

Compiler Design

Amey Karkare
Department of Computer Science and Engineering
IIT Kanpur
karkare@iitk.ac.in
<http://www.cse.iitk.ac.in/~karkare/Courses/cs335/>

2021 Compilers 1

1

Acknowledgements

- Most of the text in the slide is based on classic text **Compilers: Principles, Techniques, and Tools** by **Aho, Sethi, Ullman** and **Lam**
- Slides are modified version of those created by Prof S K Aggarwal, IITK

2021 Compilers 2

2

Motivation

- Language processing is an important component of programming
- Many systems software and application programs require structured input
 - Operating Systems (command line processing)
 - Databases (Query language processing)
 - Type setting systems like Latex
 - Software quality assurance and software testing

2021 Compilers 3

3

Motivation

- Wherever input has a structure one can think of language processing
- Why study compilers?
 - Compilers use the whole spectrum of language processing technology

2021 Compilers 4

4

Expectations?

- What will we learn in the course?



2021

Compilers

5

5

What do we expect to achieve by the end of the course?

- Knowledge to design, develop, understand, modify/enhance, and maintain compilers for (even complex!) programming languages
- Confidence to use language processing technology for software development
- Become a better programmer and better software developer

2021

Compilers

6

6

Organization of the course

- Assignments 15%
- Mid semester exam 20%
- End semester exam 30%
- Course Project 35%
 - Group of 4/5 (to be decided)

2021

Compilers

7

7

Bit of History

- How are programming languages implemented?
Two major strategies:
 - Interpreters (Less studied)
 - Compilers (very well understood with mathematical foundations)
- Some environments provide both interpreter and compiler. Lisp, scheme etc. provide
 - Interpreter for development
 - Compiler for deployment
- Java
 - Java compiler: Java to interpretable bytecode
 - Java JIT: bytecode to executable image

2021

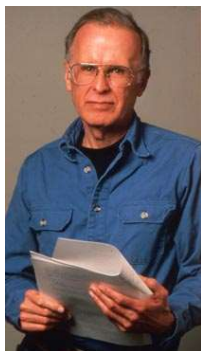
Compilers

8

8

Some early machines and implementations

- IBM developed 704 in 1954. All programming was done in assembly language. Cost of software development far exceeded cost of hardware. Low productivity.
- Speedcoding interpreter: programs ran about 10 times slower than handwritten assembly code
- John Backus (in 1954): Proposed a program that translated high level expressions into native machine code. Skepticism all around. Most people thought it was impossible
- Fortran I project (1954-1957): The first compiler was released



2021

Compilers

9

9

Fortran I

- The first compiler had a huge impact on the programming languages and computer science. The whole new field of compiler design was started
- More than half the programmers were using Fortran by 1958
- The development time was cut down to half
- Led to enormous amount of theoretical work (lexical analysis, parsing, optimization, structured programming, code generation, error recovery etc.)
- Modern compilers preserve the basic structure of the Fortran I compiler !!!

2021

Compilers

10

10

The big picture

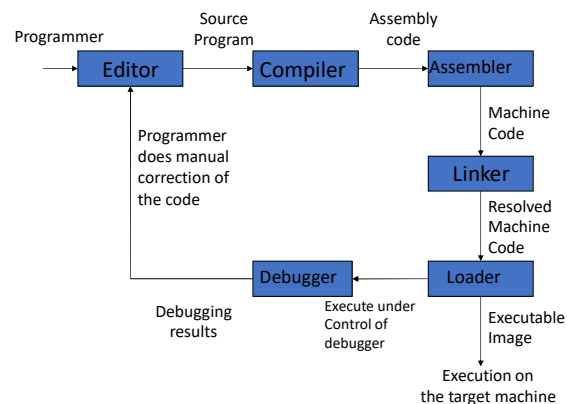
- Compiler is part of program development environment
- The other typical components of this environment are editor, assembler, linker, loader, debugger, profiler etc.
- The compiler (and all other tools) must support each other for easy program development

2021

Compilers

11

11



2021

Compilers

19

19

Normally end up with error