

```
1  /* Program to fill cache. */
2  #include <stdio.h>
3  #include <malloc.h>
4
5  #define LG_LINE_SIZE 4
6  #define LG_SET_COUNT 10
7  #define ASSOCIATIVITY 4
8
9  #define SET_COUNT (1<<LG_SET_COUNT)
10 #define CACHE_SIZE ( ASSOCIATIVITY << ( LG_LINE_SIZE + LG_SET_COUNT ) )
11
12 char a[ 2 * CACHE_SIZE ];
13
14
15 int main(int argv, char **argc)
16 {
17     int dummy = 0;
18
19     printf("Cache parameters:\n");
20     printf("  %d-way set associative, 2^%d = %d sets.\n",
21           ASSOCIATIVITY, LG_SET_COUNT, 1<<LG_SET_COUNT);
22     printf("  Line size 2^%d = %d characters, character size 8 bits = 1 byte.\n",
23           LG_LINE_SIZE, 1<<LG_LINE_SIZE);
24     printf("  Cache capacity %d bytes, %.2f MiB (megabytes) [sic].\n",
25           ASSOCIATIVITY << ( LG_LINE_SIZE + LG_SET_COUNT ),
26           ((double)(ASSOCIATIVITY << ( LG_LINE_SIZE + LG_SET_COUNT )))
27           / ( 1 << 20 ));
28
29
```

```
29
30  /* Fill up cache with minimum number of accesses. */
31
32  {
33      int s,w;
34      char *addr = &a[0];
35      int linesize = 1 << LG_LINE_SIZE;
36
37      for(w=0; w<ASSOCIATIVITY; w++)
38
39          for(s=0; s<SET_COUNT; s++)
40              {
41                  dummy += *addr;
42                  /* Below, increment index part of address. */
43                  addr += linesize;
44              }
45      }
46
```

```
46  /* Fill up cache with minimum number of accesses. (Alternate method.) */
47
48
49  {
50      int w;
51
52      for(w=0; w<ASSOCIATIVITY; w++)
53      {
54          char *b = & a[ w << ( LG_LINE_SIZE + LG_SET_COUNT ) ];
55          int s;
56
57          for(s=0; s<SET_COUNT; s++) dummy += b[ s << LG_LINE_SIZE ];
58      }
59  }
60
```

```
60
61  /* Generate lots of misses while accessing only ASSOCIATIVITY + 1
62     array elements. */
63
64  {
65     int i, w;
66     int deadly_stride = 1 << ( LG_LINE_SIZE + LG_SET_COUNT );
67
68     for(i=0; i<100; i++)
69     {
70         char *addr = &a[0];
71
72         for(w=0; w<=ASSOCIATIVITY; w++)
73         {
74             dummy += *addr;
75             /* Below: Increment tag part of address. */
76             addr += deadly_stride;
77         }
78     }
79 }
80
81 if( dummy )
82     printf("This line prevents optimizer from eliminating code above.\n");
83
84 return 0;
85 }
end
```