Debugging

2024.03.21

SWPP Practice Session

Seunghyeon Nam

Typical Bugfix Process

- Notice an error
- Narrow down the line that causes the error
  - If the program crashes, look for the assertion or invalid pointer
  - If the program yields wrong output, look for the output variable
- Traceback to the line that started to go wrong

Narrowing Down the Line

- How do you locate the exact line that crashes?
  - Guess the location
  - Insert some `std::cout` or `std::cerr` all over the code
  - Rebuild
  - Look for the last printed message
  - Repeat until you actually pinpoint the location

Narrowing Down the Line

- This is horribly inefficient!
  - Taking a wild guess in a large codebase purely depends on luck
  - Rebuilding a large codebase may take minutes or even hours
  - And you have to repeat it until you actually find the bug
- Locating a single bug may already take hours or days

Traceback

- Most of the errors cannot be fixed locally
  - It is likely that the code that ‘triggers’ the error is not a bug
  - The code that ‘leads to’ the error is the real verdict
  - But these two are usually far away from each other...
- You have to locate the code that first went wrong
  - Narrow down, take a step back, narrow down, again and again

Debugger to the Rescue

- Debugger can control the execution of your program
  - Line by line
  - In and out of function
  - Pause on assertion, throw, catch, breakpoint

Debugger to the Rescue

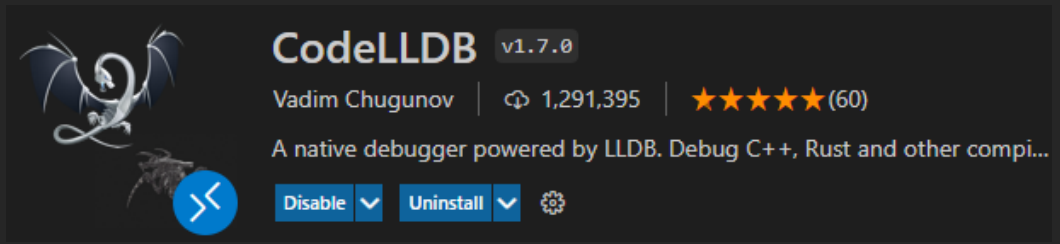
- Debugger can expose the execution context of your program
  - Call stack
  - Local/global variables and values

Using the Debugger

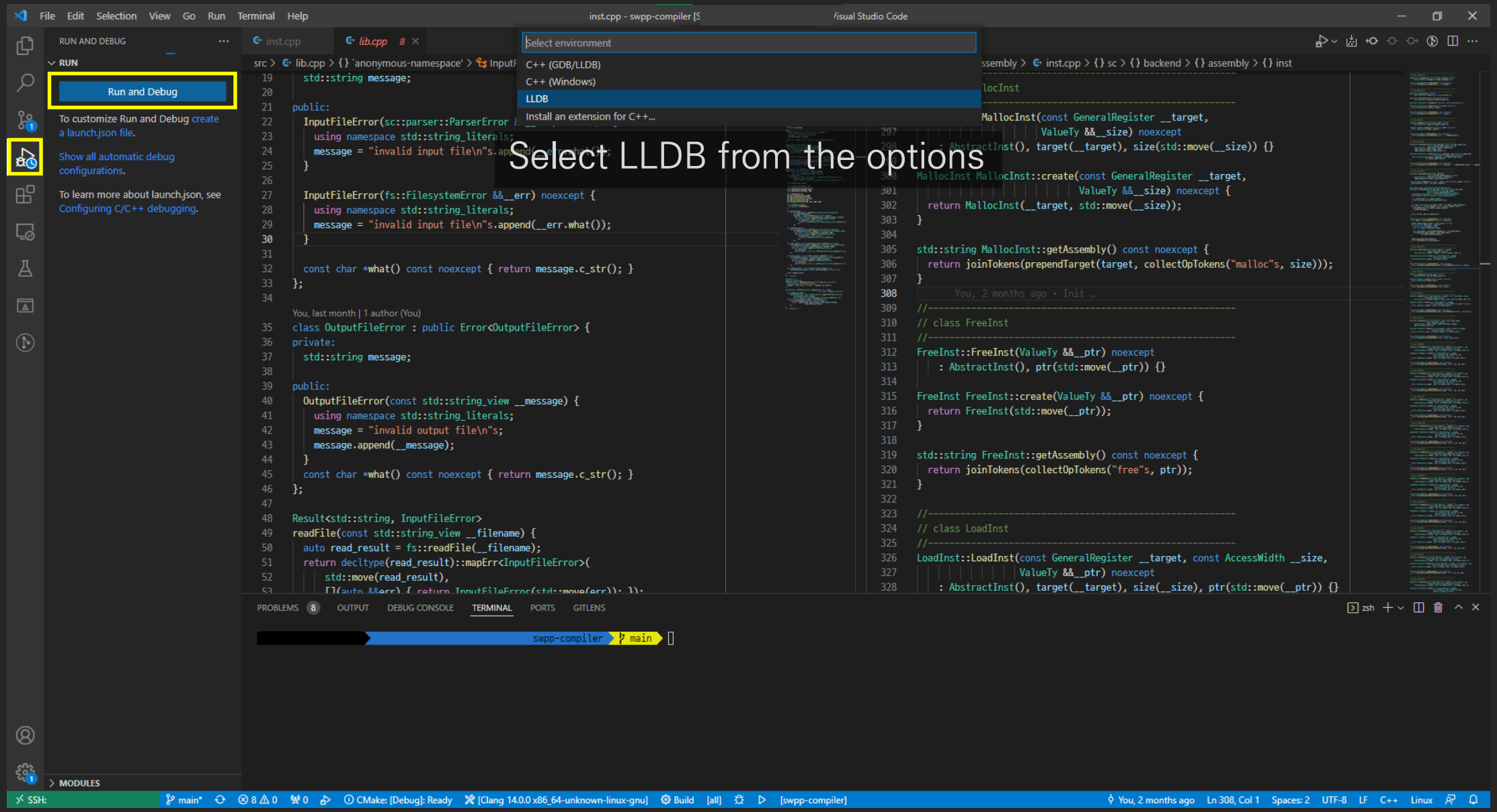
- LLDB: LLVM debugger
  - You have to enable LLDB project when building the LLVM
  - Already included if you built the LLVM using class repo script
- You must build your program with clang
- You must build your program in debug mode

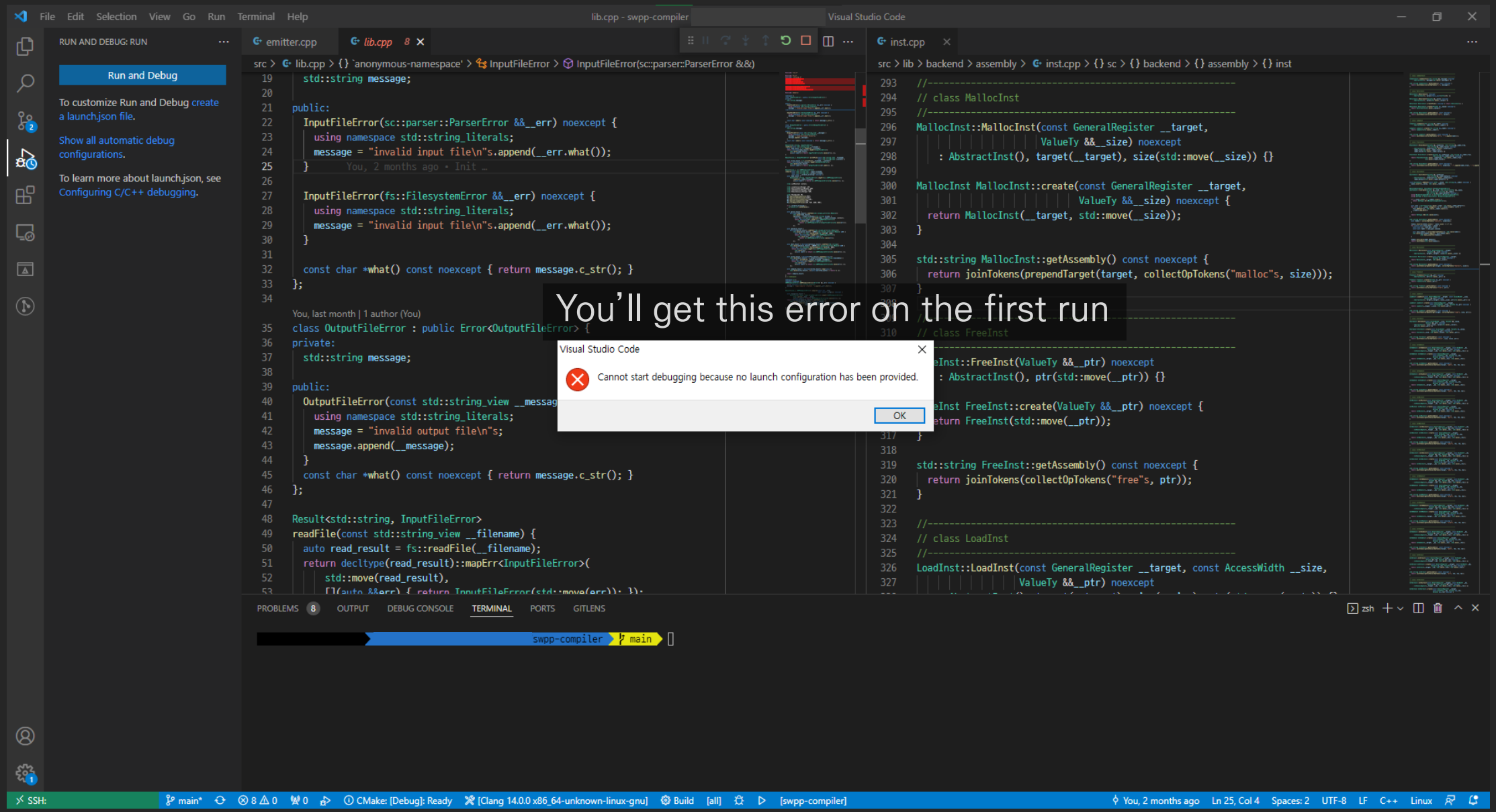
Using the Debugger

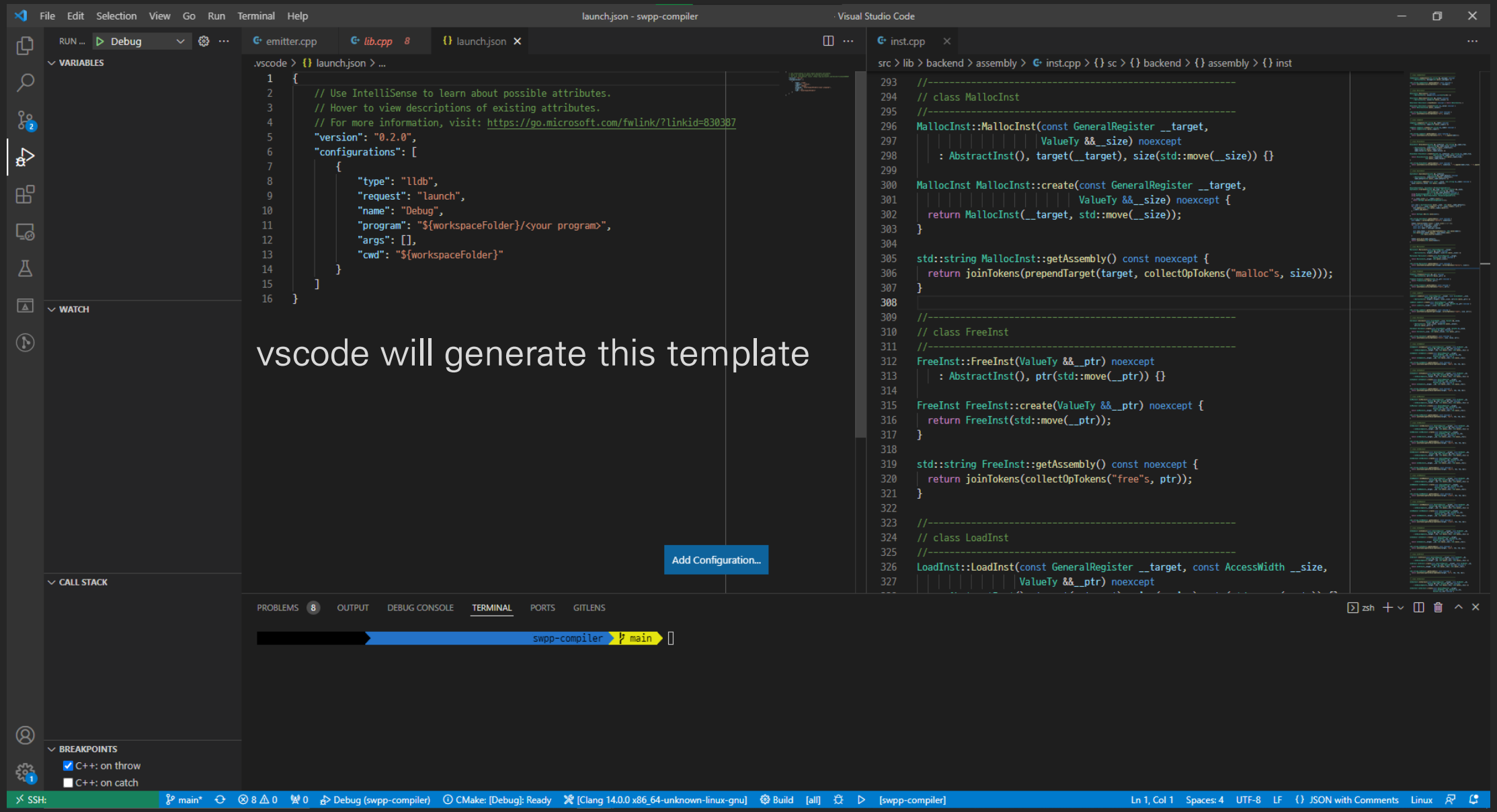
- We'll use vscode extension for convenience
  - CodeLLDB

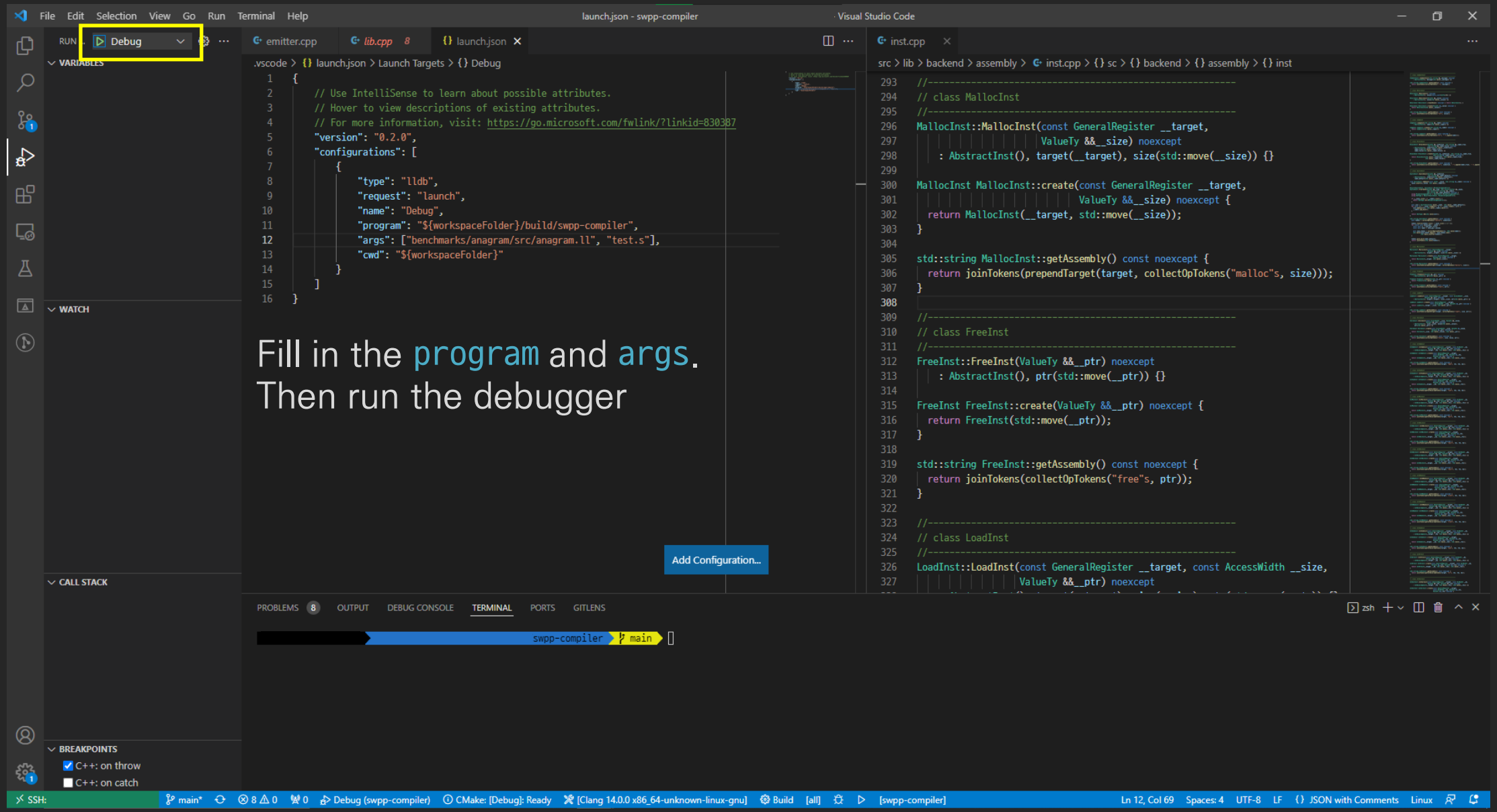


- LLDB directory should be added to your PATH
  - What is PATH?


File Edit Selection View Go Run Terminal Help
inst.cpp - swpp-compiler [S
/isual Studio Code
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
To learn more about launch.json, see Configuring C/C++ debugging.
inst.cpp
lib.cpp 8
Select environment
C++ (GDB/LLDB)
C++ (Windows)
LLDB
Install an extension for C++...
Select LLDB from the options
src > lib.cpp > {} `anonymous-namespace' > InputF
ssembly > inst.cpp > {} sc > {} backend > {} assembly > {} inst
You, last month | 1 author (You)
You, 2 months ago • Init …
PROBLEMS 8
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
zsh
swpp-compiler main
MODULES
SSH:
main*
CMake: [Debug]: Ready
[Clang 14.0.0 x86_64-unknown-linux-gnu]
Build
[all]
[swpp-compiler]
You, 2 months ago
Ln 308, Col 1
Spaces: 2
UTF-8
LF
C++
Linux

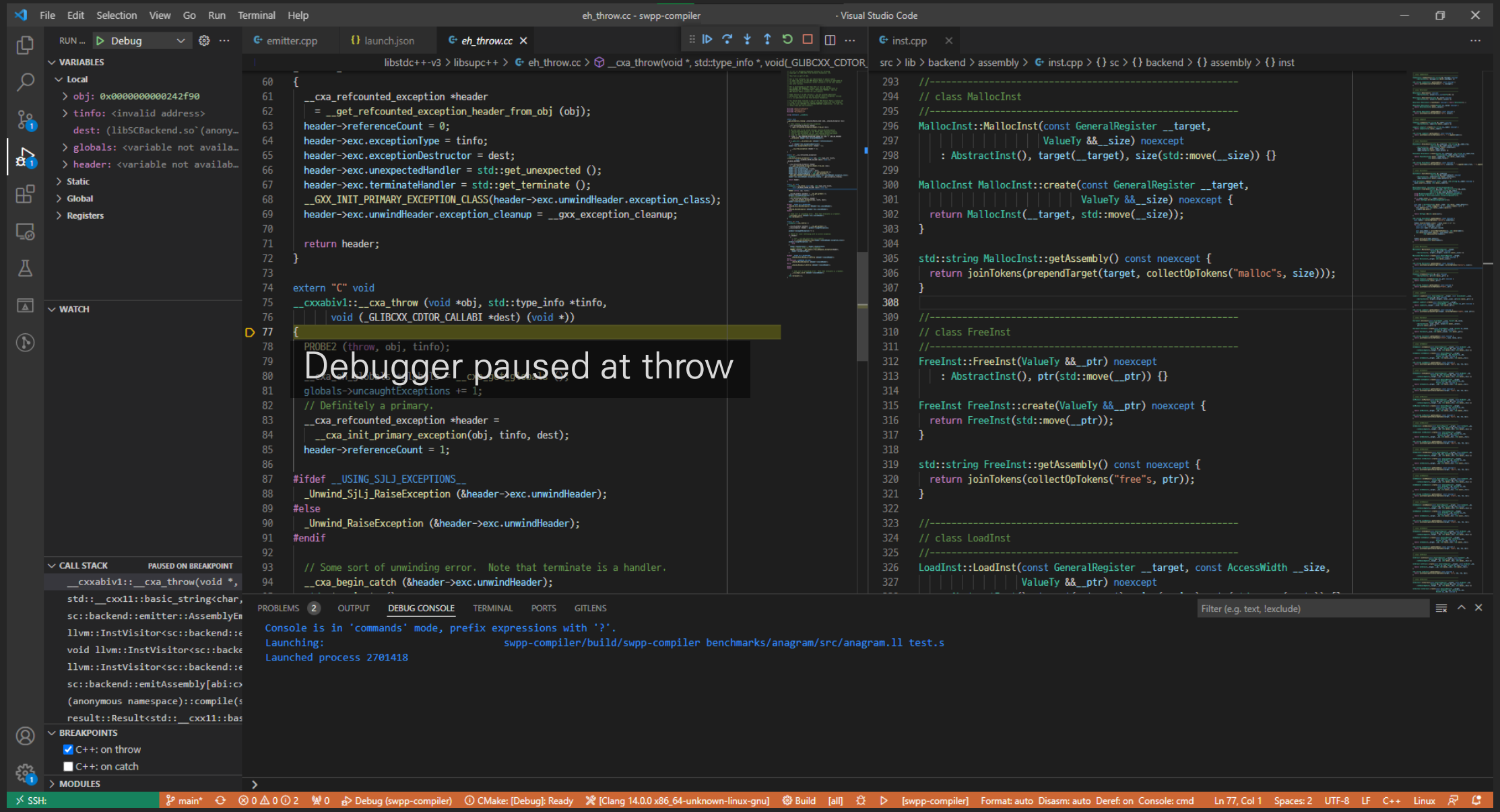

File Edit Selection View Go Run Terminal Help
lib.cpp - swpp-compiler
Visual Studio Code
RUN AND DEBUG: RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
To learn more about launch.json, see Configuring C/C++ debugging.
emitter.cpp
lib.cpp 8
inst.cpp
src > lib.cpp > {} `anonymous-namespace' > InputFileError > InputFileError(sc::parser::ParserError &&)
src > lib > backend > assembly > inst.cpp > {} sc > {} backend > {} assembly > {} inst
You'll get this error on the first run
Visual Studio Code
Cannot start debugging because no launch configuration has been provided.
OK
PROBLEMS 8
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
zsh
swpp-compiler
main
SSH:
main*
CMake: [Debug]: Ready
[Clang 14.0.0 x86_64-unknown-linux-gnu]
Build
[all]
[swpp-compiler]
You, 2 months ago
Ln 25, Col 4
Spaces: 2
UTF-8
LF
C++
Linux

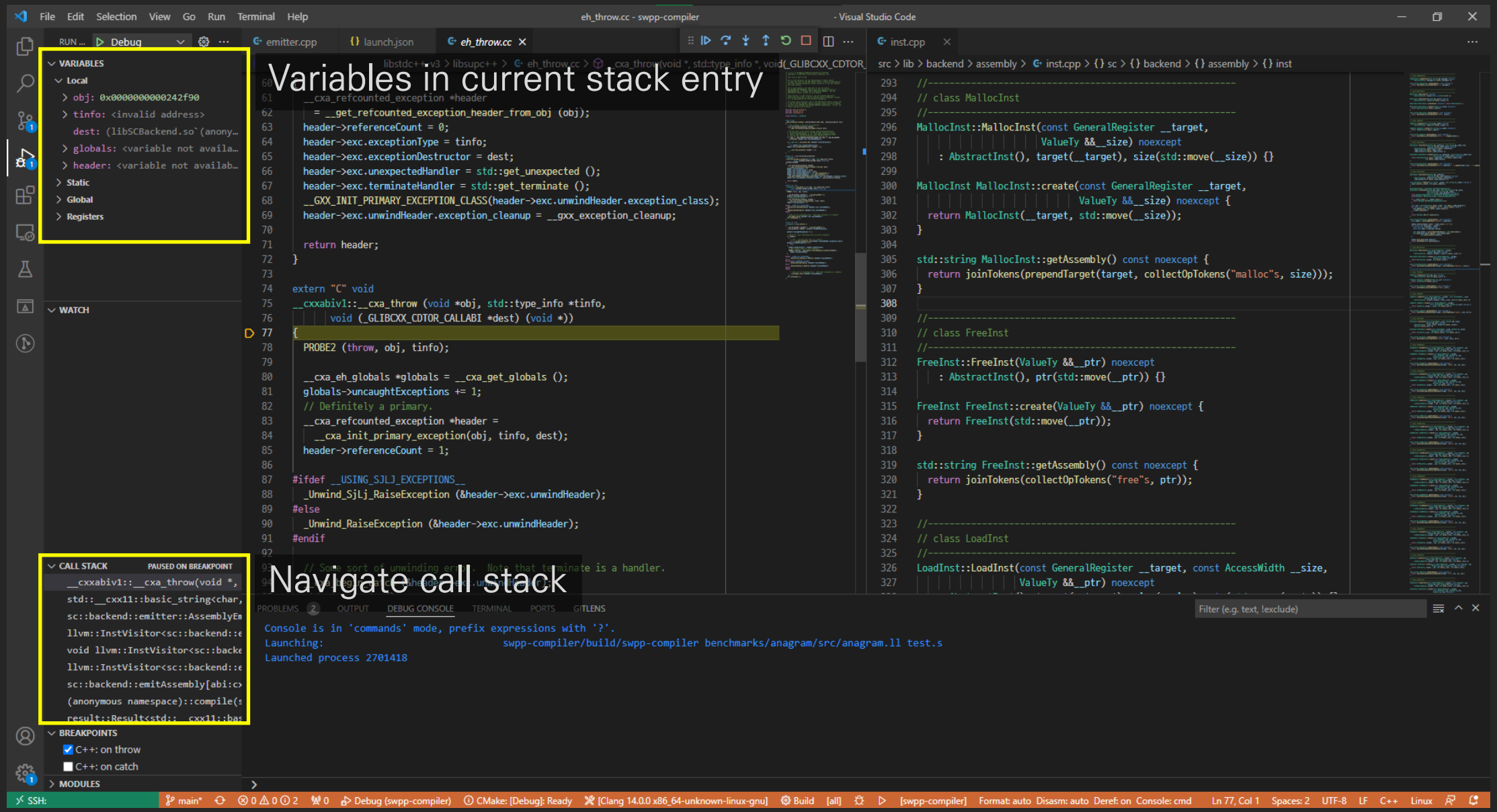

File Edit Selection View Go Run Terminal Help
launch.json - swpp-compiler
Visual Studio Code
RUN ... Debug
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
C++: on throw
C++: on catch
emitter.cpp
lib.cpp 8
launch.json
.vscode > launch.json > ...
{
// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
"version": "0.2.0",
"configurations": [
{
"type": "lldb",
"request": "launch",
"name": "Debug",
"program": "${workspaceFolder}/<your program>",
"args": [],
"cwd": "${workspaceFolder}"
}
]
}
vscode will generate this template
Add Configuration...
inst.cpp
src > lib > backend > assembly > inst.cpp > {} sc > {} backend > {} assembly > {} inst
//----------
// class MallocInst
//----------
MallocInst::MallocInst(const GeneralRegister __target,
ValueTy &&__size) noexcept
: AbstractInst(), target(__target), size(std::move(__size)) {}
MallocInst MallocInst::create(const GeneralRegister __target,
ValueTy &&__size) noexcept {
return MallocInst(__target, std::move(__size));
}
std::string MallocInst::getAssembly() const noexcept {
return joinTokens(prependTarget(target, collectOpTokens("malloc"s, size)));
}
//----------
// class FreeInst
//----------
FreeInst::FreeInst(ValueTy &&__ptr) noexcept
: AbstractInst(), ptr(std::move(__ptr)) {}
FreeInst FreeInst::create(ValueTy &&__ptr) noexcept {
return FreeInst(std::move(__ptr));
}
std::string FreeInst::getAssembly() const noexcept {
return joinTokens(collectOpTokens("free"s, ptr));
}
//----------
// class LoadInst
//----------
LoadInst::LoadInst(const GeneralRegister __target, const AccessWidth __size,
ValueTy &&__ptr) noexcept
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
zsh
swpp-compiler main
SSH: main* 8 0 0 Debug (swpp-compiler) CMake: [Debug]: Ready [Clang 14.0.0 x86_64-unknown-linux-gnu] Build [all] [swpp-compiler]
Ln 1, Col 1 Spaces: 4 UTF-8 LF JSON with Comments Linux

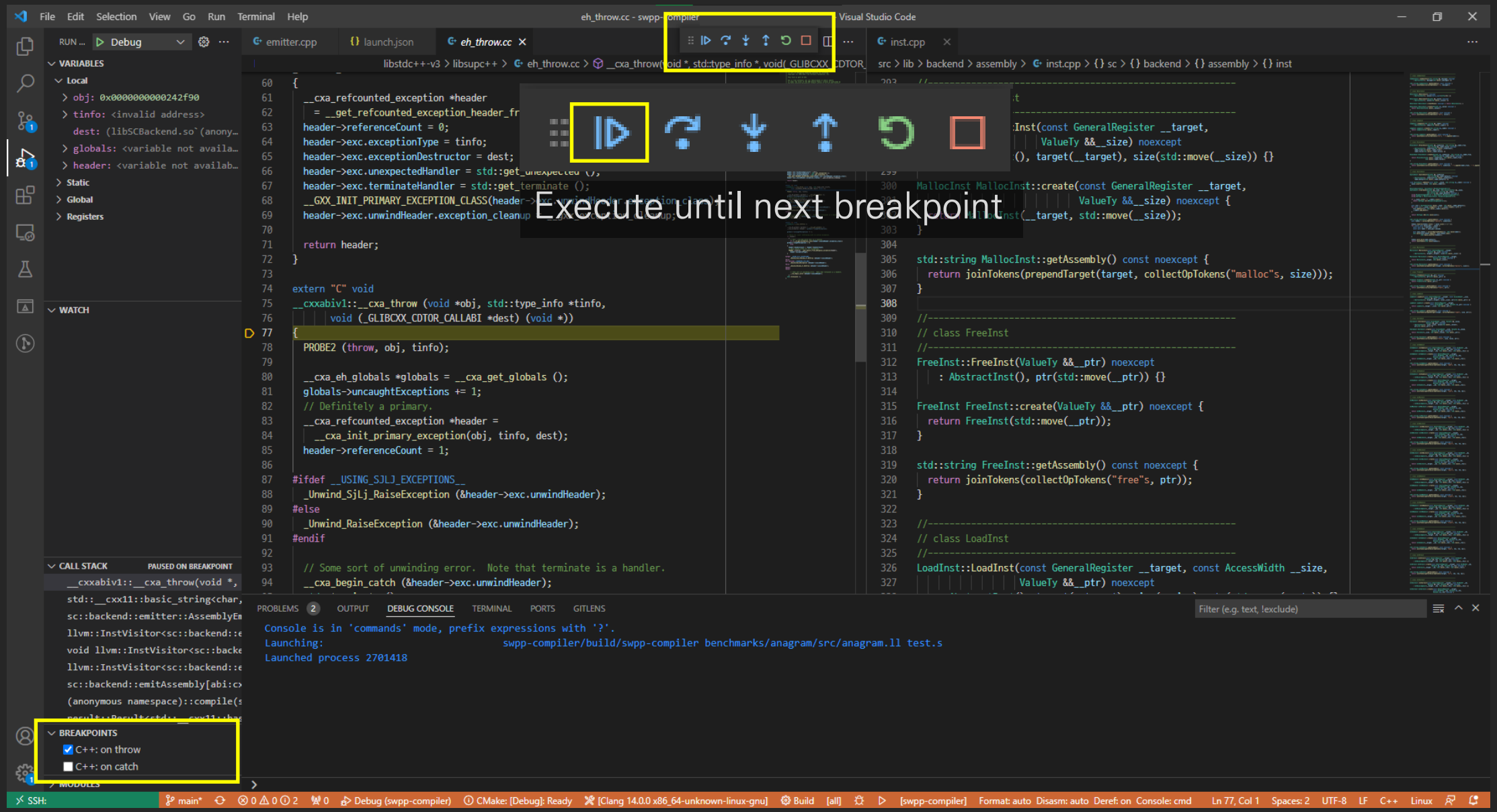


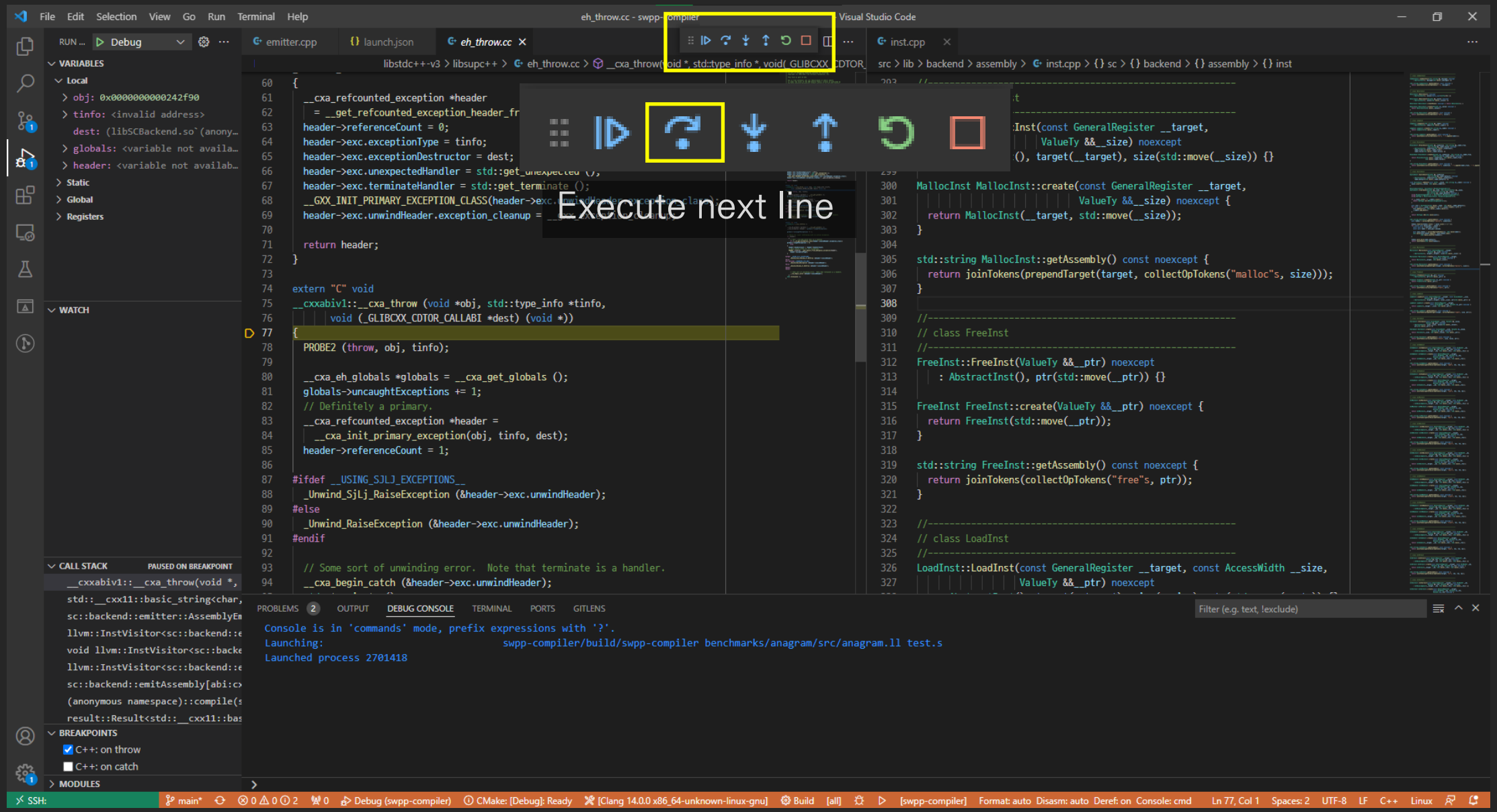

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "lldb",
            "request": "launch",
            "name": "Debug",
            "program": "${workspaceFolder}/build/swpp-compiler",
            "args": ["benchmarks/anagram/src/anagram.ll", "test.s"],
            "cwd": "${workspaceFolder}"
        }
    ]
}
```

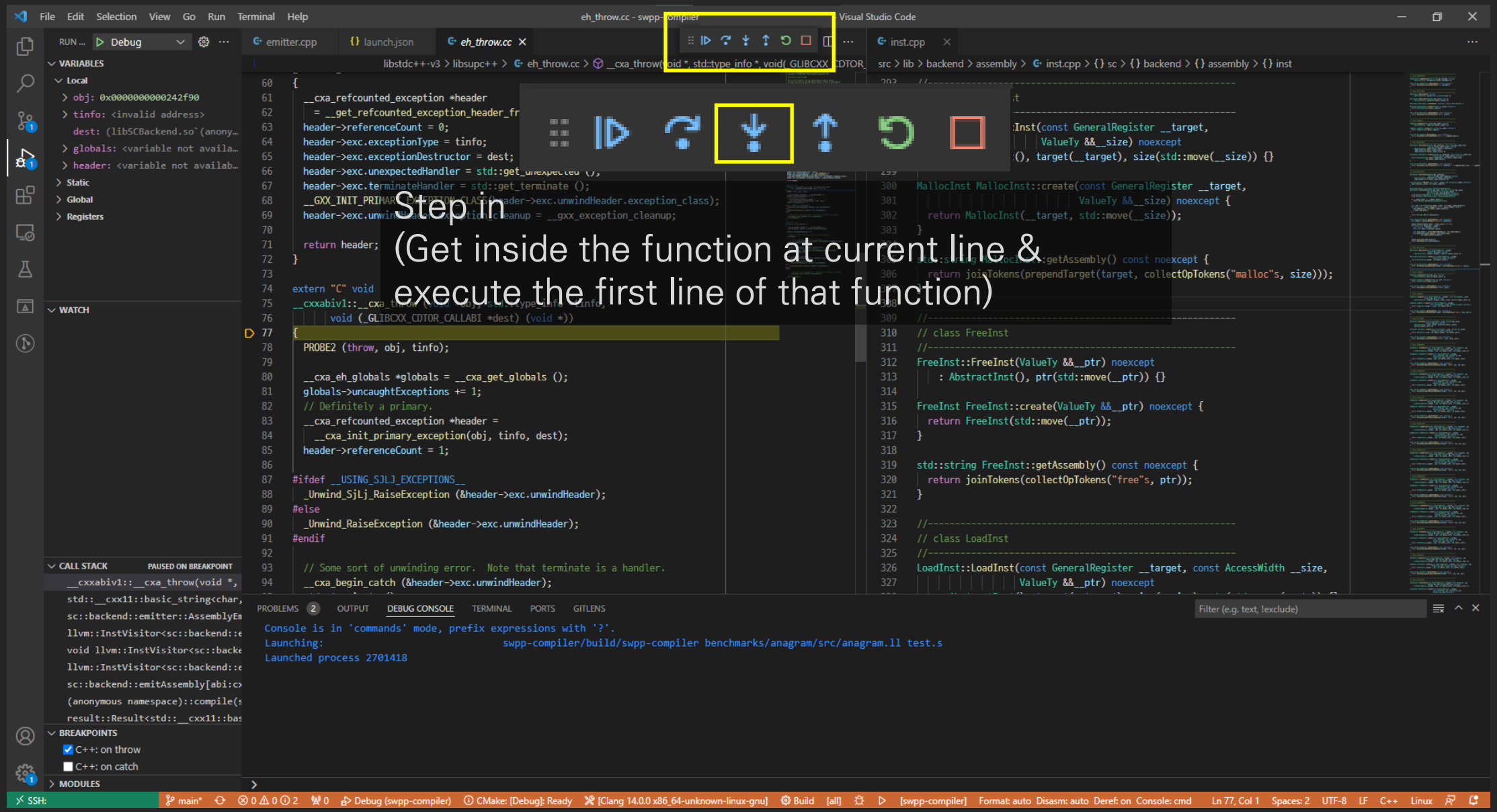
```
//------------------------------------------------------------
// class MallocInst
//------------------------------------------------------------
MallocInst::MallocInst(const GeneralRegister __target,
                       ValueTy &&__size) noexcept
    : AbstractInst(), target(__target), size(std::move(__size)) {}

MallocInst MallocInst::create(const GeneralRegister __target,
                              ValueTy &&__size) noexcept {
  return MallocInst(__target, std::move(__size));
}

std::string MallocInst::getAssembly() const noexcept {
  return joinTokens(prependTarget(target, collectOpTokens("malloc"s, size)));
}

//------------------------------------------------------------
// class FreeInst
//------------------------------------------------------------
FreeInst::FreeInst(ValueTy &&__ptr) noexcept
    : AbstractInst(), ptr(std::move(__ptr)) {}

FreeInst FreeInst::create(ValueTy &&__ptr) noexcept {
  return FreeInst(std::move(__ptr));
}

std::string FreeInst::getAssembly() const noexcept {
  return joinTokens(collectOpTokens("free"s, ptr));
}

//------------------------------------------------------------
// class LoadInst
//------------------------------------------------------------
LoadInst::LoadInst(const GeneralRegister __target, const AccessWidth __size,
                   ValueTy &&__ptr) noexcept
```

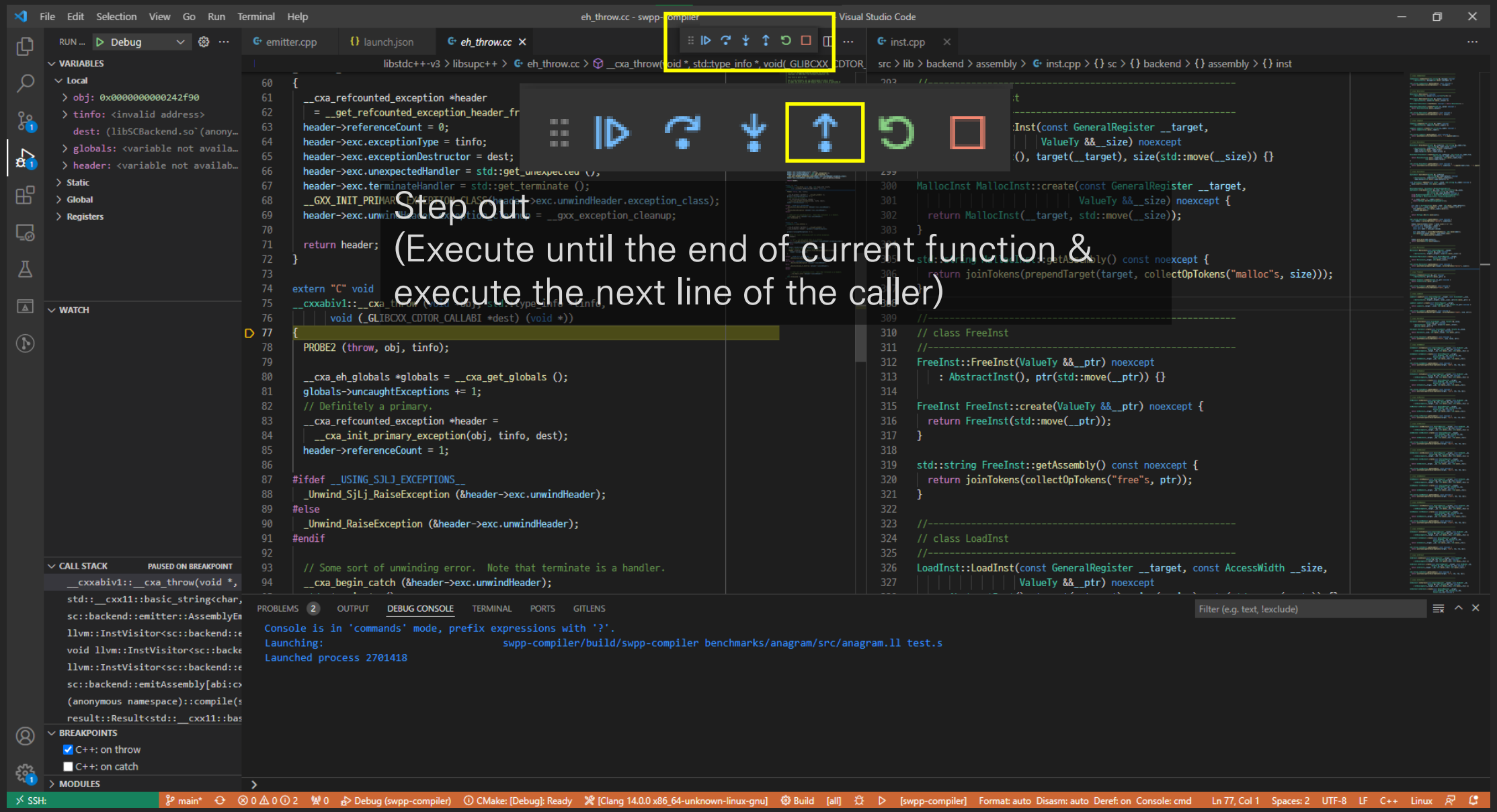

File Edit Selection View Go Run Terminal Help
eh_throw.cc - swpp-compiler - Visual Studio Code
RUN ... Debug
VARIABLES
Local
obj: 0x0000000000242f90
tinfo: <invalid address>
dest: (libSCBackend.so`(anony…
globals: <variable not availa…
header: <variable not availab…
Static
Global
Registers
WATCH
CALL STACK PAUSED ON BREAKPOINT
__cxxabiv1::__cxa_throw(void *,
std::__cxx11::basic_string<char,
sc::backend::emitter::AssemblyEm
llvm::InstVisitor<sc::backend::e
void llvm::InstVisitor<sc::backe
llvm::InstVisitor<sc::backend::e
sc::backend::emitAssembly[abi:c
(anonymous namespace)::compile(s
result::Result<std::__cxx11::bas
BREAKPOINTS
C++: on throw
C++: on catch
MODULES
emitter.cpp
launch.json
eh_throw.cc
libstdc++-v3 > libsupc++ > eh_throw.cc > __cxa_throw(void *, std::type_info *, void(_GLIBCXX_CDTOR_
{
__cxa_refcounted_exception *header
= __get_refcounted_exception_header_from_obj (obj);
header->referenceCount = 0;
header->exc.exceptionType = tinfo;
header->exc.exceptionDestructor = dest;
header->exc.unexpectedHandler = std::get_unexpected ();
header->exc.terminateHandler = std::get_terminate ();
__GXX_INIT_PRIMARY_EXCEPTION_CLASS(header->exc.unwindHeader.exception_class);
header->exc.unwindHeader.exception_cleanup = __gxx_exception_cleanup;
return header;
}
extern "C" void
__cxxabiv1::__cxa_throw (void *obj, std::type_info *tinfo,
void (_GLIBCXX_CDTOR_CALLABI *dest) (void *))
{
PROBE2 (throw, obj, tinfo);
globals->uncaughtExceptions += 1;
// Definitely a primary.
__cxa_refcounted_exception *header =
__cxa_init_primary_exception(obj, tinfo, dest);
header->referenceCount = 1;
#ifdef __USING_SJLJ_EXCEPTIONS__
_Unwind_SjLj_RaiseException (&header->exc.unwindHeader);
#else
_Unwind_RaiseException (&header->exc.unwindHeader);
#endif
// Some sort of unwinding error. Note that terminate is a handler.
__cxa_begin_catch (&header->exc.unwindHeader);
Debugger paused at throw
inst.cpp
src > lib > backend > assembly > inst.cpp > {} sc > {} backend > {} assembly > {} inst
//----------
// class MallocInst
//----------
MallocInst::MallocInst(const GeneralRegister __target,
ValueTy &&__size) noexcept
: AbstractInst(), target(__target), size(std::move(__size)) {}
MallocInst MallocInst::create(const GeneralRegister __target,
ValueTy &&__size) noexcept {
return MallocInst(__target, std::move(__size));
}
std::string MallocInst::getAssembly() const noexcept {
return joinTokens(prependTarget(target, collectOpTokens("malloc"s, size)));
}
//----------
// class FreeInst
//----------
FreeInst::FreeInst(ValueTy &&__ptr) noexcept
: AbstractInst(), ptr(std::move(__ptr)) {}
FreeInst FreeInst::create(ValueTy &&__ptr) noexcept {
return FreeInst(std::move(__ptr));
}
std::string FreeInst::getAssembly() const noexcept {
return joinTokens(collectOpTokens("free"s, ptr));
}
//----------
// class LoadInst
//----------
LoadInst::LoadInst(const GeneralRegister __target, const AccessWidth __size,
ValueTy &&__ptr) noexcept
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Filter (e.g. text, !exclude)
Console is in 'commands' mode, prefix expressions with '?'.
Launching: swpp-compiler/build/swpp-compiler benchmarks/anagram/src/anagram.ll test.s
Launched process 2701418
SSH: main* 0 0 0 2 0 Debug (swpp-compiler) CMake: [Debug]: Ready [Clang 14.0.0 x86_64-unknown-linux-gnu] Build [all] [swpp-compiler] Format: auto Disasm: auto Deref: on Console: cmd Ln 77, Col 1 Spaces: 2 UTF-8 LF C++ Linux

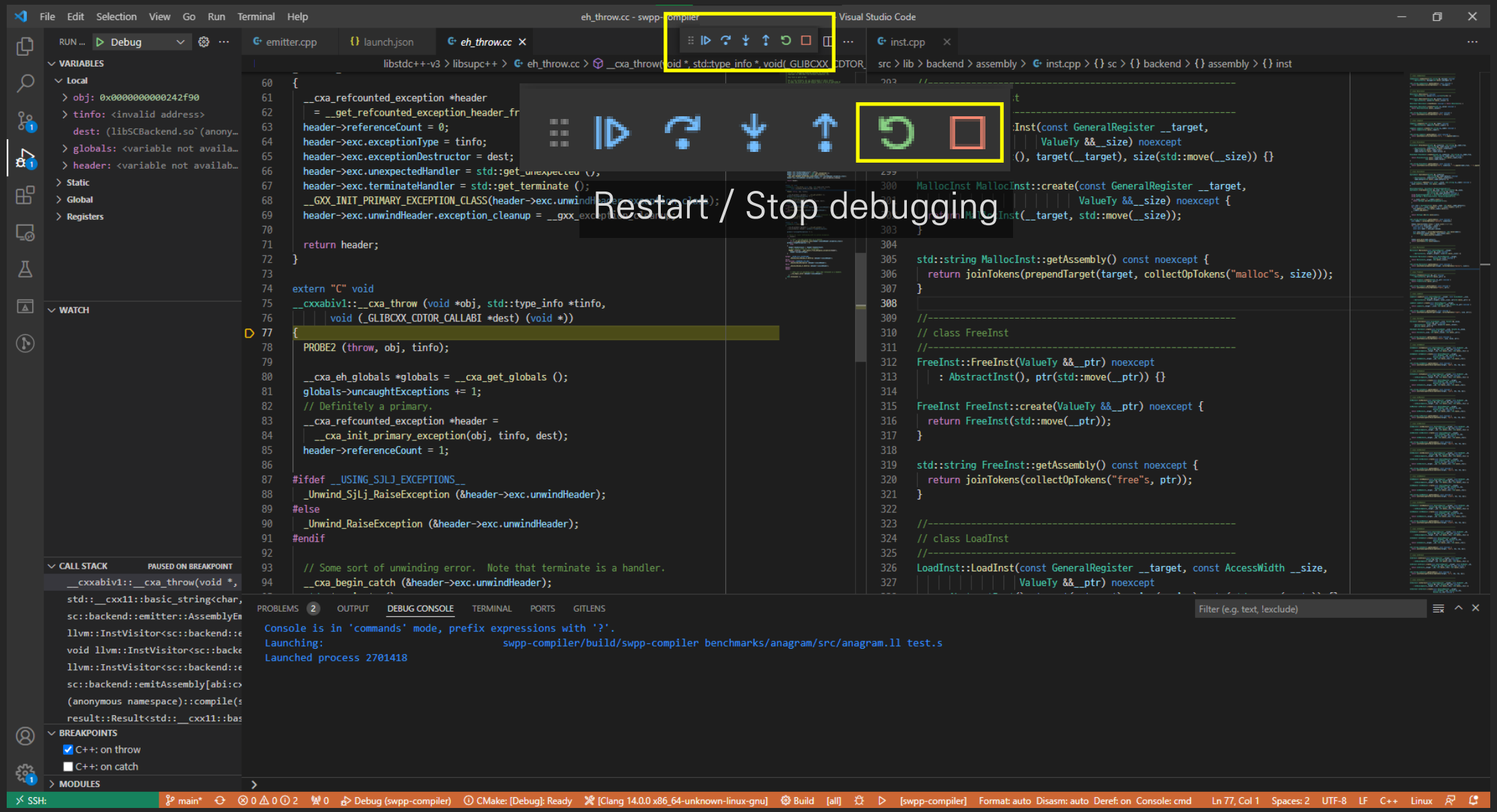

Variables in current stack entry
Navigate call stack
File Edit Selection View Go Run Terminal Help
eh_throw.cc - swpp-compiler - Visual Studio Code
RUN ... Debug
VARIABLES
Local
obj: 0x0000000000242f90
tinfo: <invalid address>
dest: (libSCBackend.so`(anony…
globals: <variable not availa…
header: <variable not availab…
Static
Global
Registers
WATCH
CALL STACK PAUSED ON BREAKPOINT
__cxxabiv1::__cxa_throw(void *,
std::__cxx11::basic_string<char,
sc::backend::emitter::AssemblyEm
llvm::InstVisitor<sc::backend::e
void llvm::InstVisitor<sc::backe
llvm::InstVisitor<sc::backend::e
sc::backend::emitAssembly[abi:c
(anonymous namespace)::compile(s
result::Result<std::__cxx11::bas
BREAKPOINTS
C++: on throw
C++: on catch
MODULES
emitter.cpp
launch.json
eh_throw.cc
inst.cpp
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Filter (e.g. text, !exclude)
Console is in 'commands' mode, prefix expressions with '?'.
Launching: swpp-compiler/build/swpp-compiler benchmarks/anagram/src/anagram.ll test.s
Launched process 2701418
SSH: main* Debug (swpp-compiler) CMake: [Debug]: Ready [Clang 14.0.0 x86_64-unknown-linux-gnu] Build [all] [swpp-compiler] Format: auto Disasm: auto Deref: on Console: cmd Ln 77, Col 1 Spaces: 2 UTF-8 LF C++ Linux


File Edit Selection View Go Run Terminal Help
eh_throw.cc - swpp-compiler - Visual Studio Code
Execute until next breakpoint
RUN ... Debug
VARIABLES
Local
Static
Global
Registers
WATCH
CALL STACK PAUSED ON BREAKPOINT
BREAKPOINTS
C++: on throw
C++: on catch
emitter.cpp
launch.json
eh_throw.cc
inst.cpp
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Console is in 'commands' mode, prefix expressions with '?'.
Launching: swpp-compiler/build/swpp-compiler benchmarks/anagram/src/anagram.ll test.s
Launched process 2701418


File Edit Selection View Go Run Terminal Help
eh_throw.cc - swpp-compiler - Visual Studio Code
RUN ... Debug
emitter.cpp
launch.json
eh_throw.cc
inst.cpp
libstdc++-v3 > libsupc++ > eh_throw.cc
src > lib > backend > assembly > inst.cpp > {} sc > {} backend > {} assembly > {} inst
VARIABLES
Local
obj: 0x0000000000242f90
tinfo: <invalid address>
dest: (libSCBackend.so`(anony...
globals: <variable not availa...
header: <variable not availab...
Static
Global
Registers
WATCH
CALL STACK
PAUSED ON BREAKPOINT
__cxxabiv1::__cxa_throw(void *,
std::__cxx11::basic_string<char,
sc::backend::emitter::AssemblyEm
llvm::InstVisitor<sc::backend::e
void llvm::InstVisitor<sc::backe
llvm::InstVisitor<sc::backend::e
sc::backend::emitAssembly[abi:c
(anonymous namespace)::compile(s
result::Result<std::__cxx11::bas
BREAKPOINTS
C++: on throw
C++: on catch
MODULES
Execute next line
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Filter (e.g. text, !exclude)
Console is in 'commands' mode, prefix expressions with '?'.
Launching: swpp-compiler/build/swpp-compiler benchmarks/anagram/src/anagram.ll test.s
Launched process 2701418
SSH:
main*
Debug (swpp-compiler)
CMake: [Debug]: Ready
[Clang 14.0.0 x86_64-unknown-linux-gnu]
Build
[all]
[swpp-compiler]
Format: auto Disasm: auto Deref: on Console: cmd
Ln 77, Col 1
Spaces: 2
UTF-8
LF
C++
Linux


Step in
(Get inside the function at current line &
execute the first line of that function)


File Edit Selection View Go Run Terminal Help
eh_throw.cc - swpp-compiler - Visual Studio Code
RUN ... Debug
emitter.cpp
launch.json
eh_throw.cc
inst.cpp
VARIABLES
Local
Static
Global
Registers
WATCH
CALL STACK
PAUSED ON BREAKPOINT
BREAKPOINTS
C++: on throw
C++: on catch
MODULES
Step out
(Execute until the end of current function &
execute the next line of the caller)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Console is in 'commands' mode, prefix expressions with '?'.
Launching: swpp-compiler/build/swpp-compiler benchmarks/anagram/src/anagram.ll test.s
Launched process 2701418


Restart / Stop debugging

Narrowing Down with Debugger

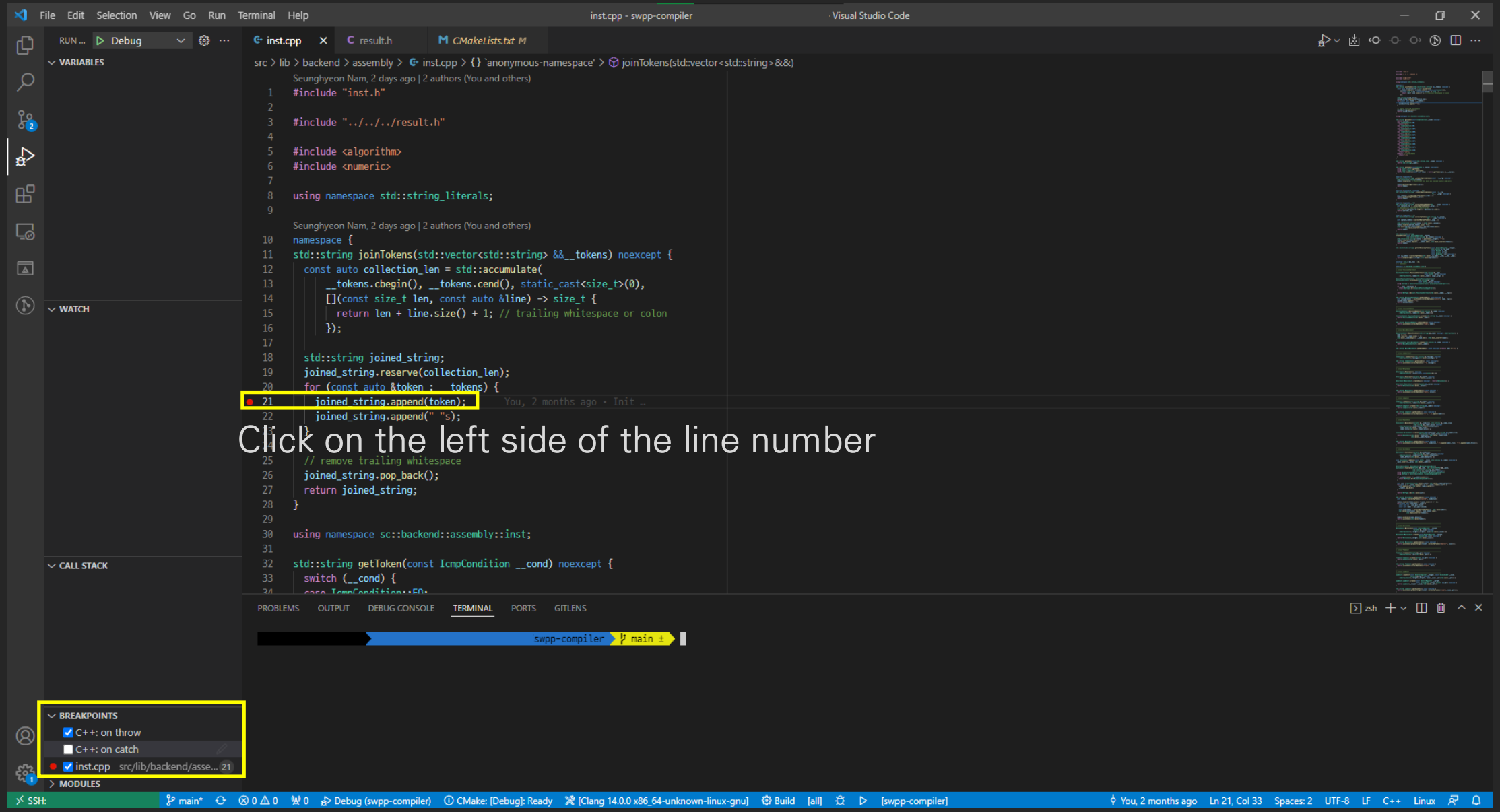
- How do you locate the exact line that crashes?
  - Debugger pinpoints (automatically pauses on) the crash site
  - No need to insert new code, rebuild, etc.

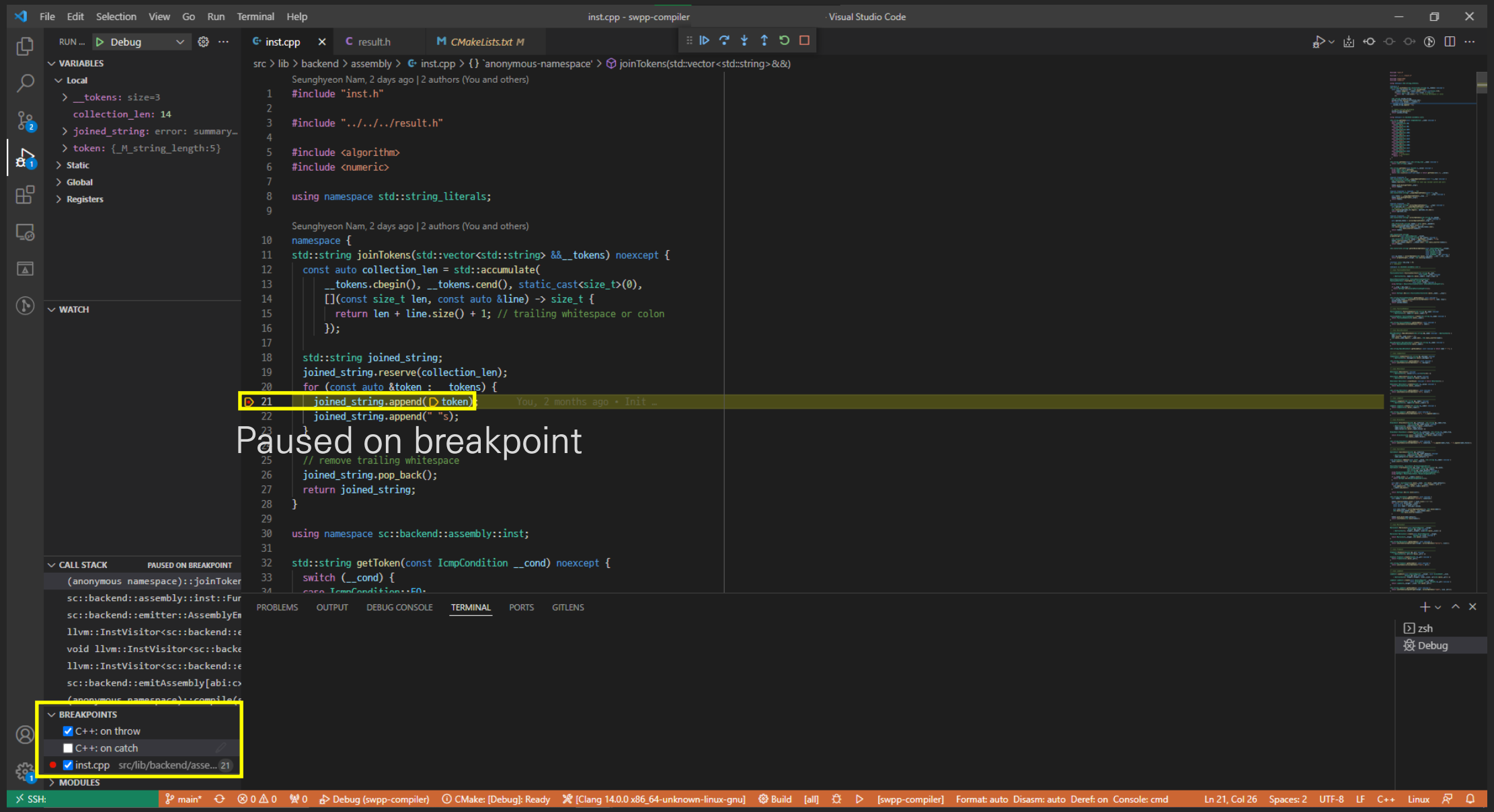
Traceback with Debugger

- You have to locate the code that first went wrong
  - Debugger shows you the call stack at the moment of the crash
  - Clicking is all you need to navigate through the call stacks
  - Find the first call stack with unexpected value or control flow

Traceback with Debugger

- Sometimes you have to monitor the change of values
  - If your code does not throw or crash, debugger won't pause
  - You can use breakpoint to pause execution at a certain point

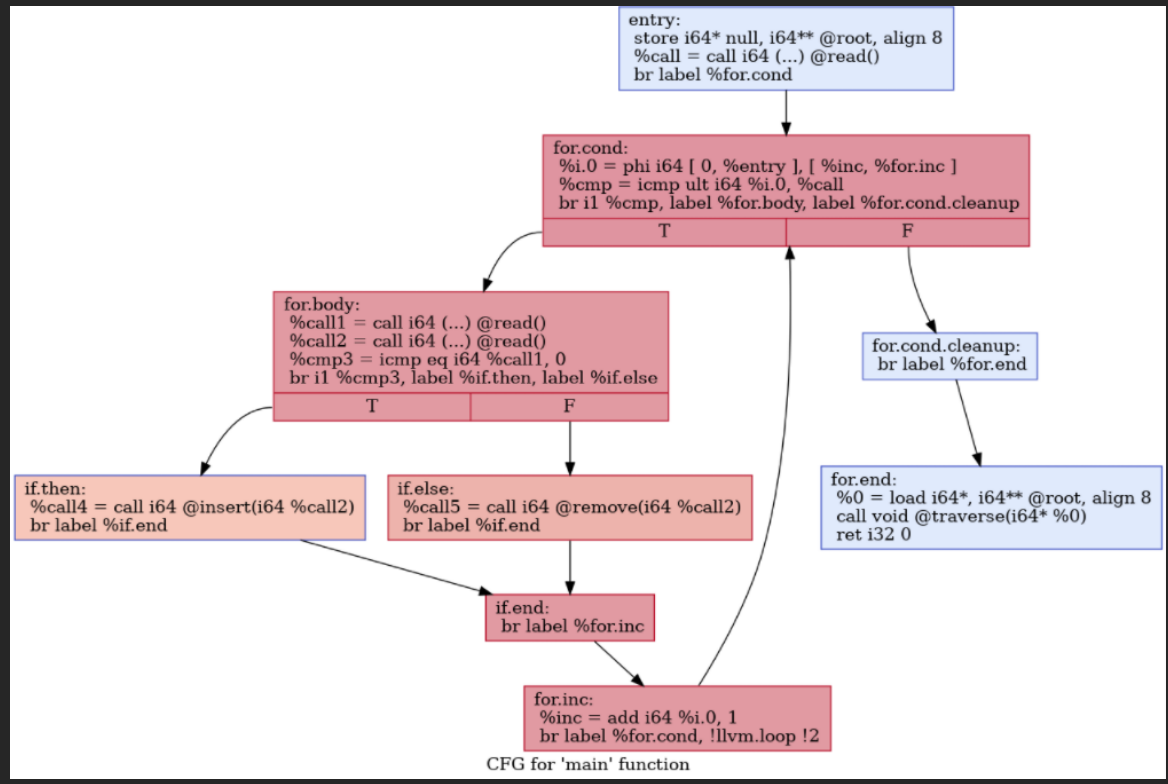

File Edit Selection View Go Run Terminal Help
inst.cpp - swpp-compiler
Visual Studio Code
RUN ... Debug
inst.cpp
result.h
CMakeLists.txt M
src > lib > backend > assembly > inst.cpp > {} `anonymous-namespace' > joinTokens(std::vector<std::string>&&)
Seunghyeon Nam, 2 days ago | 2 authors (You and others)
#include "inst.h"
#include "../../../result.h"
#include <algorithm>
#include <numeric>
using namespace std::string_literals;
Seunghyeon Nam, 2 days ago | 2 authors (You and others)
namespace {
std::string joinTokens(std::vector<std::string> &&__tokens) noexcept {
const auto collection_len = std::accumulate(
__tokens.cbegin(), __tokens.cend(), static_cast<size_t>(0),
[](const size_t len, const auto &line) -> size_t {
return len + line.size() + 1; // trailing whitespace or colon
});
std::string joined_string;
joined_string.reserve(collection_len);
for (const auto &token : __tokens) {
joined_string.append(token);
You, 2 months ago • Init …
joined_string.append(" ");
}
// remove trailing whitespace
joined_string.pop_back();
return joined_string;
}
using namespace sc::backend::assembly::inst;
std::string getToken(const IcmpCondition __cond) noexcept {
switch (__cond) {
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
C++: on throw
C++: on catch
inst.cpp src/lib/backend/asse... 21
MODULES
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
zsh
swpp-compiler main ±
Click on the left side of the line number
SSH:
main*
0 0 0
Debug (swpp-compiler)
CMake: [Debug]: Ready
[Clang 14.0.0 x86_64-unknown-linux-gnu]
Build
[all]
[swpp-compiler]
You, 2 months ago
Ln 21, Col 33
Spaces: 2
UTF-8
LF
C++
Linux


File Edit Selection View Go Run Terminal Help
inst.cpp - swpp-compiler
Visual Studio Code
RUN ... Debug
inst.cpp
result.h
CMakeLists.txt M
src > lib > backend > assembly > inst.cpp > {} `anonymous-namespace' > joinTokens(std::vector<std::string>&&)
VARIABLES
Local
__tokens: size=3
collection_len: 14
joined_string: error: summary...
token: {_M_string_length:5}
Static
Global
Registers
WATCH
CALL STACK
PAUSED ON BREAKPOINT
(anonymous namespace)::joinToker
sc::backend::assembly::inst::Fur
sc::backend::emitter::AssemblyEm
llvm::InstVisitor<sc::backend::e
void llvm::InstVisitor<sc::backe
llvm::InstVisitor<sc::backend::e
sc::backend::emitAssembly[abi:c
BREAKPOINTS
C++: on throw
C++: on catch
inst.cpp src/lib/backend/asse... 21
MODULES
Seunghyeon Nam, 2 days ago | 2 authors (You and others)
#include "inst.h"
#include "../../../result.h"
#include <algorithm>
#include <numeric>
using namespace std::string_literals;
Seunghyeon Nam, 2 days ago | 2 authors (You and others)
namespace {
std::string joinTokens(std::vector<std::string> &&__tokens) noexcept {
const auto collection_len = std::accumulate(
__tokens.cbegin(), __tokens.cend(), static_cast<size_t>(0),
[](const size_t len, const auto &line) -> size_t {
return len + line.size() + 1; // trailing whitespace or colon
});
std::string joined_string;
joined_string.reserve(collection_len);
for (const auto &token : __tokens) {
joined_string.append(token);
You, 2 months ago • Init ...
joined_string.append(" ");
}
// remove trailing whitespace
joined_string.pop_back();
return joined_string;
}
using namespace sc::backend::assembly::inst;
std::string getToken(const IcmpCondition __cond) noexcept {
switch (__cond) {
Paused on breakpoint
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
zsh
Debug
SSH:
main*
Debug (swpp-compiler)
CMake: [Debug]: Ready
[Clang 14.0.0 x86_64-unknown-linux-gnu]
Build
[all]
[swpp-compiler]
Format: auto Disasm: auto Deref: on Console: cmd
Ln 21, Col 26
Spaces: 2
UTF-8
LF
C++
Linux

IR Visualization

- Visualizing the control flow of your IR program can be helpful

```
define dso_local i32 @main() #0 {
entry:
  store i64* null, i64** @root, align 8
  %call = call i64 (...) @read()
  br label %for.cond

for.cond:                                         ; preds = %for.inc, %entry
  %i.0 = phi i64 [ 0, %entry ], [ %inc, %for.inc ]
  %cmp = icmp ult i64 %i.0, %call
  br i1 %cmp, label %for.body, label %for.cond.cleanup

for.cond.cleanup:                                 ; preds = %for.cond
  br label %for.end

for.body:                                         ; preds = %for.cond
  %call1 = call i64 (...) @read()
  %call2 = call i64 (...) @read()
  %cmp3 = icmp eq i64 %call1, 0
  br i1 %cmp3, label %if.then, label %if.else

if.then:                                          ; preds = %for.body
  %call4 = call i64 @insert(i64 %call2)
  br label %if.end

if.else:                                          ; preds = %for.body
  %call5 = call i64 @remove(i64 %call2)
  br label %if.end

if.end:                                           ; preds = %if.else, %if.then
  br label %for.inc
```



IR Visualization

- Install GraphViz
  - Use the package manager to handle dependencies for you
- Run `<llvm-dir>/opt --dot-cfg <IR-program.ll>`
  - You'll get a .dot file for each function in the program
- Run `dot <dot-file.dot> -Tpng -o <image-name.png>`