

Общие механизмы OpenMP. Расширенные возможности

```
#pragma omp threadprivate(var1[, var2 ...])
```

- Позволяет один раз объявить приватную переменную для всех параллельных секций в рамках одного файла.
- Переменная должна быть объявлена как статическая
- Директива `threadprivate` должна присутствовать до объявления первой параллельной секции
- Количество потоков в программе должно быть постоянным

Общие механизмы OpenMP. Расширенные возможности

- **void omp_init_lock(omp_lock_t *lock)**
 - инициализирует блокировку и связывает ее с параметром *lock*
- **void omp_destroy_lock(omp_lock_t *lock)**
 - деинициализирует переменную, связанную с параметром *lock*
- **void omp_set_lock(omp_lock_t *lock)**
 - блокирует выполнение потока до тех пор, пока блокировка на переменную *lock* не станет доступной.
- **void omp_unset_lock(omp_lock_t *lock)**
 - снимает блокировку с переменной *lock*
- **void omp_test_lock(omp_lock_t *lock)**
 - пытается установить блокировку и если операция выполнена успешно, возвращает ненулевое значение
 - в противном случае возвращается ноль
 - функция не блокирующая

Общие механизмы OpenMP. Расширенные возможности

Locks

```
#include <omp.h>
int main() {
    omp_lock_t lock;
    int i, p_sum = 0, res = 0;
    omp_init_lock(&lock);
    #pragma omp parallel firstprivate(p_sum)
    {
        #pragma parallel for private(i)
        for(i=0; i<100000; i++)
            p_sum +=i;
        omp_set_lock(&lock);
        res += p_sum;
        omp_unset_lock(&lock);
    }
    omp_destroy_lock(&lock);
    printf("%d\n", res);
}
```