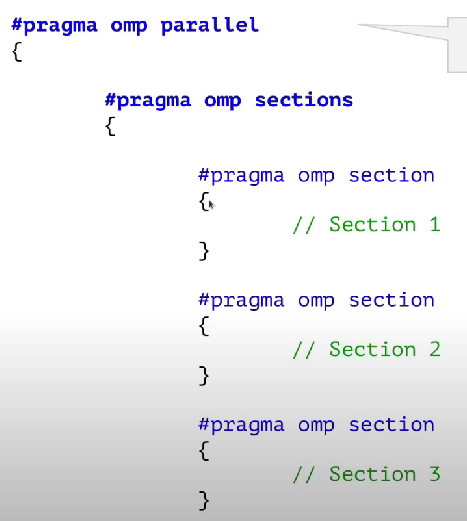
**Стандарт OpenMP. Параллелизм задач.**

Pragma omp parallel

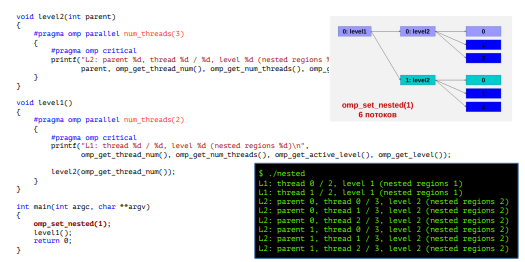
Порождает пул потоков (team of threads) и набор задач (set of tasks)



Код каждой секции выполняется одним потоком (в контексте задачи) NSECTIONS > NTHREADS Не гарантируется, что все секции будут выполняться разными потоками Один поток может выполнить несколько секций

Каждый поток поддерживает двухстороннюю очередь deque. Каждый поток может пушть(создавать) и пулить задачу. Стандарт openmp позволяет потоку красть задачи у других, что помогает нам избавиться от простоя потоков.

Мы можем создавать параллельную секцию внутри другой параллельной секции. Так, создав параллельную секцию, разделенную на три потока, и вызвав в ней еще одну параллельную секцию на два потока мы выполним блок кода в ней 6 раз.

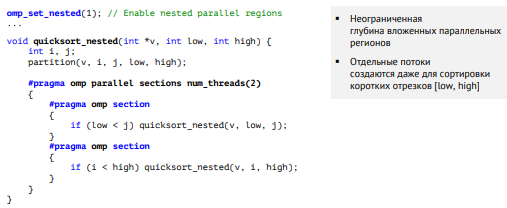


В openmp есть функция, которая отключает возможность вкладывать параллельные секции друг в друга, а также функции которая может ограничивать глубину вложений.





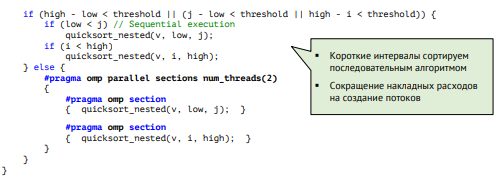
Попробуем применить параллелизм для решения задачи отсортировки массива сортировкой quicksort.



Огромный минус первой версии программы заключается в том, что мы выделяем отдельные потоки, даже на сортировку самых мелких кусочков массива, что затрачивает ресурсы.



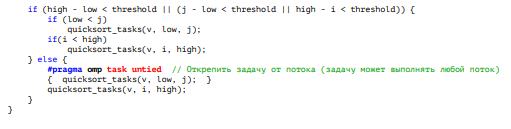
С помощью функции мы можем ограничить количество вложенй.



А короткие интервалы меньше чем threshold будем выполнять последовательно.

В конце концов мы можем избавиться от section.

Современный станадрат openmp позволяет нам использоват задачи task. Это более гибкое решение, потому что потоки могут красть их друг у друга по мере поступления.



Для левого подмассива, мы будем создавать задачу, которая благодаря untied уйдет к самому освобожденному потоку. А правый подмассив мы будем сортировать в контексте этой же задачи.