the second choice, the prefetch unit is in the load-store unit of the pipeline, and when a read operation from address “a” is read, then then a+64, a+128,

a+192 and a+256 are read by the prefetcher. Which design choice do you recommend?

The locality of memory references is a property of the user program that runs on the processor. The user program deals with virtual addresses, not physical ones. Indeed, two consecutive virtual pages may be mapped to two non-consecutive physical frames. Therefore, the prefetcher cannot be placed in the cache, or it could start fetching data that will not be needed.

3. The memory management unit contains a register that has the address of the root page table in a multi-level indexed page table scheme. Is this address virtual or real?

The address must be the real address where the root of the page table tree resides. Otherwise, we will need a TLB for the TLB!

4. Justin P. Weir is designing a cache system for a processor that supports four hardware threads. He argues that it is better to design four different cache subsystems, one for each thread, so that the threads do not interfere with one another in terms of performance. Critique this idea, arguing whether or not you believe it has merits, and explain your answers.

Pros: If the threads are running simultaneously, then separate caches will eliminate the interference between the threads' working sets in the cache. No thread will contribute to evicting a cache line that is potentially needed by the other threads.

Cons: For a given chip area, the per-thread cache will **have to be smaller**, and potentially generating more cache misses which can hurt performance. Also, if the threads are sharing memory, as is often the case, then there will be a problem with the coherence of these caches at the lowest level of the hierarchy. Finally, if the threads are not allocated, then their corresponding caches will be idle, potentially consuming static power while not doing any useful work.