

Korea University MatKor 스터디 – C 컴파일러 만들기

Introduction

Chris Ohk

utilForever@gmail.com

발표자 소개

Korea University MatKor 스터디
Introduction

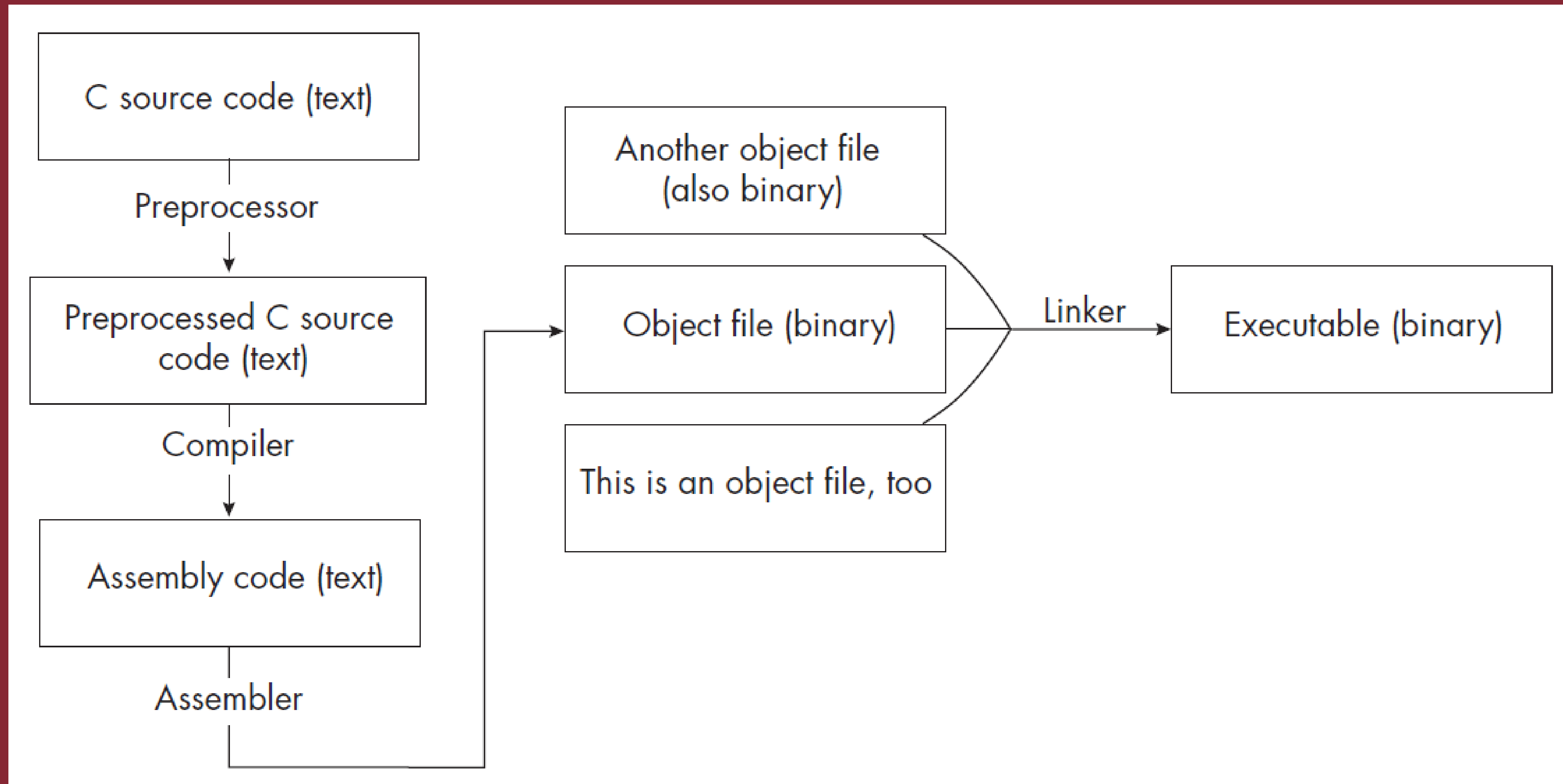
- 옥찬호 (Chris Ohk)
 - (현) 42dot Sr. Embedded System Engineer
 - (전) EJN Tech Lead
 - (전) Momenti Engine Engineer
 - (전) Nexon Korea Game Programmer
 - Microsoft Developer Technologies MVP
 - C++ Korea Founder & Administrator
 - Reinforcement Learning KR Administrator
 - IT 전문서 집필 및 번역 다수

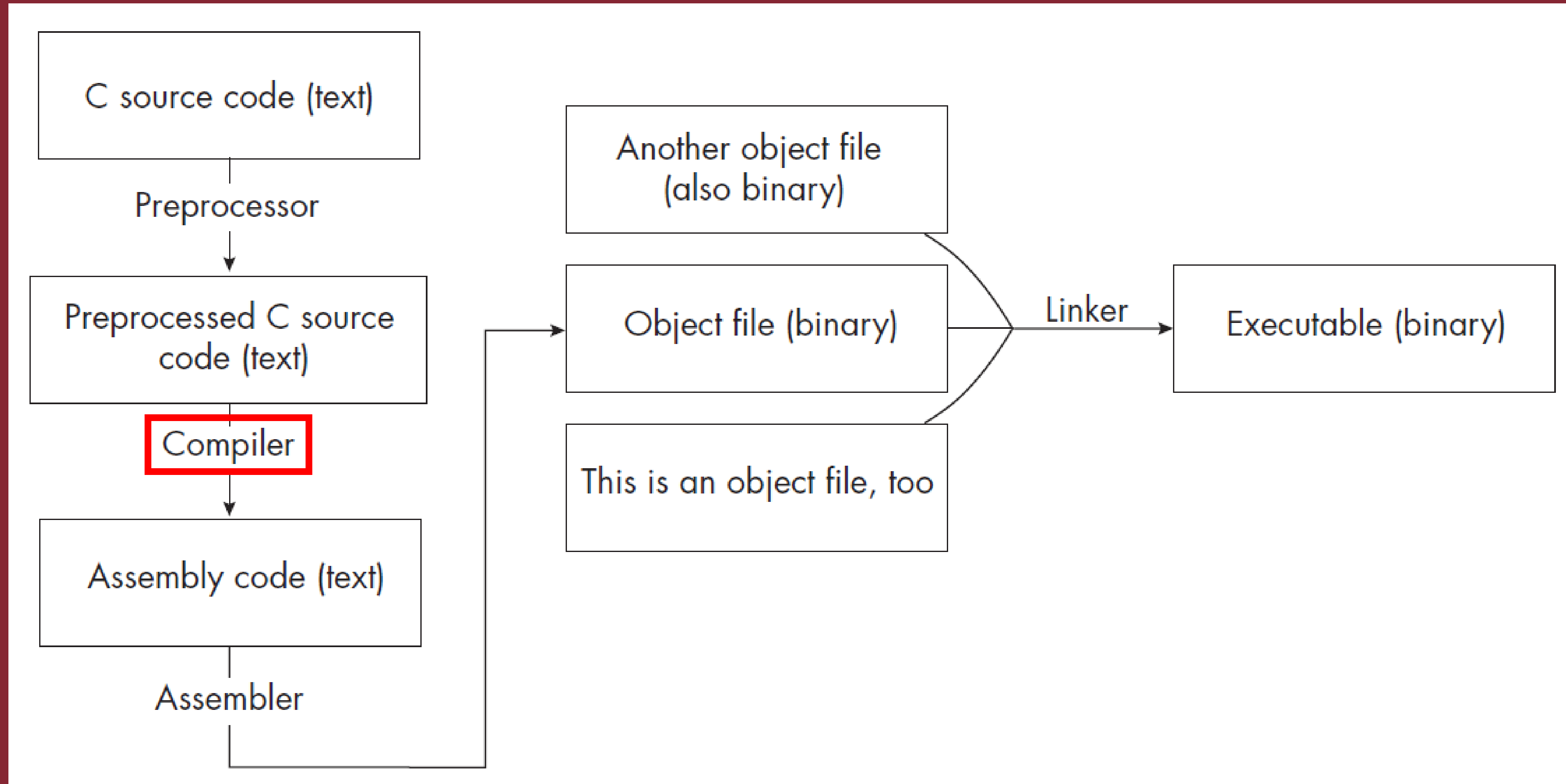
utilForever@gmail.com



utilForever







- **Writing a C Compiler, No Starch Press (2024).**
- Engineering a Compiler (3rd Edition), Morgan Kaufmann (2022).
- Compilers: Principles, Techniques, and Tools (2nd Edition), Addison Wesley (2006).
- Compiler Construction: Principles and Practice, Cengage Learning (1997).
- LLVM Code Generation: A deep dive into compiler backend development, Packt (2025).
- Writing A Compiler In Go, Thorsten Ball (2018).

- 강의 요일 및 시간
 - 요일 : 화요일 오후 8시 시작
 - 시간 : 1시간 30분 ~ 2시간
- 참고 사항
 - 온라인 + 오프라인 강의 (오프라인 강의는 월 1~2회 진행)
 - 진행자의 개인 사정에 따라 스터디 일정이 변경될 수 있음

- 강의에서는 특정 언어를 기준으로 설명하지 않습니다.
 - Pesudocode로 설명합니다.
- 매주 구현 과제가 나갑니다.
 - 강의에서 배운 내용을 구현하는 방식입니다.
 - 테스트 코드를 제공해 구현을 문제없이 했는지 확인할 수 있습니다.
 - 과제에 대한 정답 코드는 C++과 Rust로 제공합니다.
 - 과제에 대한 힌트를 얻고 싶다면, 레퍼런스 구현(NQCC2)을 참고하셔도 됩니다.

<https://github.com/nlsandler/nqcc2>

- 운영체제
 - macOS (Apple Silicon; x64 Emulator)
 - Linux (x64 Processor)
 - Windows Subsystem for Linux (WSL)
- 프로젝트 의존성
 - Python 3.8 이상
 - 실제 C 컴파일러 (GCC 또는 Xcode Clang)

- For Linux: GCC, GDB
 - 설치 : `sudo apt-get install gcc gdb`
 - 설치 확인 : `gcc -v, gdb -v`
- For macOS: Command Line Developer Tools
 - 설치 : `xcode-select --install`
 - 설치 확인 : `clang -v`
- For Apple Silicon: Rosetta 2
 - 설치 : `softwareupdate --install-rosetta --agree-to-license`
 - 설치 확인 : `arch -x86_64 zsh`

- 설치 확인
 - `git clone https://github.com/nlsandler/writing-a-c-compiler-tests.git`
 - `cd writing-a-c-compiler-tests`
 - `./test_compiler --check-setup`
- All system requirements met! 이라고 뜨면 준비 완료!

- C 표준 (C17)
<https://www.iso.org/standard/74528.html>
- System V Application Binary Interface (ABI)
<https://gitlab.com/x86-psABIs/x86-64-ABI>
- Intel 64 Software Developer's Manual
<https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html>
- Compiler Explorer
<https://godbolt.org/>

- Part 1: The Basics
 - Lecture 0: Introduction
 - Lecture 1: A Minimal Compiler
 - Lecture 2: Unary Operators
 - Lecture 3: Binary Operators
 - Lecture 4: Logical and Relational Operators
 - Lecture 5: Local Variables

- Part 1: The Basics
 - Lecture 6: if Statements and Conditional Expressions
 - Lecture 7: Compound Statements
 - Lecture 8: Loops
 - Lecture 9: Functions
 - Lecture 10: File Scope Variable Declarations and Storage-Class Specifiers

- Part 2: Types Beyond Int
 - Lecture 11: Long Integers
 - Lecture 12: Unsigned Integers
 - Lecture 13: Floating-Point Numbers
 - Lecture 14: Pointers
 - Lecture 15: Array and Pointer Arithmetic
 - Lecture 16: Characters and Strings
 - Lecture 17: Supporting Dynamic Memory Allocations
 - Lecture 18: Structures

- Part 3: Optimizations
 - Lecture 19: Optimizing TACKY Programs
 - Lecture 20: Register Allocation

감사합니다.

utilForever@gmail.com

<https://github.com/utilForever>

X, Instagram: @utilForever