

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of a Compiler

C Compilatio

Front-end

Lexical Analys

Syntax Analysis

Semantic Analysi

Intermediate C

Code Optimization

Back-end

Code Optimizatio

Sample Translatio

Summar

Module 01: CS 348: Compilers

Overview: Phases of a Compiler

Prof. Sukumar Nandi

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

sukumar@iitg.ac.in

February 17, 2023



Course Outline

Module 01

Prof. Sukuma Nandi

Objectives & Outline

Phases of Compiler

C Compilatio Front-end

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

Sample Franslation

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

Prof. Sukuma Nandi

Objectives & Outline

Phases of

C Compilat

Front-end

.

Dylitax Allalysis

Semantic Analysi

Intermediate Co

Code Optimizatio

Code Optimizatio

Sample Translatio

Summar

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



Module Outline

Module 01

Prof. Sukum Nandi

Objectives & Outline

Phases of Compiler

Front-end Lexical Analys

Syntax Analysis
Semantic Analysis
Intermediate Code

Generator
Code Optimization
Back-end
Code Optimization

Sample Translatior

Summary

1 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

Prof. Sukuma Nandi

Objectives Outline

Phases of a Compiler

C Compilation

Front-end

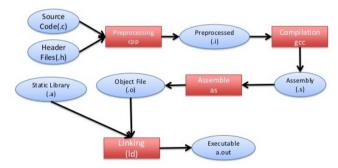
Lexical Analysis Syntax Analysis Semantic Analysis Intermediate Code Generator

Back-end
Code Optimization
Target Code

Sample Translatio

Summan

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



Compilation Flow Diagrams for gcc

 $\textbf{Source}: \ http://www.slideshare.net/Bletchley 131/compilation-and-execution (slide \#2)$



Compiling a C Program

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilation

Event and

Cexical Analysis

Semantic Analysis

Semantic Analysis

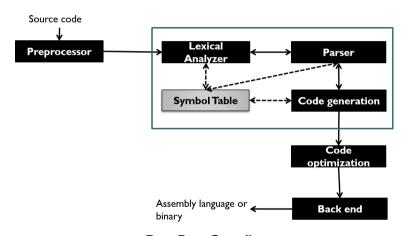
Generator

Code Optimizatio

Code Optimization Target Code

Sample

Summai



Four Pass Compiler



Phases

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilation

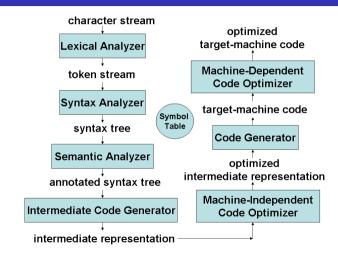
Lexical Analysis Syntax Analysis Semantic Analysis Intermediate Code Generator

Back-end

Code Optimization Target Code Generation

Translation

Summar



Source: Y N Srikant (NPTEL)

CS 348 Prof. Sukumar Nandi 01.7



Lexical Analysis Phase

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of a Compiler

Eront-and

Lexical Analysis

Syntax Analysis Semantic Analysis

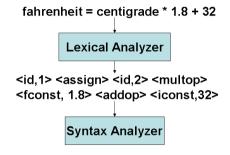
Intermediate Code

Code Optimization

Code Optimization
Target Code

Sample Translation

Summar



fahrenheit = centigrade * 1.8 + 32

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

 $\mathit{finalVelocity} = \mathit{acceleration} * \mathit{time} + \mathit{initialVelocity}$



Lexical Analysis Phase

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilat

Lexical Analysis

Syntax Analysis

Comments Annhala

Intermediate Cod

Code Ontimization

Back-end
Code Optimization
Target Code

Sample Translatio

Summai

$$f = c * 1.8 + 32$$

$$b = a*10 + a$$

$$v = a * t + u$$

$$id = id * num + num$$

$$id = id * num + id$$

$$id = id * id + id$$

$$E = E * E + E$$

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



Syntax Analysis Phase

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

Front-end

Syntax Analysis

Semantic Analys

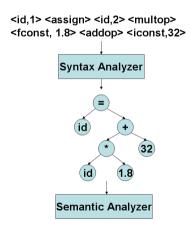
Intermediate Co

Code Optimization

Code Optimization
Target Code

Sample Translatio

Summar



Source: Y N Srikant (NPTEL)

CS 348 Prof. Sukumar Nandi 01.10



Semantic Analysis Phase

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

Front-end
Lexical Analys

Semantic Analysis

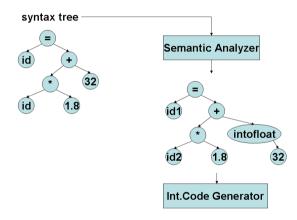
Intermediate Coc

Code Optimization

Code Optimization
Target Code

Sample Translatio

Summa



Source: Y N Srikant (NPTEL)

CS 348 Prof. Sukumar Nandi 01.11



Intermediate Code Generator

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilation

Svntax Analysis

Semantic Analysis

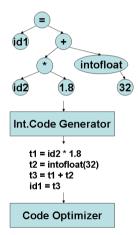
Generator

Code Optimization

Code Optimization Target Code

Sample Translatio

Summar





Code Optimization

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilat

Front-end

Lexical Analy

Syntax Analysis

Semantic Analy

Intermediate Co

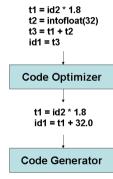
Code Optimizat

Code Optimization

Target Code Generation

Sample Translatio

Summa





Code Generation and Optimization: Practice Example

Module 01

Code Optimization

* A+B*C+D

t0=A

tI=B

t2=C

• t3=t1*t2

• t4=t0+t3

t5=D

• t6=t4+t5

* t0=A

* tI=B

* t2=C

* t | =t | *t2

* t0=t0+t1

* tI=D

* t0=t0+t1

* t0=A

* tI=B

* tl=tl*C

* t | =t0+t |

*tl=tl+D



Target Code Generation

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilation

Lexical Analys

Syntax Analysis

Semantic Analysis

Generator

Code Optimizatio
Back-end

Code Optimizati

Target Code

Generation

Sample Translatio

Summai

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



Target Code Generation

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilation

Front-end

Syntax Analysis

Semantic Analysis

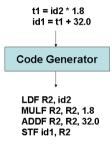
Generator

Code Optimizati

Code Optimizatio

Generation
Sample

Summai





Sample pass through Phases

position

SYMBOL TABLE

initial

rate

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

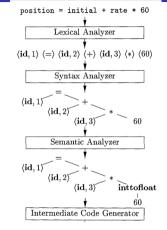
Front-end Lexical Analysis Syntax Analysis Semantic Analysis Intermediate Code

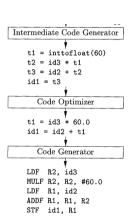
Code Optimization Back-end

Code Optimization Target Code Generation

Sample Translation

Summa





Source: Dragon Book

Figure: Translation of an assignment statement



A Typical Compiler Techniques

Module 01

Prof. Sukum Nandi

Objectives Outline

Phases of Compiler

Front-end

Lexical Analys

Semantic Analysis

Generator

Back-end Code Optimization

Sample Translatio

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

Prof. Sukum Nandi

Objectives Outline

Phases of Compiler

Front-end
Lexical Analy

Semantic Analysis Intermediate Code

Code Optimization

Code Optimizatio
Target Code
Generation

Sample Translation

Summary

Language C C++

Java

Python

Static Static Static

Static Dynamic⁷

Compilation

Typing Weak¹, Static Strong², Static³

Strong, Static⁵ Strong, Dynamic Framework

No⁴ Yes⁶ Yes⁸

¹ For example, void* breaking typing

² If typical C features are not used

³ Dynamic w/ Polymorphism

RTTI for dynamic_cast

⁵ Dynamic w/ Polymorphism

⁶ Java Virtual Machine – JVM

^{7&}lt;sub>Interpreter</sub>

⁸ Python Virtual Machine – PVM



Module Summary

Module 01

Prof. Sukuma Nandi

Objectives Outline

Phases of Compiler

C Compilation Front-end Lexical Analys Syntax Analys Semantic Anal

Semantic Analysis Intermediate Code Generator Code Optimization Back-end

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
- Outline of languages with translation types