



# Patents and Copyright Compilation Pipeline

CS 2130: Computer Systems and Organization 1  
October 22, 2025

# Announcements

- Homework 6 **due Monday at 11:59pm**

# Patents and Copyright

Remember our Toy ISA. Can we patent our ISA? Should we?

| icode | b | meaning                                                                                                    |
|-------|---|------------------------------------------------------------------------------------------------------------|
| 0     |   | $rA = rB$                                                                                                  |
| 1     |   | $rA \&= rB$                                                                                                |
| 2     |   | $rA += rB$                                                                                                 |
| 3     | 0 | $rA = \sim rA$                                                                                             |
|       | 1 | $rA = !rA$                                                                                                 |
|       | 2 | $rA = -rA$                                                                                                 |
|       | 3 | $rA = pc$                                                                                                  |
| 4     |   | $rA = \text{read from memory at address } rB$                                                              |
| 5     |   | write $rA$ to memory at address $rB$                                                                       |
| 6     | 0 | $rA = \text{read from memory at } pc + 1$                                                                  |
|       | 1 | $rA \&= \text{read from memory at } pc + 1$                                                                |
|       | 2 | $rA += \text{read from memory at } pc + 1$                                                                 |
|       | 3 | $rA = \text{read from memory at the address stored at } pc + 1$                                            |
|       |   | For icode 6, increase $pc$ by 2 at end of instruction                                                      |
| 7     |   | Compare $rA$ as 8-bit 2's-complement to 0<br>if $rA \leq 0$ set $pc = rB$<br>else increment $pc$ as normal |

# Patents and Copyright

## Copyright

- “Everyone is a copyright owner. Once you create an original work and fix it, like taking a photograph, writing a poem or blog, or recording a new song, you are the author and the owner.”  
from <https://www.copyright.gov/what-is-copyright/>

# Patents and Copyright

## Patent

- “Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.”  
from 35 U.S.C. 101

# Patents

In software and hardware, patents become messy

- Code is a description of a process we want the computer to do
- Do not have to implement the process to patent it

Question: Should we patent something like our ISA?

# Patents

In software and hardware, patents become messy

- Code is a description of a process we want the computer to do
- Do not have to implement the process to patent it

Question: Should we patent something like our ISA?

What is the current state of the art?

# Common Approaches to Software

How can we get value from what we create?

- Copyright - distribute closed source software
- License Agreements (in contract law)
- Always innovate



# Compilation Pipeline

Turning our code into something that runs

- **Pipeline** - a sequence of steps in which each builds off the last

# Why did we discuss assembly?

# C

C is a thin wrapper around assembly

- This is by design!
- Invented to write an operating system
  - Can write inline assembly in C
- Many other languages decided to look like C

# Simple C Example

```
int main() {  
    int y = 5;  
    return 0;  
}
```

# Compilation Pipeline

Earlier, we saw:

- C files (.c) compiled to assembly (.s)
- Assembly (.s) assembled into object files (.o)
- Object files (.o) linked into a program / executable

# Compiling C to Assembly

Multiple stages to compile C to assembly

- Preprocess - produces C
  - C is actually implemented as 2 languages:  
C preprocessor language, C language
  - Removes comments, handles preprocessor directives (#)
  - `#include`, `#define`, `#if`, `#else`, ...
- Lex - breaks input into individual tokens
- Parse - assembles tokens into intended meaning (parse tree)
- Type check - ensures types match, adds casting as needed
- Code generation - creates assembly from parse tree

# Compiling C to Assembly

# Compiling C to Assembly



# Errors

## Compile-time errors

- Errors we can catch during compilation (this process)
- **Before** running our program

## Runtime errors

- Errors that occur when running our programs

# Simple C Example

```
int main() {  
    return 0;  
}
```

The `main` function

- Start running the `main()` function
- `main` must return an integer - **exit code**
  - 0 = everything went okay
  - Anything else = something went wrong
- There *should* be arguments to `main`