C Introduction

CS 2130: Computer Systems and Organization 1
March 20, 2026

Announcements

- Homework 6 **due Monday at 11:59pm**
- Midterm 2 is 2 weeks away (April 3)
  - Similar format to midterm 1
  - Schedule with SDAC early if needed

C

C is a thin wrapper around assembly

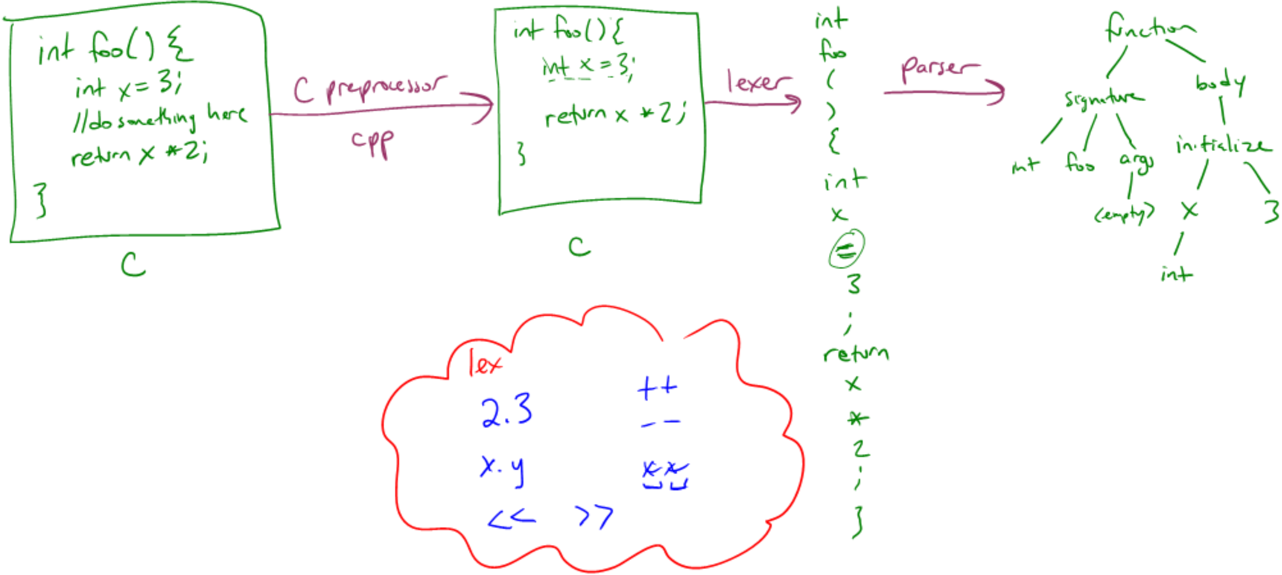
- This is by design!
- Invented to write an operating system
  - Can write inline assembly in C
- Many other languages decided to look like C

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

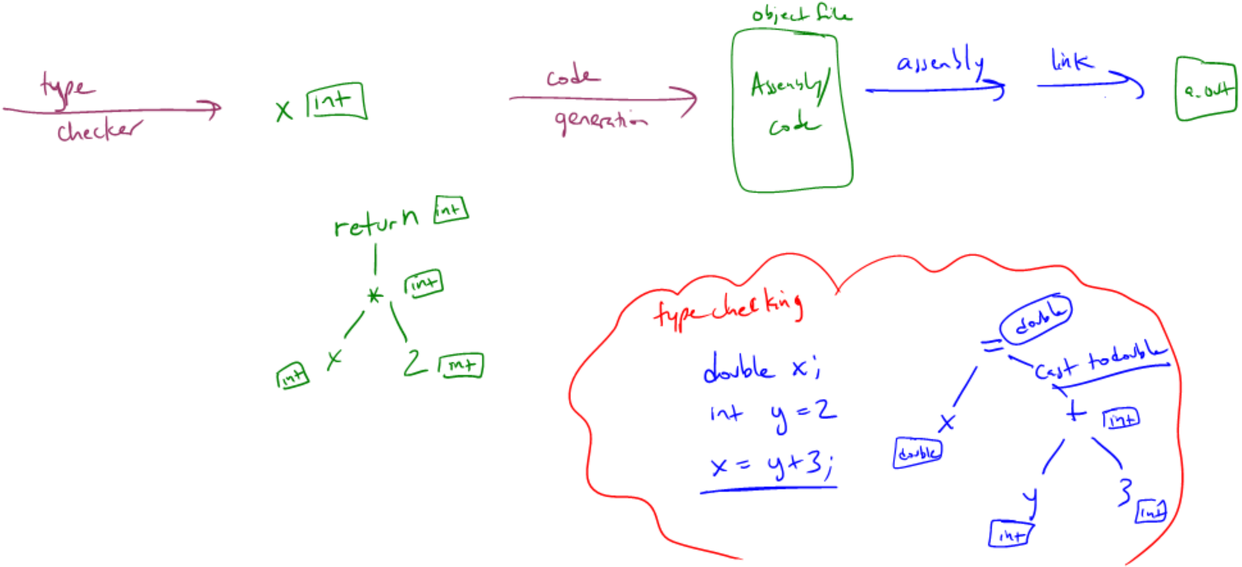


```
int foo() {
    int x=3;
    //do something here
    return x *2;
}
```

```
int foo() {
    int x = 3;

    return x * 2;
}
```

Compiling C to Assembly



```
double x;
int y=2
x = y+3;
```

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