```
static int glob = 0;
static pthread_mutex_t mtx = PTHREAD_MUTEX_INITIALIZER;

static void *threadFunc(void *arg)
{
  int j;

  for (j = 0; j < *((int *) arg); j++) {
    if ( pthread_mutex_lock(&mtx) != 0 ){
        perror("pthread_mutex_lock");
    }
    glob++;
    if ( pthread_mutex_unlock(&mtx) != 0){
        perror("pthread_mutex_unlock");
    }
  }

  return NULL;
}

int main(int argc, char *argv[])
{
    pthread_t t1, t2;
    int loops;

    loops = (argc > 1) ? atoi(argv[1]) : 10000000;

    if( pthread_create(&t1, NULL, threadFunc, &loops) != 0 ){
        perror("pthread_create");
    }

    if( pthread_create(&t2, NULL, threadFunc, &loops) != 0 ){
        perror("pthread_create");
    }

    if( pthread_join(t1, NULL) != 0 ){
        perror("pthread_join");
    }

    if( pthread_join(t2, NULL) != 0 ){
        perror("pthread_join");
    }

    printf("glob = %d\n", glob);
    exit(EXIT_SUCCESS);
}
```