

```

1  #include <stdio.h>
2  #include <unistd.h>
3  #include <pthread.h>
4
5  pthread_mutex_t mut = PTHREAD_MUTEX_INITIALIZER;
6
7  void * dec(void *a)
8  {
9      long int * c = a;
10     for(long int i = 0; i < 10000000; i++) {
11         pthread_mutex_lock(&mut);
12         *c = *c - 1;
13         pthread_mutex_unlock(&mut);
14     }
15 }
16
17 void * inc(void *a)
18 {
19     long int * c = a;
20     for(long int i = 0; i < 10000000; i++) {
21         pthread_mutex_lock(&mut);
22         *c = *c + 1;
23         pthread_mutex_unlock(&mut);
24     }
25 }
26
27 int main(int argc, char * const argv[]) {
28
29     long int licznik = 20000000;
30
31     printf("start: %i\n", licznik);
32
33     pthread_t t0, t1;
34     if(pthread_create(&t0, NULL, dec, &licznik) == -1) {
35         printf("Nie mozna utworzyc watku: %d\n", t0);
36     } else {
37         printf("Watek: %d utworzony\n", t0);
38     }
39     if(pthread_create(&t1, NULL, inc, &licznik) == -1) {
40         printf("Nie mozna utworzyc watku: %d\n", t0);
41     } else {
42         printf("Watek: %d utworzony\n", t0);
43     }
44
45     void * results;
46     if(pthread_join(t0, results) == -1) {
47         printf("Nie mozna zakonczyc watku: %d\n", t0);
48     } else {
49         printf("Watek: %d zakonczony\n", t0);
50     }
51     if(pthread_join(t1, results) == -1) {
52         printf("Nie mozna zakonczyc watku: %d\n", t1);
53     } else {
54         printf("Watek: %d zakonczony\n", t1);
55     }
56
57     printf("end: %i\n", licznik);
58
59     return 0;
60 }

```