```
const unsigned physical_thread_number = std::thread::hardware_concurrency();

std::cout << "Available physical thread = "
<< available_threads << std::endl;
```

```
~/GitHub/cpp_sandbox/multithreading/thread_sample$ ./a.out  100 10
 …
Available physical thread = 4
 …
```

Get physical threads

```
const unsigned physical_thread_number = std::thread::hardware_concurrency();

 std::cout << "Available physical thread = "
 << available_threads << std::endl;
```

```
~/GitHub/cpp_sandbox/multithreading/thread_sample$ ./a.out  100 10
 …
Available physical thread = 4
 …
```

std::async + std::future

```
std::mutex m;
using lock = std::lock_guard<std::mutex>;
std::map<std::thread::id,bool> ids;

void f(unsigned i)
{
    lock lk{m};
    auto id = std::this_thread::get_id();
    std::cout << "thread #"<<i<< " id = " << id << std::endl;
    ids.insert(std::make_pair(id, false));
}
```