Compiling C

?

Who works with C on a regular basis?

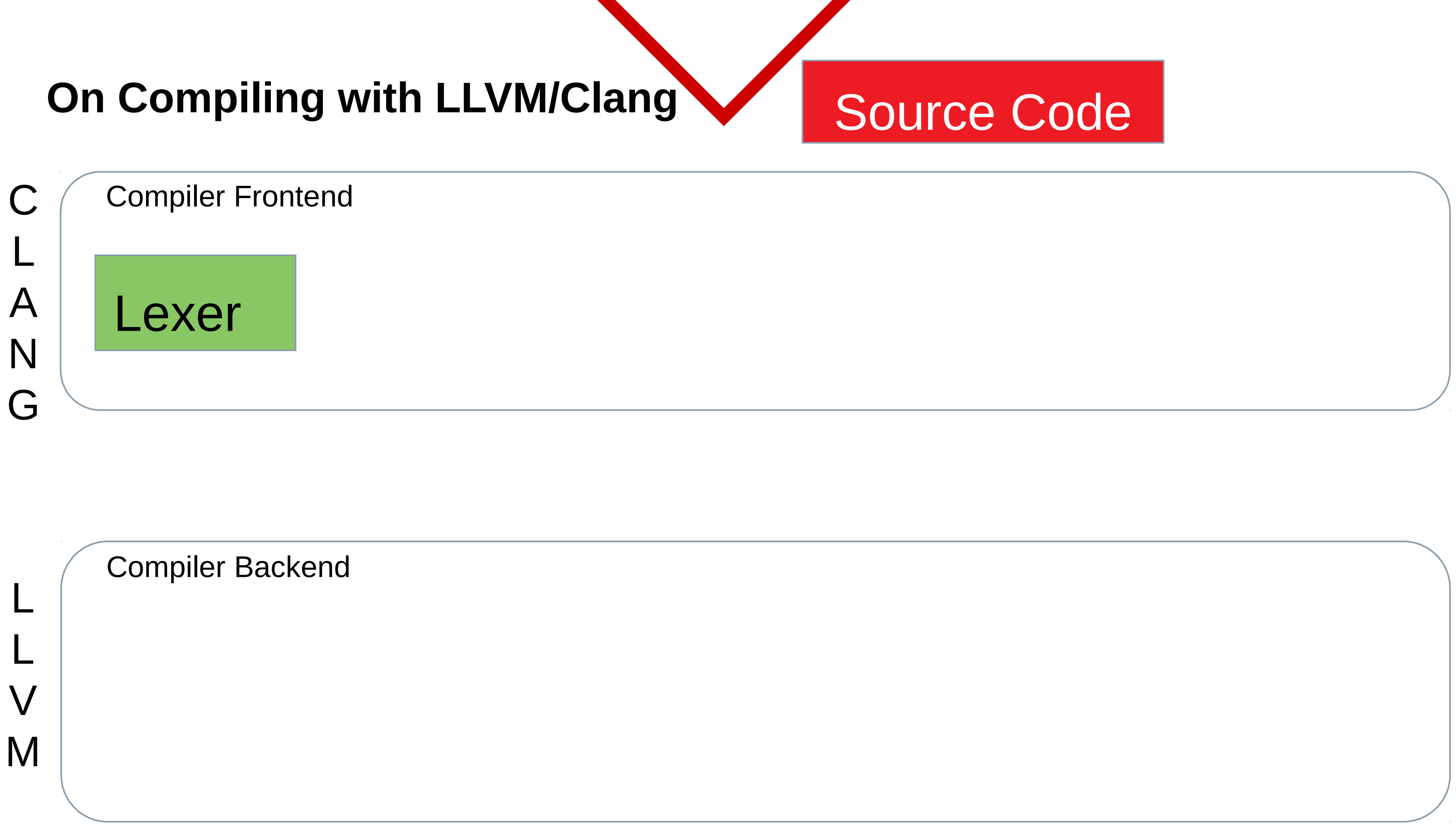
?

Who works or worked with a large C project?

?

What are compilers and why do we even need them?

On Compiling with LLVM/Clang



value=previous + increment * 37

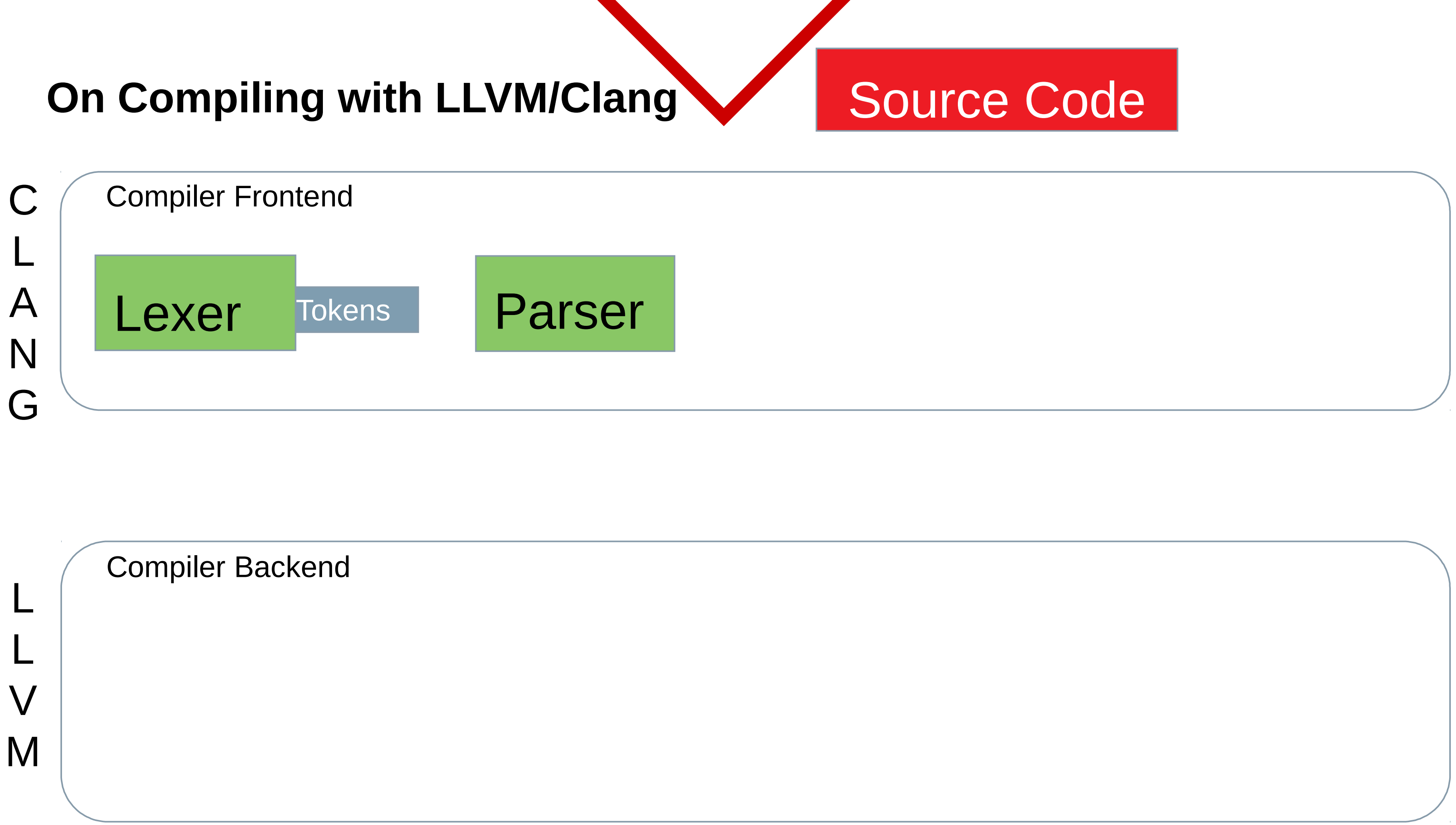
- Groups every statement in lexemes
  - For each lexeme, a token is generated in the form of

    <token-name, token-value>
  - Common token names are identifiers, operators, seperators and keywords

<id, 1><=><id, 2> <+> <id, 3> <*> <37>

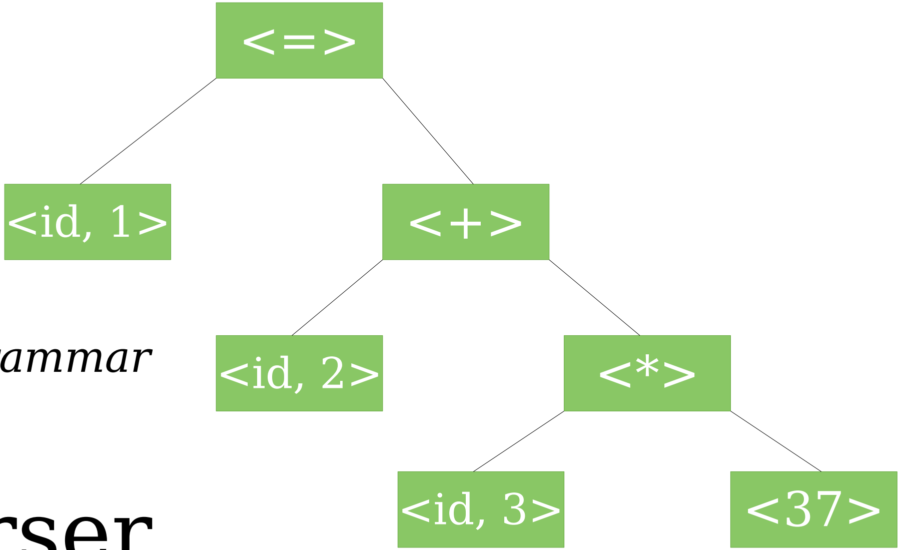
Lexical Analysis

On Compiling with LLVM/Clang

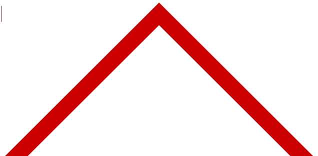


<id, 1><=><id, 2> <+> <id, 3> <*> <37>

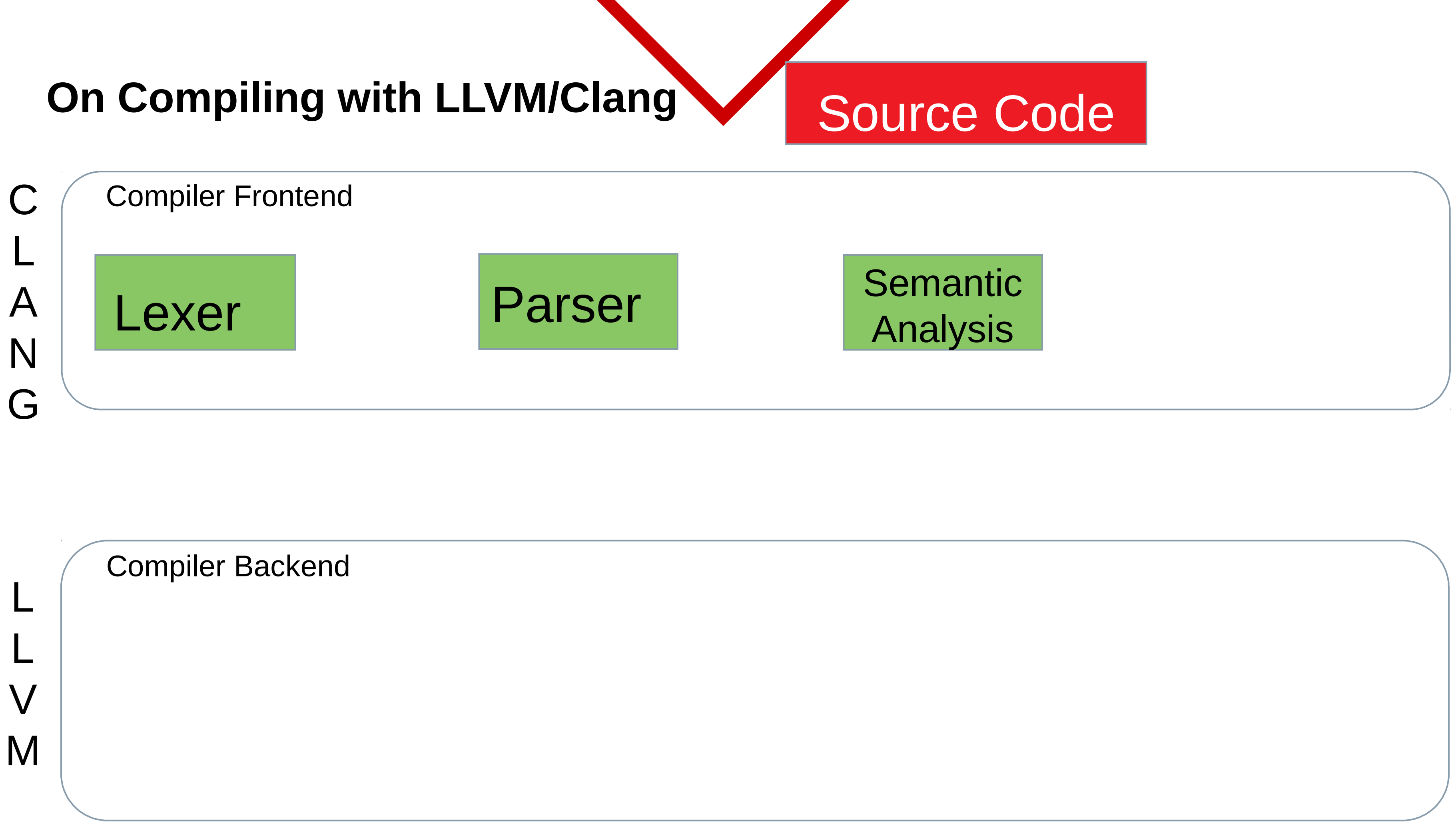
- Syntax Analysis
  - Tokes are used to create a *syntax tree*
- *Syntax Tree* uses *context-free grammar*
  - Mathematical linguistic



Parser

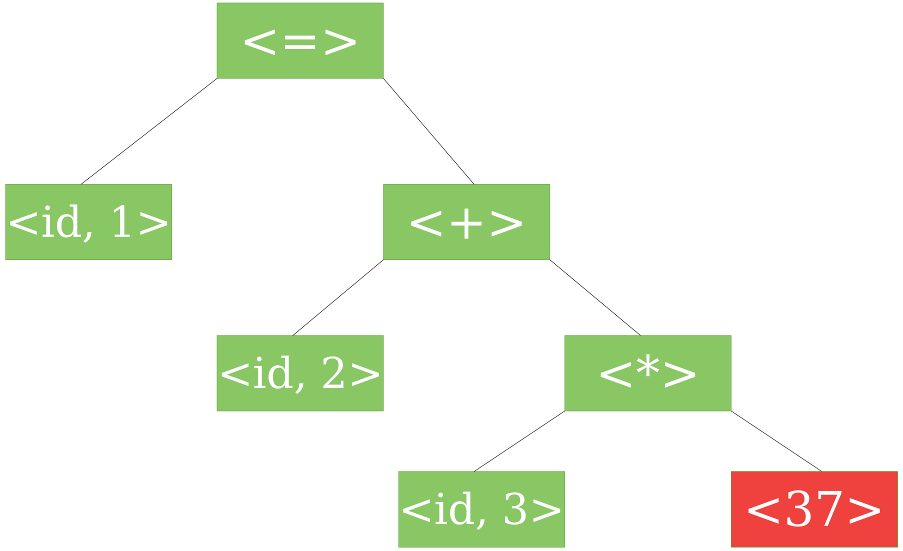


On Compiling with LLVM/Clang

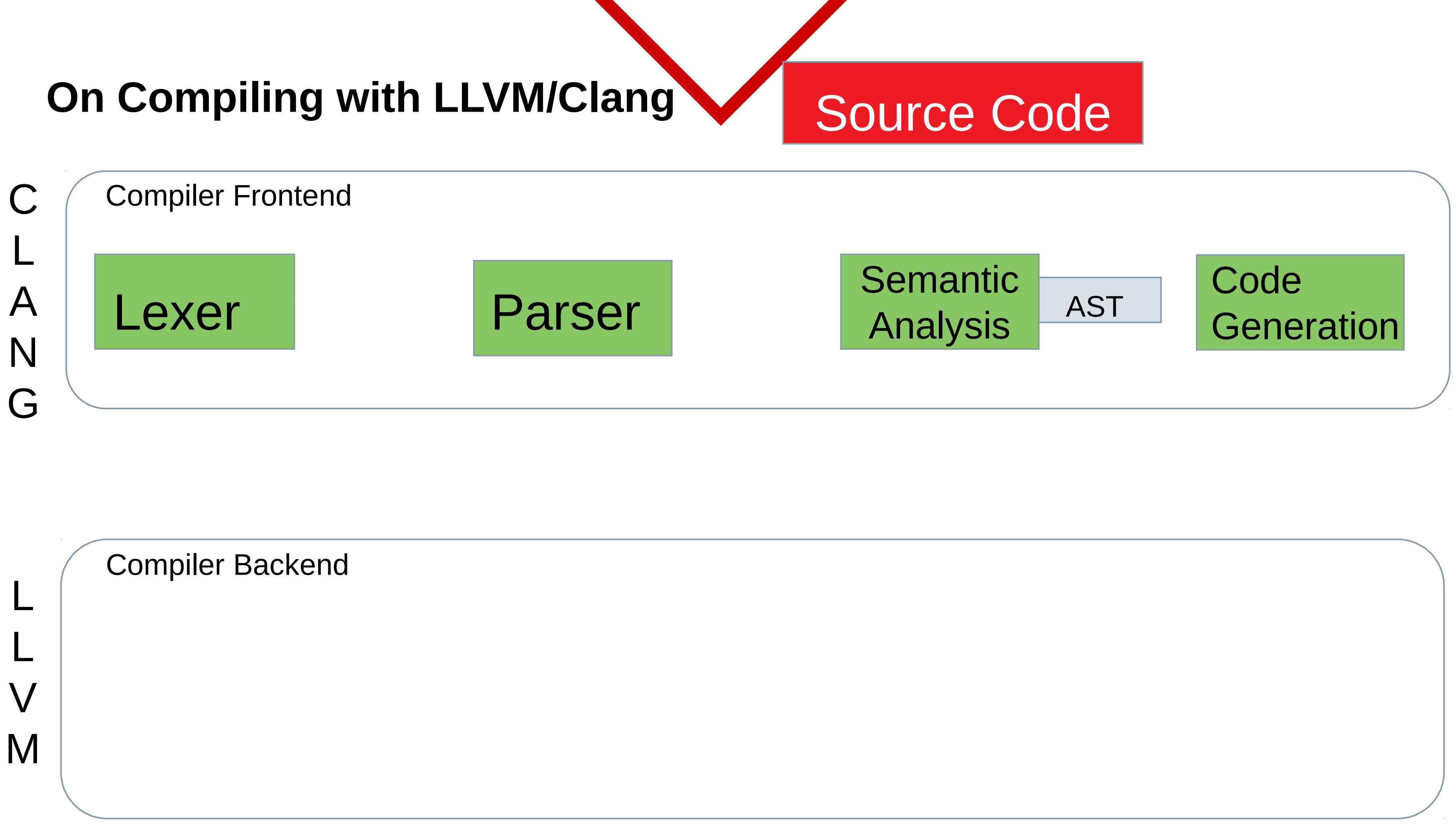


Semantic Analysis

- Checks for semantic consistency
  - initialization of variables
  - type information & checking
    - Coercions
- Context sensitive



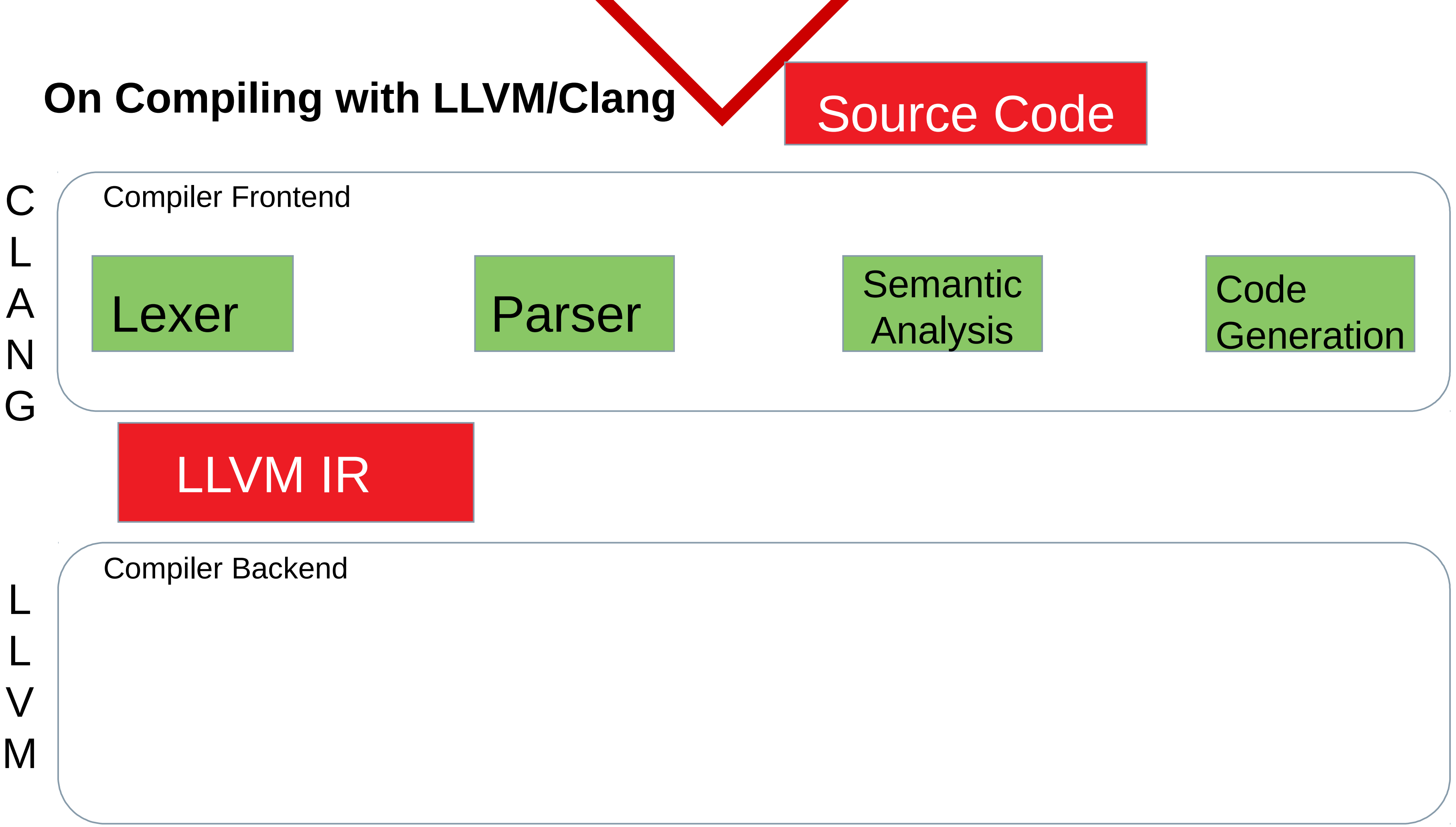
On Compiling with LLVM/Clang



- To compile *m* languages for *n* architectures (mxn), ideally, you would need m frontends and n backends.
- Low- or High-Level Representation
- Translates the AST in a form of intermediate representation:
  - Three-Address Code
  - LLVM IR
  - AST

Intermediate Code Generation

On Compiling with LLVM/Clang

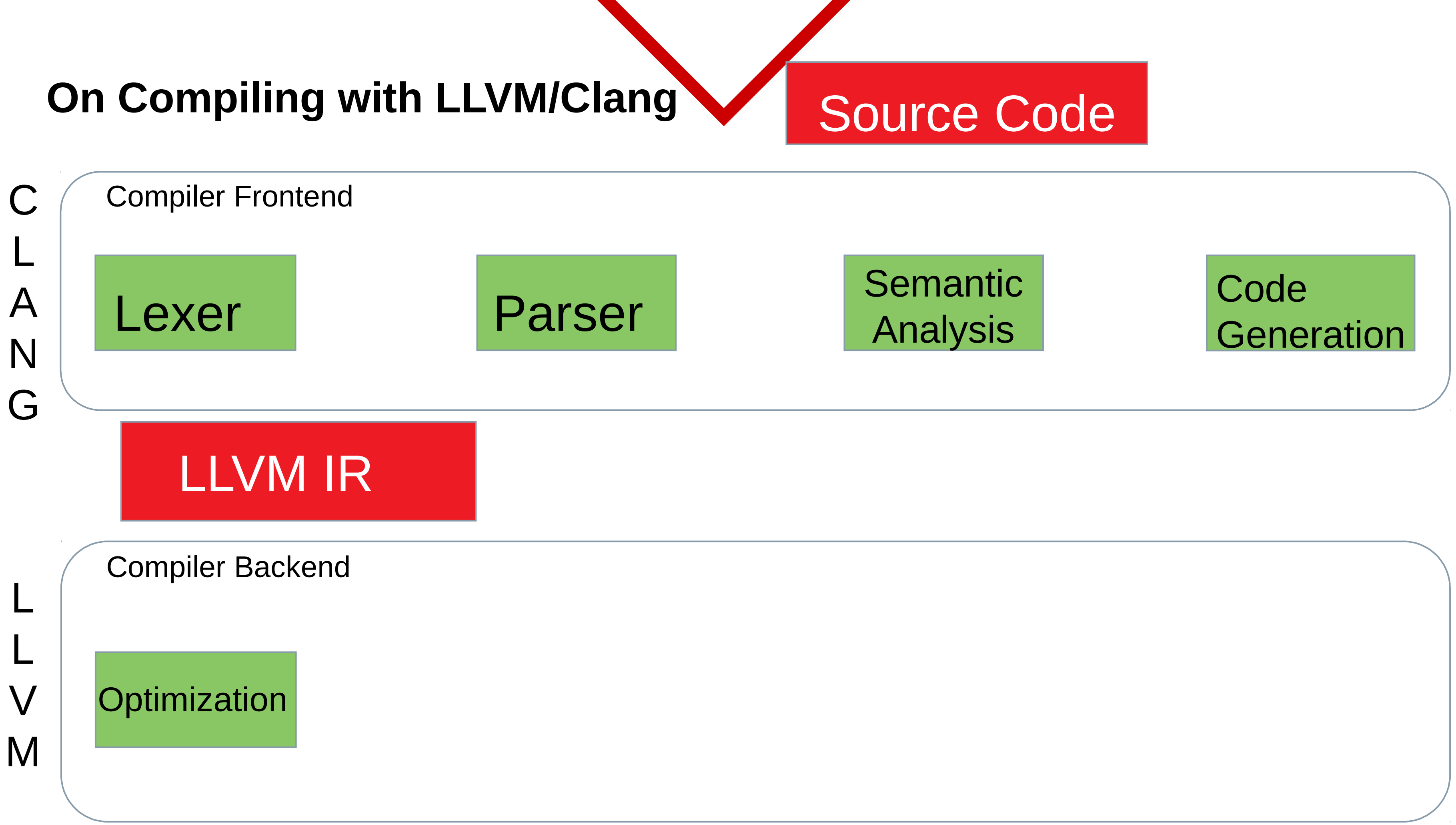


LLVM IR

- Low-level programming language (e.g. Assembly)
- Link between Front End and Back End
- Strongly typed – simple typing system
  - Integer
    - i1
    - i32
    - i1000282

- clang -S -emit-llvm file.c
- What do the following commands do?
  - %5 = mul nsw i32 %3, %4
  - %4 = alloca i32, align 4
  - %6 = call i32 @square(i32 7)

On Compiling with LLVM/Clang

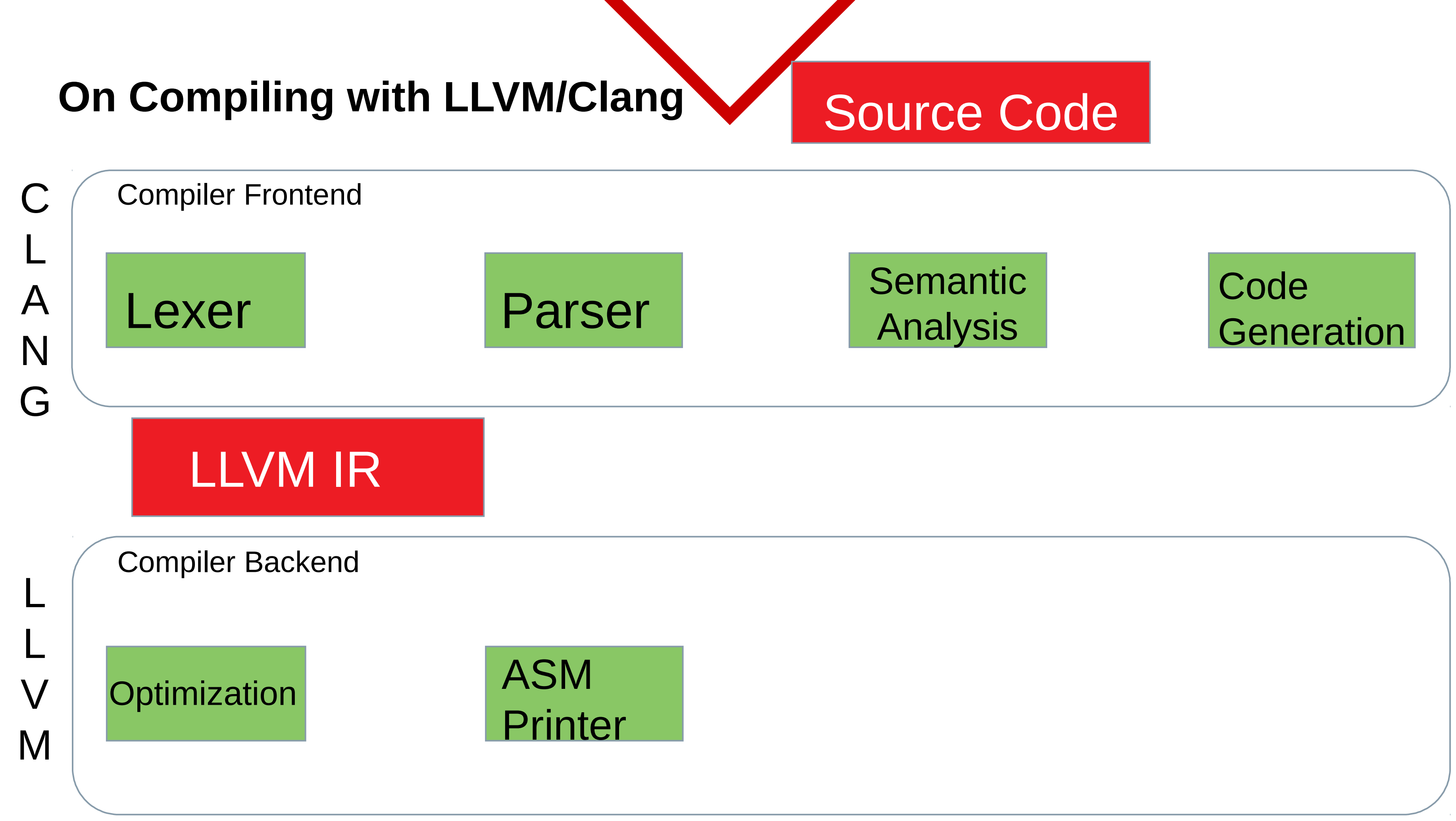


Optimizer

- Generates *better* code
  - What is better code?
    - Faster
    - Smaller
    - More secure
- Algorithms
  - Contradicting
  - Cooperating

- Different approaches in placing the Optimizer
  - Frontend
  - Backend
- try optimizing!
- *clang -S -emit-llvm file.c*
  - Optimizing for size: -Os
  - Optimizing for speed: -O1 -O2 -O3

On Compiling with LLVM/Clang



Compilers

Are they different?

```
$ diff clang/5.0.1/include/sanitizer/asan_interface.h \ gcc/x86_64-pc-linux-gnu/7.3.0/include/sanitizer/asan_interface.h

3,4d2

< //                     The LLVM Compiler Infrastructure

< //
```

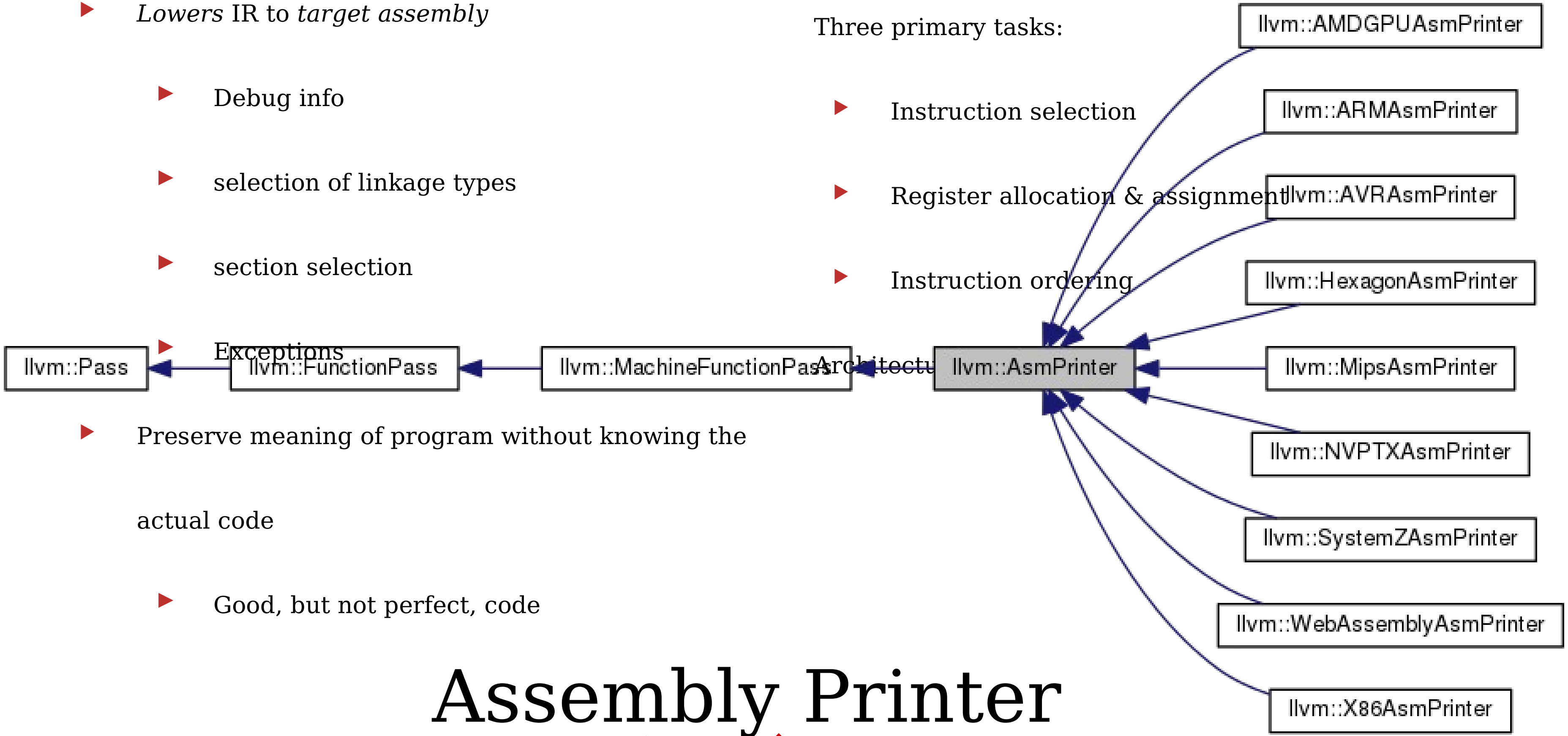
Assembly Printer

- *Lowers* IR to *target assembly*
  - Debug info
  - selection of linkage types
  - section selection
  - Exceptions
- Preserve meaning of program without knowing the actual code
  - Good, but not perfect, code

Three primary tasks:

- Instruction selection
- Register allocation & assignment
- Instruction ordering

Architectu

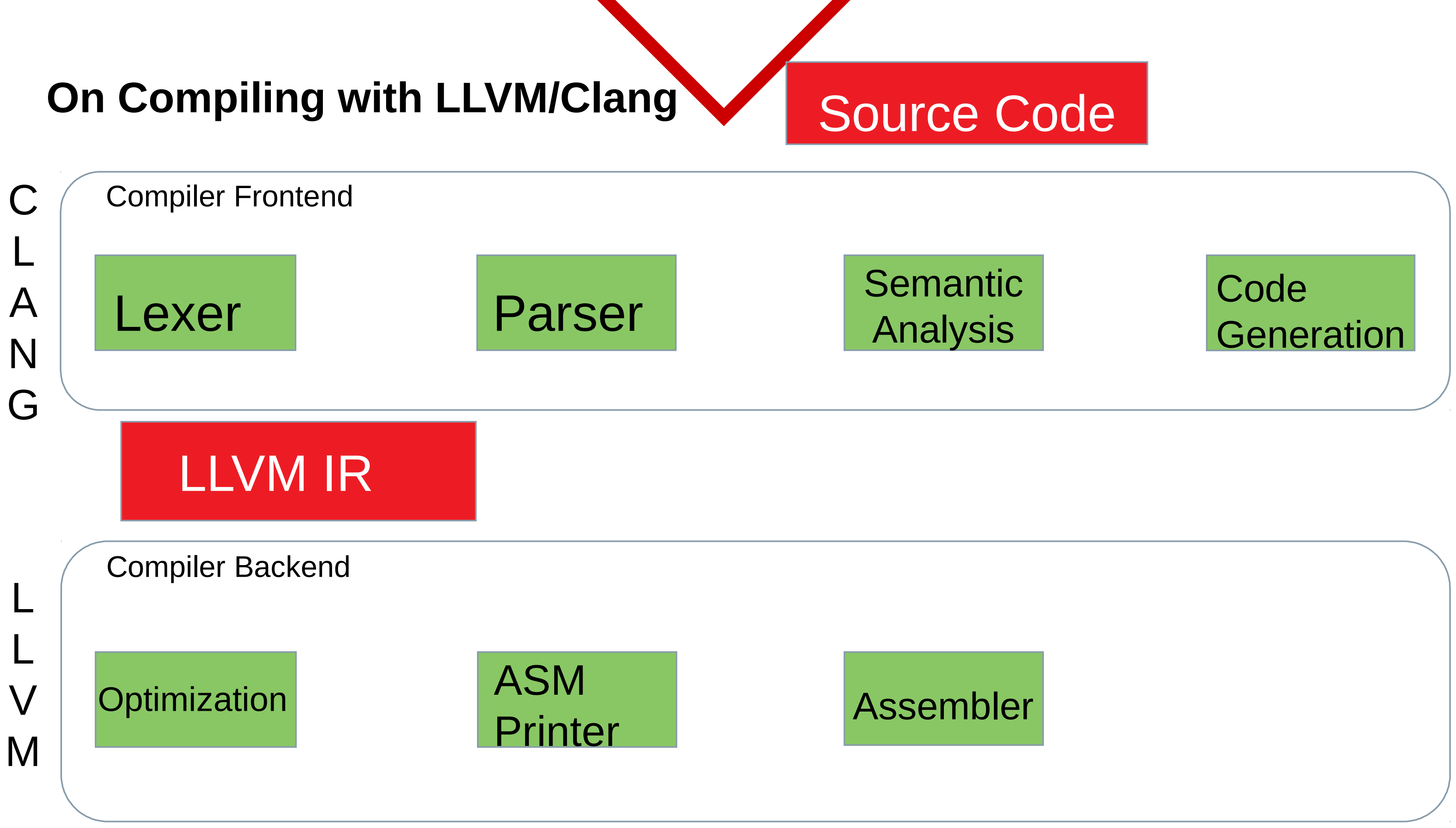


Inheritance diagram for llvm::AsmPrinter, taken from
http://llvm.org/doxygen/classllvm_1_1AsmPrinter.html

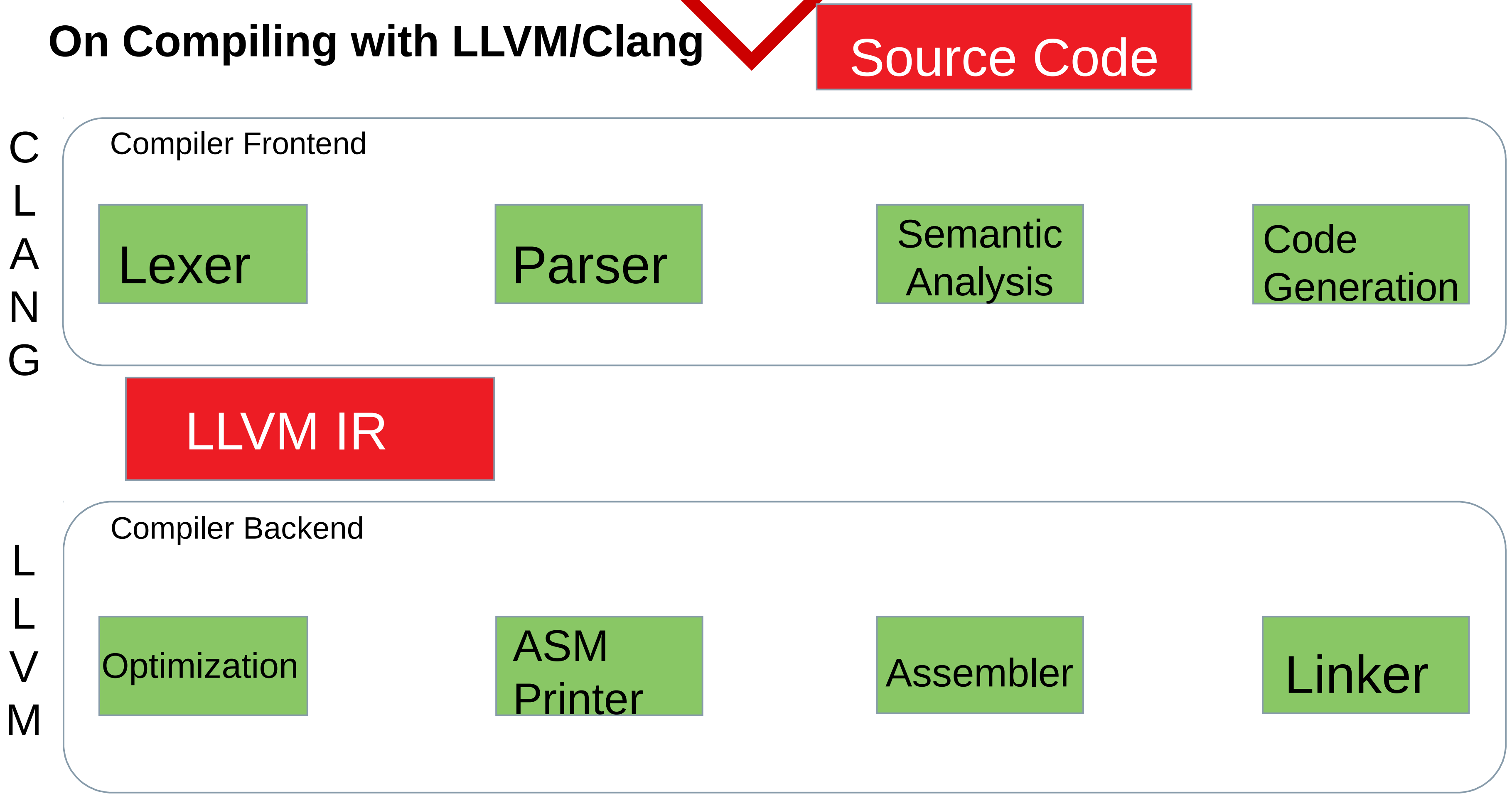
Target Architectures

RISC (ARM)	**CISC (INTEL)**
Many Registers	Few Register, different types
Few Instructions	Many Instructions
Low-level addressing structure	High-level instructions
Three-Address adressing	Different Addressing types
Big and little Endian	Little endian

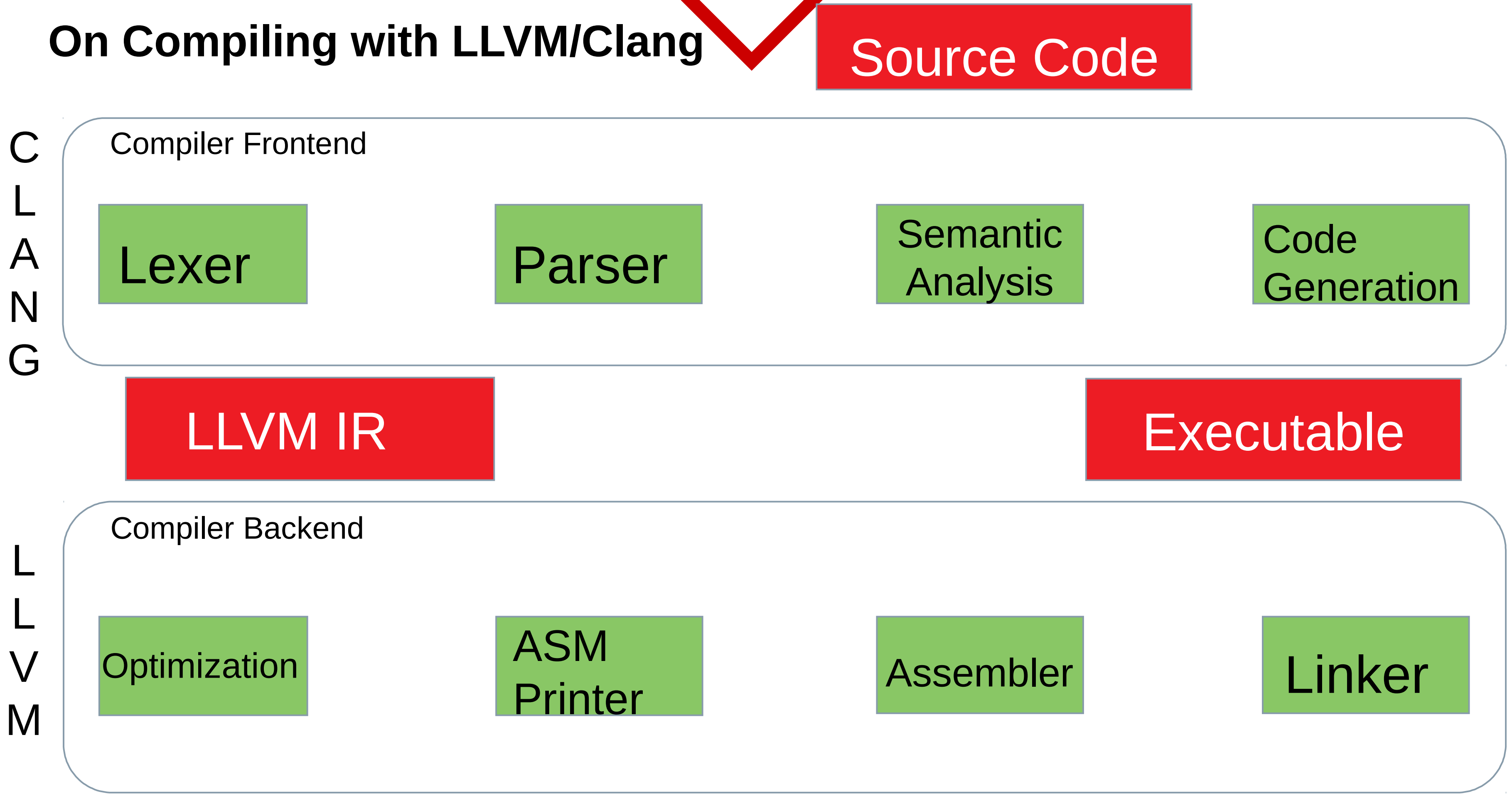
On Compiling with LLVM/Clang



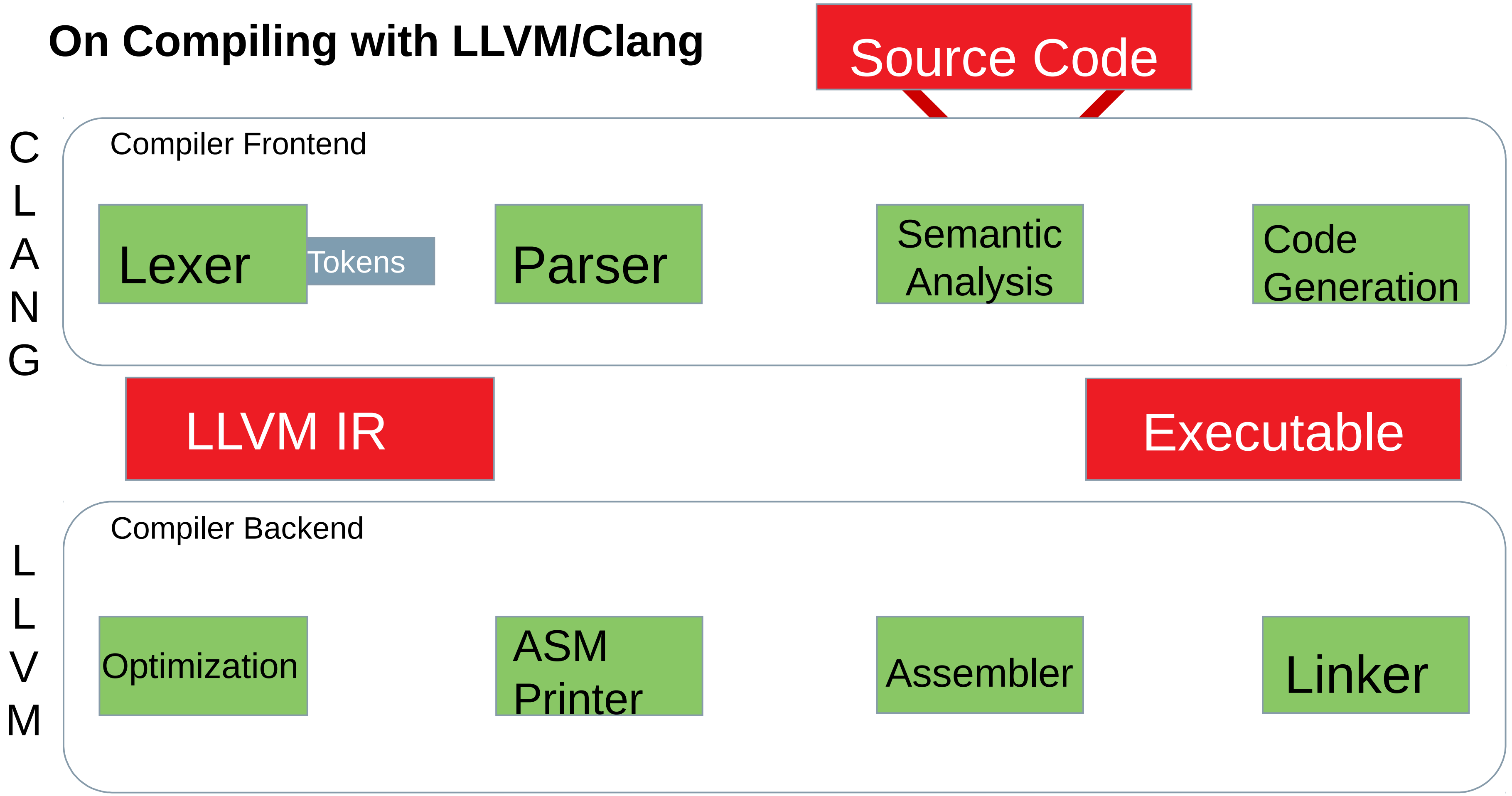
On Compiling with LLVM/Clang



On Compiling with LLVM/Clang



On Compiling with LLVM/Clang



Optimizing C

What can I optimize?

- Functions
- Memory operations, load and store
- Loops

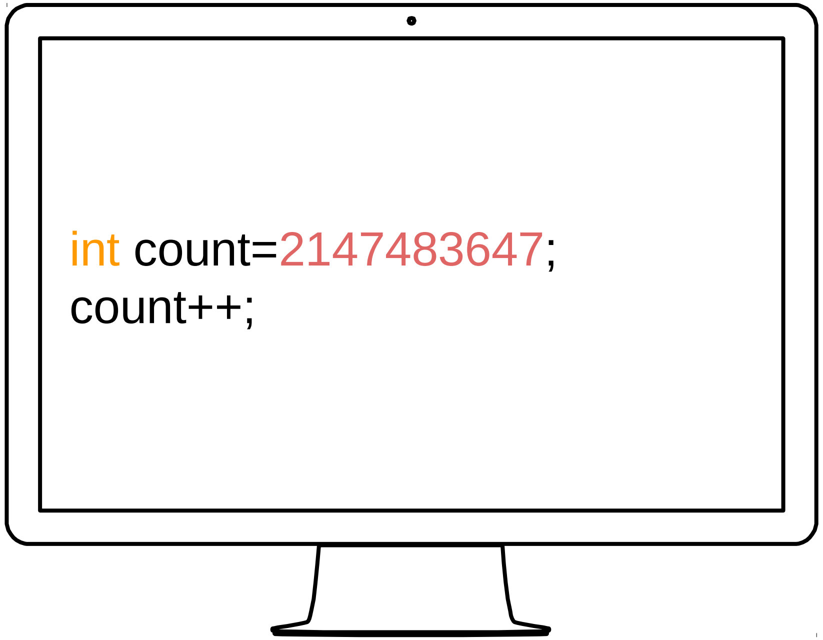
What can optimizations do?

Replace insecure library functions with secure ones:

atoi() --> strtol()

Delete function calls

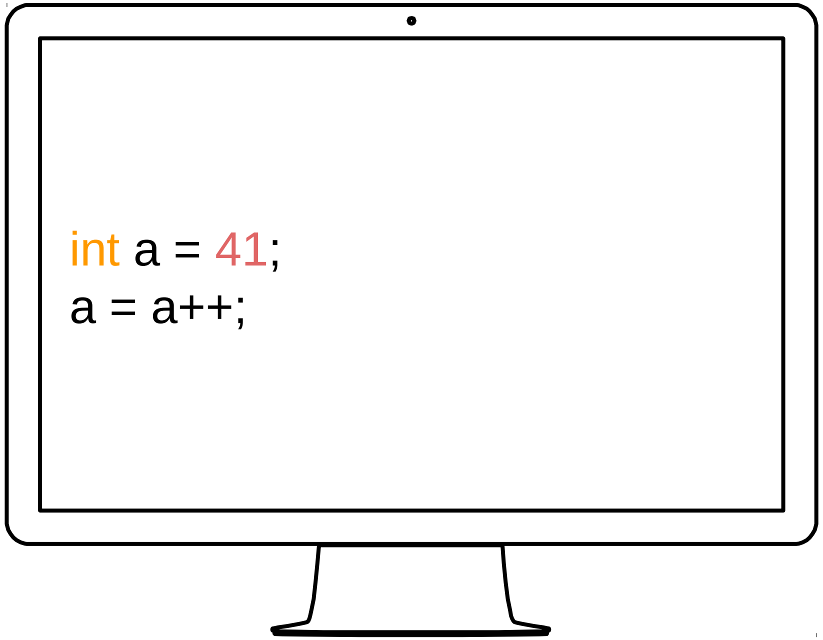
Addition



```
int count=2147483647;
count++;
```

What is going to happen?

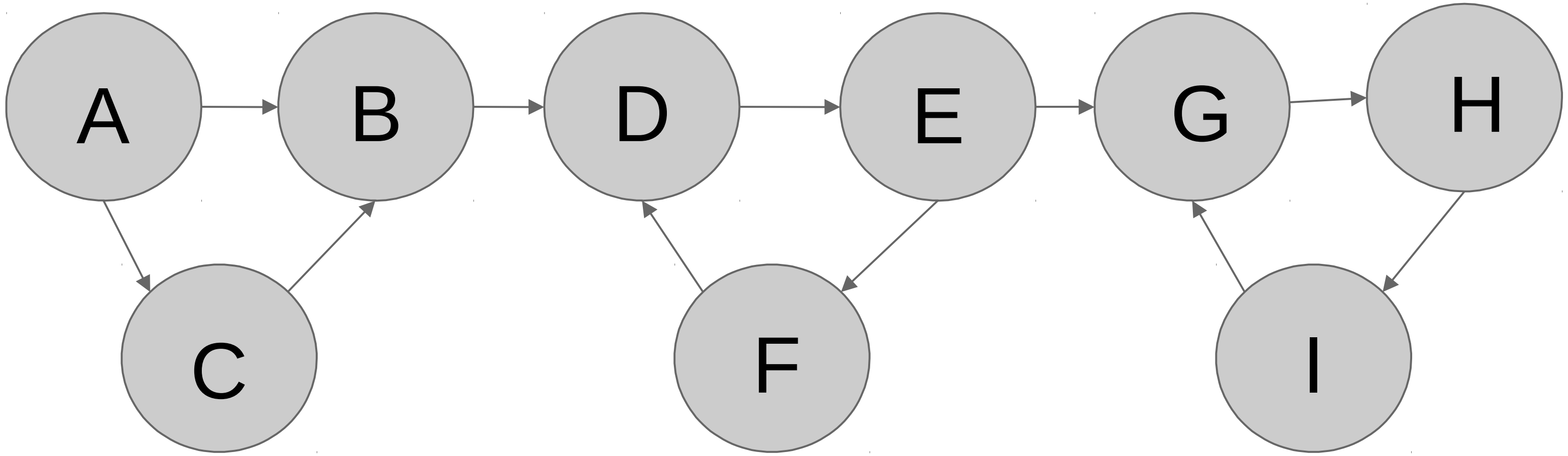
More Addition



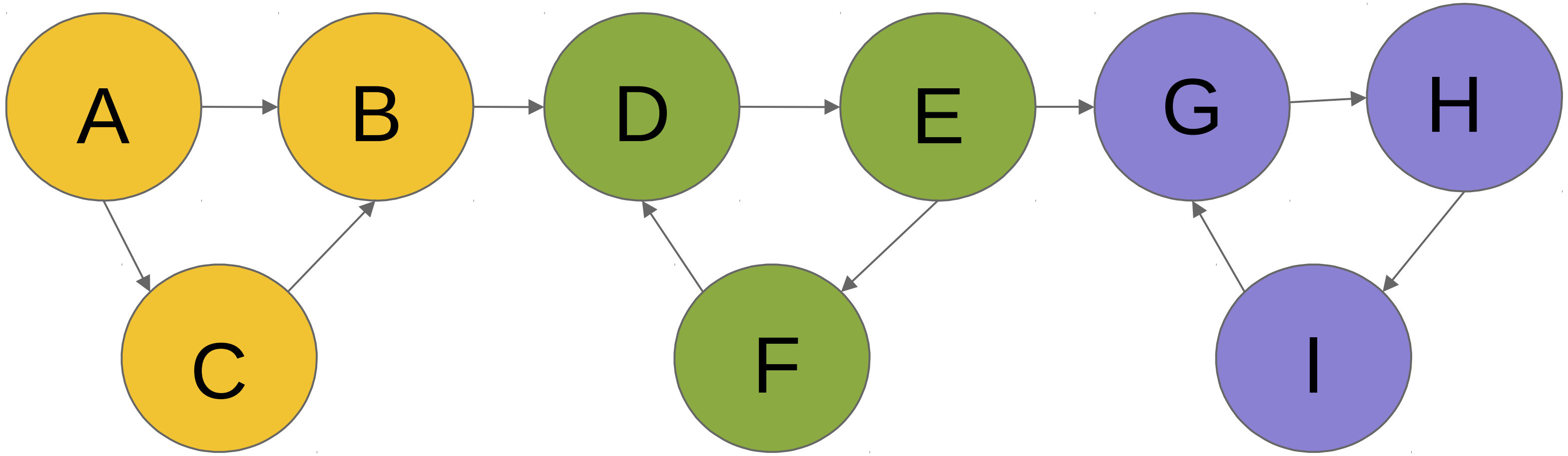
```
int a = 41;
a = a++;
```

What is going to happen?

How do I optimize?



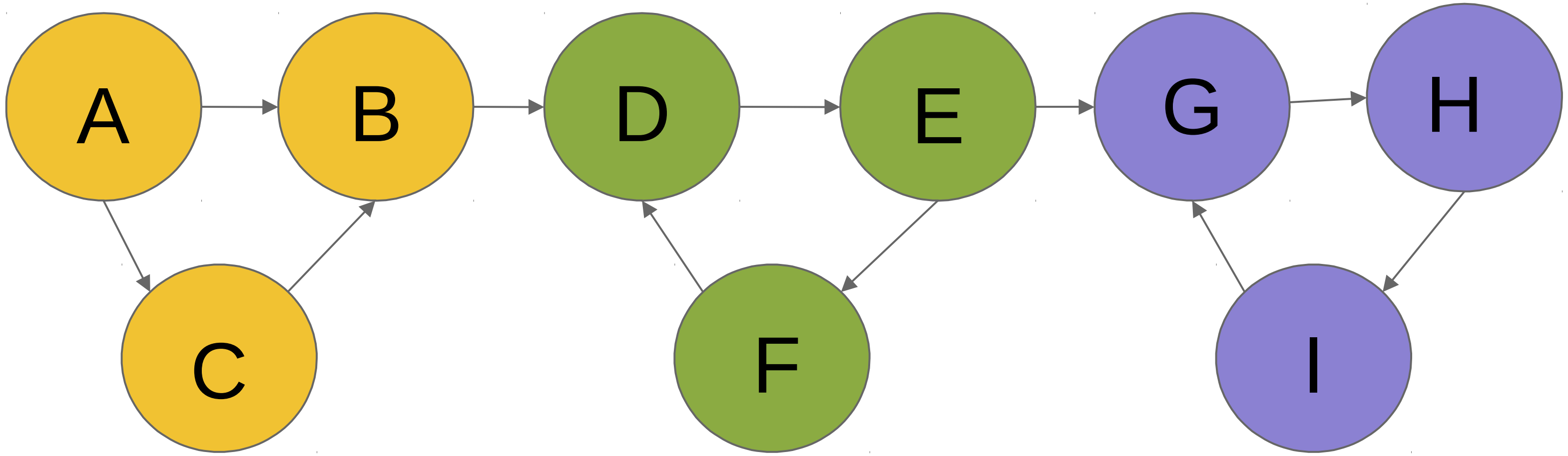
How do I optimize?



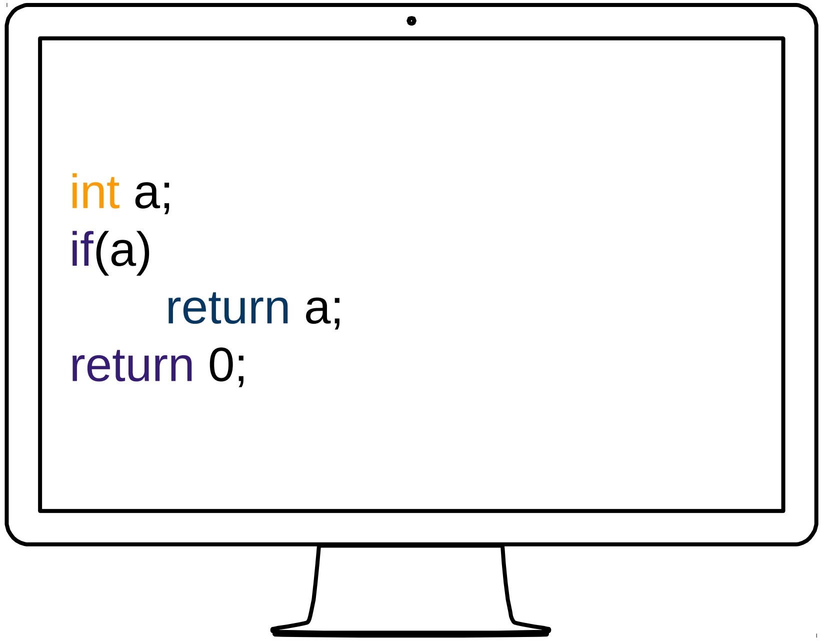
How do I optimize?

First the caller, then the callee!

Why?



Initialisation

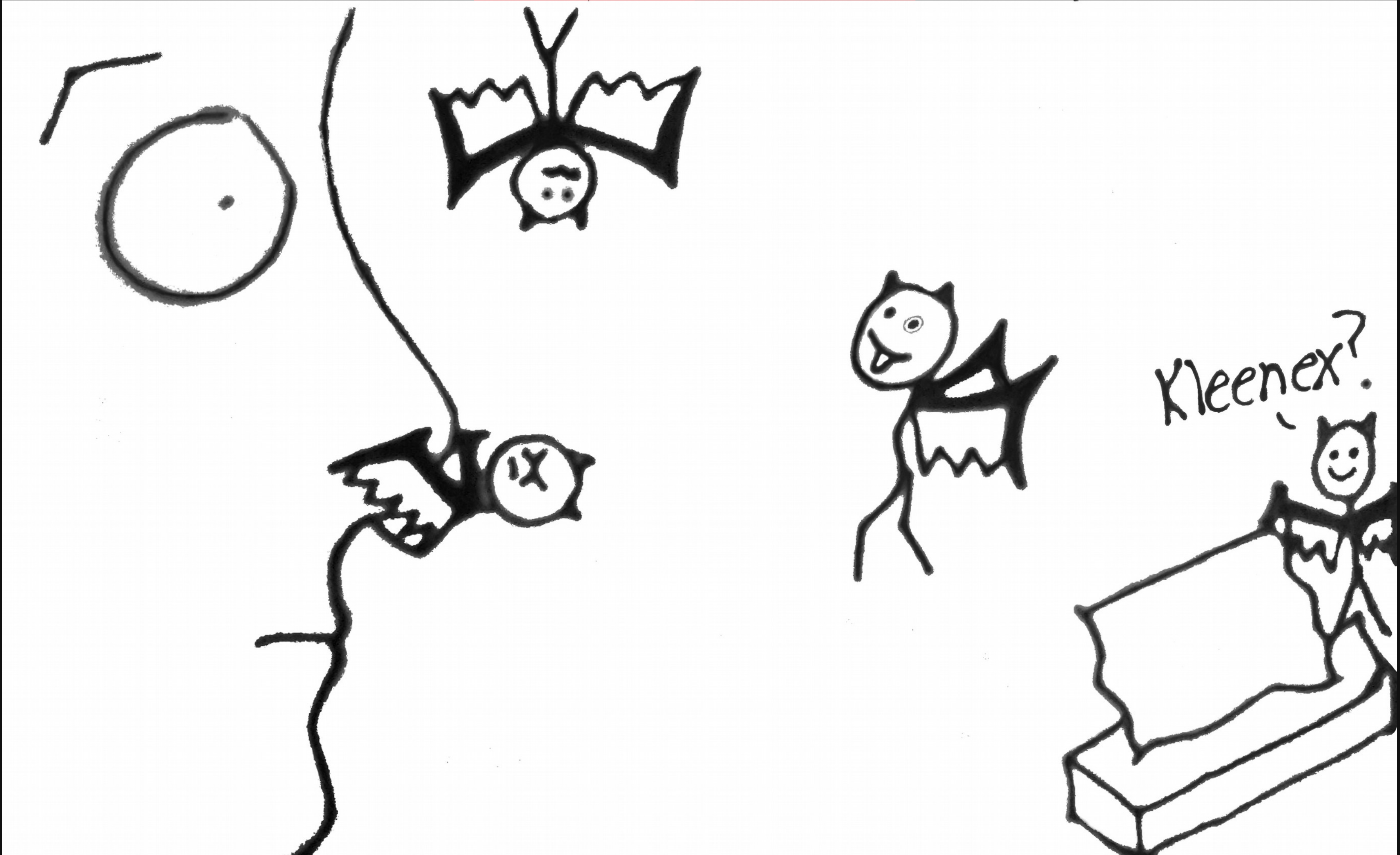


```
int a;
if(a)
        return a;
return 0;
```

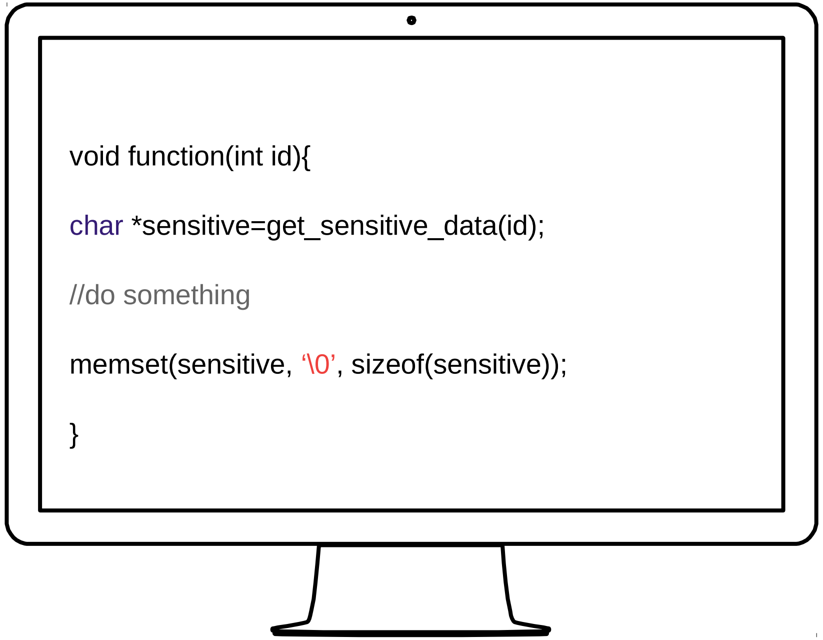
What is going to happen?

S

TERRA INCOGNITA
NVEVA FRANZA
OCEANO OCCIDENTALE
CANARIE
TROPICO DE CANCRO
MEDITERRANEO
MARE DE INDIA
MERI MACADAZO
QVITO
BRASIL PRO
ATACAMA
OCEANO ORIENT
Here Be Dragons


Kleenex?

Memset



```
void function(int id){

char *sensitive=get_sensitive_data(id);

//do something

memset(sensitive, '\0', sizeof(sensitive));

}
```

What is going to happen?

Thanks!

Any questions?

You can reach me at:
s1510239012@students.fh-hagenberg.at
lena.heimberger@gmail.com
Please send me encrypted E-mails
GPG fingerprint: 390D F1A5 06ED 3FBA EE44 74E7 0DF6 79E9
29BC B4EA