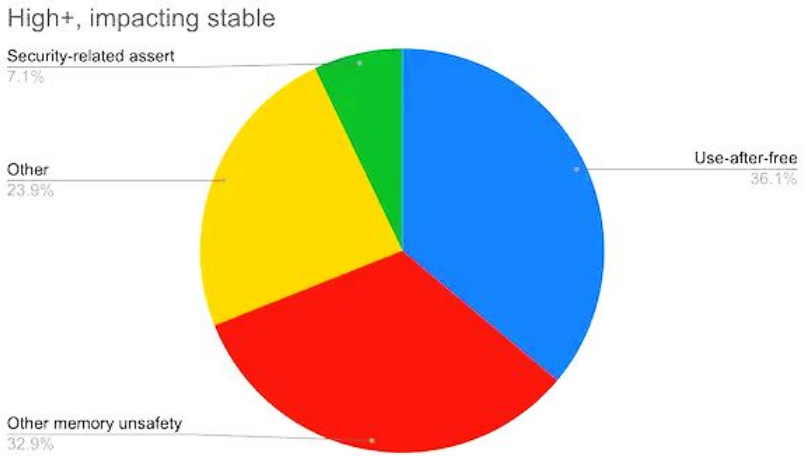
Undefined Behavior

Roope Salmi

Otaniemen lukio 14.11.2022

Muistiturvallisuus

- **70%** vakavista haavoittuvuuksista tuotteissa, kuten Google Chromessa, johtuu C++:n **muistivirheistä**
- Rust on muistiturvallinen, paitsi jos käyttää unsafe-lohkoja
- **Osoittimet**
- Python, Java, muut kielet?



Määritelmä

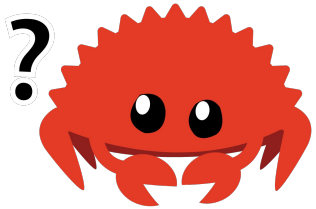
undefined behavior

behavior for which this International Standard imposes no requirements

- C- ja C++-standardien abstrakti kone
- Kun ajon aikana ohjelma tulee “UB” tilaan, ei standardi enää vaadi ohjelman käyttäytymiseltä mitään
  - UB:tä tulee välttää, sillä se tekee ohjelmasta virheellisen
- Rustilla ei standardia eikä spesifikaatiota, mutta termiä käytetään silti

Esimerkkejä

- Muistivirheet
  - taulukon ulkopuolella operointi
  - alustamattoman muistin lukeminen
  - “dangling pointer”
- Laittomat arvot
  - bool, jonka muistiesitys ei ole 0 tai 1
  - enum
- Synkronoimaton monisäikeisyys
  - data race
- C/C++: virheelliset laskutoimitukset
  - etumerkillisen kokonaisluvun ylivuoto



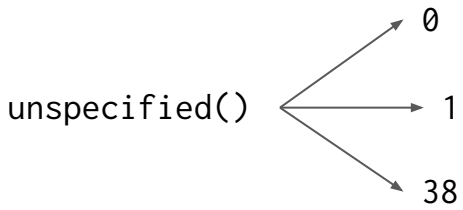

```
let y;
{
    let x = 123;
    y = &x;
}
println!("{y}");
```

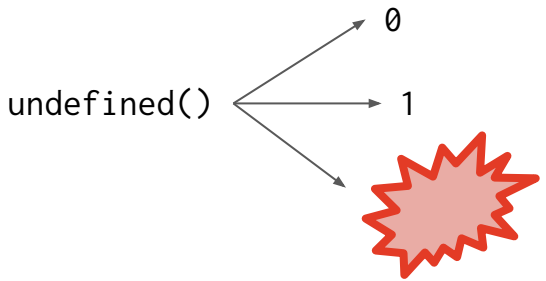
Seuraukset

Vertaa: Unspecified behavior, implementation-defined behavior

Usein UB johtaa suoraan kaatumiseen, esim. muistivirheestä seuraa segmentaatiovirhe (Segmentation fault).

Kääntäjä saa kuitenkin tehdä mitä lystää, ja sitä paitsi, käytös voi muuttua saman kääntäjän tulevissa versioissa.





Näyttää pahalta

Permissible undefined behavior ranges from ignoring the situation completely with unpredictable results, to having demons fly out of your nose.

— John F. Woods Usenet-uutisryhmässä comp.std.c, 1992

Ovatko standardin tai kääntäjien laatijat laiskoja, tai jopa ilkeitä?

Olemassa myös tarkistusjärjestelmiä, kuten UndefinedBehaviorSanitizer ja MIRI (Rust)

Optimointi

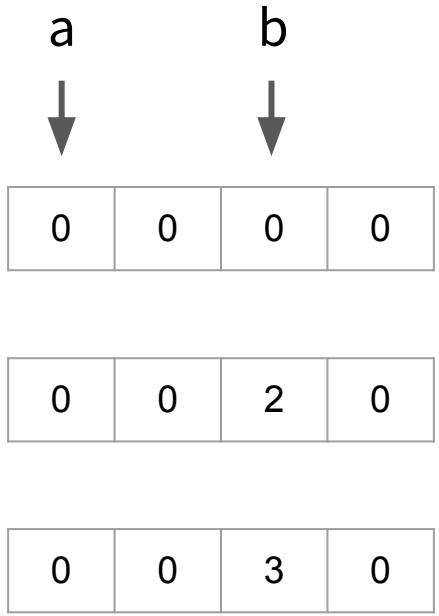
Rust compilers, including rustc, will perform optimizations. The reference does not specify what optimizations are allowed or disallowed. Instead, think of the compiled program as a black box. You can only probe by running it, feeding it input and observing its output. Everything that happens that way must conform to what the reference says.

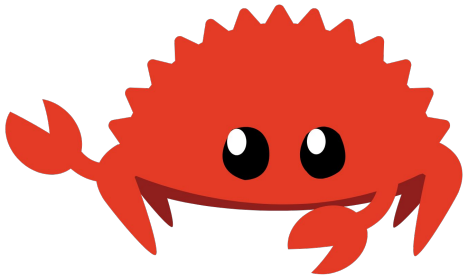
Kääntäjä voi olettaa, että mitään määrittelemätöntä käytöstä ei voi tapahtua, ja optimoida koodia sen perusteella.

Demo

Taulukon rajojen ulkopuolelle kirjoitus
https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=66b15645d1936c061f770241e1c5f2f4

Monisäikeinen luvun kasvatus: race condition vs. data race
https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=414d35e3c7ce1874b55976dbcdb1fe7f


a
b
0 0 0 0
0 0 2 0
0 0 3 0
?



Useimmiten…

Säie 1	Säie 2
LUE LASKURI X = 0	
KASVATA X X = 1	
KIRJOITA LASKURI LASKURI = 1	
	LUE LASKURI X = 1
	KASVATA X X = 2
	KIRJOITA LASKURI LASKURI = 2

Mutta välillä…

Säie 1	Säie 2
LUE LASKURI X = 0	
KASVATA X X = 1	
	LUE LASKURI X = 0
KIRJOITA LASKURI LASKURI = 1	
	KASVATA X X = 1
	KIRJOITA LASKURI LASKURI = 1

Data race

- Vaatii vähintään kaksi säiettä
- Kaksi säiettä käsittelee yhteistä muistia ja vähintään toinen kirjoittaa
- Muistin käsittely ei ole **synkronoitua**
- **On aina UB**, joten kääntäjä voi olettaa, että data raceja ei tapahdu

Race condition

- Vaatii vähintään kaksi säiettä
- Yleisempi virhe kuin **data race**
- On olemassa, kun oletetaan tapahtumien järjestys, mutta sitä ei ole taattu
- Ei yleensä ole UB, vaan looginen virhe

Rust: Oletuksena turvallinen

- Koodissa, joka ei käytä unsafe-lohkoja, ei voi esiintyä UB
- Kaikki laskutoimitukset on määritelty
- unchecked-versiot useista funktioista
- Entä muut kielet?
  - C, C++
  - Roskienkeruu
  - Global interpreter lock

```
assert!(oletus, "oletus ei pätenyt");


if !oletus {
    unsafe { unreachable_unchecked(); }
}
```

Lisämateriaalia

Behavior considered undefined — The Rust Reference
https://doc.rust-lang.org/reference/behavior-considered-undefined.html

Undefined Behavior deserves a better reputation — Ralf Jung
https://www.ralfj.de/blog/2021/11/18/ub-good-idea.html

CppCon 2016: Garbage In, Garbage Out: Arguing about Undefined Behavior — Chandler Carruth
https://www.youtube.com/watch?v=yG1OZ69H_-o

Rust Belt Rust Conference 2018: Building on an unsafe foundation — Jason Orendorff
https://www.youtube.com/watch?v=rTo2u13lVcQ