

Course Title: Assembly Language and System Programming

Course Type: Main Course

Unit Type: Theoretical

Units: 3

Prerequisites: None

Project: Yes

Teaching Hours: 48

Objectives:

This course intends to introduce the student to the assembly language programming and the assembly language way of thinking. In this course students are also introduced to the concepts related to the PC system programming which includes system programming related to the MS-DOS operating system. Although the course is focused on the MS-DOS operating system, students will be also introduced to the concepts such as computer viruses and low level system programming. In this course, PC components such as video cards and their utilizations will be discussed. Course will have a seminar and a project as its course work.

Syllabus:

1. Introduction to some logic design components such as registers.
2. Introduction to the 8086 microprocessor, its internal architecture, addressing space and IBM PC memory architecture
3. Tasks assigned to the assembler, linker and loader. How to build an executable file and how does the MS-DOS operating system loads programs and executes them.
4. Data types and addressing modes in 8086 assembly language.
5. Introduction to the 8086 assembly language instruction set, its status flags and methods for passing parameters to the procedures.
6. Assembler pseudo instructions and how to write a code understandable by the assembler.
7. Introduction to the PSP (how to use information included in PSP), the .EXE and .COM programs and how to build these types of executable files. Example codes will be provided.
8. The Numerical Co-processor and its instructions/programming. Concepts such as co-processor synchronization will be discussed.
9. Device driver concept and device drivers in MS-DOS (block and character device drivers).
10. Viruses in computer systems.
11. MACRO programming in IBM PC assembly language.
12. Connecting the assembly languages with high level languages. Concept of the multi-language programming. Calling routines and passing parameters to them in C, PASCAL, BASIC and FORTRAN languages.
13. Interrupts and exceptions in 80x86 and how to write an Interrupt Service Routine (ISR) (i.e. interrupt handler) and memory resident programs in MS-DOS.
14. Introduction to the BIOS and MS-DOS interrupts and their functionalities. How to open file (FCB, file handle), write to the screen, format disk, etc in assembly language using these interrupts.
15. MS-DOS disk structure, Graphic Cards and some system programming concepts related to them.
16. Advance system programming concepts if time permits.

Recommended References

1. Peter Abel, *IBM PC Assembler Language and Programming*, Prentice-Hall International Editions, 1997.
2. M. Tischer, B. Jennrich, *IBM PC Intern*, 6th Edition, Abacus, 1996.