Alex Sobiek

COMP 264

Dr. Greenberg

Homework 7-1

Determine the cache performance of the following code when the cache is a 2048 byte direct-mapped cache with 32 byte blocks:

```
for (i=0; i<16; i++) {
    for (j=0; j<16; j++) {
        square[i][j].c = 0;
        square[i][j].m = 0;
        square[i][j].y = 0;
        square[i][j].k = 0;
    }
}
```

(a) What is the total number of writes Since it's a 16x16 matrix of a struct that has 4 elements each, the number of writes for the entire matrix would be `16*16*4 = 1024`

(b) What is the total number of writes that miss in cache? We're accessing the data in a linear manner and since each cache block can hold 8 structs and when we access a new cache block we always miss the first, the number of writes that miss will be 128.

(c) What is the miss rate? $128/1024 = 0.125$ or 12.5%

Homework 7-2

With the same assumptions as in the previous problem, determine the cache performance of the following code:

```
for (j=0; j<16; j++) {
    for (i=0; i<16; i++) {
        square[i][j].c = 0;
        square[i][j].m = 0;
        square[i][j].y = 0;
        square[i][j].k = 0;
    }
}
```

(a) What is the total number of writes? Since it's the same 16x16 matrix with structs of 4 elements, the number or writes for the entire matrix is `16*16*4 = 1024`

(b) What is the total number of writes that miss in cache? Because each iteration jumps cache blocks, we will miss each time. By the time we finish the inner loop and return to a cache block that we once accessed before, we can assume that it no longer has the previous values that would have been loaded. Therefore, we will miss 1024 writes.

(c) What is the miss rate? The miss rate will be 100%