

Manual implementation of a `Fn*` trait.

Erroneous code example:

```
struct MyClosure {
    foo: i32
}

impl FnOnce<()> for MyClosure { // error
    type Output = ();
    extern "rust-call" fn call_once(self, args: ()) -> Self::Output {
        println!("{}", self.foo);
    }
}
```

Manually implementing `Fn`, `FnMut` or `FnOnce` is unstable and requires `#![feature(fn_traits, unboxed_closures)]`.

`#![feature(fn_traits, unboxed_closures)]`

```
struct MyClosure {
    foo: i32
}

impl FnOnce<()> for MyClosure { // ok!
    type Output = ();
    extern "rust-call" fn call_once(self, args: ()) -> Self::Output {
        println!("{}", self.foo);
    }
}
```

The arguments must be a tuple representing the argument list. For more info, see the tracking issue: