Grant Davis

12/8/21

CS 446 Homework 4

Chapter 12 - OS Design Essay

Operating system design choices for the different paradigms, architectures, and hardware are tough decisions to make, each having major trade offs. I will be making Operating System design choices for UI paradigm, execution paradigm, system structure, binding time, data structure type, implementation approach, and communication approach. Each design choice will be made with Operating System design goals in mind, like performance, optimization, and project management.

My design choices for the UI paradigm will make the user interaction as simple and efficient as possible. Having a point-and-click, point-and-double-click, and dragging features will be important to allow users to interact with the operating system. Including a menu bar with options like file, edit, and other common entries will allow users to quickly learn the program. Not all devices use the same hardware for interaction , devices like smart-phones, tablets, and some laptops and monitors have a touch screen feature to allow interaction. Other devices may have voice activation and pen based interfacing as options. It is important to determine the main interaction paradigm for the operating system that will unite the whole user interface. This main paradigm will be used for every program available. Tools and libraries should be provided for developers, so they can create programs that are uniform with the operating systems user interface.

My design choices for the execution paradigm will include both algorithmic and event-driven paradigm. For an algorithmic paradigm the program will need to know the function in advance or acquire it from its parameters. Whatever the program does it will make system calls, get user input, use OS services, etc. This is effective for programs that will do a specific task. For an event-driven paradigm the program will wait for the OS to tell it an event has occurred, like with a mouse movement, click, or keystroke. This is very effective for interactive programs.

My design choices for the system structure is to make the operating system as easy to design, implement and maintain. These specifications are directed at making the job of those who will create, maintain, and operate the operating system easier. Different system structures like with layered systems, exokernels, microkernel-based client-server systems, extensible systems, and kernel threads will be more beneficial for specific devices.

My design choices for binding time choose between early and late binding. Early binding will be more simplistic but not flexible and best for an ordering system or pre purchasing. Late binding will be more flexible but more complicated and best for a system with no advanced notice. The system can use either binding times or both binding times.

My design choices for data structure type would be between arrays, hash tables, stacks, queues, linked lists, trees, and graphs. These seven data structures are each unique and have advantages over each other for implementation of an operating system.

My design choice for implementation approach would be to use a high-level language to write and implement the systems. The use of a high-level language would allow for the operating system to be functional on a wide variety of systems rather than using a low-level language like assembly, which would only work for specific hardware.

My design choices for communication approach would include both independent and co-operating processes. Independent processes are not affected by the execution of new or existing processes, while co-operating processes are. Communication between processes would be done with shared memory or message passing. Shared memory would allow processes that are executed to share resources and store the record in the shared memory. Message passing allows for sending and receiving messages between each process.

When designing an operating system, the design choices are very important for the machine you are creating. Having a system that works together efficiently and effectively will make the programmer and end users lives much easier. Each design choice will have a massive effect on the operating system as a whole. Carefully weighing the pros and cons of each part will result in a better end product for everyone involved with the operating system.