

Course Description

Tianjin International Engineering Institute

Course Name (Chinese): 计算机系统 1
(English): Computer System1

Course Name: Computer System1			Course Code:		
Semester: 1			Credit: 2.5		
Program: Computer Science					
Course Module: Specialized Compulsory					
Responsible: Xuewei Li			E-mail: lixuwei@tju.edu.cn		
Department: School of Computer Science & Technology, Tianjin University					
Time Allocation (1 credit hour = 45 minutes)					
Exercise	Lecture	Lab-study	Project	Internship (days)	Personal Work
8	14	18			15

Course Description

A required course designed for Engineering Master of Computer Science in TIEI. This course will systematically describe how computer systems run programs, store information and communicate. The main components of a computer system will be introduced. Emphasis will be given to the key points, such as machine instruction, computer system architecture, and hierarchical memory. Besides, some experimental projects will be combined to cultivate students to develop simple systems. Students will have a preliminary and extensive understanding about computer system, which enables students to become more effective Programrs. In addition, this course also serves as a foundation for courses on compilers, networks, operating systems, and computer architecture, where a deeper understanding of systems-level issues is required.

Pre-requisite

- Basic programming knowledge: the ability to write programs independently,
- Knowledge about Computer Composition Principles: know about the essential components of a computer, instruction systems, etc, and to
- Knowledge about compiler: understand the process of translating a source program to target language.

Course Objectives

This course discusses basic concepts of computer system to help students understand the program better and enhance their professional skills. After taking this course, students should be able to:

- Understand computer system architecture,
- Understand what really happens when a program runs,
- Have the intellectual tools to find out and solve the problem when something goes wrong in their program, and to
- Systematically understand operating systems and computer networks.

Course Syllabus

- Basics of computer system: representation of the program, compiler building, storage structure, foundation of operating system,
- Information represents: data storage, integer arithmetic, floating-point arithmetic,
- Instruction and assembly: instruction and data formats, addressing mode, arithmetic and logical operations, control instructions, procedure call, buffer overflow,
- Processor: Y86 instruction set, hardware description language solutions, pipeline, data collision,
- Program performance optimization: cycle, procedure call, data references, parallelism, and to
- Memory structure: storage technologies, principle of locality, storage hierarchy, Cache structure.

Textbooks & References

- Randal E Bryant and David R O'Hallaron. *Computer Systems: A Programmer's Perspective*, 2nd ed), Prentice Hall, 2011.
- Brian W Kernighan and Dennis M Ritchie. *The C Programming Language* (2nd ed). Prentice Hall, 1988.
- Hayes John P. *Computer architecture and organization*. 2002.
- Hennessy J L, Patterson D A. *Computer architecture: A quantitative approach* (3rd ed). Message Passing Interface Forum, 2003.

Capability Tasks

CT2: To understand the basic concepts of computer system, be able to analyze system or software with professional knowledge.

CT3: To master execution process of a program, and be able to debug fast and accurately.

CT4: To use tools to analyze and optimize a system or software.

CS1: To master the basic theories of computer system, and know its development status and trends.

CS2: To gain a comprehensive and solid foundation of computer system architecture to develop a system.

CS3: To have the ability to analyze and design a system or software, and be able to propose a solution for a practical background.

Achievements

- To understand information storage in computer - Level: N
- To master common assembly instructions, and be able to write simple assembly programs - Level: M
- To master the pipeline technique and process mode of data collision - Level: M

- To master Program optimization methods, and be able to optimize programs of moderate complexity - Level: M
- To understand memory architecture and storage technologies - Level: N

Students: Computer Science, Year 1