9.1 What is an Operating System and 9.2 Operating system function: application loading, file systems & memory managment Reading

Notebook:	How Computers Work [CM1030]		
Created:	2019-10-09 10:09 AM	**Updated:**	2019-11-27 1:33 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 9.1 What is an Operating System? 9.2 Operating system function: application loading, file systems, & memory management Reading	Course: BSc Computer Science Class: How Computer Work [CM1030]-Reading Date: November 27, 2019

Essential Question:

What is an operating system and what are the main functions of an operating system?

Questions/Cues:

- What is Application software?
- What is System software?
- What is Utility software?
- What is an User interface?
- What is a shell?
- What is a GUI?
- What is the window manager?
- What is the kernel?
- What is the file manager?
- What is a file/directory?
- What is a directory path?
- What is device driver?
- What is the memory manager?
- What is paging?
- What is the scheduler/dispatcher?
- What is ROM?
- What is a boot-loader and what is meant by the term "booting the computer"?
- What is a firmware update?

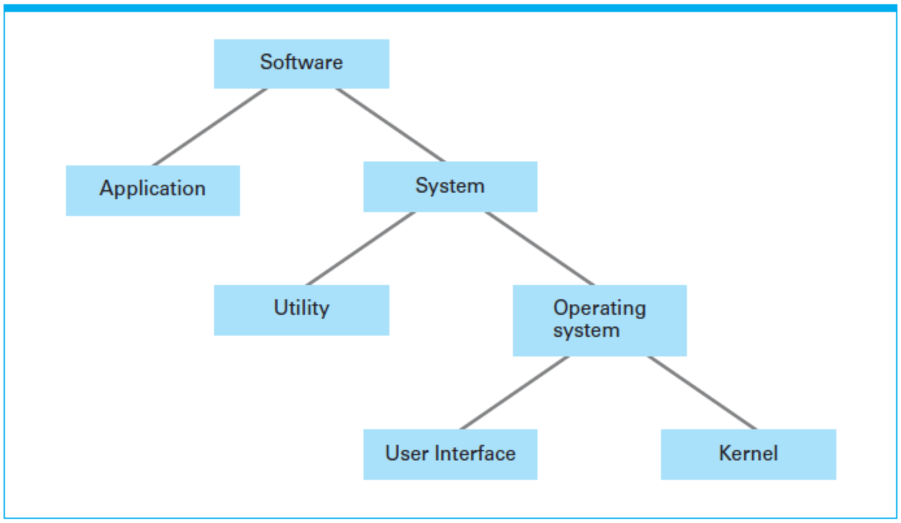
Notes

- Application software = consists of programs for performing task particular to the machine's utilization
  - Examples include spreadsheets, accounting systems, games and much more
- System software = performs tasks that common to comp systems in general. System software provides the infrastructure that the app software requires

    - Within system software, there are two categories of division, Operating system & utility software

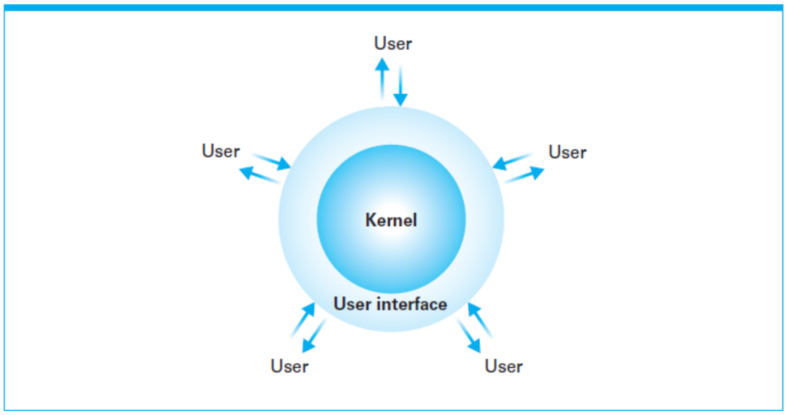
- Utility software = programs for performing activities fundamental to comp installations, but not included in OS or consists of software units that extend (or customize) the capabilities of OS, for example the ability to format HDD or copy file from HDD to CD, functionality not implemented by OS itself but instead provided by the means of a utility program.

Figure 3.3 Software classification



- User Interface = portion of OS that handles comp's ability to communicate with users
- Shell = communicated with users through textual messages using a keyboard and monitor screen
- Graphical user interface (GUI) = in which objs to be manipulated such as files & programs, are rep'ed pictorially on the display as icons. These allows users to issue commands via one of several common input devices such mouse or styluses. Advances in touch screen tech has allowed to user to use their finger as input device.

Figure 3.4 The user interface acts as an intermediary between users and the operating system's kernel



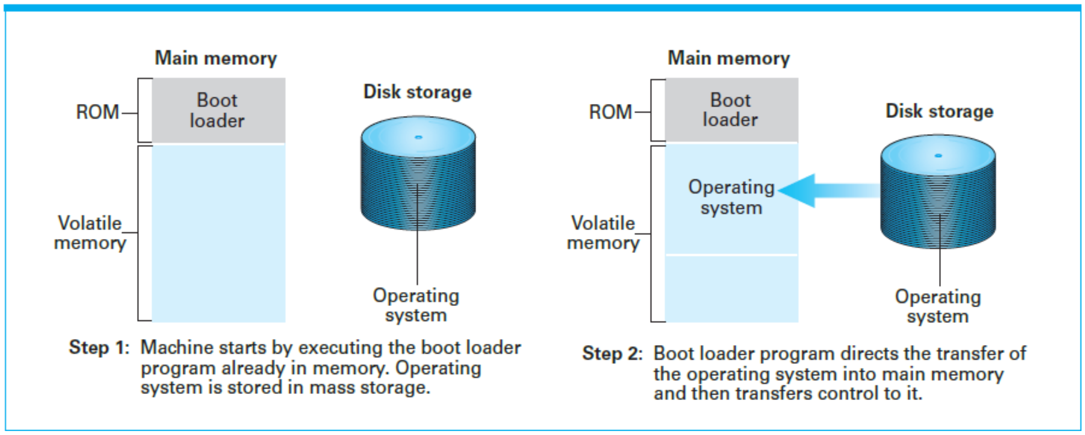
- Window manager = allocates blocks of space on screen called windows & keeps track of which app is associated with each window.
    - When app wants to display something on screen, it notify window manager and WM places desired image in window assigned to app. When mouse is clicked, WM computes mouse's location and notifies appropriate app of mouse action
- Kernel = internal part of OS, contains software components that perform the very basic functions required by comp installation.
- Kernel coordinates or overlooks file management, memory management & use of device drivers
- File Manager = coordinates the use of comp's mass storage facilities; maintains records of all the files stored in mass storage. Also in the records of such files, the FM

tells us where file is located, which users are allowed to access the various files, and which portions of mass storage are available for new files or extensions to existing files.

    - The records are kept in the same location as the files they speak of, so that when the storage goes online, FM can retrieve the record and know what's stored in that portion of mass storage.

- File/directory = grouping of files into a bundle
    - This allows for users to organize files according their purpose by placing related file in same folder
    - Folder/directory can contain other folders/directories called sub-directories which help to form a hierarchy
- Directory path = A chain of directories within directories
    - paths are expressed by listing the directories along the path separated by slashes
- Any access to a file by other software is regulated by FM, it obtain by process called "opening the file". If FM approves requested access, it provides info needed to find and manipulate file
- Another component of the kernel consists of a collection of device drivers
- Device driver = software units that communicate with the controllers (or directly with peripheral devices) to carry out operations on peripheral devices attached to comp.
    - Each device driver is designed particularly for its particular device & translates generic requests into more tech step required by device assigned to the particular driver.
- Memory manager = coordinates the comp's use of main memory
    - When comp is performing one task at a time, prog for such task is placed at predetermined location in main memory, executed, & then replaced by prog for performing next task
    - In case of multi-tasking needs, many progs & blocks of data must reside in main memory at same time. MM must find find & assign memory space for these needs and ensure that actions of each prog are restricted to prog's allotted space. Moreover, MM must keep track of memory area no longer occupied
- Paging = when total main memory space required exceeds space available in comp, MM creates additional temp memory by rotating progs & data back and forth between main memory and mass storage. This large temp memory space created by paging is called virtual memory.
- Scheduler = in a multi-programming system, it determines which activities are to be considered for execution
- Dispatcher = controls the allocation of time to the activities deemed ready for execution by the scheduler
- Boot strapping (booting) = performed by a comp each time it is turned on, transfers OS from HDD into Main memory
- Read-only memory (ROM)= small portion of a comp's main memory where the CPU expects to find its initial program (in most cases the OS) is constructed by special non-volatile memory cells. Its contents can be read not altered.
    - This is done to solve the dilemma where main memory is wiped of data stored in it when comp turned off.
- Boot-loader = stored permanently in ROM and is the prog initially executed when comp turned on. Instructions in boot-loader direct CPU to transfer OS from predetermined location into volatile area of main memory, then boot-loader directs CPU to execute jump instruction to that area of memory. Overall process of executing boot-loader and hence starting OS is called booting the comp.

Figure 3.5 The booting process



Step 1: Machine starts by executing the boot loader program already in memory. Operating system is stored in mass storage.

Step 2: Boot loader program directs the transfer of the operating system into main memory and then transfers control to it.

- Firmware update = act of updating the OSes and boot-loaders stored in ROM

Summary

In this week, we learned about what an Operating system is, the different functionalities/services it provides. Also, we delved deeper into what functions of the kernel (core of OS) are, for example file and memory management. Lastly, we touch on the process of actually starting the OS and what's involved.