DBLN CPSI 2295

This course is being offered at Griffith College, CAPA's academic partner in Dublin. The Irish academic system differs from the US, particularly with grading. Griffith College professors expect students to undertake a good deal of independent study to achieve a high mark in their classes. For additional information about this class, please contact the Boston Program Advising Team at 1-800-793-0334.

Systems Software

Continuous Assessment: 60%
Exam: 40%

Intended Module Learning Outcomes

On successful completion of this module learners will be able to:

1. Explain the role and functioning of low-level software such as the BIOS, boot loaders, device drivers and and firmware
2. Design and implement basic assembly language programs

3. Explain the role played by the operating system and outline the characteristics of a typical operating system
4. Differentiate between programming languages with respect to the level and features of the language
5. Explain how program assembly, compilation and linking is performed
6. Critically analyse approaches to input and output
7. Compare approaches using virtual machines with natively compiled software

Module Objectives

This module aims to support participants as they develop their understanding of computer software and in particular system software. Participants develop a broadly based and intellectually challenging framework in system software and low-level programming languages such as assembly language. Participants have an awareness of current technologies, literature and research in the area. Participants are expected to apply the principles of system software to solve problems in current and in developing areas. Participants achieve this through developing knowledge and skills in computer system software. Further, they cultivate an understanding of how the insights and practice from system software and assembly language programming contribute to the current state of the art in the wider Computer Science landscape.

Module Curriculum

System Software and the the Basic Input/Output System

- Differentiation of application and system software.
- System software types and uses.
- Interaction between hardware devices and low-level software.
- The BIOS, bootloaders, device drivers and firmware.
- User and kernel mode execution.

Assembly Language Programming

- Basic program assembly and execution.
- Registers and instructions.
- Operand types.
- Variables and labels.
- Working with the stack.
- Working with numbers.
- Arithmetic and logic operations, comparison.
- Left and right shifting. Bitwise operations.
- Using the processors instruction set specification.
- Communicating with peripheral devices.

Introduction to Operating Systems

- Operating system roles.
- Operating system services.
- Processes and programs.
- Concurrent execution. Scheduling.
- Features of a typical operating system.

Programming Languages

- History and development of programming languages.
- High and low-level languages.
- Language generations.
- Common features of programming languages including variable creation and use, arithmetic and logic operations, iteration, conditional statements and abstraction mechanisms such as objects.

Program Compilation & Linking

- Assembly of low level programs.
- Compilation of high level programs.
- Static and dynamic linking.
- How to create a library file.

Input & Output

- Input and output devices.
- Serial and parallel movement of data.
- Movement of data to and from devices.
- Interrupts and polling.
- Programmed I/O, I/O instructions, memory-mapped I/O.
- Direct Memory Access (DMA).

Virtual Machines

- Advantages and disadvantages.
- Differentiating the virtual machine and native platform approaches.
- The characteristics of a typical virtual machine.