

# CS480

## Compilers

### Chapter 1, 2 (and section 7.6)

Pages 1-51, 60 - 62, 429 - 440

February 9, 2009

# Compilers

- Definition (again)
- Difference between compiler and
  - Eclipse
  - MS Visual Studio
  - Parts of each?
- Other source code analysis tools?

# Two Jobs

- Analysis
- Synthesis/Generation

# Compiler, in detail

- Lexical Analysis `taxes = salary * 0.30 + 500;`
- Syntactic Analysis
  - AST
- Semantic Analysis

# Compiler, in detail

- Intermediate code generator

```
taxes = salary * 0.30 + 500;
```

- Code generation
  - Optimization

- What's the difference between assembly lang and machine lang?

# Example?

What does the compiler need to do here?

```
1#include <stdio.h>
2#if SYS == RH5.1
3#define LIB "rh51.h"
4#elif SYS == RH6.0
5#define LIB "rh60.h"
6#else
7#define LIB "rh62.h"
8#endif
9#include LIB
10
11#define MAX(x,y) ((x) > (y) ? (x) : (y))
12
13main ()
14{
15    int i = 3, j = 7;
16    printf ("%d", MAX(i,j));
17}
```

# Compiler Terms

- Front End/Back End
  
  
  
  
  
  
  
  
  
  
- Passes

# Linker/Loader

- What's a link error?



# Ch 2: One-Pass Compiler

- Convert infix to postfix (RPN) expressions
  - why is RPN useful?
- Infix:  $5 + 2 / 4 + 1$
- Postfix:

# Language

- Context Free Grammar
  - components?
- Can you give a grammar for postfix notation?
  - Assume single digit ints
  
- Parse Tree?
  - Components?
  - Parse tree for  $1\ 2\ +\ 2\ -$

# Grammars

**bin**  $\rightarrow$  **bin** + **bin**

**bin**  $\rightarrow$  **bin** - **bin**

**bin**  $\rightarrow$  0 | 1

- Ambiguous?
- Associativity?
  - Is + left or right associative?

# Grammars

`expr -> expr + term | expr - term | term`

`term -> term * factor | term / factor | factor`

`factor -> digit | ( expr )`

- Ambiguous?
- Left or Right associative

+          -          \*          /

- **9 + 2 \* 5**

# Syntax Directed Translation

- Grammar for infix
- Grammar of postfix
- ???
- Translation

# Annotated Parse Trees

- Semantic rules
- Synthesized Attributes
- Traversal
- Emit Translation

# Parsing!

- Top Down
  
  
  
  
  
  
  
  
  
  
- Bottom Up

# Symbol Tables (p 60-62, 429-440)

- Lexeme
- Runtime location
  - Where could a variable be stored at runtime?
- Type
- Level/Scope
- Array dimension/Number of parameters



# Symbol Table

- Cross Reference Information
- Data structures
  - Unordered linear list
  - Ordered linear list
  - Binary search tree
  - AVL tree
  
  - advantages/disadvantages
  - Insert/Delete/Search  $O(??)$