



Presented by Chester Wyke

At Vancouver Rust Meetup 2025-03-19

[Bacon Website](#)

[Bacon GitHub](#)

Designed Purpose

- Background Code Checker

How I use bacon

- Background Code Checker
- Effortless application Restarts
- Frequently used commands storage
- Long command completion notifications (using sound)
- Simplified Compiler Driven Refactoring

Why use bacon

- Get full screen error messages
- Get errors before warnings
- Get error feedback in small terminal window
- Improve signal to noise ratio (only update after subsequent compiles)

cargo clippy

```
onelap% cargo clippy
    Checking a v0.1.0 (/home/one/tmp/a)
warning: variable does not need to be mutable
--> src/main.rs:10:9
10 |         let mut stop = 0;
    |         -----^^^^^
    |         help: remove this `mut`
= note: `[warn(unused_mut)]` on by default

error: this arithmetic operation will overflow
--> src/main.rs:16:17
16 |         let x = 1 - 2;
    |                 ^^^^^ attempt to compute `1_u64 - 2_u64`, which would overflow
= note: `[deny(arithmetic_overflow)]` on by default

warning: useless conversion to the same type: `u64`
--> src/main.rs:17:59
17 |         std::thread::sleep(std::time::Duration::from_secs(1u64.into()));
    |                                                                 ^^^^^^^^^^^^^ help: consider removing `.into(): 'u64'`
= help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#useless\_conversion
= note: `[warn(clippy::useless_conversion)]` on by default

warning: manual implementation of an assign operation
--> src/main.rs:19:9
19 |         i = i - 1;
    |         ^^^^^^^^^ help: replace it with: `i -= 1`
= help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#assign\_op\_pattern
= note: `[warn(clippy::assign_op_pattern)]` on by default

warning: `a` (bin "a") generated 3 warnings
error: could not compile `a` (bin "a") due to 1 previous error; 3 warnings emitted
onelap% 
```

cargo clippy

```
onelap% cargo clippy
    Checking a v0.1.0 (/home/one/tmp/a)
warning: variable does not need to be mutable
  --> src/main.rs:10:9
10 |         let mut stop = 0;
   |         ^^^^^
   |         help: remove this `mut`
   = note: `[warn(unused_mut)]` on by default

error: this arithmetic operation will overflow
  --> src/main.rs:16:17
16 |         let x = 1 - 2;
   |         ^^^^^ attempt to compute `1_u64 - 2_u64`, which would overflow
   = note: `[deny(arithmetic_overflow)]` on by default

warning: useless conversion to the same type: `u64`
  --> src/main.rs:17:59
17 |         std::thread::sleep(std::time::Duration::from_secs(1u64.into()));
   |                                                                 ^^^^^^^^^^^^^ help: consider removing `.into(): 'u64'`
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#useless_conversion
   = note: `[warn(clippy::useless_conversion)]` on by default

warning: manual implementation of an assign operation
  --> src/main.rs:19:9
19 |         i = i - 1;
   |         ^^^^^^^^^ help: replace it with: `i -= 1`
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#assign_op_pattern
   = note: `[warn(clippy::assign_op_pattern)]` on by default

warning: `a` (bin "a") generated 3 warnings
error: could not compile `a` (bin "a") due to 1 previous error; 3 warnings emitted
onelap%
```

bacon

```
a clippy 1 error 3 warnings

1 error: this arithmetic operation will overflow
--> src/main.rs:16:17
16 |         let x = 1 - 2;
    |         ^^^^^ attempt to compute `1_u64 - 2_u64`, which would overflow
= note: `[deny(arithmetic_overflow)]` on by default

2 warning: variable does not need to be mutable
--> src/main.rs:10:9
10 |         let mut stop = 0;
    |         ^^^^^
    |         help: remove this `mut`
= note: `[warn(unused_mut)]` on by default

3 warning: useless conversion to the same type: `u64`
--> src/main.rs:17:59
17 |         std::thread::sleep(std::time::Duration::from_secs(1u64.into()));
    |                                                                    ^^^^^^^^^^^^^ help: consider removing
    |                                                                    `.into()`: `1u64`
= help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#useless_conversion
= note: `[warn(clippy::useless_conversion)]` on by default

4 warning: manual implementation of an assign operation
--> src/main.rs:19:9
19 |         i = i - 1;
    |         ^^^^^^^^^ help: replace it with: `i -= 1`
= help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#assign_op_pattern
= note: `[warn(clippy::assign_op_pattern)]` on by default
```

bacon

```
a clippy 1 error 3 warnings

1 error: this arithmetic operation will overflow
--> src/main.rs:16:17
16 |         let x = 1 - 2;
    |         ^^^^^ attempt to compute `1_u64 - 2_u64`, which would overflow
= note: `[deny(arithmetic_overflow)]` on by default

2 warning: variable does not need to be mutable
--> src/main.rs:10:9
10 |         let mut stop = 0;
    |         ^^^^^
    |         help: remove this `mut`
= note: `[warn(unused_mut)]` on by default

3 warning: useless conversion to the same type: `u64`
--> src/main.rs:17:59
17 |         std::thread::sleep(std::time::Duration::from_secs(1u64.into()));
    |                                                                    ^^^^^^^^^^^^^ help: consider removing
    |                                                                    `.into()`: `1u64`
= help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#useless_conversion
= note: `[warn(clippy::useless_conversion)]` on by default

4 warning: manual implementation of an assign operation
--> src/main.rs:19:9
19 |         i = i - 1;
    |         ^^^^^^^^^ help: replace it with: `i -= 1`
= help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#assign_op_pattern
= note: `[warn(clippy::assign_op_pattern)]` on by default
```


bacon

```
a  clippy  1 error  3 warnings

1 error: this arithmetic operation will overflow
--> src/main.rs:16:17
16 |         let x = 1 - 2;
    |         ^^^^^ attempt to compute `1_u64 - 2_u64`, which would overflow
= note: `[deny(arithmetic_overflow)]` on by default

2 warning: variable does not need to be mutable
Hit / to search, s to toggle summary mode, h for help, w to not wrap lines, p to pause, q to quit
```