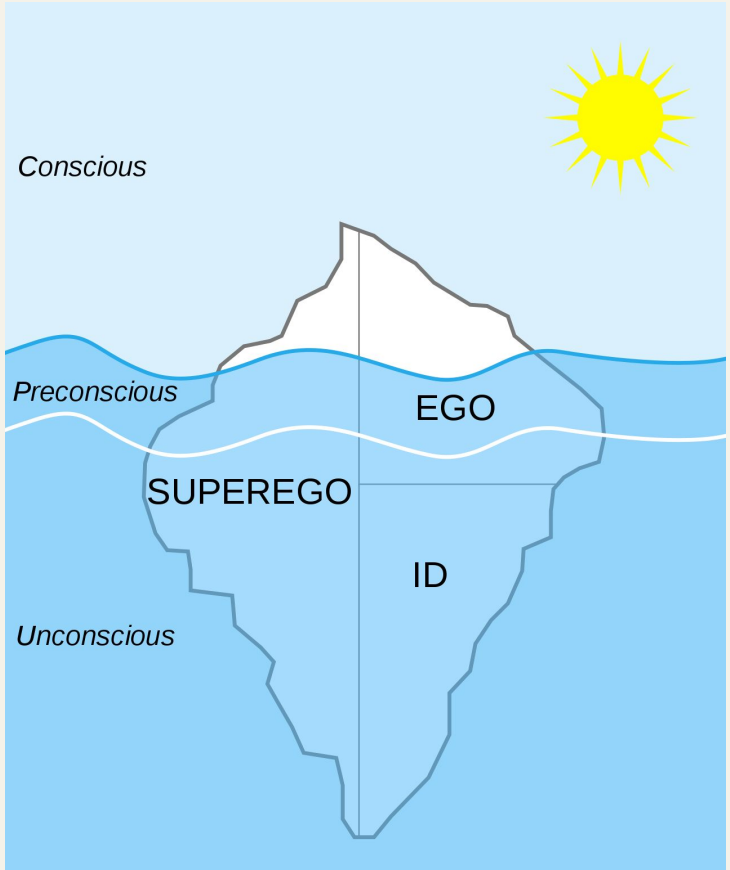
死者甦醒，佛洛伊德

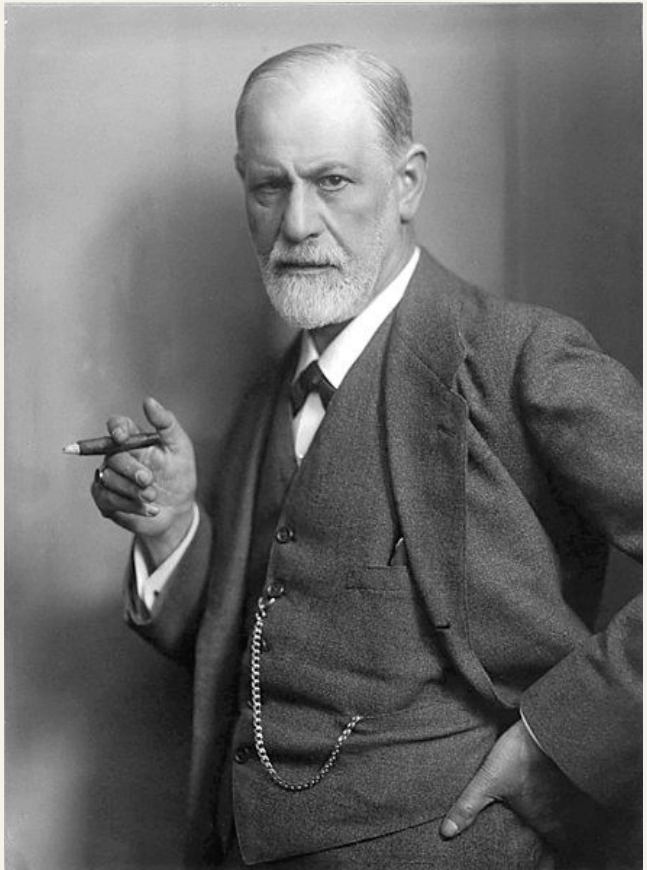
Id, ego and super-ego

scc@teamt5.org

TEAMT5
杜 浦 數 位 安 全
Persistent Cyber Threat Hunters

As everyone knows





Why named ego?

TEAMT5
杜 浦 數 位 安 全

EG- word

- Egg
- Ego

exception_guarding_observatory

TL;DR

TEAMT5
杜浦數位安全

```cpp
#include "../single_include/ego.hpp"
auto foo() -> ego::expected<int, ego::ego_error> {
    EGO(int);
    throw std::runtime_error("foo");

    return 0;
}
```

TL;DR

TEAMT5
杜浦數位安全

```cpp
#include "../single_include/super_ego.hpp"
auto foo() -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    int *p = nullptr;
    *p = 42;
    return 0;
}
```

TL;DR

TEAMT5
杜浦數位安全

```
auto foo(int index=10)-> ego::expected<int, ego::ego_error> {
    EGO(int);
    if (index == 0)
        throw std::runtime_error("foo");

    return foo(--index);
}
```

TL;DR

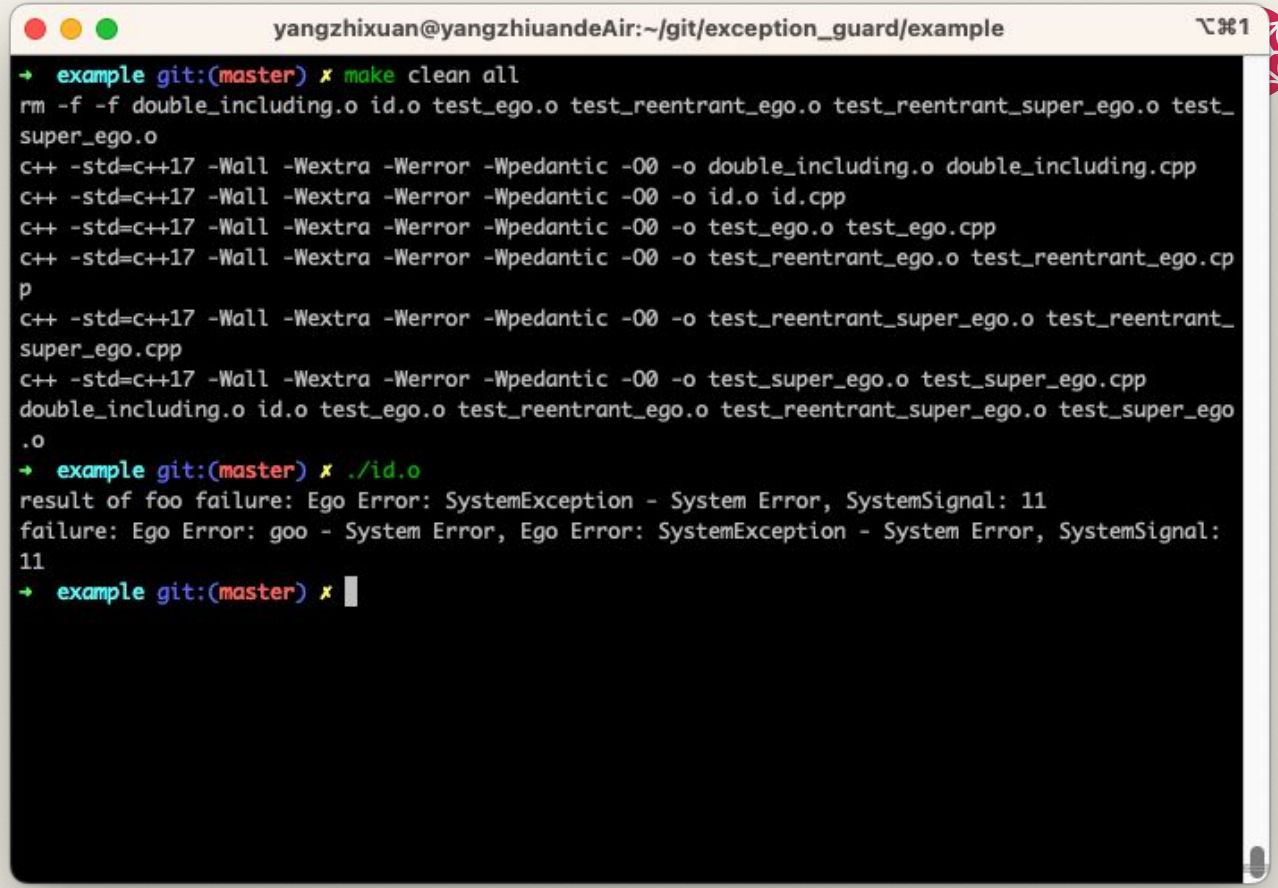
TEAMT5
杜浦數位安全

```
auto foo(int index=10)-> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    if (index == 0) {
        int *p = nullptr;
        *p = 42;
    }
    return foo(--index);
}
```

TL;DR - it can be …

TEAMT5
杜 浦 數 位 安 全

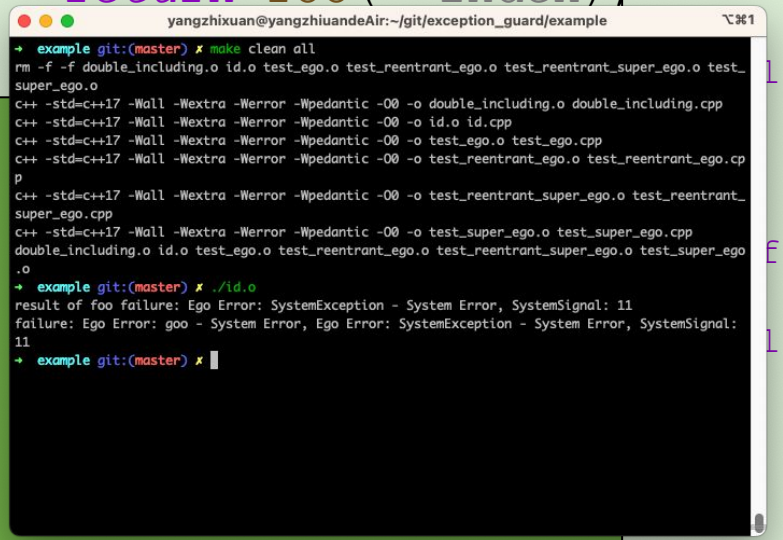
1. Single included
2. Thread-safe
3. Reentrant
4. Following in C++11 standard



```
auto foo(int index = 10)
    -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    if (index == 0) {
        int *p = nullptr;
        *p = 42;
    }
    return foo(--index);
}
```

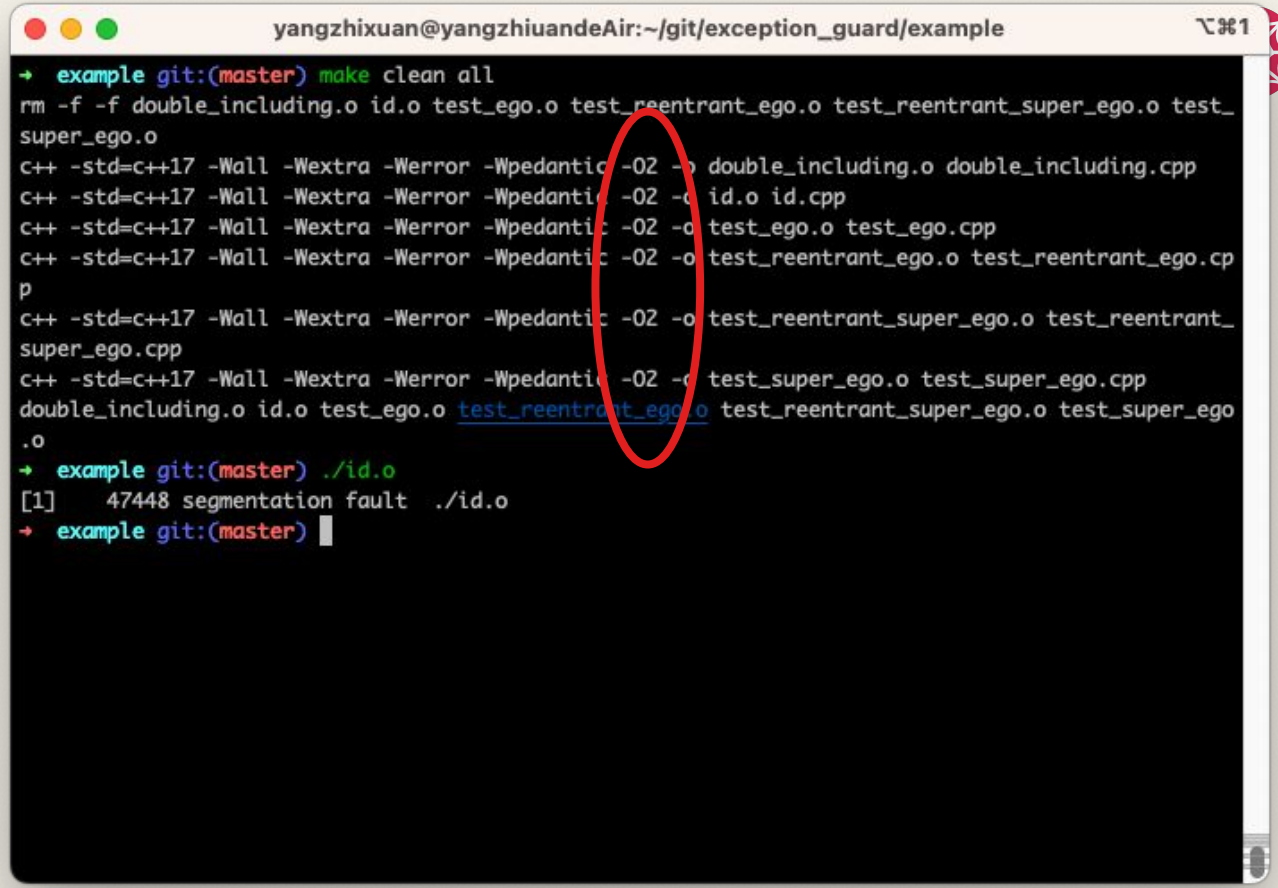
TEAMT5
杜浦數位安全

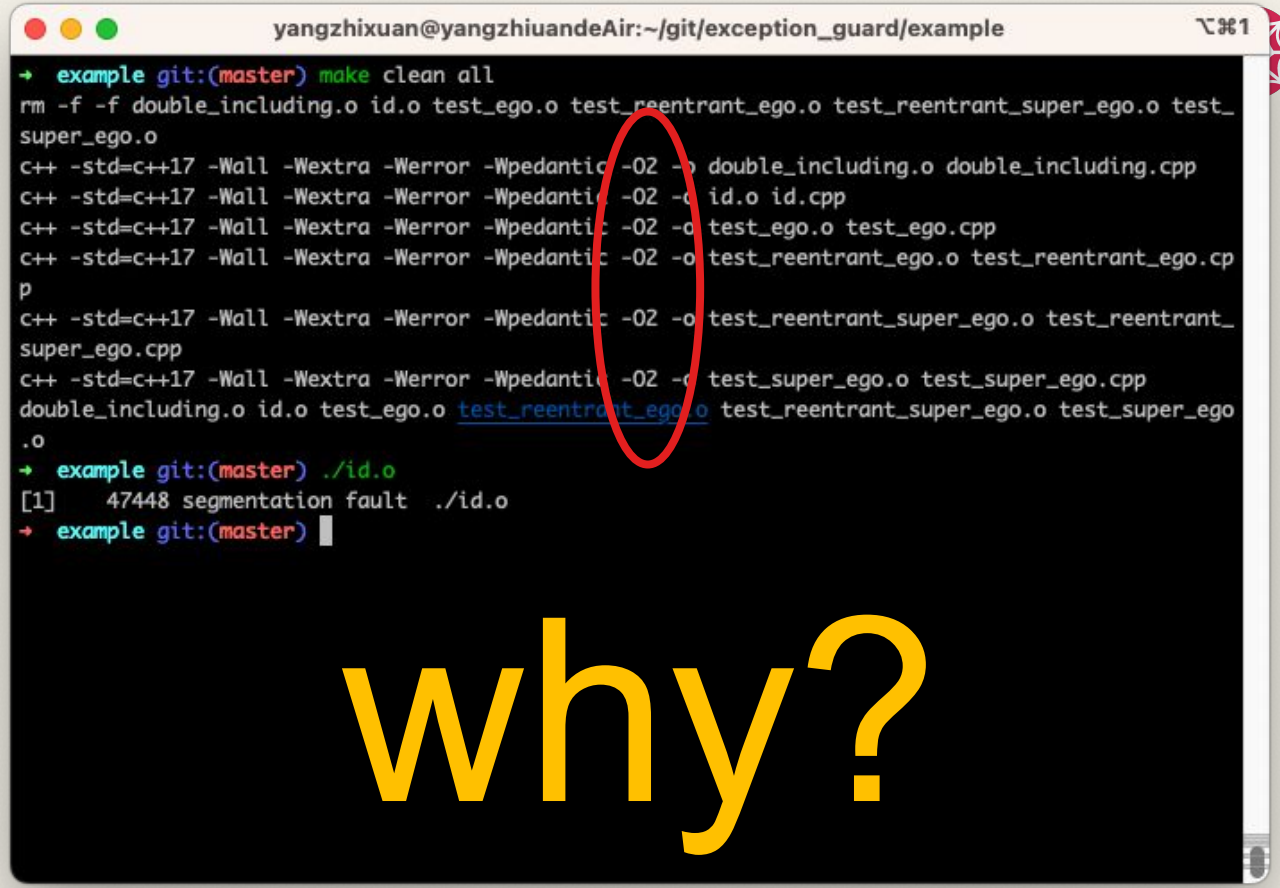
```cpp
auto foo(int index = 10)
    -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    if (index == 0) {
        int *p = nullptr;
        *p = 42;
    }
    return foo(--index);
}
```




```cpp
auto goo()
    -> ego::expected<std::string, ego::ego_error> {
    EGO(std::string);
    auto res_foo = foo();

    if (res_foo)
        std::cout << "result of foo success: "
                  << res_foo.value() << std::endl;
lse
    std::cout << "result of foo failure: "
              << res_foo.error().message()
              << std::endl;

f (res_foo)
    return std::string("goo");
lse {
    auto res_foo_error = res_foo.error();
    return ego::make_ego_error(res_foo_error,
    res_foo_error.message(), "goo");

}
```

But, clang





TEAMT5
杜 浦 數 位 安 全

```
auto foo(int index = 10)
    -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    if (index == 0) {
        int *p = nullptr;
        *p = 42;
    }
    return foo(--index);
}
```

TEAMT5
杜浦數位安全

Undefined behavior

```
auto foo(int index = 10)
    -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    if (index == 0) {
      int *p = nullptr;
      *p = 42;
    }
    return foo(--index);
}
```

Unreachable

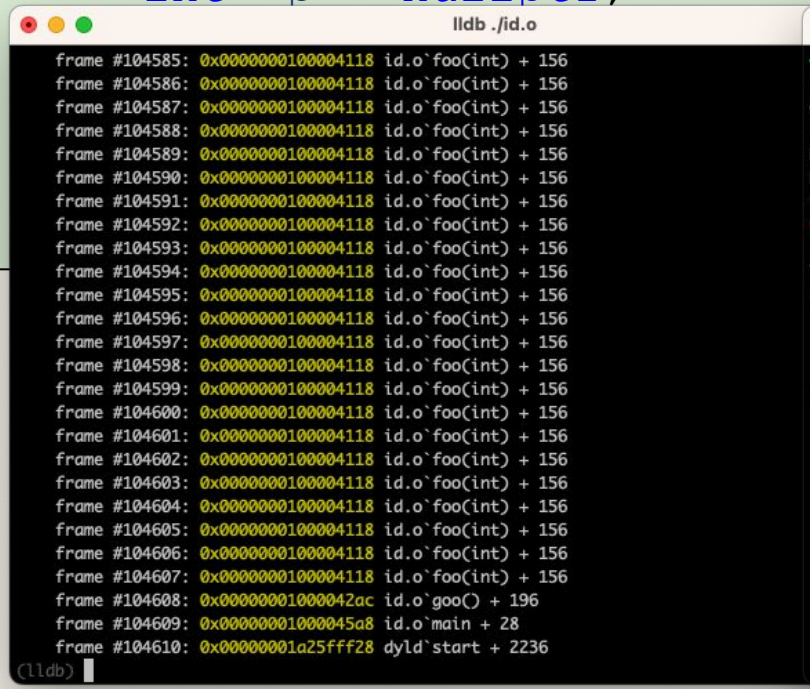
TEAMT5
杜浦數位安全

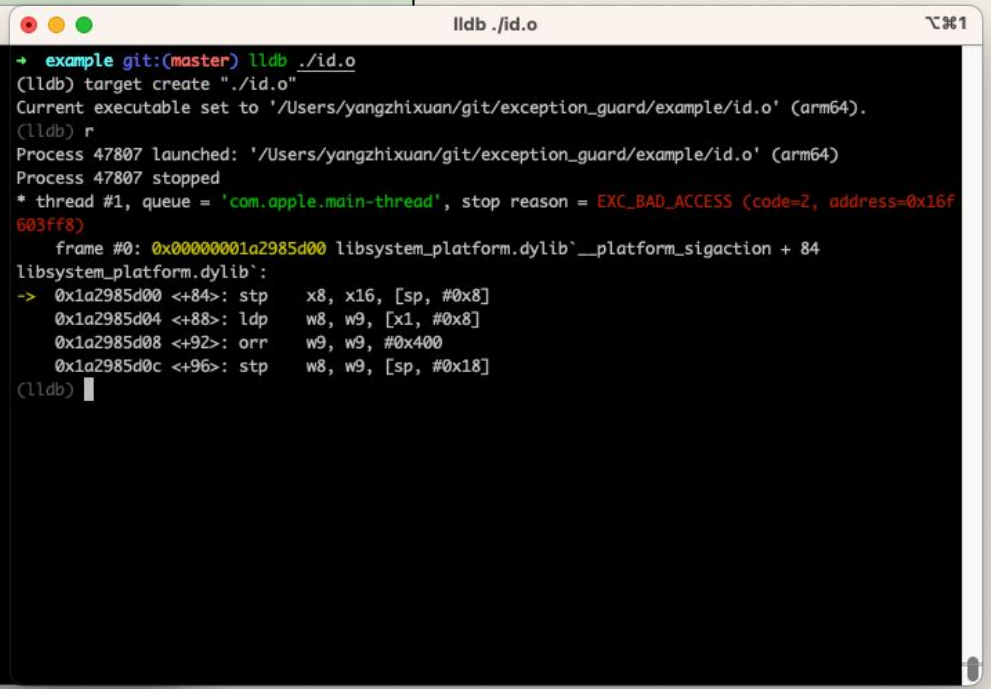
```
auto foo(int index = 10)
    -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    if (index == 0) {
      int *p = nullptr;
      *p = 42;
    }
    return foo(--index);
}
```

Unreachable

TEAMT5
杜浦數位安全

```
auto foo(int index = 10)
    -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    if (index == 0) {
        int *p = nullptr;
}
```





TL;DR - it can be …

TEAMT5
杜 浦 數 位 安 全

1. Single included
2. Thread-safe
3. Reentrant
4. Following in C++11 standard
5. But, at it's wits' end with UB.

Growth zone

TMP

TEAMT5
杜 浦 數 位 安 全

ego::detail::expected_impl__

As we introduced on the previous meeting.

```
template <typename T, typename E>
constexpr bool can_be_expected =
    std::is_default_constructible<T>::value &&
    std::is_base_of<std::error_code, E>::value;
```

How to catch uncaught exception

If the exception handling mechanism handling an uncaught exception directly invokes a function that exits via an exception, the function `std::terminate` is invoked.

- §14.2.7 Throwing an exception [except.throw.7]

How to catch uncaught exception

C++ | Diagnostics library

std::get_terminate

Defined in header <exception>

```
std::terminate_handler get_terminate() noexcept;    (since C++11)
```

Returns the currently installed `std::terminate_handler`, which may be a null pointer.

This function is thread-safe. Prior call to `std::set_terminate` *synchronizes-with* (see `std::memory_order`) this function. (since C++11)

Parameters

(none)

Return value

The currently installed `std::terminate_handler`.

See also

terminate_handler	the type of the function called by `std::terminate` (typedef)
set_terminate	changes the function to be called by `std::terminate` (function)

```
struct exception_guard {
private:
    std::terminate_handler th;
    static constexpr auto custom_terminate_handler = []() {
        std::cout << "hello custom_terminate_handler"
                  << std::endl;
        std::longjmp(jump_buffer, 1);
    };

public:
    exception_guard() : th(std::get_terminate()) {
        std::cout << __PRETTY_FUNCTION__ << std::endl;
        std::set_terminate(custom_terminate_handler);
    }
    ~exception_guard() {
        std::cout << __PRETTY_FUNCTION__ << std::endl;
        std::set_terminate(th);
    }
};
```

```
struct exception_guard {
private:
    std::terminate_handler th;
    static constexpr auto custom_terminate_handler = [] () {
        std::cout << "hello custom_terminate_handler"
                  << std::endl;
        std::longjmp(jump_buffer, 1);
    };

public:
    exception_guard() : th(std::get_terminate()) {
        std::cout << __PRETTY_FUNCTION__ << std::endl;
        std::set_terminate(custom_terminate_handler);
    }
    ~exception_guard() {
        std::cout << __PRETTY_FUNCTION__ << std::endl;
        std::set_terminate(th);
    }
};
```

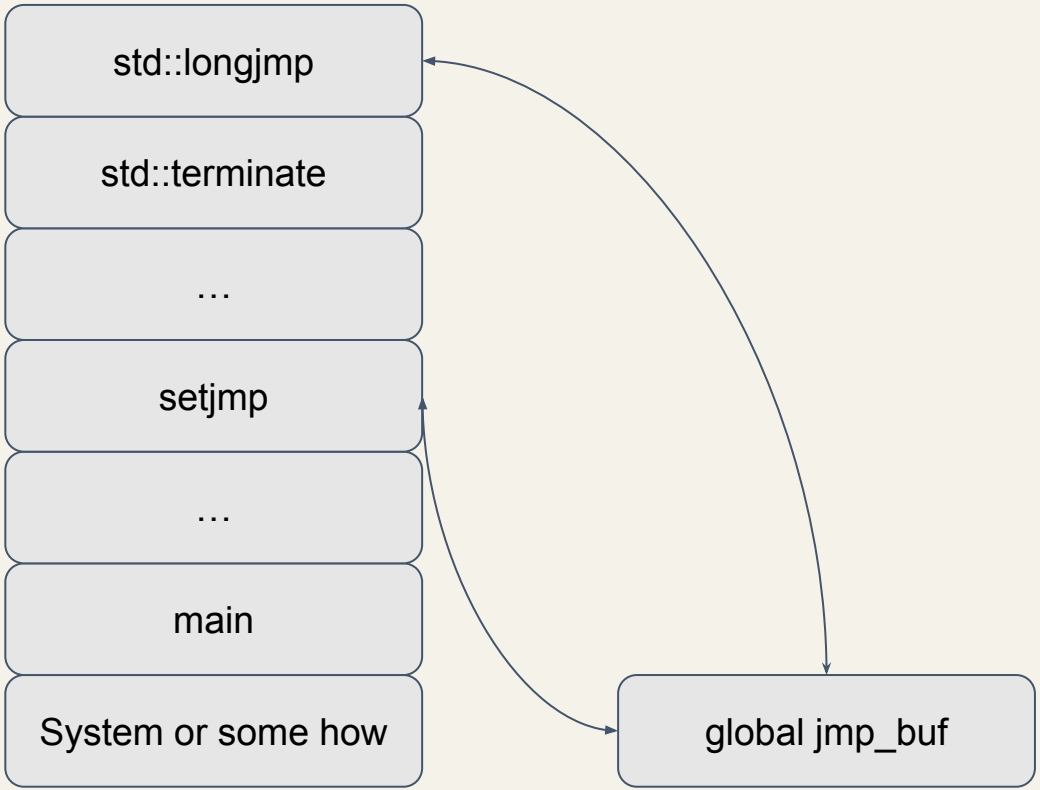
```
struct exception_guard {
private:
    std::terminate_handler th;
    static constexpr auto custom_terminate_handler = []() {
        std::cout << "hello custom_terminate_handler"
                  << std::endl;
        std::longjmp(jump_buffer, 1);
    };

public:
    exception_guard() : th(std::get_terminate()) {
        std::cout << __PRETTY_FUNCTION__ << std::endl;
        std::set_terminate(custom_terminate_handler);
    }
    ~exception_guard() {
        std::cout << __PRETTY_FUNCTION__ << std::endl;
        std::set_terminate(th);
    }
};
```

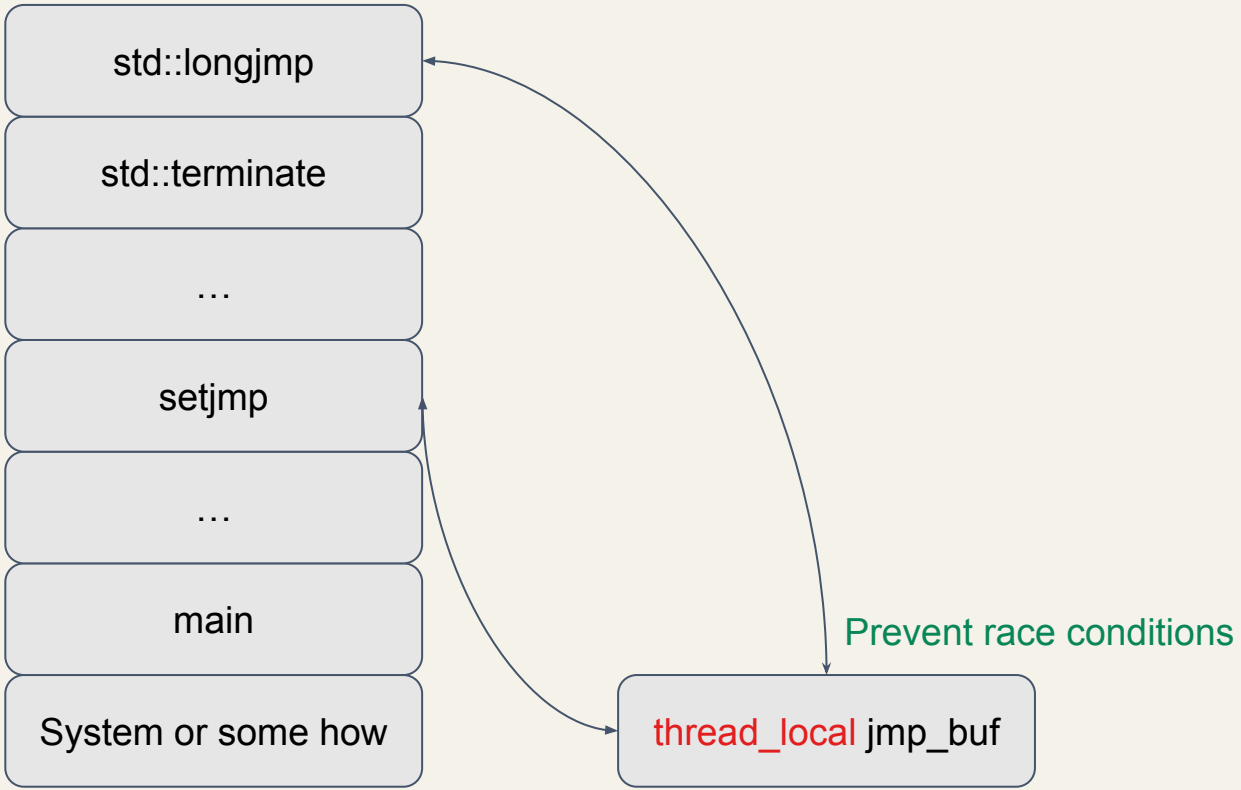
https://godbolt.org/z/f7GT3asxT

TEAMT5 杜浦數位安全

```
struct exception_guard {
private:
    std::terminate_handler th;
    static constexpr auto custom_terminate_handler = []() {
        std::cout << "hello custom_terminate_handler"
                  << std::endl;
        std::longjmp(jump_buffer, 1);
    };

public:
    exception_guard() : th(std::get_
        std::cout << __PRETTY_FUNCTIO
        std::set_terminate(custom_ter
    }
    ~exception_guard() {
        std::cout << __PRETTY_FUNCTIO
        std::set_terminate(th);
    }
};
```

```
#define EXCEPTION_GURAD()   \
exception_guard guard;      \
do {                        \
    if (setjmp(jump_buffer))\
    return guard;           \
} while(0)
```


std::longjmp
std::terminate
…
setjmp
…
main
System or some how
global jmp_buf


std::longjmp
std::terminate
…
setjmp
…
main
System or some how
thread_local jmp_buf
Prevent race conditions

How to know typeid?

TEAMT5 杜浦數位安全

C++ source #1

A Save/Load + Add new... Vim CppInsights Quick-bench

```
1  #include <iostream>
2  #include <typeinfo>
3
4  struct Base {};
5  struct Derived : Base {};
6
7  int main() {
8      Base* p = new Derived;
9      std::cout << typeid(p).name() << std::endl;
10     return 0;
11 }
```

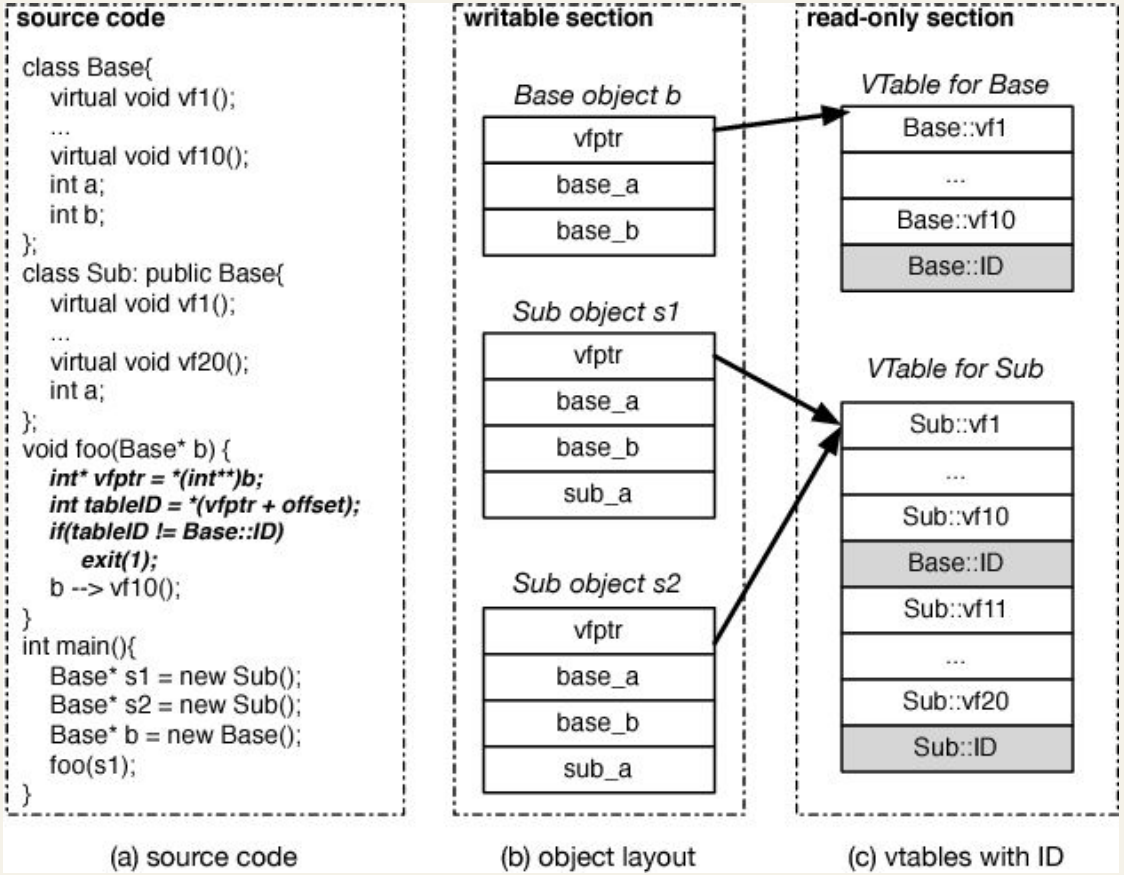
Output of x86-64 gcc (trunk) (Compiler #1)

A Wrap lines Select all

```
ASM generation compiler returned: 0
Execution build compiler returned: 0
Program returned: 0
    P4Base
```

Details of RTTI wouldn't be introduced here.

Parse the vtable



Zhang, Chao, Chengyu Song, Kevin Zhijie Chen, Zhaofeng Chen and Dawn Xiaodong Song. “VTint: Protecting Virtual Function Tables' Integrity.” *Network and Distributed System Security Symposium* (2015).

```
template <typename T>
static auto get_exception_dtor(T target_exception) noexcept
-> exception_dtor {
    const auto &target_vtable = *reinterpret_cast<void **>
                                (&target_exception);
    auto target_dtor = *reinterpret_cast<exception_dtor *>
                                (target_vtable);
    return target_dtor;
}
```

TEAMT5
杜 浦 數 位 安 全

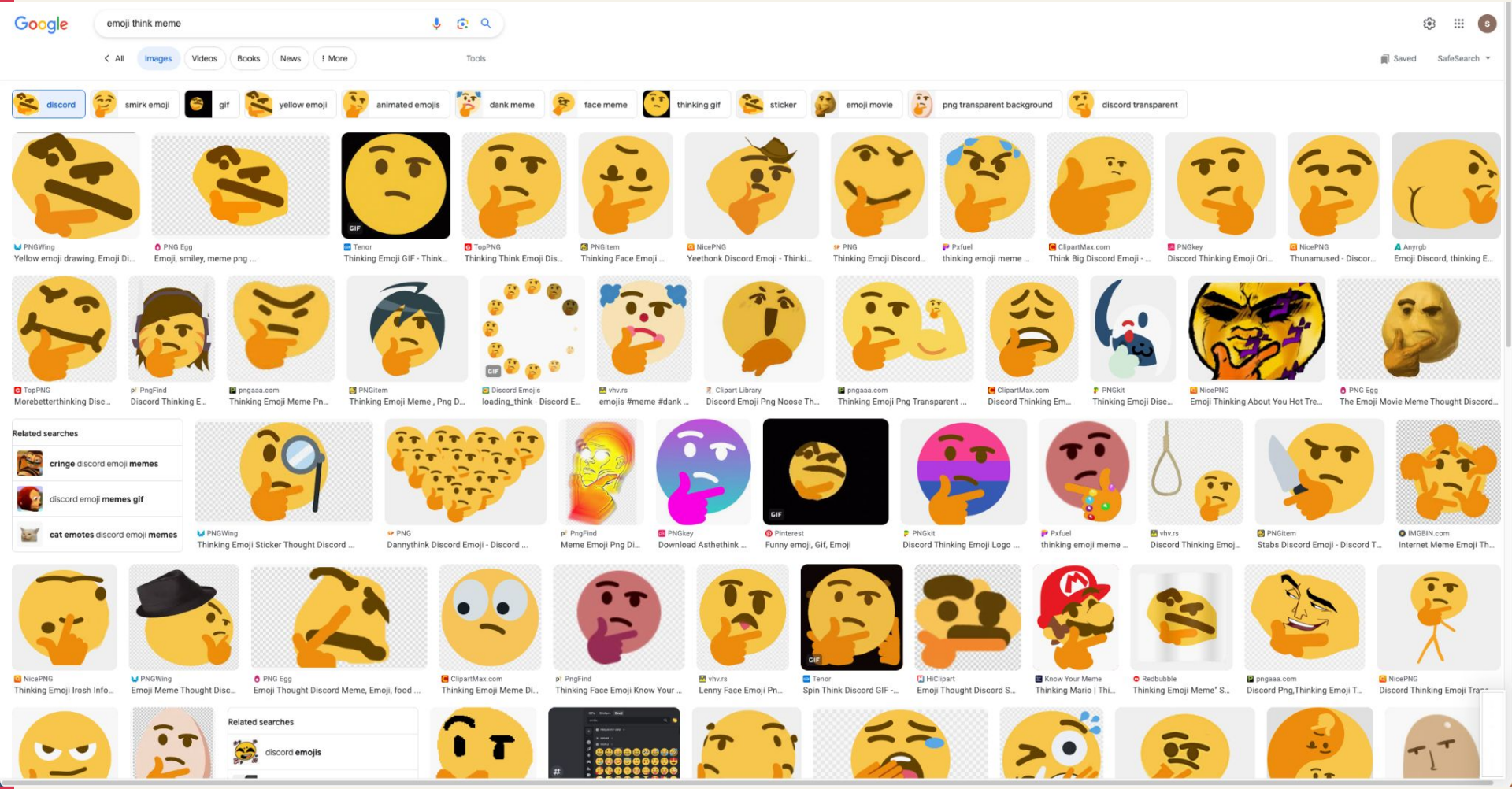
```
static void ego_custom_terminate_handler() noexcept {
    try {
        std::rethrow_exception(std::current_exception());
    } catch (const std::exception &e) {
        auto dtor = ego::detail::get_exception_dtor(e);
        ego_cur_exception_info__ =
                        cur_exception_info(dtor, e.what());
    }
    std::longjmp(exception_guard_jump_buffer, 1);
}
```

TEAMT5
杜 浦 數 位 安 全

```
static void ego_custom_terminate_handler() noexcept {
    try {
        std::rethrow_exception(std::current_exception());
    } catch (const std::exception &e) {
        auto dtor = ego::detail::get_exception_dtor(e);
        ego_cur_exception_info__ =
                        cur_exception_info(dtor, e.what());
    }
    std::longjmp(exception_guard_jump_buffer, 1);
}
```

TEAMT5
杜 浦 數 位 安 全

```
static void ego_custom_terminate_handler() noexcept {
    try {
        std::rethrow_exception(std::current_exception());
    } catch (const std::exception &e) {
        auto dtor = ego::detail::get_exception_dtor(e);
        ego_cur_exception_info__ =
                        cur_exception_info(dtor, e.what());
    }
    std::longjmp(exception_guard_jump_buffer, 1);
}
```


Google
emoji think meme
All
Images
Videos
Books
News
More
Tools
Saved
SafeSearch
discord
smirk emoji
gif
yellow emoji
animated emojis
dank meme
face meme
thinking gif
sticker
emoji movie
png transparent background
discord transparent
PNGWing
Yellow emoji drawing, Emoji Di...
PNG Egg
Emoji, smiley, meme png ...
Tenor
Thinking Emoji GIF - Think...
TopPNG
Thinking Think Emoji Dis...
PNGitem
Thinking Face Emoji ...
NicePNG
Yeethonk Discord Emoji - Thinki...
PNG
Thinking Emoji Discord...
Pxfuel
thinking emoji meme ...
ClipartMax.com
Think Big Discord Emoji - ...
PNGkey
Discord Thinking Emoji Ori...
NicePNG
Thunamused - Discor...
Anyrgb
Emoji Discord, thinking E...
TopPNG
Morebetterthinking Disc...
PngFind
Discord Thinking E...
pngaaa.com
Thinking Emoji Meme Pn...
PNGitem
Thinking Emoji Meme , Png D...
Discord Emojis
loading_think - Discord E...
vhv.rs
emojis #meme #dank ...
Clipart Library
Discord Emoji Png Noose Th...
pngaaa.com
Thinking Emoji Png Transparent ...
ClipartMax.com
Discord Thinking Em...
PNGkit
Thinking Emoji Disc...
NicePNG
Emoji Thinking About You Hot Tre...
PNG Egg
The Emoji Movie Meme Thought Discord...
Related searches
cringe discord emoji memes
discord emoji memes gif
cat emotes discord emoji memes
PNGWing
Thinking Emoji Sticker Thought Discord ...
PNG
Dannythink Discord Emoji - Discord ...
PngFind
Meme Emoji Png Di...
PNGkey
Download Asthethink ...
Pinterest
Funny emoji, Gif, Emoji
PNGkit
Discord Thinking Emoji Logo ...
Pxfuel
thinking emoji meme ...
vhv.rs
Discord Thinking Emoj...
PNGitem
Stabs Discord Emoji - Discord T...
IMGBIN.com
Internet Meme Emoji Th...
NicePNG
Thinking Emoji Irosh Info...
PNGWing
Emoji Meme Thought Disc...
PNG Egg
Emoji Thought Discord Meme, Emoji, food ...
ClipartMax.com
Thinking Emoji Meme Di...
PngFind
Thinking Face Emoji Know Your ...
vhv.rs
Lenny Face Emoji Pn...
Tenor
Spin Think Discord GIF -...
HiClipart
Emoji Thought Discord S...
Know Your Meme
Thinking Mario | Thi...
Redbubble
Thinking Emoji Meme" S...
pngaaa.com
Discord Png,Thinking Emoji T...
NicePNG
Discord Thinking Emoji Tra...
Related searches
discord emojis

The signal handler

TEAMT5 杜浦數位安全

```
#if defined(__linux__) || defined(__APPLE__)
static void posix_exception_handler(bool enable = true)
{
    auto callback = [](int signo,siginfo_t *info, void *context) {
        ego::detail::super_ego_cur_exception_info__ =
        ego::detail::cur_exception_info(signo);
        std::longjmp(jump_buffer, 1);
    };

    static int supporting_signals[] = {
        SIGSEGV,
        SIGFPE,
        SIGILL,
        SIGBUS,
        SIGABRT,
    };

    ...
}
```

The VEH

TEAMT5 杜浦數位安全

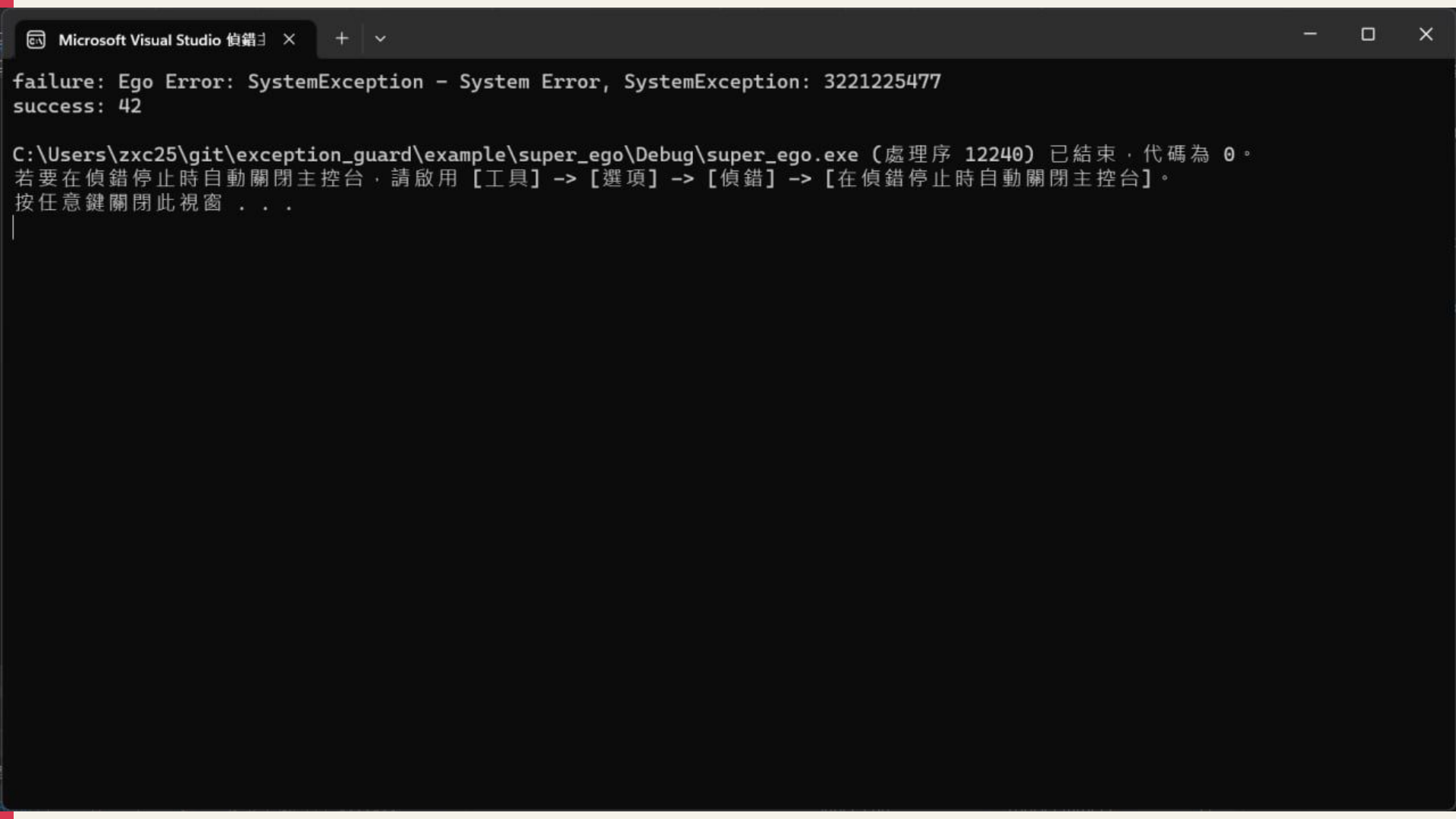
```
#if defined(_WIN32) || defined(_WIN64)
static void windows_exception_handler(bool enable = true) {
    auto callback = [](EXCEPTION_POINTERS *ep) -> LONG {
    // supporting ExceptionCodes under windows
    static DWORD supporting_exceptions[] = {
        ...
    };

    auto exception_code = ep->ExceptionRecord->ExceptionCode;
    for (auto code : supporting_exceptions)
        if (code == exception_code) {
            ego::detail::super_ego_cur_exception_info__ =
            ego::detail::cur_exception_info(exception_code);
            std::longjmp(jump_buffer, 1);
        }

    return EXCEPTION_CONTINUE_SEARCH;
    };
...
}
```

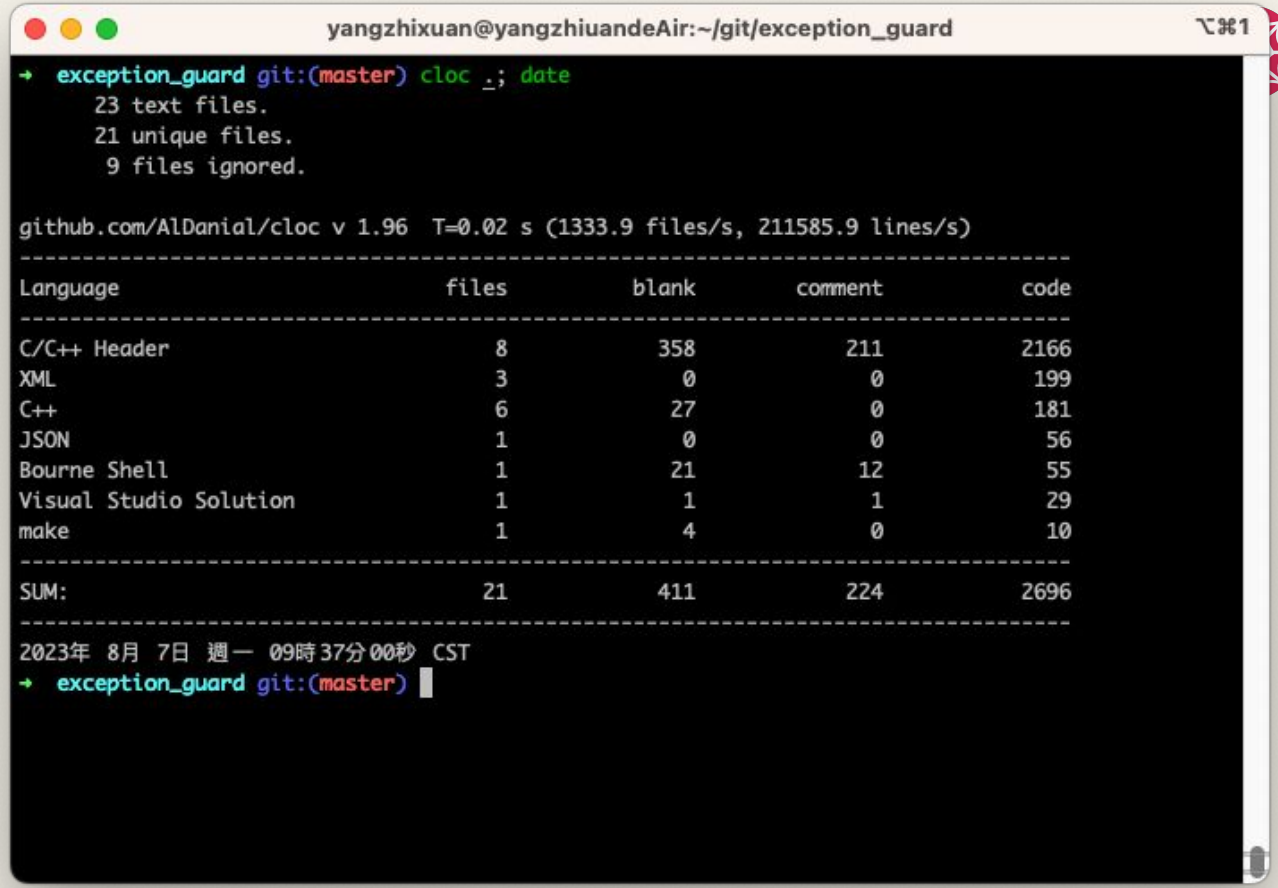
The VEH

TEAMT5 杜浦數位安全

```
#if defined(_WIN32) || defined(_WIN64)
static void windows_exception_handler(b
    auto callback = [](EXCEPTION_POINTE
    // supporting ExceptionCodes under w
    static DWORD supporting_exceptions[]
        ...
    };

    auto exception_code = ep->Exception
    for (auto code : supporting_exceptio
        if (code == exception_code) {
            ego::detail::super_ego_cur_
            ego::detail::cur_exception_
            std::longjmp(jump_buffer, 1
        }

    return EXCEPTION_CONTINUE_SEARCH;
    };
...
}
```

```
static DWORD supporting_exceptions[] = {
    EXCEPTION_ACCESS_VIOLATION,
    EXCEPTION_ARRAY_BOUNDS_EXCEEDED,
    EXCEPTION_DATATYPE_MISALIGNMENT,
    EXCEPTION_FLT_DENORMAL_OPERAND,
    EXCEPTION_FLT_DIVIDE_BY_ZERO,
    EXCEPTION_FLT_INEXACT_RESULT,
    EXCEPTION_FLT_INVALID_OPERATION,
    EXCEPTION_FLT_OVERFLOW,
    EXCEPTION_FLT_STACK_CHECK,
    EXCEPTION_FLT_UNDERFLOW,
    EXCEPTION_ILLEGAL_INSTRUCTION,
    EXCEPTION_IN_PAGE_ERROR,
    EXCEPTION_INT_DIVIDE_BY_ZERO,
    EXCEPTION_INT_OVERFLOW,
    EXCEPTION_INVALID_DISPOSITION,
    EXCEPTION_NONCONTINUABLE_EXCEPTION,
    EXCEPTION_PRIV_INSTRUCTION,
};
```


Microsoft Visual Studio 偵錯
failure: Ego Error: SystemException - System Error, SystemException: 3221225477
success: 42
C:\Users\zxc25\git\exception_guard\example\super_ego\Debug\super_ego.exe (處理序 12240) 已結束，代碼為 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [在偵錯停止時自動關閉主控台]。
按任意鍵關閉此視窗 . . .



Recall for conclusion:

TEAMT5
杜 浦 數 位 安 全

```cpp
#include "../single_include/super_ego.hpp"
auto foo() -> ego::expected<int, ego::ego_error> {
    SUPER_EGO(int);
    int *p = nullptr;
    *p = 42;
    return 0;
}
```

Thank you

scc@teamt5.org