



Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

Module 01: CS31003: Compilers:

Overview: Phases of a Compiler

Indranil Sengupta
Partha Pratim Das

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

isg@iitkgp.ac.in
ppd@cse.iitkgp.ac.in

September 01 & 05, 2020



Course Outline

Module 01

I Sengupta &
P P Das

Objectives & Outline

Phases of a Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample Translation

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)
 - Compiler Design in C by Allen Holub
 - Advanced Compiler Design and Implementation by Steven Muchnick



Module Objectives

Module 01

I Sengupta &
P P Das

Objectives & Outline

Phases of a Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample Translation

Summary

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



Module Outline

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

1 Objectives & Outline

2 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

3 Sample Translation

4 Summary



Compiling a C Program

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code

Generator

Code Optimization

Back-end

Code Optimization

Target Code

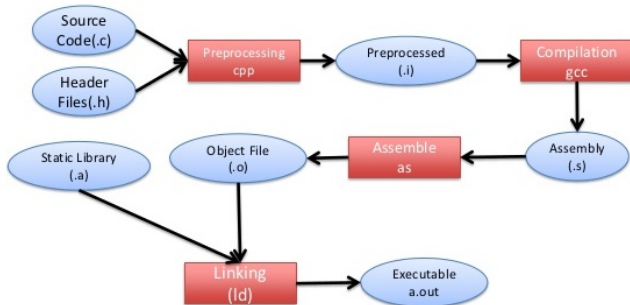
Generation

Sample

Translation

Summary

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



Compilation Flow Diagrams for gcc

Source: <http://www.slideshare.net/Bletchley131/compilation-and-execution> (slide #2)



Compiling a C Program

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

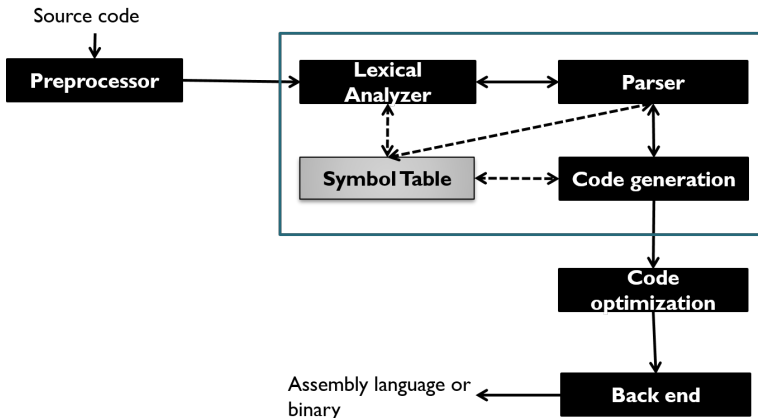
Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary



Four Pass Compiler



Phases

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

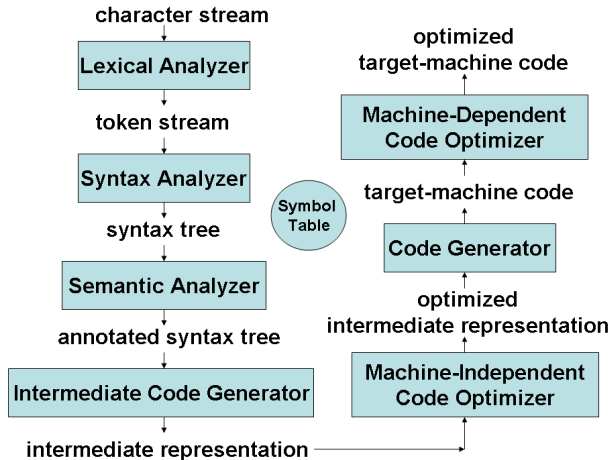
Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary



Source: Y N Srikant (NPTEL)



Lexical Analysis Phase

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

fahrenheit = centigrade * 1.8 + 32

Lexical Analyzer

**<id,1> <assign> <id,2> <multop>
<fconst, 1.8> <addop> <iconst,32>**

Syntax Analyzer

fahrenheit = *centigrade* * 1.8 + 32
totalAmount = *principalAmount* * 10 + *principalAmount*
finalVelocity = *acceleration* * *time* + *initialVelocity*

Source: Y N Srikant (NPTEL)



Lexical Analysis Phase

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

$$f = c * 1.8 + 32$$

$$b = a * 10 + a$$

$$v = a * t + u$$

$$id = id * num + num$$

$$id = id * num + id$$

$$id = id * id + id$$

$$E = E * E + E$$

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



Syntax Analysis Phase

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

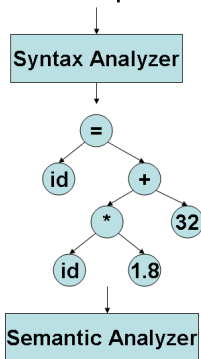
Code Optimization

Target Code
Generation

Sample
Translation

Summary

$\langle id, 1 \rangle \langle assign \rangle \langle id, 2 \rangle \langle multop \rangle$
 $\langle fconst, 1.8 \rangle \langle addop \rangle \langle iconst, 32 \rangle$



Source: Y N Srikant (NPTEL)



Semantic Analysis Phase

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

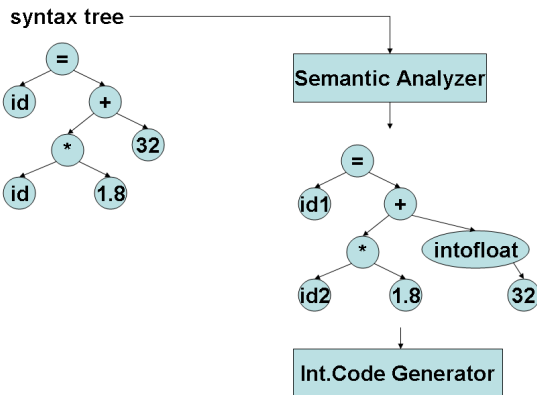
Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary



Source: Y N Srikant (NPTEL)



Expression Quads

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

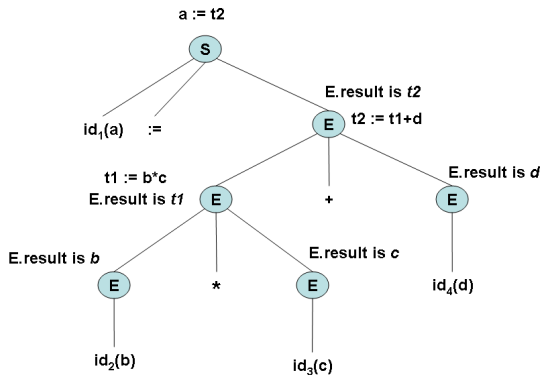
Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary



Source: Y N Srikant (NPTEL)



Intermediate Code Generator

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

**Intermediate Code
Generator**

Code Optimization

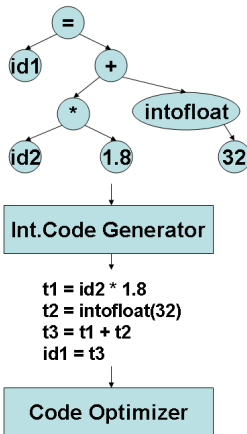
Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary



Source: Y N Srikant (NPTEL)



Code Optimization

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

```
t1 = id2 * 1.8  
t2 = intfloat(32)  
t3 = t1 + t2  
id1 = t3
```

Code Optimizer

```
t1 = id2 * 1.8  
id1 = t1 + 32.0
```

Code Generator

Source: Y N Srikant (NPTEL)



Code Optimization

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

```
t1 = id2 * 1.8  
t2 = intfloat(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

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

* $A+B*C+D$

• $t0=A$

• $t1=B$

• $t2=C$

• $t3=t1*t2$

• $t4=t0+t3$

• $t5=D$

• $t6=t4+t5$

* $t0=A$

* $t1=B$

* $t2=C$

* $t1=t1*t2$

* $t0=t0+t1$

* $t1=D$

* $t0=t0+t1$

* $t0=A$

* $t1=B$

* $t1=t1*C$

* $t1=t0+t1$

* $t1=t1+D$



Target Code Generation

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

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



Target Code Generation

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

t1 = id2 * 1.8
id1 = t1 + 32.0

Code Generator

LDF R2, id2
MULF R2, R2, 1.8
ADDF R2, R2, 32.0
STF id1, R2

Source: Y N Srikant (NPTEL)



Sample pass through Phases

Module 01

I Sengupta &
P P Das

Objectives & Outline

Phases of a Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample Translation

Summary

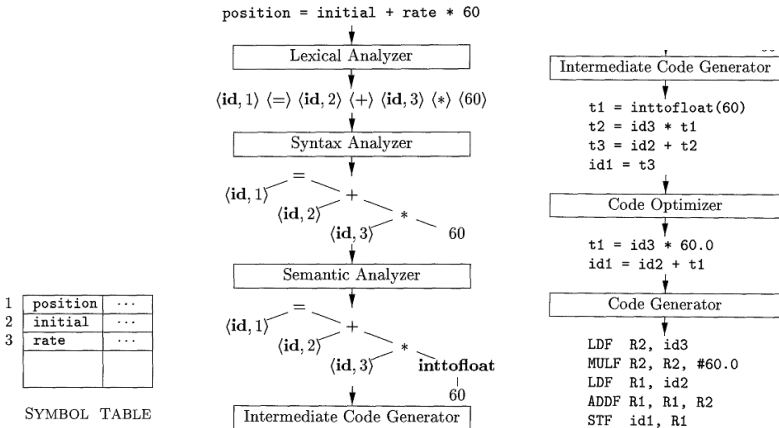


Figure: Translation of an assignment statement



Sample Translation

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

```
{  
    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

I Sengupta &
P P Das

Objectives & Outline

Phases of a Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample Translation

Summary

Promote high level languages by
minimizing the execution overhead

Support HPC systems

Compiler

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

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

Syntax Analysis

Semantic Analysis

Intermediate Code
Generator

Code Optimization

Back-end

Code Optimization

Target Code
Generation

Sample
Translation

Summary

Language	Compilation	Typing	Framework
C	Static	Weak ¹ , Static	No
C++	Static	Strong ² , Static ³	No ⁴
Java	Static	Strong, Static ⁵	Yes ⁶
Python	Dynamic ⁷	Strong, Dynamic	Yes ⁸

¹ For example, void* breaking typing

² If typical C features are not used

³ Dynamic w/ Polymorphism

⁴ RTTI for `dynamic_cast`

⁵ Dynamic w/ Polymorphism

⁶ Java Virtual Machine – JVM

⁷ Interpreter

⁸ Python Virtual Machine – PVM



Module Summary

Module 01

I Sengupta &
P P Das

Objectives &
Outline

Phases of a
Compiler

C Compilation

Front-end

Lexical Analysis

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
 - Back-end flow: TAC to Machine
- Infix to Postfix Translation
- Outline of languages with translation types