CS202 – System Software

Dr. Manish Khare

Lecture 1

About Me

Dr. Manish Khare

M.Sc. (Computer Science), M.Tech. (Computer Science and Engineering), PhD (Computer Science)

Address

Mail: manish_khare@daiict.ac.in

Phone: +91-9452081912

https://sites.google.com/site/mkharejk/

Course Objective

- This course introduces design and implementation of various types of system software and their relationship with machine architecture.

Introduction – CS202

- Course Name: System Software
- Course ID: CS202
- Credits: 3(L)-0(T)-0(P)-3(C)

Schedule– MC212

- Tuesday
  - 10:45 - 11:45 – Section 2
  - 12:00 - 13:00 – Section 1
- Wednesday
  - 10:45 - 11:45 – Section 1
  - 12:00 - 13:00 – Section 2
- Friday
  - 09:30 - 10:30 – Section 2
  - 10:45 - 11:45 – Section 1

Course Evaluation Distribution (Tentative)

- Evaluation Distribution:
  - Mid-Sem Exam – 25%
  - End Semester Exam – 45%
  - Assignments – 15%
  - Class Quizzes/Viva – 15%

Suggested Books/Literatures

- System Software – An introduction to System Programming, Leland L. Beck, 3rd Edition, Pearson Education.
- Systems Programming and Operating Systems, D. M. Dhamdhere, Tata McGraw Hill Publication.
- System Programming, John Donovan, McGraw Hill Publication.
- System Software, Santanu Chattopadhyay, Prentice Hall India Publication.
- Other Online References

Course Content

- ***Introduction to System Software:*** Definition, System software, Machine structure, Components of a programming system, Assemblers, linker, loader, compiler, Macros, text Editor, Debugger, Program development Flow, Introduction to Operating System, Language Processor, Assembly Language, Introduction to CISC and RISC machine architecture

- **Assembler:** Basic Assembler Functions, Machine Dependent Features, Machine Independent Features, One pass and Multi pass Assembler.

Course Content

- **Linkers, Loaders, Macros and Macro Processors**: Basic Loader Function, Loader Design Options, Relocation and Linking Concepts, Design of a Linker, Case study for Linker and Loader, Macro definition, Macro expansion, Basic Macro Processor Functions and Features, Macro Processor Design Options, Implementation example for Macro Processor

- **Compilers:** Aspects of Compilation, Compiler Features, Memory Allocation, Grammar, Parsing Techniques, Compiler Design Options, Intermediate Code Generation and Optimization Techniques

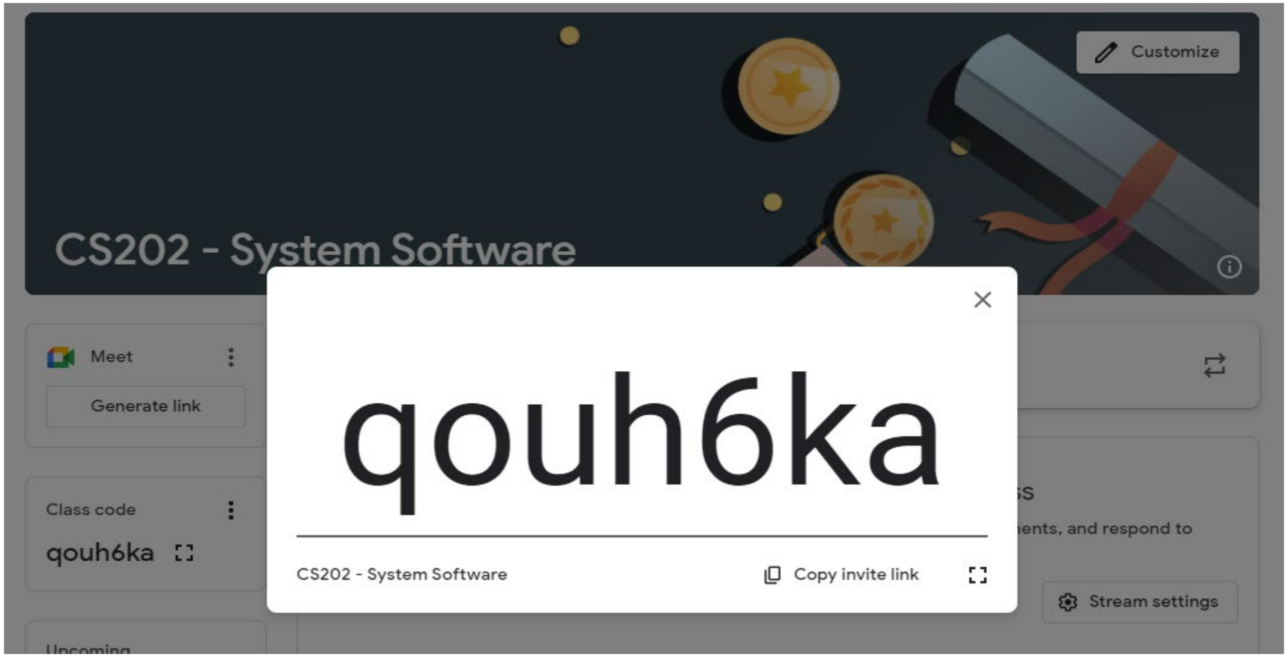
- **Scanning and Parsing**: Programming language grammars, Scanning, parsing, language processor development tools

- **Interpreters:** Overview of interpretation, benefits of interpretation

- **Software Tools:** Text Editors, Debuggers, User Interfaces.

Platform used for Lecture slides

- For Lecture Slides and Other Reference Matter
  - Google Classroom
    - Class Code: (qouh6ka)



CS202 - System Software

qouh6ka

CS202 - System Software

Class code

qouh6ka

Learning Outcomes

- On successful completion of this course, students should be able to:
  - List relationship between machine architecture and system software.
  - Analyze different types of software processors viz. assemblers, compilers, loaders.
  - Able to differentiate between top down and bottom up parsing and understand syntax directed translation techniques.

Introduction to System Software

- System software includes numerous programs that assist the operations of a computer.
- This makes it probable for the consumer to concentrate on an application or other difficulty to be solved, without requiring to know the details of how the machine functions internally.
- Instances of system software are text-editors, compilers, loaders or linkers, debuggers, assemblers, and operating systems.

System Software Concept

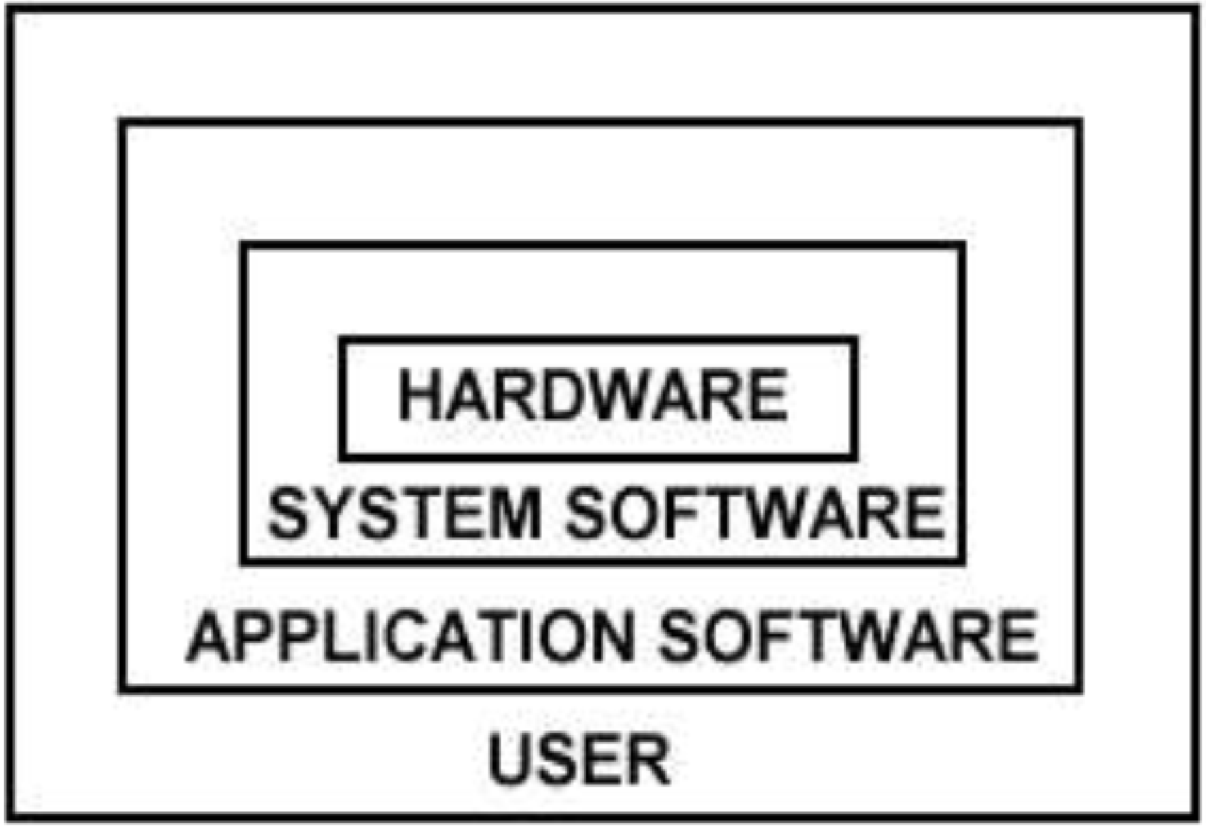
- System Software is a set of programs to carry out a number of system functions such as file editing, resource organization, I/O management and storage organization.

- The whole gamut of softwares present in a computer system can broadly be classified into two categories:
  - Application Softwares: Written by individual users for their particular computational requirements, e.g. scientific calculations, database applications, etc
  - System Softwares: Generally written by the manufacturer or the system administrator for proper and easy use of the system and its maintenance.

- It is very difficult to draw the demarcation line between these two categories of softwares.
- because in most of the systems, softwares are designed following a layered approach-a software available at lower level may be used to design a higher level one.
- But, we can broadly say that a system software is much closer to the actual hardware than the application programs.
- System softwares help in the development of application programs and system maintenance.

- The major system softwares are listed below:
  - Operating system
  - Compiler
  - Assembler
  - Linker
  - Loaded
  - Text editor
  - Debugger

Relation between Hardware, Software and user



HARDWARE

SYSTEM SOFTWARE

APPLICATION SOFTWARE

USER

Machine Dependency of System Software

- An assembler is considered as a system software which converts mnemonic instructions into machine code; the instruction formats, addressing modes, etc., are of direct relation in assembler design.
- Likewise, compilers must produce machine language code, considering such hardware traits as the number and the types of registers and machine instructions obtainable.
- Operating systems are directly related with the management of almost all of the resources of a computer system.

- When you took the initial programming course:
  - Text editor: creates and alters the program
  - Compiler: convert programs into machine language
  - Loader or linker: load machine language program into memory and geared up for implementation
  - Debugger: assist in detecting errors in the program.

- When you engraved programs in assembler language:
  - Assembler: convert assembly program into machine language
  - Macro processor: convert macros instructions into its definition.

- When you manage all of these processes
  - By networking with the OS
  - The significant machine structures utilized in the design of system software are:
    - Memory structure
    - Registers
    - Data formats
    - Instruction formats
    - Addressing modes
    - Instruction set.

SIC Machine

- SIC points to Simplified Instruction Computer which is a imaginary computer that has been intended to comprise the hardware traits most frequently found on real machines, while averting unusual and immaterial complexities.
- This permits to evidently separate the central concepts of a system software from the execution details related with a specific machine.