

Intro to OS

Sunday, October 23, 2016

2:17 PM

- Operating System (aka. Virtual Machine)
 - Virtualization: takes a physical resource such as the processor, or memory, or a disk and transforms it into a more general, powerful and easy-to-use virtual form of itself.
- Interface stability
 - Rewards for standardization to interface
 - Work as a programmer is simplified
 - Program becomes likely portable to any system since all provide the same file services
 - Training time for new programmers is reduced
 - A software interface specification is a form of contract
 - The contract describes an interface and the associated functionality
 - Implementation providers agree that their systems will conform to the specification
 - Developers agree to limit their use of the functionality to what is described in the interface specification
 - Making an incompatible change to the interfaces of one computer has the potential to break other components in the same system.
 - Interfaces are not graven in stone!
 - Free to *add* features that do not change interface specification.
 - Add *upwards-compatible* extensions, meaning all old programs will work the same way but new interfaces enable new software to access new functionality
 - Can be changed if all the clients of the interface will receive the changes at the same time
- Systems and Complexity
 - Problems encountered in many kinds of systems
 - Emergent properties (surprises)
 - Properties not evident in the individual components of a system, but show up when combining those components
 - e.g. bridge swaying to the synchronized footsteps of pedestrians

- Propagation of effects
 - "there are no small changes in a large system"
 - e.g. tire size 13 to 15 inches to improve ride comfort leads to enlarging trunk space, moving seats forwards, knee room too small, back of seats made thinner, reduces ride comfort.
- Incommensurate scaling
 - As a system increases in size or speed, not all parts of it follow the same scaling rules so things stop working
 - e.g. the captain of a modern oil supertanker finds that the ship is so massive that when underway at full speed it takes 12 miles to bring it to a straight line stop— but 12 miles is beyond the horizon as viewed from the ship's bridge
- Trade-offs
 - Waterbed effect: pushing down on a problem at one point causes another problem to pop up somewhere else
 - Binary classification: wish to classify a set of things into two categories based on the presences or absence of some property but we lack a direct measure of the property
- System
 - System
 - a set of interconnected components that has an expected behavior observed at the interface with its environment
 - Depending on point of view we choose to ignore or place importance on certain system components or interfaces
 - Passengers, flight attendants are weight to engine; Engine is discomfort in noise to passengers
 - Sub-system: when a system in one context is a component in another
- Coping with complexity
 - Modularity
 - def: Divide and conquer technique by diving the program into K modules, reduces debugging time by a factor of K.
 - Unyielding foundations rule
 - it is easier to change a module than to change the modularity
 - Robustness principle
 - be tolerant of inputs and strict on outputs
 - Robustness - a system should not be sensitive to modest, long term shifts in its environment

to modest, long-term shifts in its environment

- Resilience - a system should continue operating correctly in the face of transient adversity
- Safety margin principle
 - it is important to track and report out-of-tolerance inputs even if the robustness principle would allow them to be interpreted successfully
- Bad-news diode
 - good news flows rapidly throughout the organization but bad news often gets confined to the part of the organization that discovers it and does not travel upward
- Abstraction
 - separation of interface from internals, of specifications from implementation
- Layering
 - In designing layers, one builds on a set of mechanisms that is already complete (a lower layer) and uses them to create different complete set of mechanisms (upper layer).
- Hierarchy
 - Arrange a small group of modules and assemble them into a stable, self-contained subsystem with a well defined interface. Then assemble a group of subsystems to produce a larger subsystem.
 - Hierarchy constrains interactions by permitting them only among the components of a subsystem.
 - Binding: choice of a specific implementation among many that are available
- Computer Systems vs other systems
 - The complexity of a computer system is not limited by physical laws and only limited by the designer's ability to understand
 - The rate of change of computer system technology is unprecedented.
 - Rapidly improving technology means that brute-force solutions (buy more memory, wait for a faster processor, use a simpler algorithm) are often the right approach in computer systems.
 - When analog components are composed into a system, the noise from individual components accumulates, thus dominating the system at a certain point limiting the number of components to be cascaded
 - Noise is always a limit on the complexity of analog systems.
 - Static discipline

- requiring that the range of analog values that a device accepts as meaning the digital value of one or zero be wider than the range of analog values that the device puts out when it means digital one or zero
 - Version of the robustness principle
- Maximum depth of composition is the number of devices one can string together
 - For digital systems, noise does not accumulate, and does not constrain depth of composition
 - For digital systems, there are no physical limitation that hardware is subject to such as the speed of light, etc.
- Libraries
 - A collection of object modules with a well-defined interface by which the behavior is invoked.
 - Static libraries (early binding)
 - Linking is performed during the creation of an executable or another object file.
 - All of the modules required by a program are sometimes statically linked and copied into the executable file.
 - Shared library
 - Libraries loaded by programs when they start
 - Advantages
 - Can update libraries and still support programs that want to use older, non-backwards-compatible versions of those libraries.
 - Can override specific libraries or even specific functions in a library when executing a particular program
 - Can do all this while programs are running existing libraries
 - Dynamic library
 - Doesn't require code to be copied and instead just places the name of the library in the binary file only to actually link when the program is run.
 - Must load the library into memory and free after using.
- Application Programming Interface
 - A specification make contain information for routines, data structures, object classes, variables, remote calls etc.
 - Makes it easier for developers to use certain technologies in building applications
 - Reduces the cognitive load on a programmer by abstracting underlying implementations
- Application Binary Interface

- Application Binary Interface

- Interface between two program modules, one of which is often a library or operating system at the level of machine code.
- Contains details about how functions are called and in which binary format information should be passed from one program component to the next, or to the operating system in the case of a system call
- ABI similarly defines interfaces between program components but does so at the source code level.