thread\_local

When needed, thread local data will be created for each thread. thread-local data exclusively belongs to the thread and behaves like static data. That means, it will be created at its first usage and its lifetime is bound to the lifetime of the thread. Often thread local data is called thread local storage.

Dealing with thread local data is straightforward.

// threadLocal.cpp

#include <iostream>

#include <string>

#include <mutex>

#include <thread>

std::mutex coutMutex;

thread\_local std::string s("hello from ");

void addThreadLocal(std::string const& s2){

s+=s2;

// protect std::cout

std::lock\_guard<std::mutex> guard(coutMutex);

std::cout << s << std::endl;

std::cout << "&s: " << &s << std::endl;

std::cout << std::endl;

}

int main(){

std::cout << std::endl;

std::thread t1(addThreadLocal,"t1");

std::thread t2(addThreadLocal,"t2");

std::thread t3(addThreadLocal,"t3");

std::thread t4(addThreadLocal,"t4");

t1.join();

t2.join();

t3.join();

t4.join();

}