

Computer Organization

C++ Cache Simulator:

這次的Lab是要用c++模擬出n-way associative cache，以下解釋如何實作：

首先是變數名稱，大部分沿用助教的code，只是搞懂這些變數代表的意思花了一些時間，而我在void simulate一開始變數附值加進了asso的變數，用來計算n-way associative的index；而line*asso就是cache內的block數，所以開了二元陣列用以表示cache。再來就是要計算Miss Rate和Hit/Miss instruction的hit/miss_times, instruction_num等等。

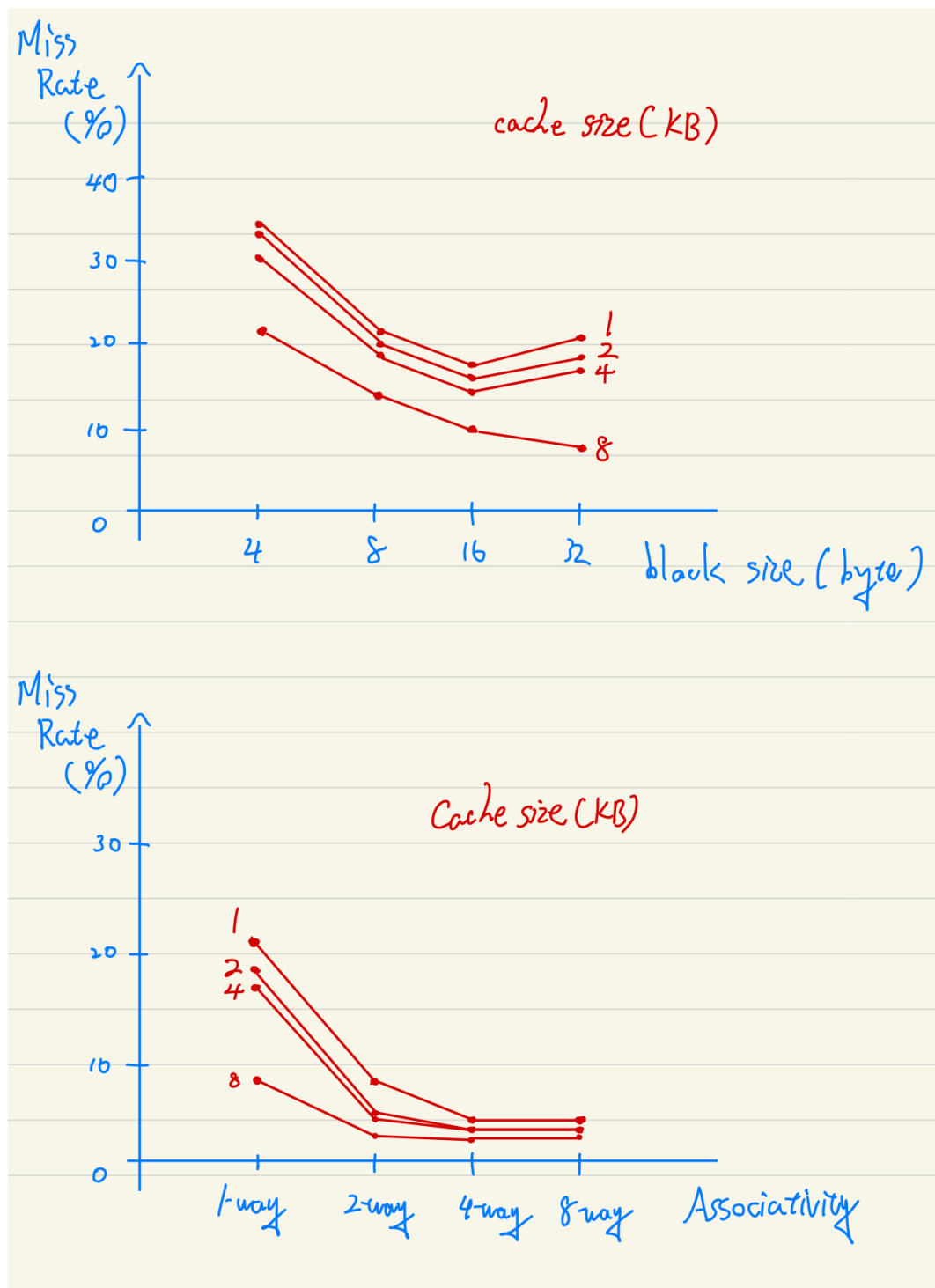
再來是LRU Replacement Policy，我的方法是新增變數LRU_num = 0，在cache_content加上一個變數LRU，每當一個cache裡的block被讀取到，不管是hit或miss，都令他的LRU為LRU_num++，這樣一來可以發現，當我需要找最不被使用到的entry時，只要找LRU最小的就可以實現。

處理完LRU，接下來就是沿著助教的code寫下來，為了記錄hit和miss的instruction，我開了兩個vector表示，只要hit或miss就把instruction_num push進去，當cache[index][I].v && cache[index][i].tag == tag，hit++等等，而當沒有hit到實則進入miss的處理，首先先判斷是否有空的entry(full != false)，有的話就直接填入tag和index，沒有的話就照LRU來找要替換的值，可是我做到後來突然想到不用這麼麻煩？因為如果是空的那LRU就會是0，一定會是最小的，所以其實應該只需要做LRU的部分就行了。

最後就是計算Miss Rate和print出 hit_instruction和 miss_instruction，就OK啦～

Result:

以下就是自己繪製的圖表，上圖為固定associativity = 1，下圖為固定block size = 32byte。



大致上也符合講義所述：For large block size, may increase miss rate due to pollution，以及對於associative的敘述：Increased associativity decreases miss rate but with diminishing returns.

Summary:

寫完這次作業發現最花我時間的其實是搞懂變數的意義和回想cache的架構，然後就是makefile，身為一個資工系學生這竟然我第一次接觸makefile，實在是有點尷尬，不過也只有一個cpp要編譯所以短短幾行就結束了，寫完這次Lab這學期的課業也跟著結束啦（希望能過~