CS354: Compiler Construction

Introduction

Madnia ashraf
Madniaashraf178@gmail.com

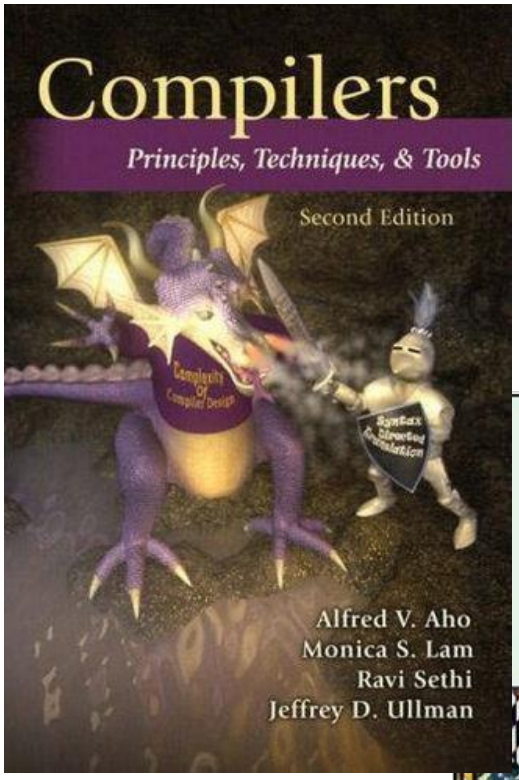
- Consult LMS for:
  - Lecture Notes, Quizzes, Assignments, Labs & Term Project

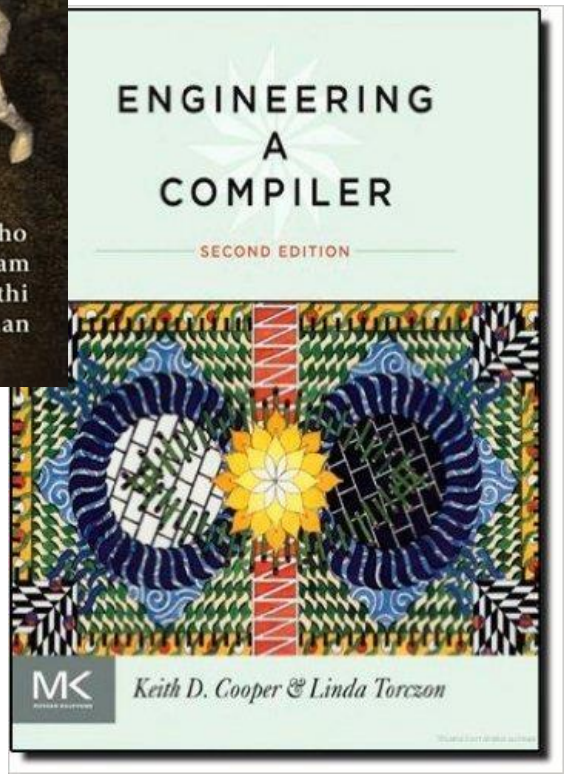
Attendance Policy

- *75%* required at the end of semester!
- Attendance will be taken at any moment
- No compensation for attendance! No Bonuses!
- Late Arrivals will be dealt according to situation:

Text Books

- Alfred V. Aho et al., **C**ompilers: **P**rinciples, **T**echniques and **T**ools, 2/E, Pearson, 2007 also known as the **"Dragon Book"**
- Keith D. Cooper and Linda Torczon, **E**ngineering **A** **C**ompiler, 2/E, Morgan Kaufmann, 2012





Assessment Plan (Tentative)

- Theory
  - 35% Final Exam (Closed Books, Closed Notes, Comprehensive)
  - 25% OHTs (Closed Books, Closed Notes, Comprehensive)
  - 10% Quizzes (*Surprise!*)
  - 5% Assignments
  - 5% Term Project

Homework Policy

- All homework assignments must be done *individually* or as directed
- Hardcopy to be submitted in class on the due date
  - May also require uploading a soft-copy on LMS (for *plagiarism* check)
- Cheating
  - Helping others, getting help, looking up websites for solutions etc.
- Late Submissions
  - Late submission will get a *–10%* penalty per late day.
    - For example: if your assignment is 02 days late, you will get 80% of your earned marks in that assignment.
  - Extensions may be permitted under extraordinary circumstances
    - Contact the instructor at least *01 week* before the deadline
- Any deviation from the above rule will be considered cheating and will be subject to the SEECS academic *dishonesty* policy http://seecs.nust.edu.pk/Internal/downloads/downloads/SEECS_Plagiarism_Policy_Dec2010_v1.0.pdf

Lecture Basics

"I hear and I forget. I see and I remember. I do and I understand." – Confucius

- Classes will involve both *Slides + Board* (to roughly equal degrees)
  - Lectures will be available on LMS
  - However, no scribes from the class will be made available
  - So, take your own notes in the class
- For latest/updated slides, download before each use
  - As I might update/correct slides at any stage

Why take this course?

Reason #1: Understand compilers and languages

- Understand the code structure
- Understand language semantics
- Understand relation between source code and generated machine code
- Become a better programmer

Why take this course?

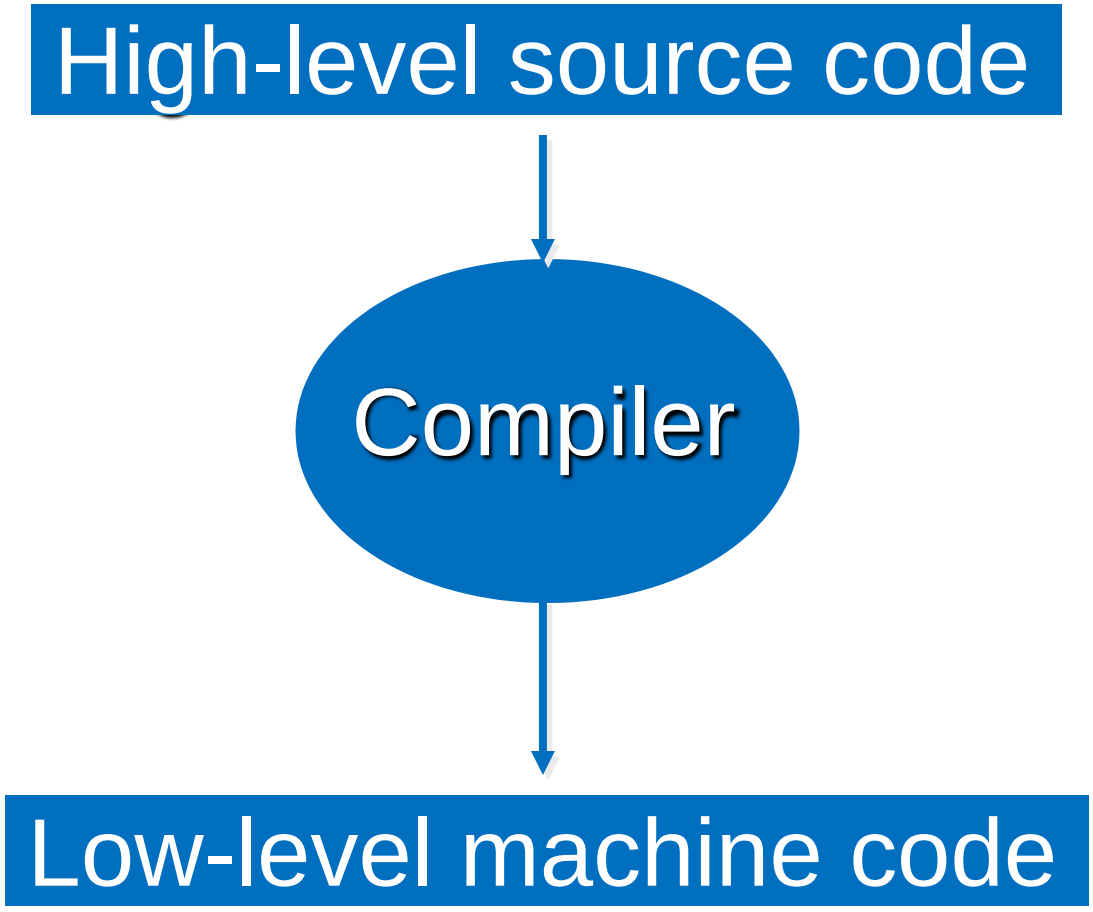
Reason #2: Nice balance of theory and practice

- Theory
  - mathematical models: regular expressions, automata, grammars, graphs
  - algorithms that use these models
- Practice
  - Apply theoretical notions to build a real compiler and/or its components

What are Compilers?

- Translate information from one representation to another; usually *information = program*
- A program that reads a program written in one language *(source language)* and translates it into an equivalent program in another language *(target language)* -- Aho et al
- Compilers are also known as *Language Processors*

Typical Compilation



High-Level Source Code

```
int expr( int n )
{
 int d;
 d = 4*n*n*(n+1)*(n+1);
 return d;
}
```

- Optimized for human readability
- Matches human notions of grammar
- Uses named constructs such as variables and procedures

Low-Level Assembly Code

```
.globl _expr
_expr:
 pushl %ebp
 movl %esp,%ebp
 subl $24,%esp
 movl 8(%ebp),%eax
 movl %eax,%edx
 leal 0(,%edx,4),%eax
 movl %eax,%edx
 imull 8(%ebp),%edx
 movl 8(%ebp),%eax
 incl %eax
 imull %eax,%edx
 movl 8(%ebp),%eax
 incl %eax
 imull %eax,%edx
 movl %edx,-4(%ebp)
 movl -4(%ebp),%edx
 movl %edx,%eax
 jmp L2
 .align 4
L2:
 leave
 ret
```

Low-Level Assembly Code

Optimized for hardware

- Consists of machine instructions
- Uses registers and unnamed memory locations
- Much harder to understand by humans

How to Translate ?

Correctness:

the generated machine code must execute precisely the same computation as the source code

- Is there a unique translation? *No!*
- Is there an algorithm for an "ideal translation"? *No!*

How to Translate ?

- Translation is a *complex process*
- Source language and generated code are *very different*
- Need to *structure* the translation process
  - More about structure of compilers: Later!