Яндекс Такси

Встреча в ISO C++ в Белфаст и Цифры

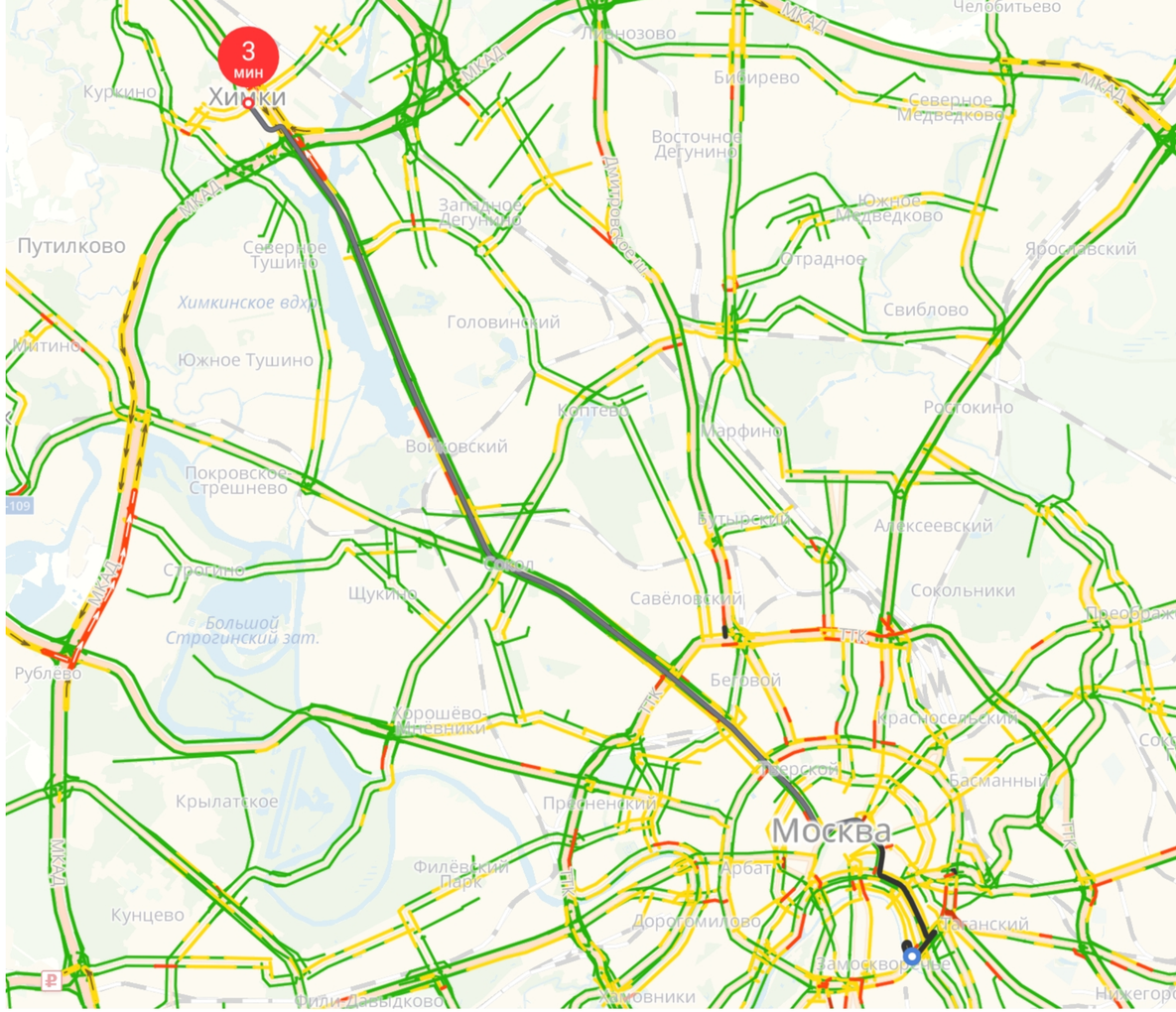
Полухин Антон

Antony Polukhin

Яндекс Такси

Содержание

- Баги
- Баги
- Баги
- NTTP
- Баги
- Numbers



RU 006

RU 006 (std::atomic)

```cpp
#include <atomic>


std::atomic<int> x{};
```

RU 006 (std::atomic)

```cpp
#include <atomic>


std::atomic<int> x{};  // does NOT zero initialize
```

RU 006 (std::atomic)

```cpp
#include <atomic>


std::atomic<int> x{};  // does NOT zero initialize


struct counter {
  int external_counters = 0;
  int count = 1;
};


std::atomic<counter> y;
```

RU 006 (std::atomic)

```cpp
#include <atomic>

std::atomic<int> x{};  // does NOT zero initialize

struct counter {
  int external_counters = 0;
  int count = 1;
};

std::atomic<counter> y;  // does not initialize with 0 and 1
```

RU 006 (std::atomic)

```cpp
#include <atomic>


std::atomic<int> x{};  // does NOT zero initialize


struct counter {
  int external_counters = 0;
  int count = 1;
};


std::atomic<counter> y;  // does not initialize with 0 and 1
std::atomic<counter> z{};
```

RU 006 (std::atomic)

```cpp
#include <atomic>


std::atomic<int> x{};  // does NOT zero initialize


struct counter {
  int external_counters = 0;
  int count = 1;
};


std::atomic<counter> y;  // does not initialize with 0 and 1
std::atomic<counter> z{};  // does not initialize with 0 and 1
```

RU 007

RU 007 (std::launder)

```
#include <new>

std::launder(x);  // ???
```

RU 007 (std::launder)

```
#include <vector>
```

RU 007 (std::launder)

```
#include <vector>

struct C {

  const int c;

};
```

RU 007 (std::launder)

```
#include <vector>
struct C {
  const int c;
};

void sanity_check() {
  std::vector<C> v = {C{1}};




}
```

RU 007 (std::launder)

```
#include <vector>
struct C {
  const int c;
};

void sanity_check() {
  std::vector<C> v = {C{1}};
  v.pop_back();



}
```

RU 007 (std::launder)

```cpp
#include <vector>
struct C {
  const int c;
};

void sanity_check() {
  std::vector<C> v = {C{1}};
  v.pop_back();
  v.push_back(C{2});

}
```

RU 007 (std::launder)

```cpp
#include <vector>
struct C {
  const int c;
};

void sanity_check() {
  std::vector<C> v = {C{1}};
  v.pop_back();
  v.push_back(C{2});
  assert(v.back().c == 2);
}
```

RU 007 (std::launder)

```cpp
#include <vector>
struct C {
  const int c;
};

void sanity_check() {
  std::vector<C> v = {C{1}};
  v.pop_back();
  v.push_back(C{2});
  assert(v.back().c == 2);  //  ]:->
}
```

RU 007 (std::launder)

```cpp
#include <vector>
struct C {
  const int c;
};

void sanity_check() {
  std::vector<C> v = {C{1}};
  v.pop_back();
  v.push_back(C{2});
  assert(std::launder(v.back()).c == 2);  //  ]:->
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      //
      C{1}                  //
  };                        //
  v.pop_back();             //
  v.push_back(C{2});        //
                            //
  assert(v.back().c == 2);  //
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  //
  };                        //
  v.pop_back();             //
  v.push_back(C{2});        //
                            //
  assert(v.back().c == 2);  //
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};
  };                        //
  v.pop_back();             //
  v.push_back(C{2});        //
                            //
  assert(v.back().c == 2);  //
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};
  };                        //
  v.pop_back();             // data_->~C();
  v.push_back(C{2});        //
                            //
  assert(v.back().c == 2);  //
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};
  };                        //
  v.pop_back();             // data_->~C();
  v.push_back(C{2});        // new (data_) C{2};
                            //
  assert(v.back().c == 2);  //
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};
  };                        //
  v.pop_back();             // data_->~C();
  v.push_back(C{2});        // new (data_) C{2};
                            //
  assert(v.back().c == 2);  // data_->c == 2;
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};              <== (*data_ == 1)
  };                        //
  v.pop_back();             // data_->~C();
  v.push_back(C{2});        // new (data_) C{2};
                            //
  assert(v.back().c == 2);  // data_->c == 2;
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};              <== (*data_ == 1)
  };                        //
  v.pop_back();             // data_->~C();                   <== v.pop_back()
  v.push_back(C{2});        // new (data_) C{2};
                            //
  assert(v.back().c == 2);  // data_->c == 2;
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};              <== (*data_ == 1)
  };                        //
  v.pop_back();             // data_->~C();                   <== v.pop_back()
  v.push_back(C{2});        // new (data_) C{2};              <== v.push_back(C{2})
                            //
  assert(v.back().c == 2);  // data_->c == 2;
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};              <== (*data_ == 1)
  };                        //
  v.pop_back();             // data_->~C();                   <== v.pop_back()
  v.push_back(C{2});        // new (data_) C{2};              <== v.push_back(C{2})
                            //
  assert(v.back().c == 2);  // data_->c == 2;                 <== (*data_ == 1) == 2
}
```

RU 007 (std::launder)

```
void sanity_check() {
  std::vector<C> v = {      // data_ = new char[sizeof(C)];
      C{1}                  // new (data_) C{1};              <== (*data_ == 1)
  };                        //
  v.pop_back();             // data_->~C();                   <== v.pop_back()
  v.push_back(C{2});        // new (data_) C{2};              <== v.push_back(C{2})
                            //
  assert(v.back().c == 2);  // data_->c == 2;                 <== 1 == 2
}
```

US 233

US 233 (std::span)

```
void send_ints(const int* data, unsigned size);
```

US 233 (std::span)

```
void send_ints(const int* data, unsigned size);
```

```
send_ints(varaible.data(), variable.size());
```

US 233 (std::span)

```cpp
#include <span>


void send_ints(std::span<int> data);
```

US 233 (std::span)

```cpp
#include <span>

void send_ints(std::span<int> data);

send_ints(variable);
```

US 233 (std::span)

```cpp
#include <span>


boost::container::vector<int>         a;
boost::container::small_vector<int>   b;
boost::container::stack_vector<int>   c;
boost::array<int, 1024>               d;
std::vector<int>                      e;
std::array<int, 1024>                 f;


void send_ints(std::span<int> data); /* ??? */
```

NTTP

NTTP

```
#include <array>

template <std::array<char, 1024> A>
void foo();
```

Много других багов

Цифры (P1889)

Цифры

Цифры

– SG6 Numerics долго работала

Цифры

- SG6 Numerics долго работала
- Новые предложения по классам чисел не пересылались в LEWG ...

Цифры

- SG6 Numerics долго работала
- Новые предложения по классам чисел не пересылались в LEWG …
- … хотели убедиться, что вместе они работают верно (см. P0101)

Цифры

- SG6 Numerics долго работала
- Новые предложения по классам чисел не пересылались в LEWG …
- … хотели убедиться, что вместе они работают верно (см. P0101)
- Накопился большой объём предложений

Цифры

- SG6 Numerics долго работала
- Новые предложения по классам чисел не пересылались в LEWG …
- … хотели убедиться, что вместе они работают верно (см. P0101)
- Накопился большой объём предложений
- P1889 «Numbers» или «SG6 черновик»

Цифры. Примеры.

Цифры (пример 1)

```
using int_t = /* ??? */;

int_t count_atoms(auto... args);
```

Цифры

Цифры

– В человеческой тушке ~10^13 атомов

Цифры

– В человеческой тушке ~10^13 атомов

– Планета Земля состоит из ~2*10^26 атомов

Цифры (пример 1)

```
using int_t = /* ??? */;

int_t count_atoms(auto... args);
```

Цифры (пример 1)

```
using int_t = std::integer;

int_t count_atoms(auto... args);
```

Цифры (пример 2)

```
using int_t = std::integer /* too slow */;

int_t count_atoms(auto... args);
```

Цифры (пример 2)

```
using int_t = std::wide_uint<128>;

int_t count_atoms(auto... args);
```

Цифры (пример 2)

```
using int_t = std::wide_uint<128>; // WRONG!

int_t count_atoms(auto... args);
```

Цифры (пример 3)

```
using int_t = std::least_2int<128>; // std::wide_uint<128> !!!

int_t count_atoms(auto... args);
```

Цифры (пример 4)

```
bool try_compute_something_very_important(
      unsigned a, unsigned b, unsigned c, unsigned& result)
{
  result = a * b + c;
  return true;
}
```

Цифры (пример 4)

```
bool try_compute_something_very_important(
      unsigned a, unsigned b, unsigned c, unsigned& result)
{
  result = a * b + c;
  return true; // TODO: detect overflows and return false!
}
```

Цифры (пример 4)

```
bool try_compute_something_very_important(
      unsigned a, unsigned b, unsigned c, unsigned& result)
{
  return !std::overflow_mul(&result, a, b)
         && !std::overflow_add(&result, result, c);
}
```

Цифры. Вместо итогов

Есть замечания к C++20?

Too late!

Есть идеи для C++23?

https://stdcpp.ru

C++
РГ21 C++ РОССИЯ

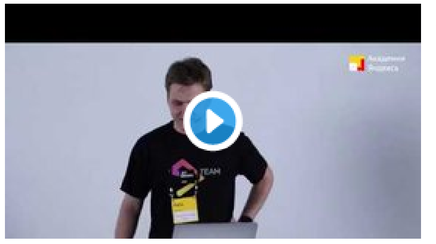
RSS Ru En yndx-antoshkka, выход

О проекте
Новости
Предложения
Вопросы и ответы
Инструкция по подготовке proposal

stdcppru
@stdcppru

Для тех, кто пропустил встречу в декабре:
– Обзор встречи Комитета по стандартизации C++ в Сан-Диего, США – Антон Полухин
youtube.com/watch?v=QaDO9L…
– Модули в C++ – Дмитрий Кожевников
youtube.com/watch?v=p8MkTJ…

YouTube @YouTube



9 янв. 2019 г.

Помогаем готовить предложения для защиты перед рабочей группой Комитета ISO C++

Новости

27 ноября 2018 29 мая 2018 14 марта 2018

Спасибо

Полухин Антон

Эксперт-разработчик C++

antoshkka@gmail.com

antoshkka@yandex-team.ru

https://github.com/apolukhin

https://stdcpp.ru/

РГ21 C++ РОССИЯ