Яндекс Такси

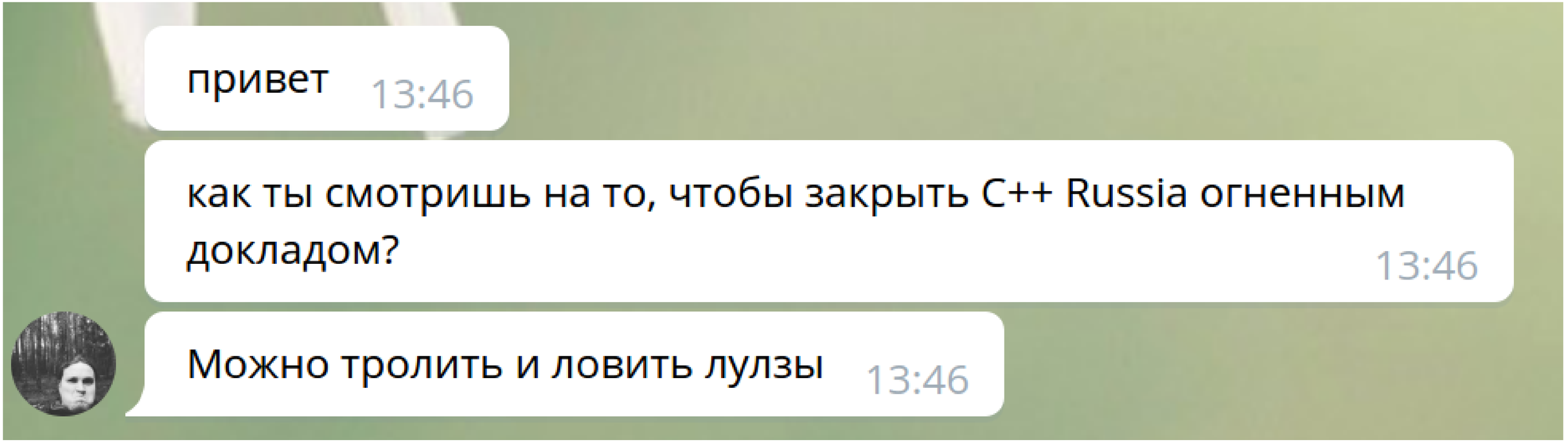
Незаменимый C++

Полухин Антон

Antony Polukhin

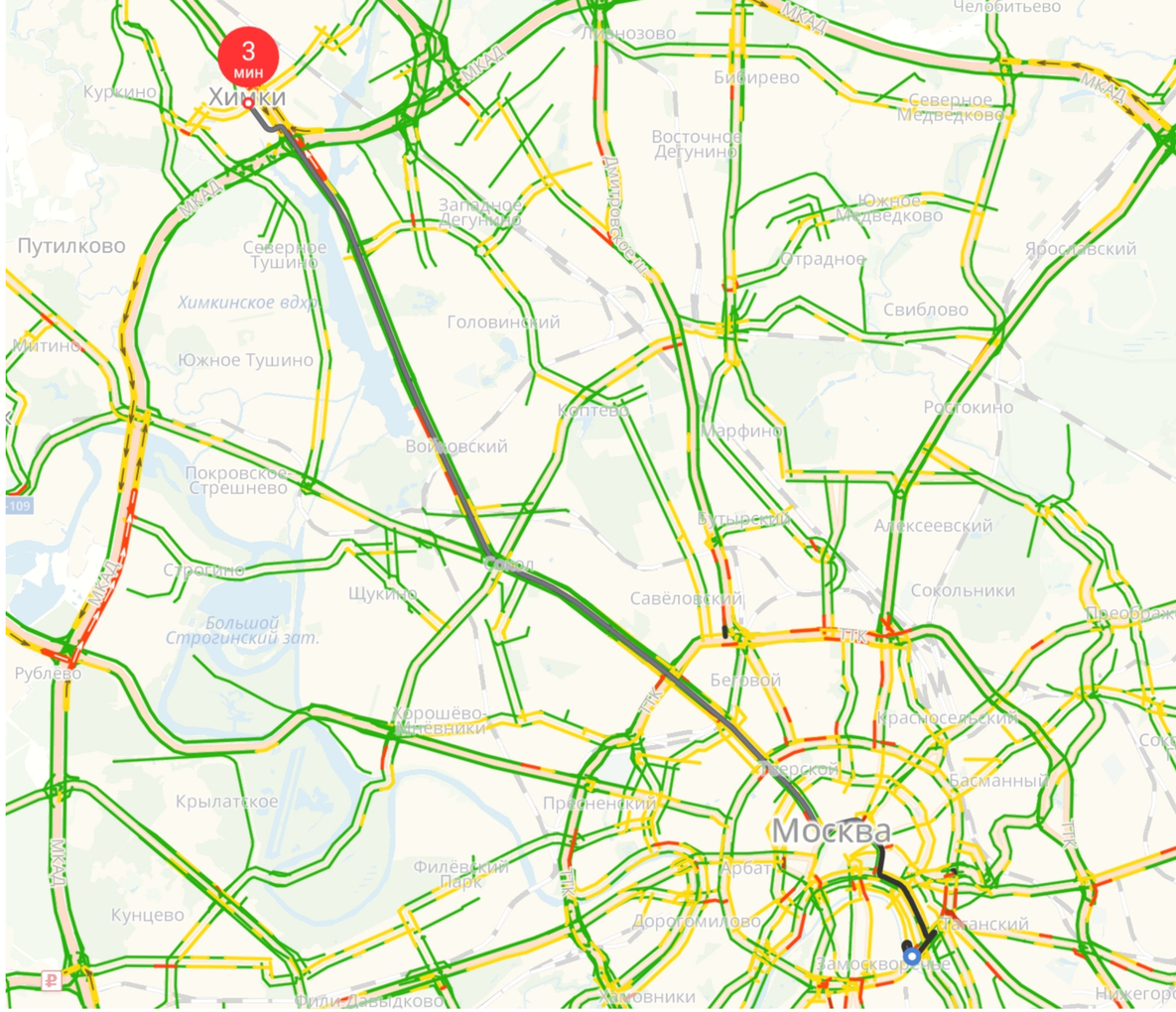
Яндекс Такси

Disclaimer


привет 13:46
как ты смотришь на то, чтобы закрыть C++ Russia огненным докладом? 13:46
Можно тролить и ловить лулзы 13:46

Содержание

- На C++ больше не пишут?
- C++ vs. Asm
- C++ vs. Rust
- C++ vs. Go
- C++ vs. (Java + C#)
- Слабые места C++
- Что с этим делать



Заблуждение №1

На C++ больше не пишут программ

Программы

Программы

- Поисковые движки

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)
- «Виртуальные машины»

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)
- «Виртуальные машины»
- Научные программы (CERN и Бозон Хиггса)

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)
- «Виртуальные машины»
- Научные программы (CERN и Бозон Хиггса)
- Части ОС (Драйверы, userspace)

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)
- «Виртуальные машины»
- Научные программы (CERN и Бозон Хиггса)
- Части ОС (Драйверы, userspace)
- Автопром

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)
- «Виртуальные машины»
- Научные программы (CERN и Бозон Хиггса)
- Части ОС (Драйверы, userspace)
- Автопром
- Заводы

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)
- «Виртуальные машины»
- Научные программы (CERN и Бозон Хиггса)
- Части ОС (Драйверы, userspace)
- Автопром
- Заводы
- Биржа

Программы

- Поисковые движки
- Большинство высоконагруженных програм (Такси)
- Игры
- Браузеры
- Спецэффекты и анимация
- Компиляторы (не только компиляторы для C++)
- «Виртуальные машины»
- Научные программы (CERN и Бозон Хиггса)
- Части ОС (Драйверы, userspace)
- Автопром
- Заводы
- Биржа
- Офисные приложения

ОК, чем C++ привлекает сегодня?

C++

C++

+ zero-overhead

C++

+ zero-overhead
+ неограниченные возможности

C++

+ zero-overhead
+ неограниченные возможности
+ поддержка огромного количества платформ

C++

+ zero-overhead
+ неограниченные возможности
+ поддержка огромного количества платформ
+ безопасность

C++

+ zero-overhead
+ неограниченные возможности
+ поддержка огромного количества платформ
+ безопасность?

C++

+ zero-overhead
+ неограниченные возможности
+ поддержка огромного количества платформ
+ безопасность?
+ небольшой рантайм

Сравним с другими языками!

C++ vs. Asm

C++ vs. Asm

C++ vs. Asm

+ Ассемблер позволяет выжать максимум из железа

C++ vs. Asm

+ Ассемблер позволяет выжать максимум из железа

- непереносимый код

C++ vs. Asm

+ Ассемблер позволяет выжать максимум из железа

- непереносимый код

- очень медленная разработка

C++ vs. Asm

+ Ассемблер позволяет выжать максимум из железа

- непереносимый код
- очень медленная разработка
- не всегда быстрее

C++ vs. Asm

+ Ассемблер позволяет выжать максимум из железа

- непереносимый код
- очень медленная разработка
- не всегда быстрее

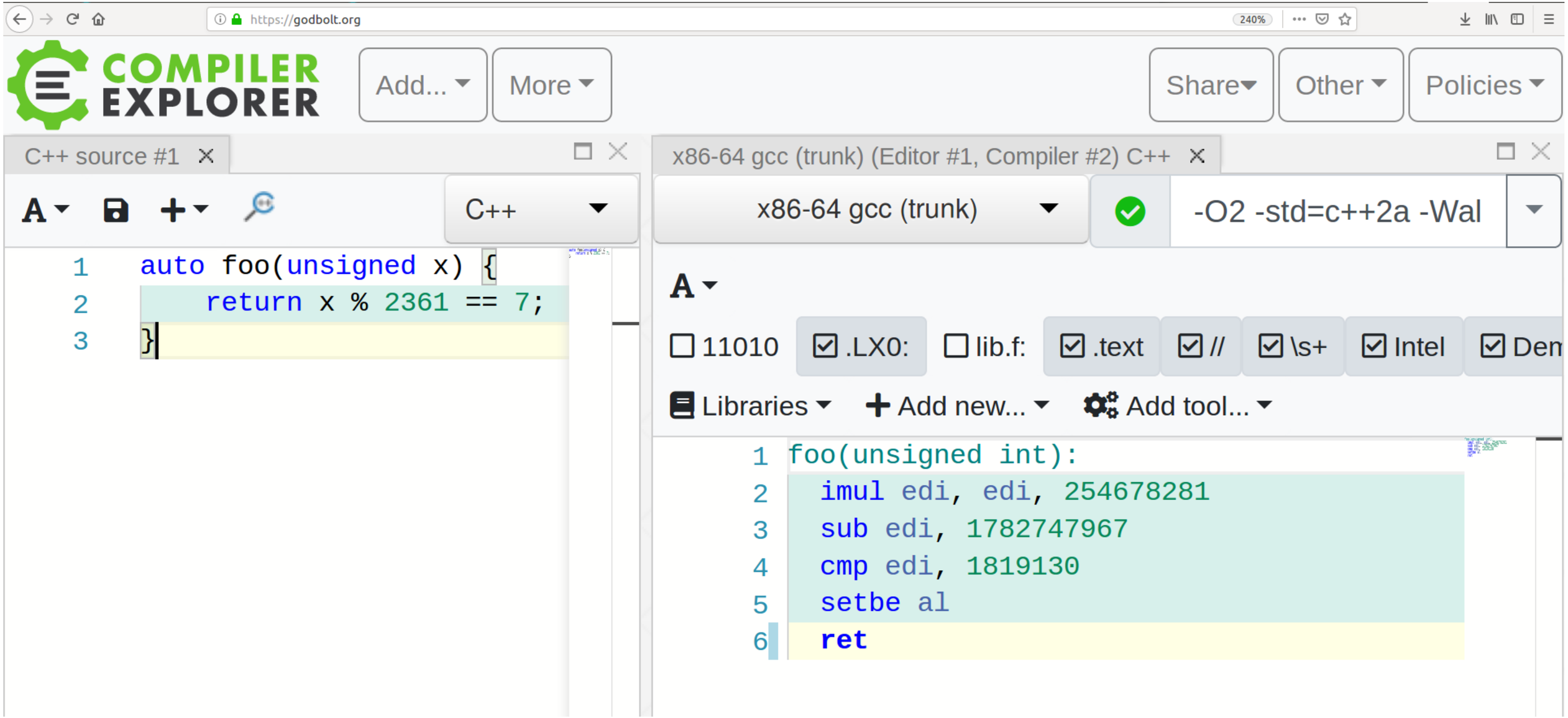
Заблуждение №2

Большие программы на C++ работают медленнее чем на ASM

C++ vs. Asm (или как завалить Asm разработчика)

```
x % 2361 == 7
```

C++ vs. Asm (или как завалить Asm разработчика)



C++ vs. Asm (или как завалить Asm разработчика)

```
11750      unsigned).
11751      (X * C3) + C4 <= 2 * C4, where
11752      C3 is modular multiplicative inverse of (unsigned) C1 and 1<<prec and
11753      C4 is ((1<<(prec - 1) - 1) / C1).
11754      If C1 is even, S = ctz(C1), use
11755      ((X * C3) + C4) r>> S <= (C4 >> (S - 1))
11756      where C3 is modular multiplicative inverse of (unsigned)(C1>>S) and 1<<prec
11757      and C4 is ((1<<(prec - 1) - 1) / (C1>>S)) & (-1<<S).
11758
11759      See the Hacker's Delight book, section 10-17.  */
11760   enum tree_code
11761   maybe_optimize_mod_cmp (enum tree_code code, tree *arg0, tree *arg1)
11762   {
```

C++ vs. Rust

C++ vs. Rust

C++ vs. Rust

+ Отличная безопасность?

C++ vs. Rust

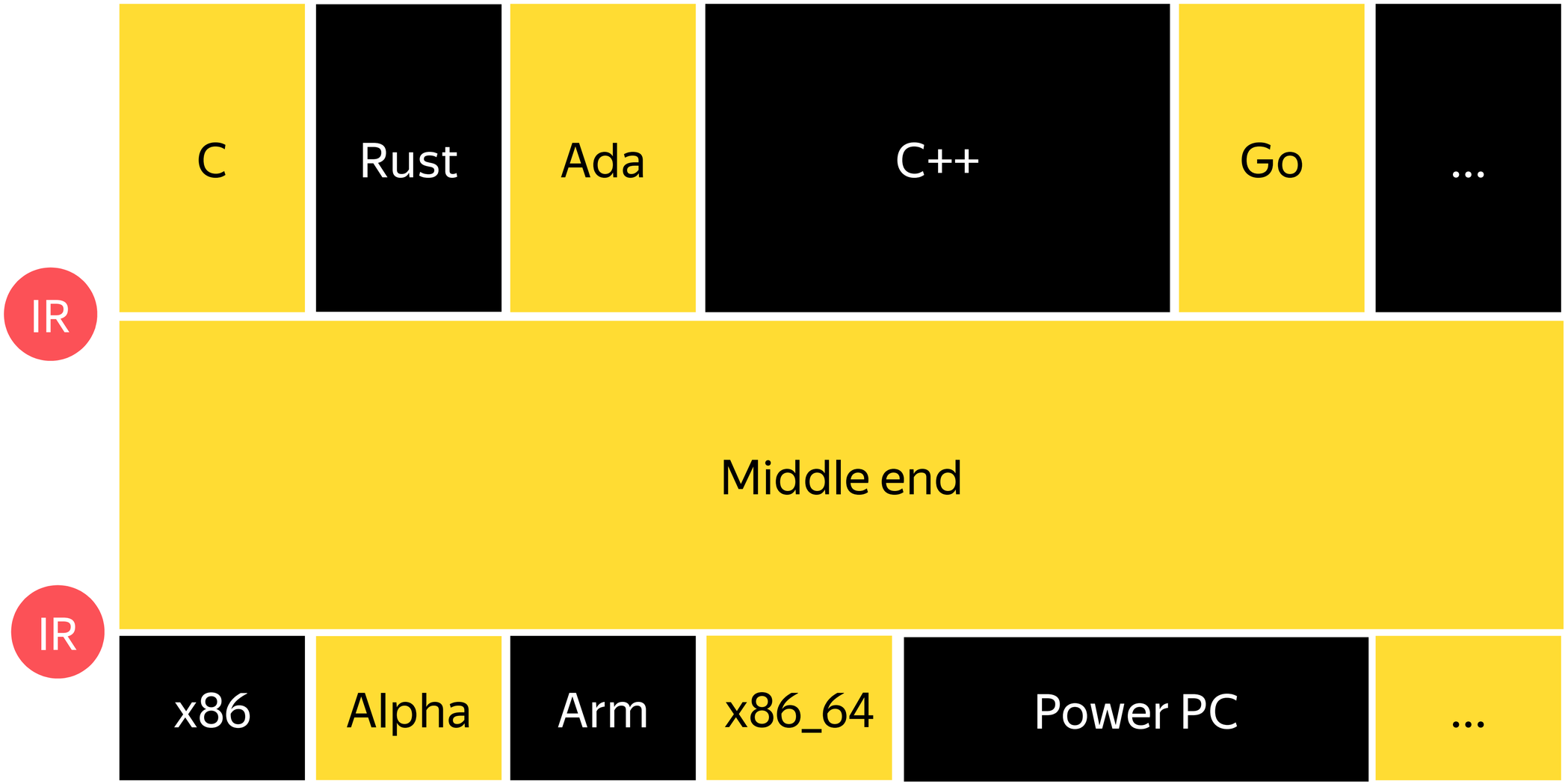
+ Отличная безопасность?

+ Потенциальная возможность оптимизировать лучше C++

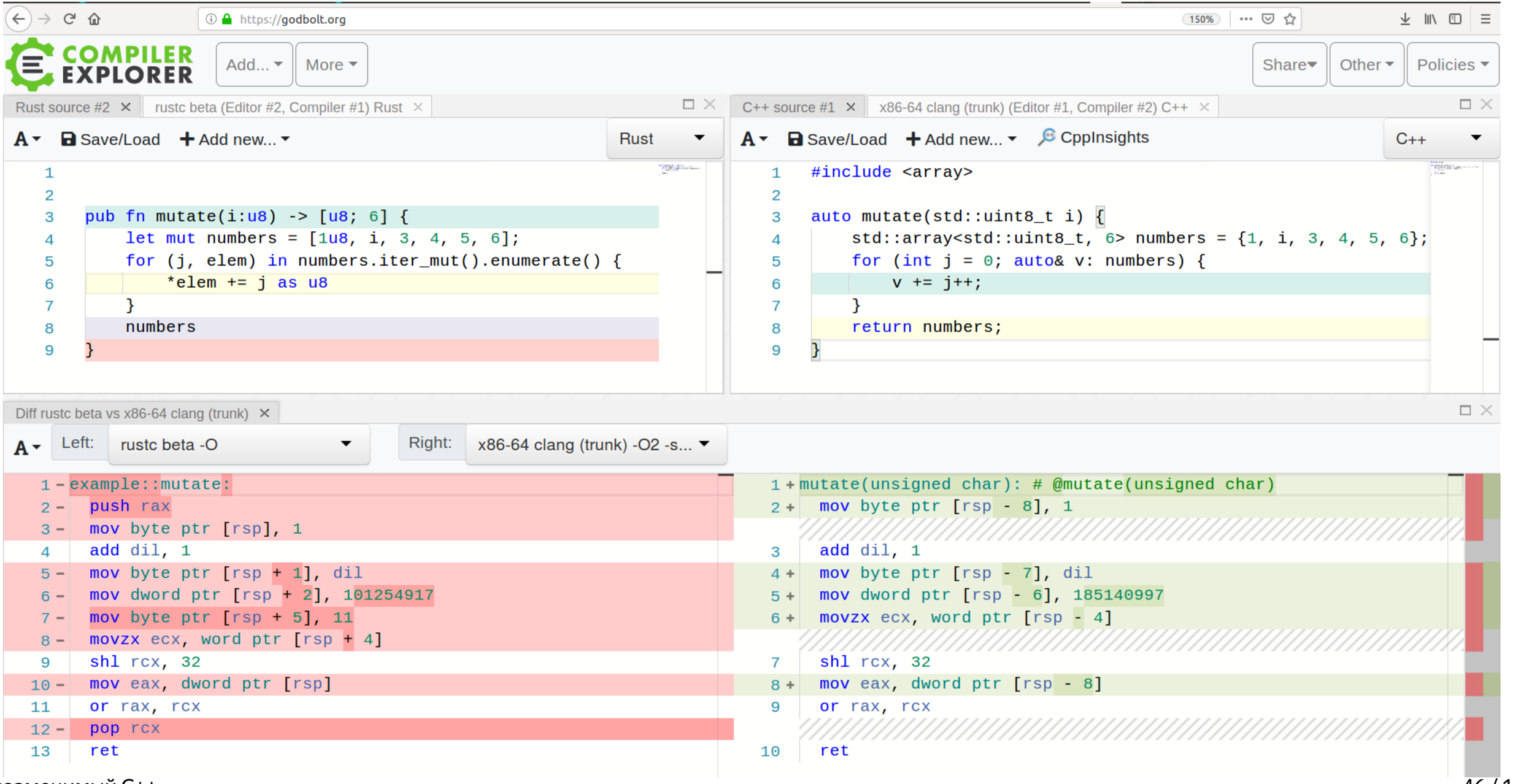
Заблуждение №3

Rust более/такой же производительный как C++

Анатомия компилятора (упрощённо)



C++ vs. Rust


https://godbolt.org
COMPILER EXPLORER
Add...
More
Share
Other
Policies
Rust source #2
rustc beta (Editor #2, Compiler #1) Rust
Save/Load
Add new...
Rust
pub fn mutate(i:u8) -> [u8; 6] {
let mut numbers = [1u8, i, 3, 4, 5, 6];
for (j, elem) in numbers.iter_mut().enumerate() {
*elem += j as u8
}
numbers
}
C++ source #1
x86-64 clang (trunk) (Editor #1, Compiler #2) C++
Save/Load
Add new...
CppInsights
C++
#include <array>
auto mutate(std::uint8_t i) {
std::array<std::uint8_t, 6> numbers = {1, i, 3, 4, 5, 6};
for (int j = 0; auto& v: numbers) {
v += j++;
}
return numbers;
}
Diff rustc beta vs x86-64 clang (trunk)
Left:
rustc beta -O
Right:
x86-64 clang (trunk) -O2 -s...
example::mutate:
push rax
mov byte ptr [rsp], 1
add dil, 1
mov byte ptr [rsp + 1], dil
mov dword ptr [rsp + 2], 101254917
mov byte ptr [rsp + 5], 11
movzx ecx, word ptr [rsp + 4]
shl rcx, 32
mov eax, dword ptr [rsp]
or rax, rcx
pop rcx
ret
mutate(unsigned char): # @mutate(unsigned char)
mov byte ptr [rsp - 8], 1
add dil, 1
mov byte ptr [rsp - 7], dil
mov dword ptr [rsp - 6], 185140997
movzx ecx, word ptr [rsp - 4]
shl rcx, 32
mov eax, dword ptr [rsp - 8]
or rax, rcx
ret

C++ vs. Rust

Вроде норм.

C++ vs. Rust

Вроде норм.

Переходим на Rust?

C++ vs. Rust

Вроде норм.

Переходим на Rust?

Oh, wait!..

C++ vs. Rust

Вроде норм.

Переходим на Rust?

Oh, wait!..

C → C++ — noop

C → Rust — PAIN!!!!!!!

C → Rust

- unsafe {} → нет безопасности

C → Rust

- unsafe {} → нет безопасности
- Нет возможности использовать C headers
  - Надо генерировать свои
    - Обновление библиотек — БОЛЬ!
    - Надо headers руками допатчивать
    - Мучительные страдания с borrow checker на сложных C проектах [https://hackernoon.com/why-im-dropping-rust-fd1c32986c88]

C → C++

Берёте и используете C headers

- Оборачиваете в классы по необходимости

Заблуждение №4

Программа написанная на языке ~~Rust~~ X не содержит ошибок

Anything

- unsafe или аналоги → нет безопасности

Anything

- unsafe или аналоги → нет безопасности
- если ваша программа компилится, это ещё не значит что всё ОК

C++ vs. Go

C++ vs. Go

C++ vs. Go

- <*великое множество*>

C++ vs. Go

- <*великое множество*>

+ Асинхронность и многопточность на основе корутин

C++ vs. Go

- <*великое множество*>

+ Асинхронность и многопточность на основе корутин

- Boost.Fibers

C++ vs. Go

- <*великое множество*>

+ Асинхронность и многопточность на основе корутин

- Boost.Fibers
- Yandex.Taxi userver

C++ vs. Go

- <*великое множество*>

+ Асинхронность и многопточность на основе корутин

- Boost.Fibers
- Yandex.Taxi userver
- Quantum

C++ vs. Go

- <*великое множество*>

+ Асинхронность и многопточность на основе корутин

- Boost.Fibers
- Yandex.Taxi userver
- Quantum
- Folly fibers

C++ vs. Go

- <*великое множество*>

+ Асинхронность и многопточность на основе корутин

- Boost.Fibers
- Yandex.Taxi userver
- Quantum
- Folly fibers
- Coroutines TS

C++ vs. Go

- <*великое множество*>

+ Асинхронность и многопточность на основе корутин

- Boost.Fibers
- Yandex.Taxi userver
- Quantum
- Folly fibers
- Coroutines TS
- C++20

Python vs. Go

Go скорее конкурент Python, чем C++

Заблуждение №5

Бенчмарки показывают что прогарммы на X быстрее C++

Добро пожаловать в мир «честных» бенчмарков!

Типичные ошибки

- Отключается сборщик мусора

Типичные ошибки

- Отключается сборщик мусора
- Код написан не на C++

Типичные ошибки

- Отключается сборщик мусора
- Код на C++ написан в стиле float* f = new float;

Типичные ошибки

- Отключается сборщик мусора
- Код на C++ написан в стиле float* f = new float;
- На X написана другая программа

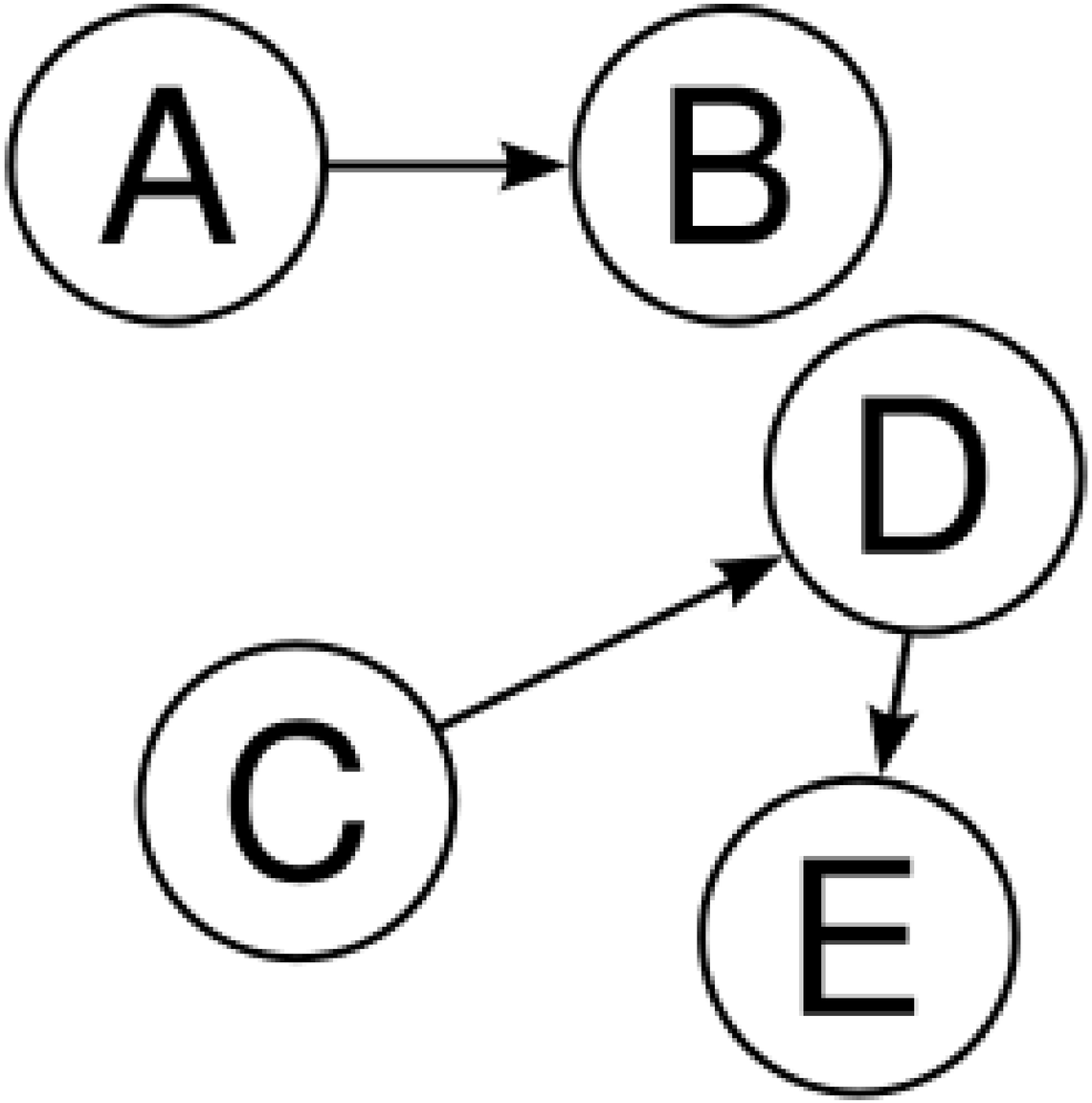
Типичные ошибки

- Отключается сборщик мусора
- Код на C++ написан в стиле float* f = new float;
- На X написана другая программа
- Тестируется библиотека а не язык

Заблуждение №6

Сборщик мусора не добавляет накладных расходов

Mark and sweep



structures

```
struct list_node {
    list_node* next;
    list_node* prev;
};
struct slist_node {
    slist_node* next;
};
```

structures

```
vector<void*> root;
```

structures

```
vector<void*> root; // root[0] — это slist_node или list_node?
```

structures

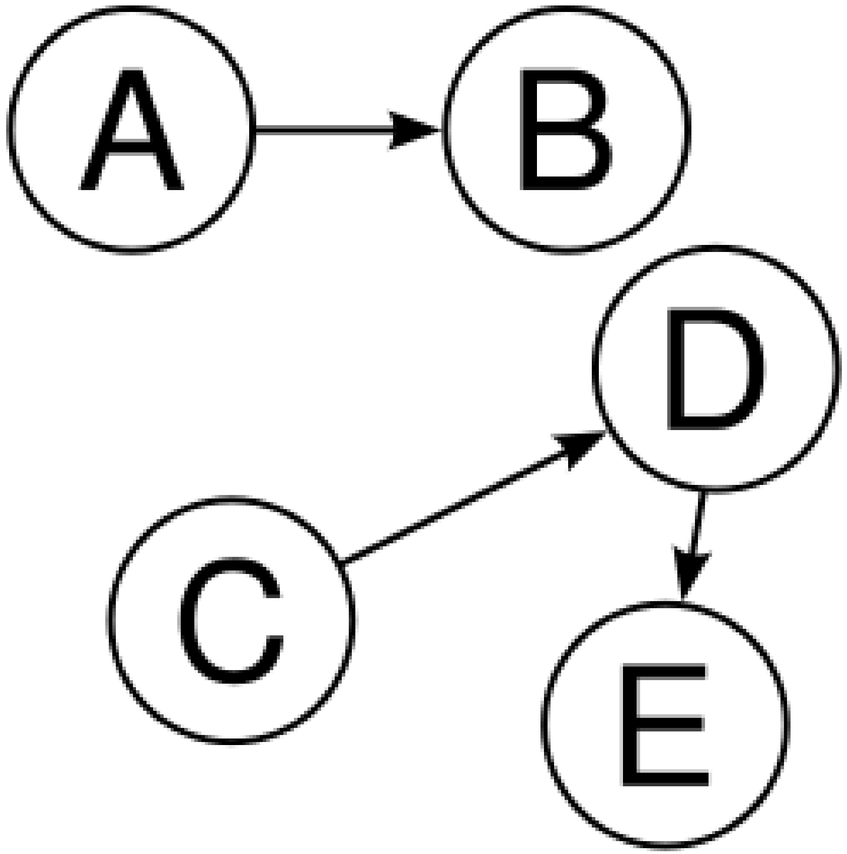
```
vector<void*> root; // Сколько указателей и где они?

vector<Object*> root;
```

structures

```
struct list_node {
    __meta vptr;
    list_node* next;
    list_node* prev;
};
struct slist_node {
    __meta vptr;
    slist_node* next;
};
```

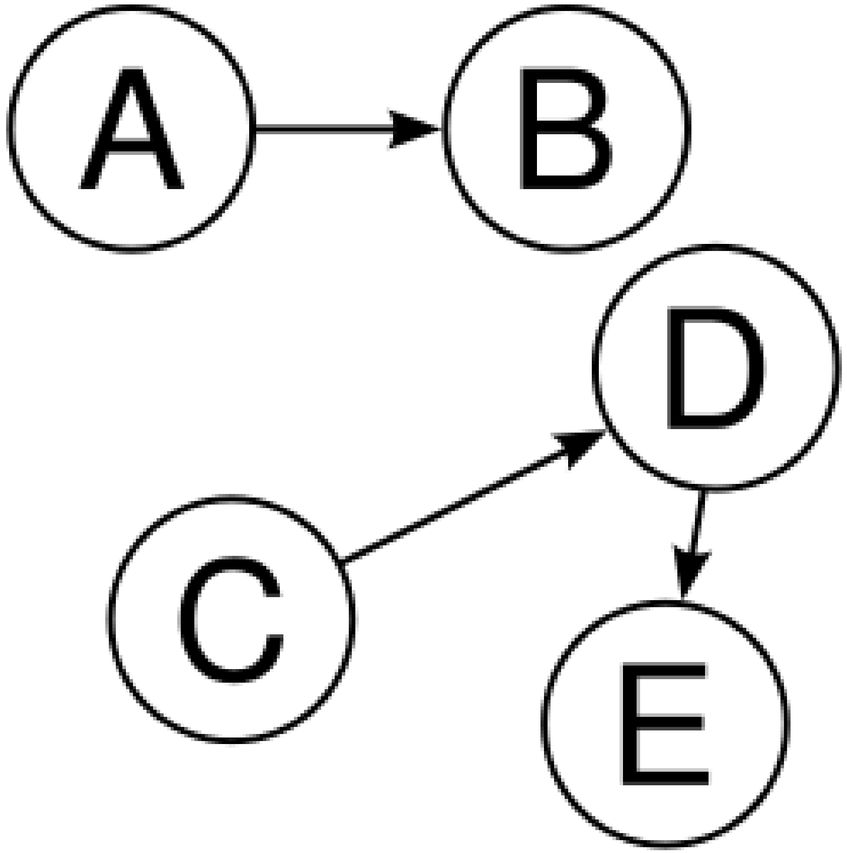
Stop the world

- Все потоки останавливаются



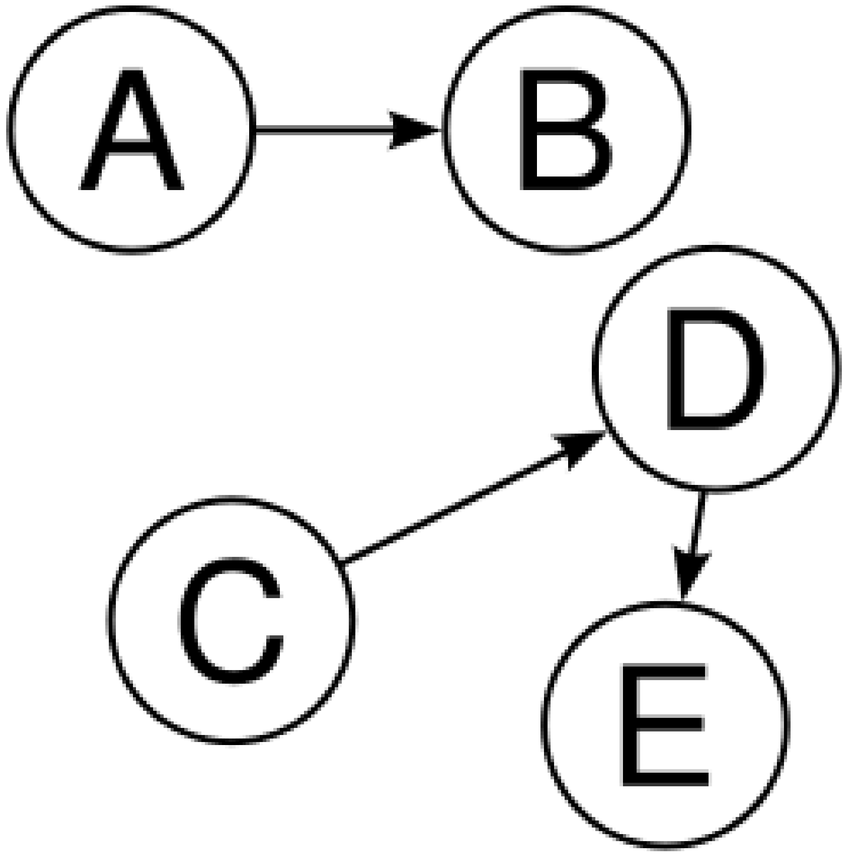
Stop the world

- Все потоки останавливаются
  - Проходим по всем узлам графа, помечая достижимые узлы

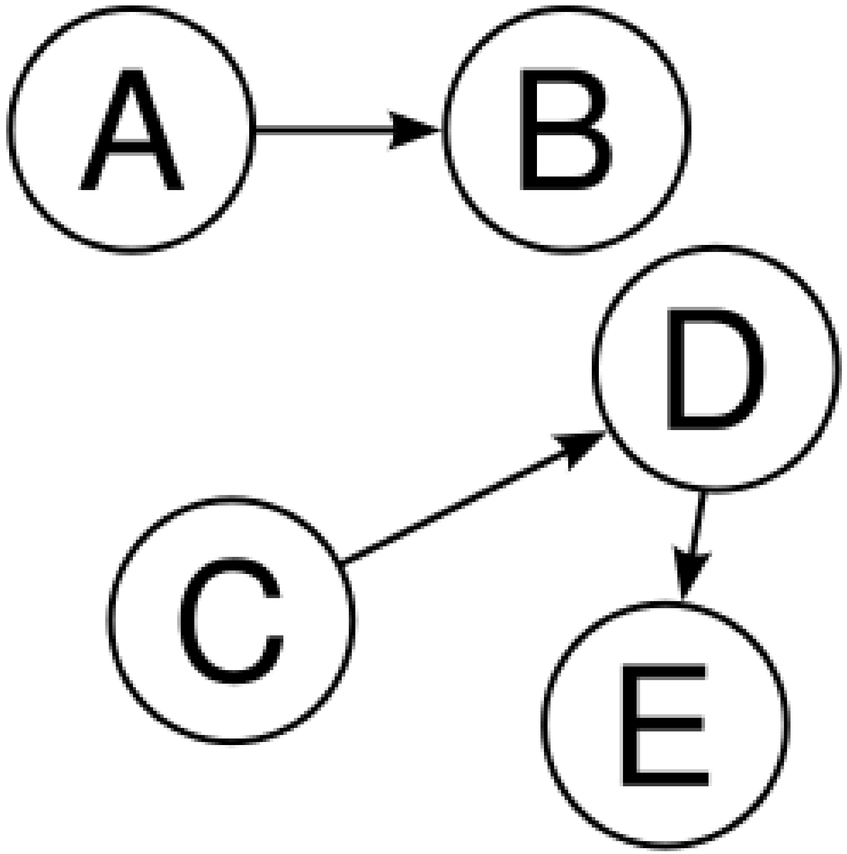


Stop the world

- Все потоки останавливаются
  - Проходим по всем узлам графа, помечая достижимые узлы
  - ????

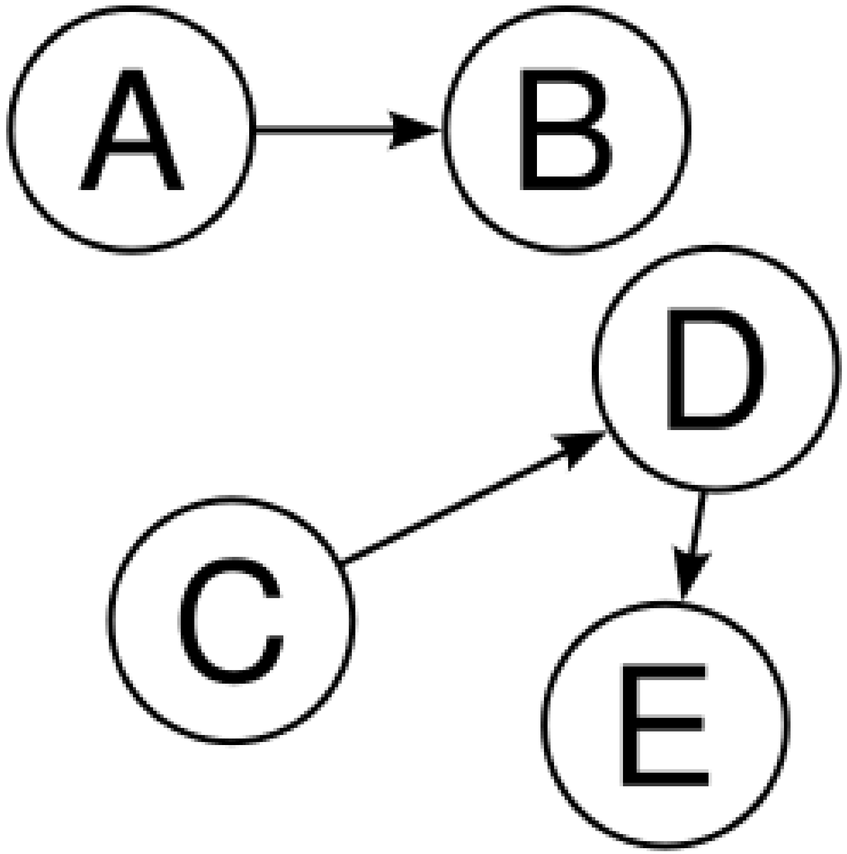


Stop the world



- Все потоки останавливаются
  - Проходим по всем узлам графа, помечая достижимые узлы
  - ????
  - Profit

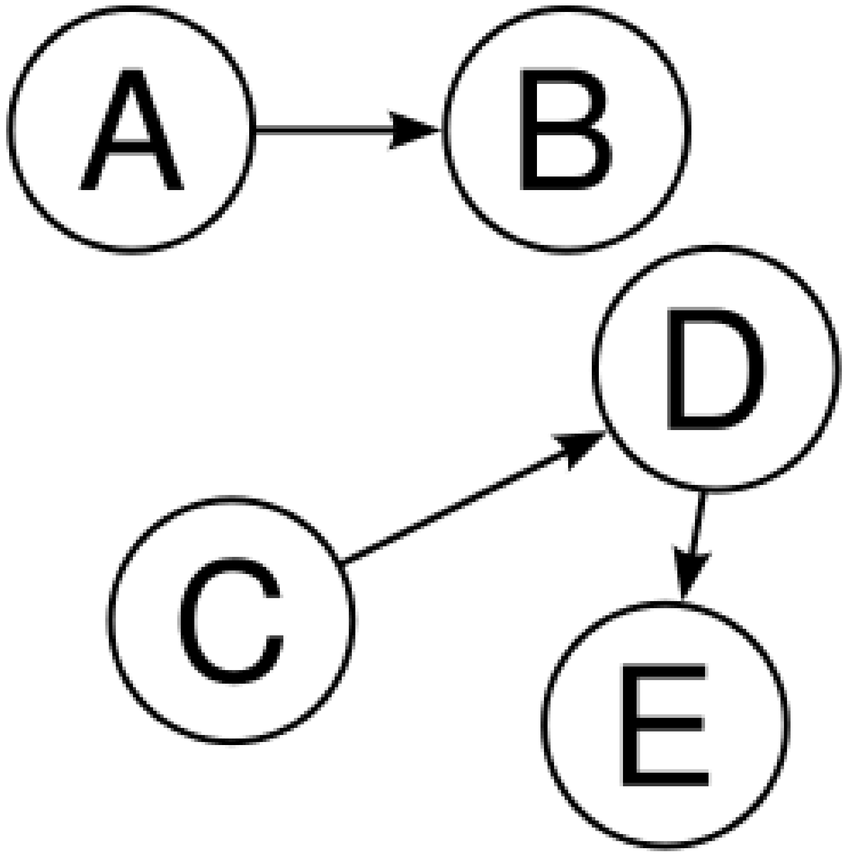
Stop the world



- Все потоки останавливаются
  - Проходим по всем узлам графа, помечая достижимые узлы
  - ????
  - Profit
- На современном железе больше 1 ядра!

Stop the world

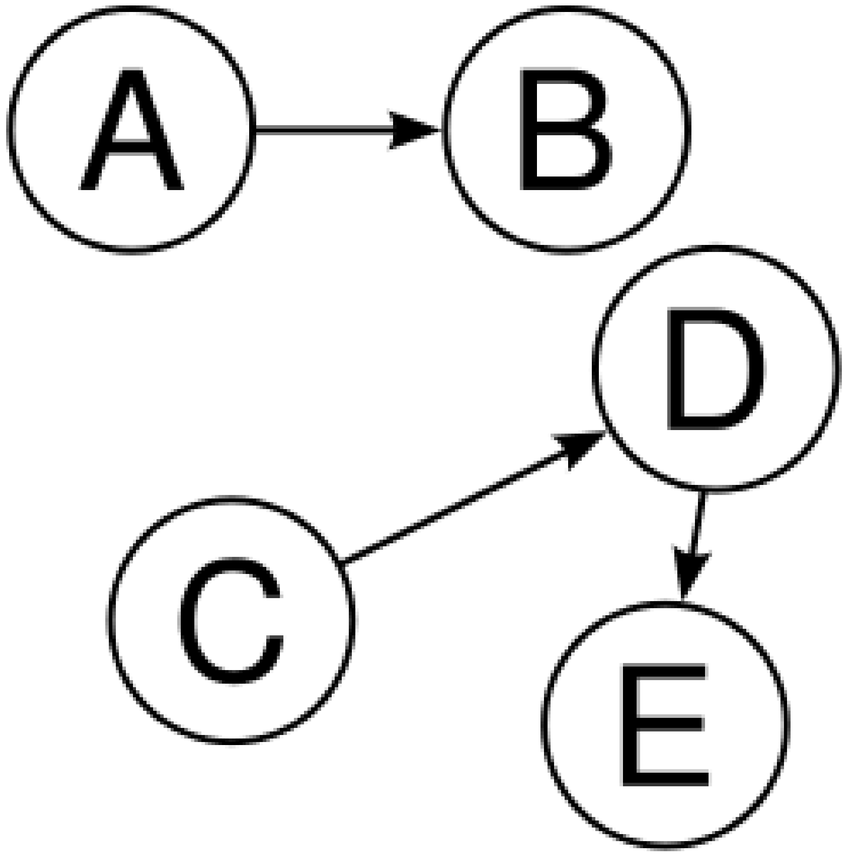
- Все потоки останавливаются
  - Проходим по всем узлам графа, помечая достижимые узлы
  - ????
  - Profit



- На современном железе больше 1 ядра!
- При каждой сборке мусора мы проходимся по всем узлам → постоянно перепроверяя живые узлы

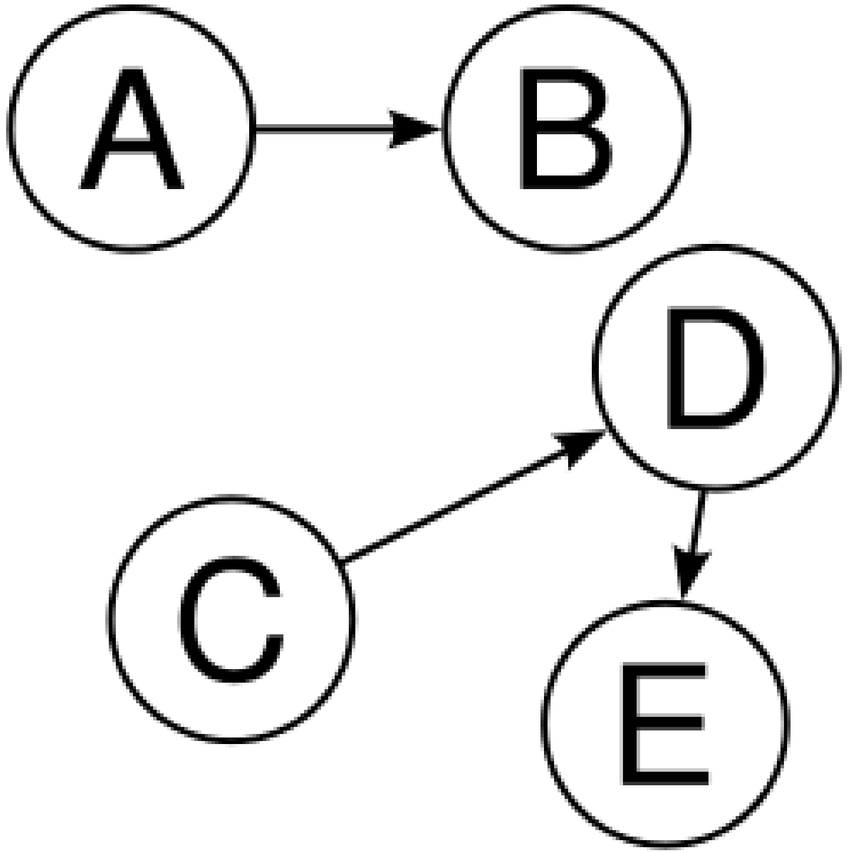
No stop the world

- Ничего не останавливается



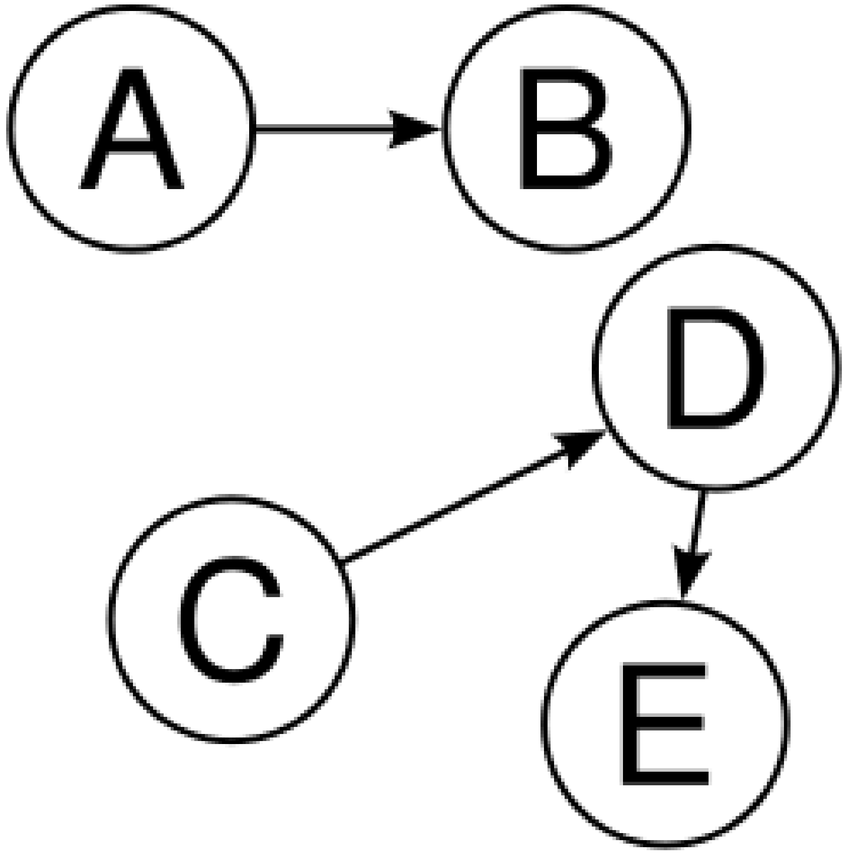
No stop the world

- Ничего не останавливается
  - Многопоточно разбираем мусор

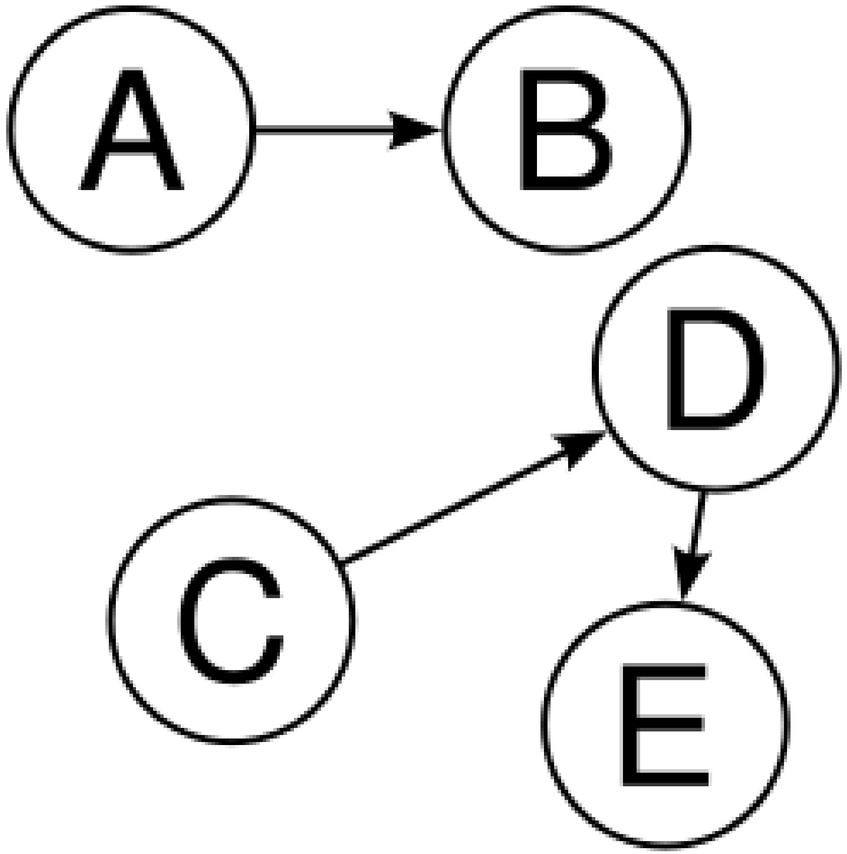


No stop the world

- Ничего не останавливается
  - Многопоточно разбираем мусор
  - Ура!

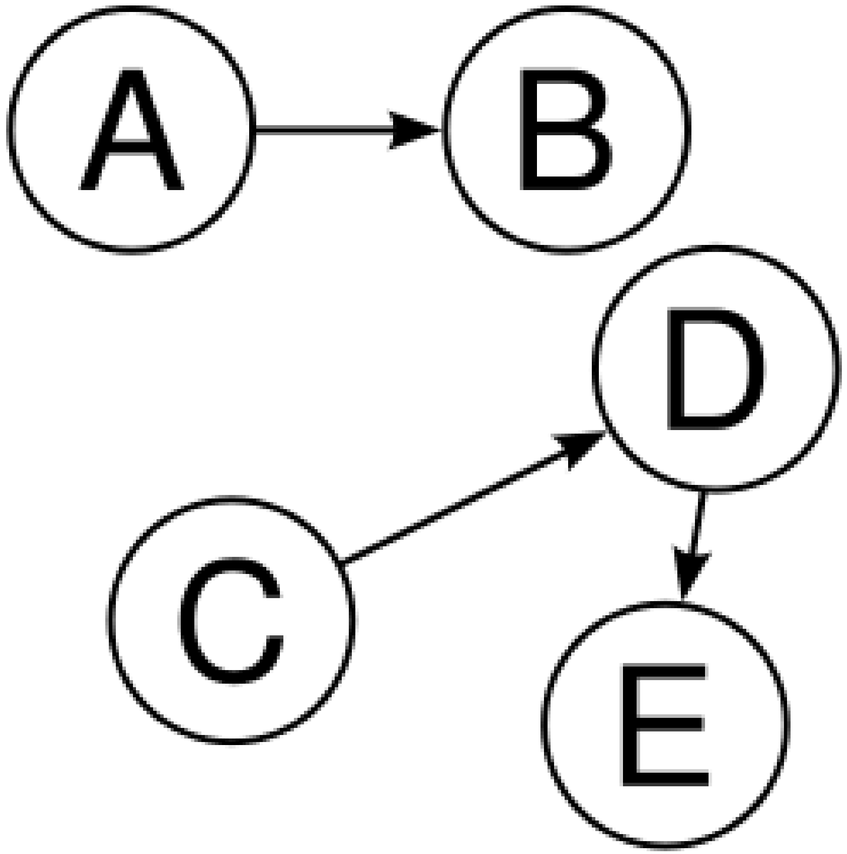


No stop the world



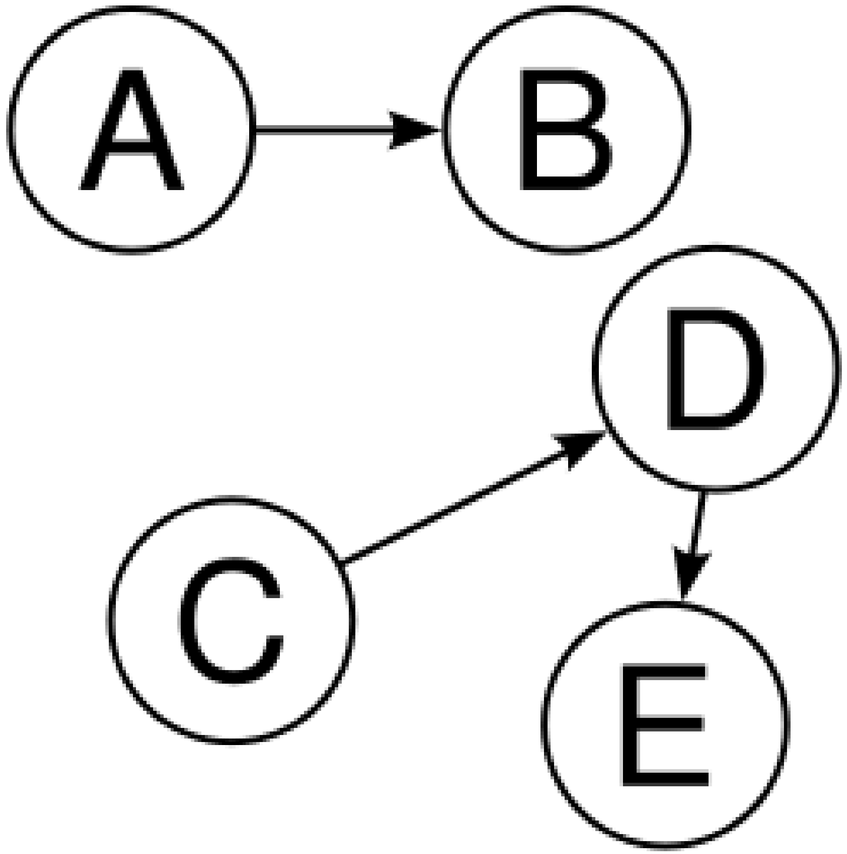
- Ничего не останавливается
  - Многопоточно разбираем мусор
  - Ура!
  - НО

No stop the world



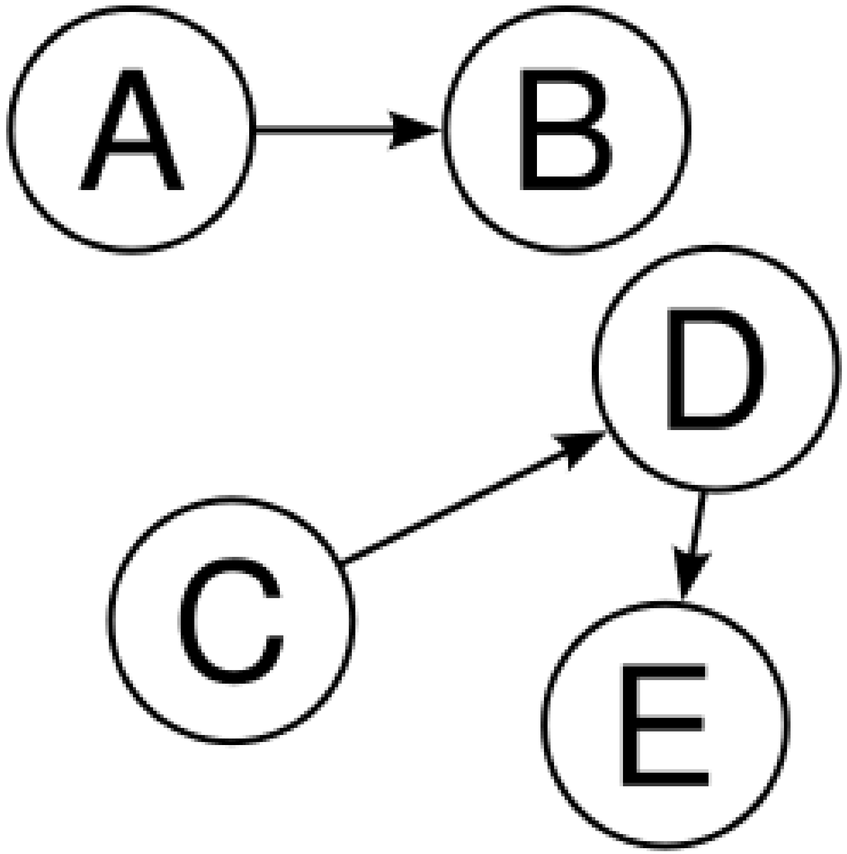
- Ничего не останавливается
  - Многопоточно разбираем мусор
  - Ура!
  - НО
  - Нам нужно синхронизировать все потоки через атомарные инструкции

No stop the world



- Ничего не останавливается
  - Многопоточно разбираем мусор
  - Ура!
  - НО
- Нам нужно синхронизировать все потоки через атомарные инструкции
- Это очень дорого, если не мы последними трогали ресурс

No stop the world



- Ничего не останавливается
  - Многопоточно разбираем мусор
  - Ура!
  - НО
- Нам нужно синхронизировать все потоки через атомарные инструкции
- Это очень дорого, если не мы последними трогали ресурс
- Мы делаем это при каждой сборке мусора!

+ Не проходим лишние разы

structures

```
struct list_node {
    shared_ptr<list_node> next;     // atomic refcounter
    shared_ptr<list_node> prev;     // atomic refcounter
};
struct slist_node {
    shared_ptr<slist_node> next;    // atomic refcounter
};
```

Это ещё не всё!

```
shared_ptr<Object> some;
```

Это ещё не всё!

```
shared_ptr<Object> some;          // Скоро будет больно!
```

Это ещё не всё!

```
shared_ptr<Object> some;           // Скоро будет больно!
shared_ptr<Object> prev = some.a; // ...
```

Это ещё не всё!

```
shared_ptr<Object> some;           // Скоро будет больно!
shared_ptr<Object> prev = some.a; // ...
prev.a = some;                     // Циклические ссылки!
```

C++ vs. (Java + C#)

C# ?? Java

C# ?? Java

- См. «Сборщики Мусора»

Logstash

– Программа для сбора, трансформации и складирования логов.

Бесплатное и очень популярное Open Source приложение на Java.

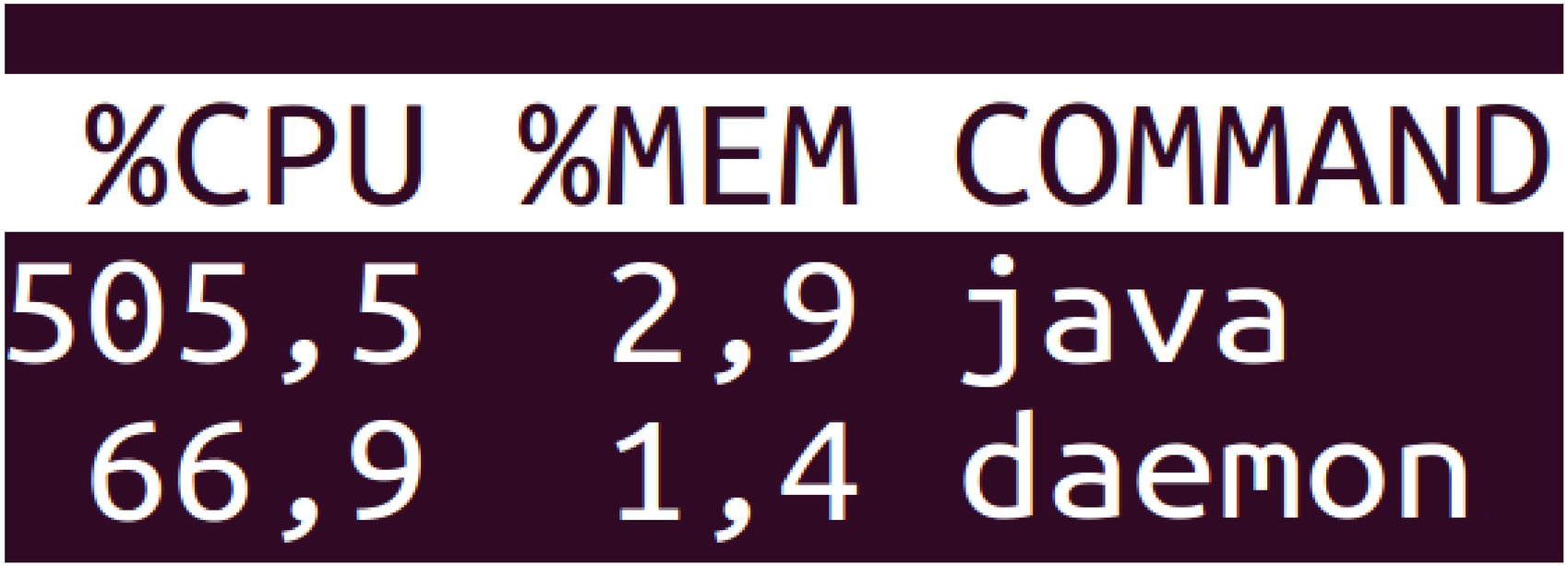
Что может пойти не так?

Logstash

– Программа для сбора, трансформации и складирования логов.

Бесплатное и очень популярное Open Source приложение на Java.

Что может пойти не так?



Слабые места C++

C++

C++

- скрытый высокий порог вхождения
  - неограниченные возможности
  - проблемы с безопасностью

C++

- скрытый высокий порог вхождения
  - неограниченные возможности
  - проблемы с безопасностью

- отсутствие изкоробочности
  - крошечная стандартная библиотека
  - отсутствие готовой инфраструктуры

ОК, а что делать то?

https://stdcpp.ru

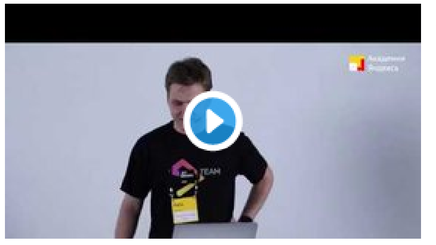
C++
РГ21 C++ РОССИЯ

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

Итоги

Итоги

Итоги

- C++ везде

Итоги

- C++ везде
- C++ крайне популярен

Итоги

- C++ везде
- C++ крайне популярен
- C++ - самый производительный инструмент для написания больших приложений

Итоги

- C++ везде
- C++ крайне популярен
- C++ - самый производительный инструмент для написания больших приложений
- Относитесь со скепсисом к бенчмаркам

Итоги

- C++ везде
- C++ крайне популярен
- C++ - самый производительный инструмент для написания больших приложений
- Относитесь со скепсисом к бенчмаркам
- Не верьте managed языкам, которые говорят что они быстрее C++

Итоги

- C++ везде
- C++ крайне популярен
- C++ - самый производительный инструмент для написания больших приложений
- Относитесь со скепсисом к бенчмаркам
- Не верьте managed языкам, которые говорят что они быстрее C++
- C++ не идеален!..

Итоги

- C++ везде
- C++ крайне популярен
- C++ - самый производительный инструмент для написания больших приложений
- Относитесь со скепсисом к бенчмаркам
- Не верьте managed языкам, которые говорят что они быстрее C++
- C++ не идеален!..
    - ...но это исправимо

Спасибо

Полухин Антон

Старший разработчик Yandex.Taxi

antoshkka@gmail.com

antoshkka@yandex-team.ru

https://github.com/apolukhin

https://stdcpp.ru/

РГ21 C++ РОССИЯ