

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <unistd.h>
4 #include <sys/mman.h>
5 #include <sys/wait.h>
6 #include <errno.h>
7 #include <string.h>
8 #include <ctype.h>
9
10 #include "spin.h"
11
12 #define NUM_PROC 64
13 #define MYPROCS 4
14 #define NUMITR 1000000
15
16 void throwError(char *message, char *file)
17 {
18     if (file)
19         fprintf(stderr, "%s [%s]: Error code %i: %s\n", message, file, errno,
strerror(errno));
20     else
21         fprintf(stderr, "%s\n", message);
22     exit(-1);
23 }
24
25 int main(int argc, char const *argv[])
26 {
27     int pid[MYPROCS], myPID = 0;
28
29     unsigned long long idealCt = (MYPROCS * NUMITR);
30     unsigned long long *counter = (unsigned long long *)mmap(NULL,
sizeof(unsigned long long), PROT_READ | PROT_WRITE, MAP_SHARED |
MAP_ANONYMOUS, -1, 0);
31     unsigned long long *counterTAS = (unsigned long long *)mmap(NULL,
sizeof(unsigned long long), PROT_READ | PROT_WRITE, MAP_SHARED |
MAP_ANONYMOUS, -1, 0);
32
33     char *lock = (char *)mmap(NULL, sizeof(char), PROT_READ | PROT_WRITE,
MAP_SHARED | MAP_ANONYMOUS, -1, 0);
34
35     for (int i = 0; i < MYPROCS; i++)
36     {
37         if ((pid[i] = fork()) < 0)
38         {
39             throwError("Error: Failed to fork process.", NULL);
40         }
41         else if (pid[i] == 0)
42         {
43             myPID = 0;
44         }
45     }
```

```
45         for (i = 0; i < NUMITR; i++)
46         {
47             *counter += 1;
48         }
49
50         for (i = 0; i < NUMITR; i++)
51         {
52             spin_lock(lock);
53             *counterTAS += 1;
54             spin_unlock(lock);
55         }
56         break;
57     }
58     else
59         myPID = 1;
60 }
61
62 if (myPID)
63 {
64     for (int i = 0; i < MYPROCS; i++)
65     {
66         if (waitpid(pid[i], NULL, 0) < 0)
67         {
68             throwError("Error: Unable to wait for child process to
complete", NULL);
69         }
70     }
71     // PRINT OUT RESULTS
72     fprintf(stderr, "IDEAL COUNT: %llu | NON-TAS COUNT: %llu | TAS COUNT:
%llu\n", idealCt, *counter, *counterTAS);
73
74     if ((munmap(counter, sizeof(unsigned long long)) < 0) ||
(munmap(counterTAS, sizeof(unsigned long long)) < 0))
75         throwError("Error: Unable to munmap counter(s)", 0);
76
77     if ((munmap(lock, sizeof(char)) < 0))
78         throwError("Error: Unable to munmap lock", 0);
79 }
80
81 return 0;
82 }
83
```