

Автор:

09.03.18 06:46 -

[1.1](#) , [Аноним](#) , 09:51, 09/03/2018 [[ответить](#)] [[смотреть все](#)] +3 [+](#) / [-](#) Давайте теперь посмотрим на тесты скорости по сравнению с обычной библиотекой ...
весь текст скрыт

[
[показать](#)
]

[2.3](#)

[A.Stahl](#)

[^](#)

[ответить](#)

[смотреть все](#)

[показать ветку](#)

[+](#)

[-](#)

Зачем? Оно не ради скорости делалось. А ради... блин, я х.з. ради чего, но точно не ради какого

[3.5](#) ,

[Аноним](#) , 10:31, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

Для того чтобы показать, что тестирование скорости выполнения на [показано](#) если код...

[4.8](#) ,

[Bvz](#) , 10:39, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

А скорость всегда будет быстрее. Если отключить всякие проверки [показано](#) за гра...

[5.10](#) ,

[Аноним](#) , 10:44, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

Если отключить проверки, тогда speed нужен [раст ... [показать](#)]

[6.14](#) ,

[Аноним](#) , 11:02, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

В Rust все проверки при компиляции проходят.

Автор:
09.03.18 06:46 -

[7.15](#) ,

[A.Stahl](#) , 11:08, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

Тогда насколько они эффективны? Это мне подсказывает, что [90vaxx00](#) за пред...

[8.41](#) ,

[Аноним](#) , 16:28, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

Спасибо за Мудрость [ветрилокрыши](#) и знача[ьных Законов Вселенной. Хотелось бы зна...

[8.68](#) , [0xd34df00d](#) , 05:12, 10/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [[к модератору](#)]

[+](#)

/

[-](#)

Доказать, что индекс всегда внутри правильного диапазона, например. Про rust не знаю, но в каком-нибудь Idris это довольно легко.

[7.16](#) , [Аноним](#) , 11:20, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] +1 [+](#) / [-](#) ну это как

вставить `srccheck` какой-нибудь в препроцессор ...

весь текст скрыт

[

[показать](#)

]

[8.20](#)

[Аноним](#)

[^](#)

[-](#)

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

Замечательно Теперь [предложить](#) логическую цепочку Почему `srccheck` какой-нибу...

[9.21](#) ,

[Аноним](#) , 12:12, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

Как это не эффективно? Простая идея: лоялен к тем кто делает [скалоп](#)ину Сд...

[10.31](#) ,

[Аноним](#) , 14:48, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

Автор:
09.03.18 06:46 -

±

=

На раз поймается и valgrind не заскрит [[показать](#)]

[9.27](#) , [Crazy Alex](#) , 13:31, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] -1 [+](#) / [-](#) Он точнее, более живые анализаторы очень даже эффективен для отлова отхода от ...
весь текст скрыт

[

[показать](#)

]

[7.66](#)

,

[Alex](#)

, 04:13, 10/03/2018 [

[^](#)

][

[ответить](#)

][

[смотреть все](#)

] [

[к модератору](#)

]

±

/

=

Выход за границы массива бросает исключение в расте. Проверка в рантайм.

[2.4](#) , [Sunderland93](#) , 10:29, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [[показать ветку](#)]

+7

±

/

=

Давайте. Но только когда проект дорастёт до первого стабильного релиза.

[2.9](#) , [Аноним](#) , 10:39, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [[показать ветку](#)] +2 [+](#)

/

=

Скорее всего всё ок, т к в rust управление памятью не создаёт оверхеда все про...
весь текст скрыт

[

[показать](#)

][

[показать ветку](#)

]

[3.23](#)

[Аноним](#)

[^](#)

[ответить](#)

Автор:
09.03.18 06:46 -

[смотреть все](#)

±

—

Даже для динамических массивов, ребята, ну что же это такое [Почему](#) не ра...

[4.32](#) ,

[VladSh](#) , 14:54, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

±

—

То есть проверок для динамических массивов в рантайме лучше не делать?

[5.33](#) ,

[Аноним](#) , 15:07, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

±

—

Мы же про статический анализатор динамических массивов разве нет? :)

[4.35](#) , [pda](#) , 15:08, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [+](#) / [—](#) Иногда и для них. Rust предоставляет достаточно информации llvm, чтобы тот мог удалять проверку диапазона в некоторых случаях. Например (насколько я помню), когда вы делаете цикл по неизменяемому объекту (например неизменяемой ссылке) и только что получили длину массива, llvm может понять, что вы никогда не выйдете за границу диапазона и удалить проверку.

[5.53](#) ,

[angra](#) , 21:12, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

±

—

А если я в цикле присвою счетчику другое значение? А если обращусь к a[i+1] на последней ите

[6.60](#) ,

[Аноним](#) , 22:31, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

±

—

Дада, это всё очень просто отлавливается компилятором.

[7.63](#) ,

[angra](#) , 00:53, 10/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

Автор:
09.03.18 06:46 -

±
=

Разве что в самых простейших случаях, типа тела цикла из одного выражения. При наличии дру

[6.65](#), [Ordu](#), 01:58, 10/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [±](#) / [=](#) for mut i in 0..10
println!("{}", i); i += 1; резуль...

весь текст скрыт

[
[показать](#)
]

[7.67](#)

[angra](#)

^
=

[ответить](#)

[смотреть все](#)

[к модератору](#)

±
=

Ох уж эти титеретики

```
fn main() {  
    let a: [i32;5]=[1,2,3,4,5];  
    for i in 0..a.len() {  
        if i == a.len() - 1 {  
            println!("{}", a[i+1]);  
        }  
    }  
}
```

Compiling playground v0.0.1 (file:///playground)

Finished release [optimized] target(s) in 0.59 secs

Running 'target/release/playground'

thread 'main' panicked at 'index out of bounds: the len is 5 but the index is 5', /checkout/src/libcore/slice/mod.rs:726:5
note: Run with 'RUST_BACKTRACE=1' for a backtrace.

[3.48](#), [Аноним](#), 18:16, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [±](#) / [=](#) И как статически
проверить конкатенацию 2 строк вводимых юзером, например ...

весь текст скрыт

[
[показать](#)
]

[1.6](#)

Автор:

09.03.18 06:46 -

,
[фывфыв](#)
, 10:32, 09/03/2018 [
[ответить](#)
][
[смотреть все](#)
]
+6
[+](#)
/
[=](#)

> Код проекта написан на языке Rust

Бла-бла-бла, математика использует openlibm, а оно на C.

Алокаторы памяти и прочие низкоуровневые вещи регулярно пестрят unsafe блоками, что автоматически нивелирует все "плюшки" Rust'a.

[2.7](#) , [Аноним](#) , 10:36, 09/03/2018 [[^](#)] [
[ответить](#)
[смотреть все](#)
[показать ветку](#)
[+](#)
[=](#)
Есть примеры?

[3.11](#) , [Аноним](#) , 10:51, 09/03/2018 [[^](#)] [
[ответить](#)
[смотреть все](#)
[+](#)
[=](#)
Модуль реализации строк посмотри.

[3.12](#) , [Аноним](#) , 10:51, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [+](#) / [=](#) [https github com](https://github.com/redox-os/relibc/search?utf8=E2%9C%93qunsafe+type+...)

редокс-ос релибс поиск utf8 E2 9C 93 q unsafe type ...

весь текст скрыт

[
[показать](#)
]
[3.18](#)

,
[lfx](#)
, 11:33, 09/03/2018 [
[^](#)
[=](#)

Автор:

09.03.18 06:46 -

][
[ответить](#)
][
[смотреть все](#)
]
+8
[+](#)
/
=
Лучше молчи... Когда я сказал что без unsafe на rust далеко не уедешь, любители смузи меня тапками забросали. Им что то объяснять себе дороже.
[2.13](#) , [Аноним](#) , 10:53, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [[показать ветку](#)] -6
[+](#)
/
=
Вообще касательно самого языка. Идея хорошая, только зря они его назвали rust ...
весь текст скрыт
[
[показать](#)
][
[показать ветку](#)
]
[3.24](#)
[Аноним](#)
[^](#)
=
[ответить](#)
[смотреть все](#)
[+](#)
=
Надо ([кому?]) переписать игру Rust на язык Rust.

[2.37](#) , [pda](#) , 15:20, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [[показать ветку](#)] -1 [+](#) / [=](#)
И что Похоже вы как и многие не правильно понимают назначение unsafe так же ка...
весь текст скрыт
[
[показать](#)
][
[показать ветку](#)
]
[3.54](#)
[sadasd](#)
[^](#)
=
[ответить](#)
[смотреть все](#)
[+](#)

Автор:
09.03.18 06:46 -

=

О чем и речь, что в коде там дофига unsafe и смысла писать на Rust нет.

А вообще, еще до Rust'a были "безопасные" варианты C, Cyclon например.
Да и вообще, к тому-же GCC / Clang приписать пару пачку проверок и поставить их в -Werror.

[4.62](#) ,

[Тот же Аноним](#) , 00:31, 10/03/2018 [

[^](#)

[ответить](#)

[смотреть все](#)

[±](#)

=

А -fpermissive можно? А то на с++ такое бывает...

блаблабла предупреждение: декларация ничего не описывает [-fpermissive]

```
# define off_t long
```

Да можно typedef-ом, но за что препроцессор?

[3.55](#) , [_](#) , 21:46, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] [±](#) / [=](#) >Unsafe не что-то плохое в rust, в вполне сознательно сделанная вещь. Она позволяет создавать

... хоть как то работающие программы :) Не надо песен, у нас их тоже есть!

>С другой стороны, безопасный интерфейс к такому списку избавит вас от необходимости покрывать такими же тестами остальную программу. Там этим займётся компилятор.

Наивняк! :) Это уже было. Да - Java promise :) Чем всё кончилось? Покрывают тестами __всё__! И это правильно, ибо нефиг. И у вас тоже самое будет, всё суета сует :-)

>Хорошим примером такого подхода является языки C/C++, где программисты регулярно лажают от лени или усталости забывая вставлять проверку

Автор:
09.03.18 06:46 -

Оу-е бэйби! А ржавчики будут кодячить без усталости и лени! :-)))) Гив ми море! :)

Может оно и взлетит, но взлетит вопреки. Ибо нафиг оно - до сих пор не очевидно.

[4.64](#) , [pda](#) , 01:07, 10/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[+](#)

[-](#)

Вы так ничего и не поняли.

> Это уже было. Да - Java promise :)

Бессмысленное сравнение. В Java (визитной карточкой которого является NullPointerException) т

> Покрывают тестами __всё__!

Я не знаю как там в Java, не видел много Java, но судя по регулярным статьям о PVSSStudio - нет.

> А ржавчики будут кодячить без усталости и лени! :-))))

Как я говорил - вы ни фиги не поняли. "Ржавчики" такие же люди, по этому язык исключает ситу

[1.26](#) , [Crazy Alex](#) , 13:29, 09/03/2018 [[ответить](#)] [[смотреть все](#)] +4 [+](#) / [-](#) "Работа над проектом началась неделю назад и функциональность библиотеки пока сильно ограничена" - ну и смысл в таких новостях? Когда что-то хотя бы слегка живое будет - тогда и поговорим. И даже после этого - в реальном применении возникнет миллион нюансов, частных случаев и прочего, и только после возни с ними станет понятно, жизнеспособна идея или нет.

[2.28](#) , [Аноним](#) , 14:23, 09/03/2018 [[^](#)] [

Автор:
09.03.18 06:46 -

[ответить](#)
[смотреть все](#)
[показать ветку](#)

±
=

[3.30](#) , [Anonymous Coward](#) , 14:47, 09/03/2018 [[^](#)] [[ответить](#)] [[смотреть все](#)] +1 [±](#) / [=](#)
exit(EXIT_SUCCESS);

[4.36](#) , [A.Stahl](#) , 15:13, 09/03/2018 [[^](#)] [

[ответить](#)
[смотреть все](#)

±
=

Вот! Уже можно ещё одну новость писать про значительные улучшения, про сообщество с патча

[1.38](#) , [Аноним](#) , 15:30, 09/03/2018 [[ответить](#)] [[смотреть все](#)] -1 [±](#) / [=](#) Пиарят Rust как
могут Только лучше бы толковых библиотеке понаписали и примеров ...
весь текст скрыт

[
[показать](#)
]

[2.42](#)
[Аноним](#)
[^](#)
-
[ответить](#)
[смотреть все](#)
[показать ветку](#)

±
=

С нетерпением жду когда питонисты подхватят знамя и напишут стандартную либу для сишников
[показать ветку](#)

[3.45](#) , [Аноним84701](#) , 17:32, 09/03/2018 [[^](#)] [

[ответить](#)
[смотреть все](#)

±
=

> С нетерпением жду когда питонисты подхватят знамя и напишут стандартную либу для сишников

Там "знамя", AFAIK, немного не о том.

Поэтому питонистам придется сначала написать свою ОСь на питоне и Pylibc под нее, а потому у

Автор:

09.03.18 06:46 -

А вообще, некоторым опеннетчикам не угодишь:

Использовали сишный – "Фи! Не осилили даже стандартный ОСевой рантайм на своей ржавчине"

Начали писать на ржавчине – и опять все не так, "Блaсфeмия! Да как они смеют!".

Как будто следующим шагом будет принудительная установка и использование под страхом отл...

[1.39](#), [Ivan_83](#), 15:47, 09/03/2018 [[ответить](#)] [[смотреть все](#)] –1 [+](#)/[–](#) Очередной пеар от раст боев.

У раста ещё меньше каких то полезных работающих проектов чем у го.

Кроме пеара в общем то у обоих новых языков ничего и нет, на фоне мира си и крестов.

[1.43](#), [анонимус](#), 17:05, 09/03/2018 [[ответить](#)] [[смотреть все](#)] –3 [+](#)/[–](#) Зачем тащить этот образчик, как не надо делать в новую ось?

[1.44](#), [Аноним](#), 17:32, 09/03/2018 [[ответить](#)] [[смотреть все](#)] +2 [+](#)/[–](#) и приобрести 5 новых типов указателей ...

весь текст скрыт

[
[показать](#)
]

[1.50](#)

,
[Ordu](#)

, 19:57, 09/03/2018 [[ответить](#)]

[
[смотреть все](#)
]

–1

[+](#)
/

[–](#)

> Автоматическое управление памятью в Rust избавляет разработчика от манипулирования указателями

Особенно при написании функции strlen, например. Или при реализации printf.

Кстати, любопытно как они собираются разруливать varargs, если rustc вечно настаивает на том, чтобы сохранять за собой возможность точно рассчитать на этапе компиляции размер стека, который требуется для программы.

Автор:
09.03.18 06:46 -

[1.51](#) , [Вареник](#) , 20:43, 09/03/2018 [[ответить](#)] [[смотреть все](#)] +1 [+](#) / [-](#) Про безопасность ранней Жавы говорил что-то схожее "безопасности" нынешнего сырого хруста в руках малолетних фанатиков :)

[2.57](#) , [_](#) , 21:50, 09/03/2018 [[^](#)] [

[ответить](#)

[смотреть все](#)

[показать ветку](#)

[+](#)

[-](#)

Вот! Просто таких старых как мы с тобой уже почти не осталось, другие забыли, а новые - и не :

[1.56](#) , [Нет ты](#) , 21:47, 09/03/2018 [[ответить](#)] [[смотреть все](#)] [+](#) / [-](#) Безопасненько

```
#[no_mangle]
```

```
pub unsafe extern "C" fn strncat(s1: *mut c_char, s2: *const c_char, n: usize) -> *mut c_char {  
    let mut idx = strlen(s1 as *const _) as isize;  
    for i in 0..n as isize {  
        if *s2.offset(i) == 0 {  
            break;  
        }  
    }
```

```
        *s1.offset(idx) = *s2.offset(i);  
        idx += 1;  
    }  
    *s1.offset(idx) = 0;
```

```
    s1  
}
```

[1.58](#) , [_](#) , 21:52, 09/03/2018 [[ответить](#)] [[смотреть все](#)] [+](#) / [-](#) Надо законодательно запретить новости о том что ржавчики ещё чего-то __начали__ переписывать! :) Реально!

Надо разрешить постить только о том что они переписывать __закончили__!

Правда тогда о них новостей __совсем__ не будет :-)

:-)

[1.59](#) , [Аноним](#) , 22:28, 09/03/2018 [[ответить](#)] [[смотреть все](#)] -1 [+](#) / [-](#) Так и запишем го для безделушек типа сервисов, rust для системных безделушек X...

Автор:
09.03.18 06:46 -

весь текст скрыт

[
[показать](#)
]

Read more <http://www.opennet.ru/opennews/art.shtml?num=48226>