

```
1 #include <pthread.h>
2 #include "apue.h"
3 #define CICLI 10
4
5 /* Struttura Condivisa fra i thread */
6 typedef struct foo{
7     int a;
8     int b;
9     pthread_mutex_t sem; /*mutex inserito all'interno della struttura da
10     controllare*/
11 } myfoo;
12 myfoo *test; // Variabile GLOBALE Puntatore alla Struttura
13
14 myfoo *init_struct(){
15     struct foo *fp; //Puntatore alla struttura
16
17     if((fp=malloc(sizeof(myfoo)))==NULL) /*Allocazione dinamica Struttura
18     */
19         return(NULL);
20
21     fp->a=0; /*inizializzazione Struttura */
22     fp->b=0;
23     pthread_mutex_init(&fp->sem,NULL);/*inizializzazione dinamica mutex*/
24     return(fp);
25 }
26 void *somma1(void *);
27 void *somma2(void *);
28
29 int main(void){
30     pthread_t som1TID, som2TID;
31     test=init_struct(); //Inizializziamo la struttura condivisa
32
33     if (pthread_create(&som1TID, NULL, somma1, NULL) != 0) {
34         printf ("Errore nella creazione del thread somma1\n");
35         exit(1);
36     }
37
38     if (pthread_create(&som2TID, NULL, somma2, NULL) != 0) {
39         printf ("Errore nella creazione del thread somma2\n");
40         exit(1);
41     }
42
43     pthread_join(som1TID, NULL);
44     pthread_join(som2TID, NULL);
45     printf("valore -- a = %d \n", test->a);
46     printf("valore -- b = %d \n", test->b);
47     pthread_mutex_destroy(&test->sem); /*distrugge il mutex*/
```

```
48     exit (0);
49 }
50
51
52 void *somma1(void *in){
53     int i;
54
55     for(i=0; i < CICLI; i++) {
56         pthread_mutex_lock(&test->sem);
57         test->a++;
58         test->b++;
59         pthread_mutex_unlock(&test->sem);
60     }
61     pthread_exit(0);
62 }
63
64 void *somma2(void *in){
65     int i;
66
67     for(i=0; i < CICLI; i++) {
68         pthread_mutex_lock(&test->sem);
69         test->a++;
70         test->b++;
71         pthread_mutex_unlock(&test->sem);
72     }
73     pthread_exit(0);
74 }
75
```