

Copyrights and Patents Compilation Pipeline

C!

CS 2130: Computer Systems and Organization 1

March 18, 2026

Announcements

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

An aside...

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		$rA =$ read from memory at address rB
5		write rA to memory at address rB
6	0	$rA =$ read from memory at $pc + 1$
	1	$rA \&=$ read from memory at $pc + 1$
	2	$rA +=$ read from memory at $pc + 1$
	3	$rA =$ 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 if $rA \leq 0$ set $pc = rB$ 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

Going back...

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

In Monday's video, 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

2.3

- 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

x.y

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`

Example

Data Types in C

Integer data types

Data type	Size
char	
short	
int	
long	
long long	

Each has 2 versions: *signed* and *unsigned*

Data Types in C

Floating point

- float
- double

Data Types in C

Data Types in C

Pointers - how C uses addresses!

Data Types in C

Pointers - how C uses addresses!

- Hold the address of a position in memory
- Need to know the kind of information stored at that location

Example

```
int main() {  
    int x = 3;  
    long y = 4;  
    int *a = &x;  
    long *b = &y;  
    long z = *a;  
    int w = *b;  
    return 0;  
}
```

Example

```
int main() {  
    int x = 3;  
    long y = 4;  
    int *a = &x;  
    long *b = &y;  
    long z = *a;  
    int w = *b;  
    return 0;  
}
```

```
0000000000000000 <main>:  
0: 55 push %rbp  
1: 48 89 e5 mov %rsp,%rbp  
4: 31 c0 xor %eax,%eax  
6: c7 45 fc 00 00 00 00 movl $0x0,-0x4(%rbp)  
d: c7 45 f8 03 00 00 00 movl $0x3,-0x8(%rbp)  
14: 48 c7 45 f0 04 00 00 movq $0x4,-0x10(%rbp)  
1b: 00  
1c: 48 8d 4d f8 lea -0x8(%rbp),%rcx  
20: 48 89 4d e8 mov %rcx,-0x18(%rbp)  
24: 48 8d 4d f0 lea -0x10(%rbp),%rcx  
28: 48 89 4d e0 mov %rcx,-0x20(%rbp)  
2c: 48 8b 4d e8 mov -0x18(%rbp),%rcx  
30: 48 63 09 movslq (%rcx),%rcx  
33: 48 89 4d d8 mov %rcx,-0x28(%rbp)  
37: 48 8b 4d e0 mov -0x20(%rbp),%rcx  
3b: 48 8b 09 mov (%rcx),%rcx  
3e: 89 4d d4 mov %ecx,-0x2c(%rbp)  
41: 5d pop %rbp  
42: c3 retq
```