

Mar 02, 10 10:00

l13-handout.txt

Page 1/1

```

1 Handout for CS 372H
2 Class 13
3 02 March 2010
4
5 Some examples related to sequential consistency
6
7 [From S.V. Adve and K. Gharachorloo, IEEE Computer, December 1996,
8 66-76. http://rsim.cs.uiuc.edu/~sadve/Publications/computer96.pdf]
9
10 1. What might p2 return if run concurrently with p1?
11
12     int data = 0, ready = 0;
13
14     void p1 () {
15         data = 2000;
16         ready = 1;
17     }
18     int p2 () {
19         while (!ready) {}
20         return data;
21     }
22
23 2. Can both critical sections run?
24
25     int flag1 = 0, flag2 = 0;
26
27     int main () {
28         tid id = thread_create (p1, NULL);
29         p2 (); thread_join (id);
30     }
31
32     void p1 (void *ignored) {
33         flag1 = 1;
34         if (!flag2) {
35             critical_section_1 ();
36         }
37     }
38
39     void p2 (void *ignored) {
40         flag2 = 1;
41         if (!flag1) {
42             critical_section_2 ();
43         }
44     }
45
46 3. If a processor can read its own writes early, then both functions
47 below can return 2:
48
49     /*
50      * keyword "register" tells compiler to place the variable in a
51      * register, not on the stack.
52      */
53
54     int flag1 = 0, flag2 = 0;
55
56     int p1 (void *ignored)      int p2 (void *ignored)
57     {                           {
58         register int f, g;      { register int f, g;
59         flag1 = 1;              flag2 = 1;
60         f = flag1;              f = flag2;
61         g = flag2;              g = flag1;
62         return 2*f + g;          return 2*f + g;
63     }                          }
64

```