Solutions, Quiz 4, CSCI 210, Spring 2004

1. When `skip` is 16,384, each access to `arr[i]` in the inner loop maps to the same set of cache lines, the lines accessed at the beginning of the inner loop will be dropped from the cache as the inner loop continues. When `k` is incremented and the program reads the array entries following those accessed in the previous loop, those lines aren't available in the cache. As a result, all array accesses miss the cache.

When `skip` is 16,352, then the accesses to `arr[i]` map to a variety of cache lines, and all the requested lines can fit into the cache. (The code would access $4000000/16362 \approx 244$ lines, and since they map to different sets, they would all fit into the cache.) When the code continues to the next value of `k`, the data is still in the cache from before, and so the array accesses usually hit the cache.

2. In an operating system supporting segments, the operating system chooses which part of memory the program will occupy when the program starts. When a program is loaded into memory, the program must contain references to specific memory addresses within the program's region of memory, but the executable file cannot specify these addresses directly because the compiler generates it without knowing where the operating system may choose to place the program. (Indeed, the operating system may be executing the same program in different regions of memory at the same time.) The *relocation problem* refers to the problem of adapting these memory addresses to the region of memory chosen by the operating system for the loaded program.

3. The OS resets the dirty bit to zero each time it loads a page into RAM, and the CPU sets the dirty bit to one each time an instruction modifies the information in the page. As a result, the "dirty bit" keeps track of whether the page has changed since it was loaded into RAM.

If there were no dirty bit, then when the OS decided to eject the page from RAM, it would have to assume that the page were changed, and so it would have to save the page onto disk before the RAM becomes available for other purposes. The time to save a page onto disk increases the expense of ejecting a page significantly. With the dirty bit, the OS can avoid this expense when possible.