 omp\_init\_lock(&lock): Инициализация замка.

 omp\_set\_lock (&lock): Установка (захват) блокировка замка.

 omp\_unset\_lock(&lock): Освобождение захваченного замка.

 omp\_destroy\_lock(&lock): Уничтожение замка после его использования.

 omp\_test\_lock(&lock): Попытка захватить замок без блокирования.

lock: Указатель на объект типа omp\_lock\_t, который должен быть инициализирован с помощью функции omp\_init\_lock.

#pragma omp critical в OpenMP используется для создания критической секции — области кода, которая может выполняться одновременно только одним потоком. Это необходимо, чтобы избежать состояния гонки при доступе к общим данным.

#pragma omp parallel — это директива OpenMP, которая создаёт параллельный регион, в котором несколько потоков выполняют код одновременно.

#pragma omp sections в OpenMP позволяет разделить код на несколько независимых секций, каждая из которых может выполняться параллельно. В случае, когда необходимо чтобы основной поток не ждал завершения остальных потоков следует использовать условие nowait.

#pragma omp section используется внутри блока #pragma omp sections в OpenMP для разделения работы между потоками. Каждая секция будет выполняться одним из потоков, участвующих в параллельном блоке.