

System Software

CS202

Evaluation Scheme

	Type of Examination	Weightage (%) (+-5%)
1	Continuous evaluation/Quizzes	25%
2	Mid-semester examination	25%
3	End-semester examination	50%

Software Types

- **Application software**

- Programs designed for end-users to perform specific tasks
 - E.g., Word, Excel, PowerPoint, Google Docs, Sheets, Slides, Web Browsers, Media Players, Graphics Editors: Adobe Photoshop, CorelDRAW, Messaging Apps: WhatsApp, Social Media Platforms: Facebook, Instagram, E-commerce Apps: Amazon, Flipkart, eBay,

- **System software**

- Programs that manage hardware and provide a platform for application software. E.g.,
 - Operating Systems: Windows, macOS, Linux,
 - Utility Software: Antivirus, Disk Cleanup,
 - Device Drivers: Printer drivers, Graphics card drivers,
 - Firmware: BIOS,
 - Server Software: Apache, IIS
 - Compilers: GCC, LLVM, Python Interpreter
 - loaders
 - Assemblers and Linkers

SYSTEM SOFTWARE

- Definition: are programs used for
 - Functioning of a computer system
 - Its maintenance
 - Providing the general user with a user-friendly development platform for their programs
- **Functions**
 - **Resource Management**: Managing CPU, memory, and I/O devices
 - **Hardware Abstraction**: Hiding the complexity of hardware operations from users
 - **Service Provision**: Enabling execution of application software

SYSTEM SOFTWARE

- Consider a machine with CPU + memory chips
 - Still perform a lot of work
 - Program may be burnt on ROM
 - CPU execute instructions one-by-one
- **Limitation:** User need to be capable of writing program in machine language
 - It motivates necessity of existence of more user-friendly interfaces

SYSTEM SOFTWARE: User-friendly interfaces

- Early Systems (in 1940s) relied on direct programming of machine code
- **Evolution of System Software**
 - **First Generation (1940s):** Manual programming with machine languages
 - **Second Generation (1950s-60s):** Assembly lang., assemblers and No true OS
 - 1950s: batch processing systems and monitor programs, automated job execution. E.g.: GM's Monitor for IBM 701 (IBM's first mainframe/commercial scientific computer)
 - 1960s: Multiprogramming (IBM OS/360) and time-sharing systems (MIT's CTSS)
 - **Third Generation (1960s-70s):** High-level languages (Fortran, LISP, COBOL, Algol) and OS: UNIX, Personal computer with simpler OS like MS-DOS
 - By 1970, editors + compilers + OS are developed by different vendors
 - **Fourth Generation (1980s-90s):** GUIs (macOS, Windows1.0), networking OS (Windows NT, Linux), and embedded systems: RTOS -(first kernal VRTX32-1981, VxWorks-1987-used in Mars path finder mission, WinCE-96, ...).
 - **Fifth Generation (2000s-Present):** Virtualization, cloud computing, and advanced mobile OS (Android, iOS)

Modern Trends

- **Virtualization:** Running multiple OS instances on one physical machine
- **Containerization:** Using lightweight environments like Docker
- **AI Integration:** Enhanced system software performance through machine learning

What Are Language Processors?

- Software Tools that translate and optimize programs written in high-level languages into machine-readable form.
- Types
 - Assembler: Converts assembly language to machine code. (e.g., GNU, NASM), Types: one-pass or multi pass
 - Compiler: Converts high-level language to machine code in one go (GCC), Types: multi-pass, cross, source-to-source,...
 - Interpreter: Executes high-level language code line-by-line during runtime (e.g., Python, JavaScript with Node.js)

Role of Language Processors in System Software

- **Translation:** Convert source code to machine-readable code.
- **Error Checking:** Identify and report syntax and semantic errors.
- **Optimization:** Optimize the code for better runtime efficiency.
- **Debugging:** Provide meaningful error messages for developers.

Most important types of system software

- **BIOS/Firmware:** Provides basic functionality to control and operate hardware
- **Operating System:** Manages hardware, transfers data, renders output, and provides a platform for running software (e.g., Windows, macOS, Linux)
- **Utility Software:** Analyzes, optimizes, and maintains the computer (e.g., Disk defragmenter, Antivirus, File Compressor, Disk Cleaner).

BIOS (Basic Input Output System)

- First code run by a PC when powered on
- Primary function is to load and start OS
- BIOS software is stored on a non-volatile, ROM chip
- Since 2008, most 32/64 bit architectures use **UEFI** (Unified Extensible Firmware Interface) - is a newer standard that replaces the legacy BIOS. Offers more features and benefits, such as faster boot times, better security, larger disk support, and GUI

Booting Process

- A successful boot depends on error-free hardware, BIOS, and OS files
- Errors are indicated by messages, beeps, or a blank screen
- When a computer is first powered on, the computer runs a small ROM program to load OS data from nonvolatile storage into RAM.

Steps in Booting Process (Linux)

- **Bootstrap:** Refers to loading part of the OS into main memory for execution
- After power-on, the CPU resets via a hardware circuit, initializes registers, and executes code at address 0xffffffff0 to start the boot process

Steps in Booting Process (Linux)

- This address is mapped by the hardware to a persistent memory chip known as ROM
- In the 80x86 architecture, ROM contains the BIOS, a set of low-level, interrupt-driven programs used during the boot process to manage hardware devices

The BIOS bootstrap procedure essentially performs the following four operations

1. Executes a series of tests on the computer hardware to establish which devices are present and whether they are working properly. This phase is often called **Power-On Self-Test (POST)**. During this phase, several messages, such as the BIOS version banner, are displayed.
2. Initializes the hardware devices.
3. Searches for an OS to boot. Actually, depending on the BIOS setting, the procedure may try to access (in a predefined, customizable order) the first sector (boot sector) of every hard disk, pen drive, and CD-ROM in the system.
4. As soon as a valid device is found, it copies the contents of its first sector (OS Kernal-1MB) into RAM, starting from physical address 0x100000, and then jumps into that address and executes the code (Boot Loader) just loaded.

Steps in Booting process (Windows)

1. POST (Power On Self Test)

- Software that initializes the processor and runs tests to check if hardware is functioning. Errors are reported via beeps, flashing LEDs, or display text.
- Modern BIOSes that follow the **Advanced Configuration and Power Interface (ACPI)** build a number of data tables that describe the devices in the computer; these tables are later used by the kernel.

Steps in Booting process (Contd.)

2. BIOS ROM Phase

- After POST, the BIOS searches for a boot device (e.g., hard drive, CD-ROM, pen drive) in a user-configured order.
- For hard disks, it reads the Master Boot Record (MBR), which contains a bootstrapping program and a partition table, then loads the MBR into memory (0x7c00) to execute its code.

Steps in Booting process (Contd.)

3. Boot Loader Phase

- NTLDR (**NT Loader** is the boot loader for Windows operating systems) uses the boot.ini file to locate the OS boot partition.
- A boot menu appears for multi-OS selection, and NTLDR passes information from the registry and boot.ini to Ntoskrnl.exe (Windows kernel)

Steps in Booting process (Contd.)

4. Operating System Configuration Phase

Ntoskrnl will begin to load the kernel, hardware abstraction layer and registry information.

After this is completed, the control is passed over to the DOS based Ntdetect.com program which collects and configures all installed hardware devices such as the video adapters and communication ports.

Ntdetect.com then searches for hardware profiles information and load the essential software drivers to control the hardware devices.

5. Security & Logon Phase

Lastly, Ntoskrnl.exe will start up Winlogon.exe which triggers the Lsass.exe or Local Security Administration which is the logon dialog interface that prompts you to select your user profile and verifies your necessary credentials before you are transferred to the Windows desktop.

