組合語言與系統程式

Teacher, I have questions

Lecturer:

- Office: science building (理學大樓) 812 room
- Email: albert.yang@nchu.edu.tw

TA: 董諺承

- Email: dong06780@gmail.com

Making an appointment through email in advance is highly recommended!!

Grading policy

Midterm exam (40%)

- Two exams for assembly language, 20% each

Final exam (30%)

- System programming

Assignment (30%)

What is Assembly Language?

- Assembly language is a low-level programming language that is very close to machine language but provides a more understandable representation
- It uses **mnemonics** (e.g. ADD, INC) to represent machine instructions, addresses, and other opcodes, allowing programmers to directly control the hardware

What is Assembly Language?

- Assembly language is specific to a particular processor architecture, which means that the assmbly language code will be designed differently for different processors
- Assembly languages are usually used in situations where precise control of hardware is required, such as embedded systems, driver development, system startup programs, etc.

What is System Programming?

- Systems programming refers to the process of designing, developing, and maintaining system software that provides a bridge between hardware and applications (e.g., operating systems, device drivers)
- Assembly language is a basic tool in system programming for communicating directly with hardware, while system programming refers to the development of underlying software that enables the hardware to execute applications.

Why Should We Learn These?

- Understand how computers work
  - how the processor executes instructions, how memory is managed and used, and how the input/output system works …
- Performance optimization
  - such as in embedded systems or high-performance computing applications
- System program development
  - system programs, such as operating systems, drivers, etc.
- Interview …

Thanks to ChatGPT ☺

Syllabus - Assembly Language

Chapter 1: Basic Concepts

Chapter 2: x86 Processor Architecture

Chapter 3: Assembly Language Fundamentals

Chapter 4: Data Transfers, Addressing and Arithmetic

Chapter 5: Procedures

Chapter 6: Conditional Processing

Chapter 7: Integer Arithmetic

Chapter 8: Advanced Procedure

Syllabus - System Programming

Chapter 1 : Background

Chapter 2 : Assemblers

Chapter 3 : Loaders and Linkers

Chapter 4 : Macro Processors

Schedule

2024/3/19: Self-study week

2024/6/11: Final exam

2024/6/18: Self-study week