

Module 01

Pralay Mitra & I

Objectives Outline

Phases of a Compiler

C Compilation

Front-end

Lexical Analys

Syntax Analysis

Computie Analysis

Intermediate Co

Generator

Code Optimization

Code Optimization

Sample Translation

Summar

Module 01: CS31003: Compilers

Overview: Phases of a Compiler

Pralay Mitra Partha Pratim Das

Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur

pralay@cse.iitkgp.ac.in ppd@cse.iitkgp.ac.in

August 10, 2021



Course Outline

Module 01

Pralay Mitra & P Das

Objectives & Outline

Phases of Compiler

C Compilation Front-end Lexical Analy

Syntax Analysis
Semantic Analysis
Intermediate Code
Generator
Code Optimization
Back-end
Code Optimization
Target Code

Sample Franslatior

Summary

- Outline of Principles
- Outline of Implementation
- Books:
 - Compilers: Principles, Techniques, and Tools (2nd Edition) by A.V. Aho, Monica S Lam, R. Sethi, Jeffrey D. Ullman (Pearson / Addison-Wesley)
 - Flex and Bison by John Levine (O'Reilly)
 - o Compiler Design in C by Allen Holub
 - o Advanced Compiler Design and Implementation by Steven Muchnick
- Quora: Is writing a compiler very easy?

It's something we teach undergrad CS majors to do in a semester. It might be the most code they've ever written that quickly (I recall mine took 10k lines in just five weeks) but most of them succeed. It might well be the most challenging program you write as an undergrad, but you should manage to go on to more difficult projects once you start your career. - Barry Rountree, Computer Scientist, LLNL: https://www.quora.com/profile/Barry-Rountree

https://www.quora.com/Is-writing-a-compiler-very-easy



Module Objectives

Module 01

Pralay Mitra & F P Das

Objectives & Outline

Phases of

C Compila

Front-end

r-ront-end

Syntax Analysis

Semantic Analysis

Introduction Cod

Generator

Code Optimization

Code Optimizatio Target Code

Sample Translatio

Summa

- Understand an outline of the course
- Understand the phases of a compiler



Module Outline

Module 01

Pralay Mitra & P Das

Objectives & Outline

Phases of Compiler

Front-end
Lexical Analys

Syntax Analysis Semantic Analysis

Intermediate Code Generator Code Optimization

Back-end

Code Optimization

Target Code

Generation

Sample Translatior

Summary

Objectives & Outline

- Phases of a Compiler
 - Overview of Compilation Process
 - Compiler Front-end
 - Lexical Analysis
 - Syntax Analysis
 - Semantic Analysis
 - Intermediate Code Generator
 - Code Optimization
 - Compiler Back-end
 - Code Optimization
 - Target Code Generation
- Sample Translation
- Summary



Compiling a C Program

Module 01

Pralay Mitra & I P Das

Objectives Outline

Phases of a Compiler

C Compilation

C Compila

Lexical Analysis
Syntax Analysis
Semantic Analysis

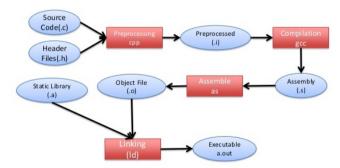
Intermediate Code Generator Code Optimization

Code Optimization
Target Code
Generation

Sample Translation

Summar

- C Pre-Processor (CPP)
- C Compiler
- Assembler
- Linker



Compilation Flow Diagrams for gcc



Compiling a C Program

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code

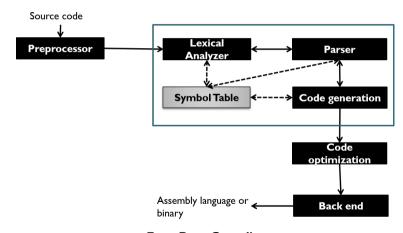
Generator

Code Optimiza

Code Optimization

Sample Translation

Summai



Four Pass Compiler



Phases

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

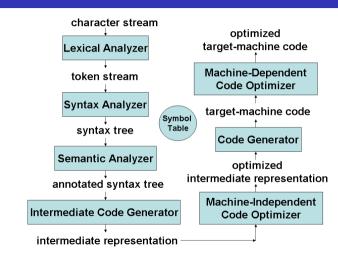
C Compilation

Lexical Analysis
Syntax Analysis
Semantic Analysis
Intermediate Code
Generator
Code Optimizatio

Code Optimization
Target Code

Sample Translation

Summai



Source: Y N Srikant (NPTEL)
Pralay Mitra & Partha Pratim Das

Compilers



Lexical Analysis Phase

Module 01

Pralay Mitra & I

Objectives of Outline

Phases of Compiler

C Compilati

Lexical Analysis

Control Annie

Syntax Analysis

Semantic Analys

Intermediate Co Generator

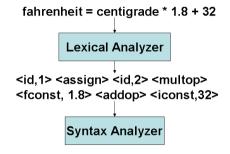
Code Optimization

Back-end
Code Optimization

Code Optimization
Target Code
Generation

Sample Translation

Summar



fahrenheit = centigrade * 1.8 + 32

 $total Amount \hspace{0.2in} = \hspace{0.2in} principal Amount * 10 + principal Amount$

 $\textit{finalVelocity} \hspace{0.2in} = \hspace{0.2in} \textit{acceleration} * \textit{time} + \textit{initialVelocity}$

Source: Y N Srikant (NPTEL)
Pralay Mitra & Partha Pratim Das



Lexical Analysis Phase

Module 01

Pralay Mitra & P Das

Objectives Outline

Compiler

C Compila

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Generator

Back-end
Code Optimization
Target Code

Sample Translatio

Summar

$$f = c * 1.8 + 32$$

$$b = a*10 + a$$

$$v = a * t + u$$

$$id = id * num + num$$

$$id = id * num + id$$

$$id = id * id + id$$

$$E = E * E + E$$

 $(E = ((E * E) + E))$



Syntax Analysis Phase

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

Front-end

Syntax Analysis

Syntax Analysis

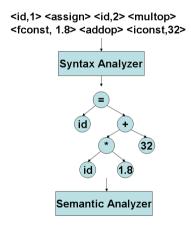
Intermediate Co

Code Optimization

Back-end
Code Optimization
Target Code

Sample Translatio

Summai



Source: Y N Srikant (NPTEL)

Compilers Pralay Mitra & Partha Pratim Das 01.10



Semantic Analysis Phase

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

Front-end

Syntax Analysis

Semantic Analysis

Schlancie / marysis

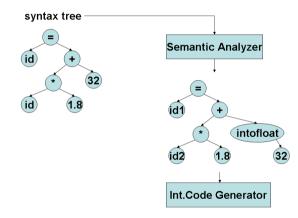
Generator

Code Optimization
Back-end

Code Optimization
Target Code
Generation

Sample Translatio

Summa



Source: Y N Srikant (NPTEL)

Compilers Pralay Mitra & Partha Pratim Das 01.11



Expression Quads

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

Front-end

Syntax Analysis

Semantic Analys

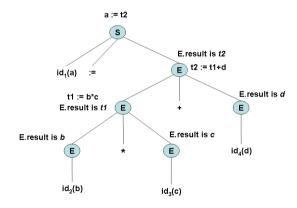
Intermediate Code Generator

Code Optimization

Code Optimization
Target Code

Sample Translation

Summa



Source: Y N Srikant (NPTEL)

Compilers Pralay Mitra & Partha Pratim Das 01.12



Intermediate Code Generator

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

C Compilati

Lexical Analy

Syntax Analysis

Semantic Analysis

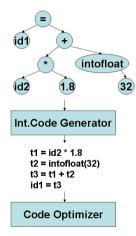
Generator

Code Optimization

Code Optimization Target Code

Sample Translatio

Summar



Source: Y N Srikant (NPTEL)

Pralay Mitra & Partha Pratim Das 01.13



Code Optimization

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

C Compilation

Tront-end

.

Syntax Analysis

Semantic Analy

Generator

Code Optimiza

Code Optimization

Sample

Summa

```
t1 = id2 * 1.8

t2 = intofloat(32)

t3 = t1 + t2

id1 = t3

Code Optimizer

t1 = id2 * 1.8

id1 = t1 + 32.0

Code Generator
```

Source: Y N Srikant (NPTEL)



Code Generation and Optimization: Practice Example

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of a Compiler

C Compilat

Lexical Analy

Syntax Analysis

Semantic Analysis

Generator

Code Optimization

Code Optimization
Target Code

Sample Translatio

Summar

* A+B*C+D

. .

• t0=A

• tI=B

• t2=C

t3=t1*t2

• t4=t0+t3

t5=D

• t6=t4+t5

* t0=A

* tl=B

* t2=C

* tl=tl*t2

* t0=t0+t1

* tl=D

* t0=t0+t1

* t0=A

* tl=B

* tl=tl*C

* t|=t0+t|

* t|=t|+D



Target Code Generation

Module 01

Pralay Mitra & I P Das

Objectives Outline

Phases of Compiler

C Compilation

Front-end

Syntax Analysis

Semantic Analysis

Intermediate Code

Code Optimizat

Back-end

Code Optimization

Target Code

Generation

Summar

- Data Flow and Control Flow Analysis
- \bullet Registration Allocation and Assignment
- Code Generation



Target Code Generation

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

C Compilatio

Front-end

Syntax Analysis

Semantic Analysis

Intermediate Co

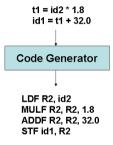
Code Optimiza

Back-end
Code Ontimization

Target Code
Generation

Sample Translatio

Summar



Source: Y N Srikant (NPTEL)



Sample pass through Phases

position

SYMBOL TABLE

initial

rate

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

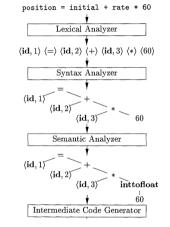
Front-end
Lexical Analysis
Syntax Analysis
Semantic Analysis
Intermediate Code

Code Optimization

Code Optimization Target Code Generation

Sample Translation

Summa



Intermediate Code Generator t1 = inttofloat(60) = id3 * t1t3 = id2 + t2id1 = t3Code Optimizer t1 = id3 * 60.0id1 = id2 + t1Code Generator R2, id3 MULF R2, R2, #60.0 R1, id2 ADDF R1, R1, R2 id1, R1

Source: Dragon Book

Figure: Translation of an assignment statement



Sample Translation

```
Module 01
Pralay Mitra &
```

Objectives of Outline

Phases of Compiler

Front-end Lexical Analysis Syntax Analysis

Semantic Analysis
Intermediate Code
Generator

Back-end

Sample Translation

Summar

```
{
   int i; int j;
   float a[100]; float v; float x;

   while (true) {
      do i=i+1; while(a[i]<v);
      do j=j-1; while(a[j]>v);
      if (i>=j) break;
      x=a[i]; a[i]=a[j]; a[j]=x;
   }
}
```

```
01: i = i + 1
02: t1 = a [ i ]
03: if t1 < v goto 01
04: j = j - 1
05: t2 = a [ j ]
06: if t2 > v goto 04
07: ifFalse i >= j goto 09
08: goto 14
09: x = a [ i ]
10: t3 = a [ j ]
11: a [ i ] = t3
12: a [ j ] = x
13: goto 01
14: .
```



A Typical Compiler Techniques

Module 01

Pralay Mitra & P Das

Objectives Outline

Phases of Compiler

C Compilati

Lexical Analys

Syntax Analysis

Intermediate Code

Code Optimization

Code Optimization

Sample Translatio

Summary

Promote high level languages by minimizing the execution overhead

Compiler

Support HPC systems

Support several source languages

Potential to translate correctly infinite set of programs written in the source language.

Support several target machines

Collection of compilers

Software engineering techniques

Generate optimal target code from source program ??



Languages by Translation Types

Module 01

Summary

Language C++

Java

Static Static Python Dynamic⁷

Compilation

Static

Typing Weak¹. Static Strong², Static³

Strong, Static⁵ Strong, Dynamic Framework Nο

 No^4 Yes⁶

Yes⁸

¹ For example, void* breaking typing

 $^{^{2}}$ If typical C features are not used

 $^{{\}rm 3}_{\rm Dynamic\ w/\ Polymorphism}$

ARTTI for dynamic_cast

⁵ Dynamic w/ Polymorphism

⁶ Java Virtual Machine – JVM

 $⁷_{\rm Interpreter}$

⁸ Python Virtual Machine – PVM



Module Summary

Module 01

Pralay Mitra & I P Das

Objectives Outline

Phases of Compiler

Front-end Lexical Analys Syntax Analys

Syntax Analysis
Semantic Analysis
Intermediate Code
Generator
Code Optimization

Back-end

Code Optimization

Target Code

Generation

Sample Translation

Summary

- Outline of Course and Material provided
- Recap on the outline of C Compilation Process
- Brief discussion on Phases of a Compiler to understand
 - Front-end flow: Language to TAC
 - o Back-end flow: TAC to Machine
- Infix to Postfix Translation
- Outline of languages with translation types