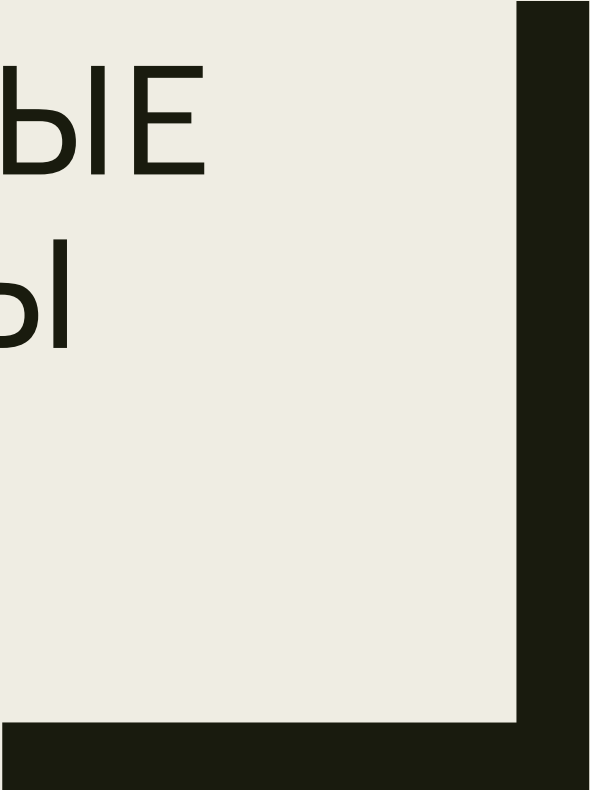




ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ

Занятие 1. Технология OpenMP



Составитель: Герасимов А.С.

OpenMP

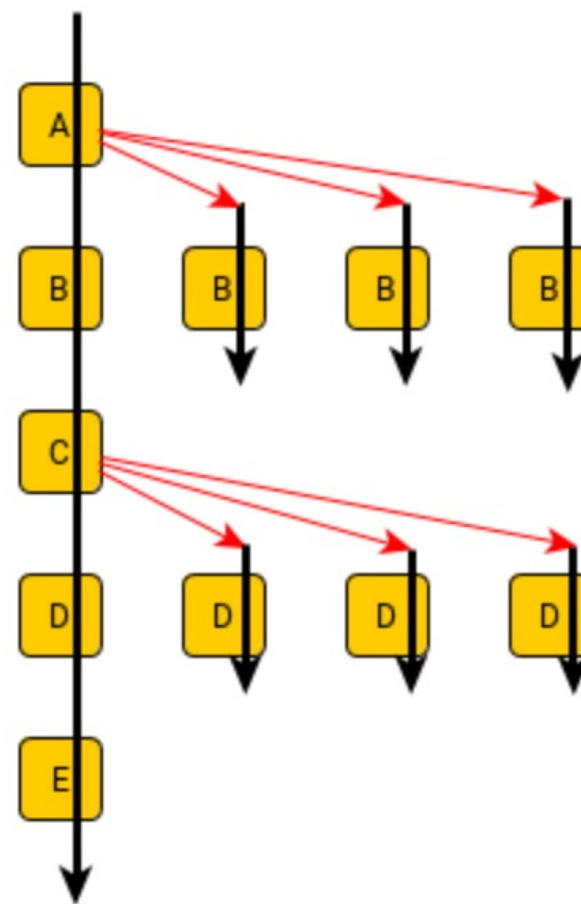
- Библиотека функций и директивы компилятора, предназначенные для написания параллельных программ
- Официально поддерживается *C, C++ и Фортран*, однако можно найти реализации для некоторых других языков, например *Паскаль* и *Java*.
- Поддерживается производителями аппаратуры (*Intel, HP, SGI, Sun, IBM*), разработчиками компиляторов (*Intel, Microsoft, KAI, PGI, PSR, APR, Absoft*)
- Ориентирован на системы с общей памятью
- Основной метод – создание потоков (модель fork-join)

Общие механизмы OpenMP. Инициализация

Основа программы

```
#include "omp.h"

int main(int argc, char** argv)
{
    // A - single thread
    #pragma omp parallel
    {
        // B - many threads
    }
    // C - single thread
}
```



Компиляция программы

```
gcc -fopenmp superhot.c -o hot
```

Запуск программы

```
OMP_NUM_THREADS=N ./hot
```

Общие механизмы OpenMP.

Параллельные регионы

- Количество порождаемых потоков для параллельных областей контролируется через переменную окружения **OMP_NUM_THREADS**, а также может задаваться через вызов функции внутри программы.
- Каждый порожденный поток исполняет код в структурном блоке. По умолчанию синхронизация отсутствует и последовательность выполнения не определена.
- После выполнения параллельного участка все потоки, кроме основного, завершаются.
- Каждый поток имеет уникальный номер, который изменяется от 0 до количества потоков - 1.
 - Идентификатор потока может быть определен с помощью функции ***omp_get_thread_num()***.
 - Общее количество потоков можно определить с помощью функции ***omp_get_num_threads()***.

Общие механизмы OpenMP. Параллельные регионы

```
#pragma omp parallel
{
    myid = omp_get_thread_num( );
    if(myid == 0)
        do_something( );
    else
        do_something_else(myid);
} ← sync
```