CS 321 Lecture Notes

Orion Sky Lawlor
Department of Computer Science, University of Alaska at Fairbanks
http://lawlor.cs.uaf.edu/ olawlor olawlor@acm.org

1 2005/01/21 Lecture Notes

As this is the first day, this was a short lecture.

The earliest machines, like small embedded machines today, had no operating system. Without an operating system, the main program must talk directly to the hardware, and must do everything itself without help; but also has complete freedom to run the machine however it likes.

Operating systems serve a variety of purposes, but they boil down to two somewhat incompatible goals. First, operating systems provide huge quantities of utility routines and libraries to make systems easier to program. Second, operating systems are designed to keep the machine running even when programs go bad.

Note that different operating systems put different emphasis on these features. The classic MacOS, for example, provided amazing quantities of utility routines, but almost zero security—a classic MacOS program could do almost anything to the machine, and the operating system would not interfere. Microkernel implementations, by contrast, are almost pure security—the OS kernel starts processes and manages how they interact, but all interesting processing is left to user code.

1.1 Utility

Without an operating system, to talk on the network you'd have to directly manipulate the network hardware; receive, reassemble, retransmit, and acknowledge individual network packets at the network, IP, and TCP levels; and perform all the other low-level processing needed for this task. But with an operating system, to talk on the network you just call a simple, standard set of routines such as BSD sockets.

Hardware-related tasks the operating system makes much easier include:

1. Booting and initializing the machine at startup (boot process).
2. Talking to all the machine's hardware (interrupts and hardware drivers).
3. Reading from the user input devices (keyboard, mouse).
4. Displaying output to the graphics card (GUI and OpenGL).
5. Allocating and deallocating memory (malloc, free, mmap).
6. Handling disk storage (virtual memory, file system).

Software-related tasks the operating system makes easier include:

1. Interacting with the machine (shells, GUIs, window managers).

2. Loading and running programs.

3. Scheduling problems that have lots of independent pieces (multiprogramming).

4. Programming the machine (interpreters, compilers, debuggers).

5. Dealing with libraries (dynamic libraries: Windows dll or UNIX .so).

6. Dealing with mathematics (math.h, fft).

7. Dealing with fonts, image handling, and sound and movie formats.

1.2 Security

The other major task of an operating system is, paradoxically, to restrict the capabilities of programs. That is, the OS is designed to limit the damage a program can cause. On a multi-user machine, this is clearly necessary—otherwise anybody could trash anybody else's files. On a single user machine, it's not so clear why this is needed. However, protection does make developing programs easier, because a runaway program can't crash the machine. Protection also helps restrict certain kinds of malicious software—for example, memory-resident viruses, common in the days of MS-DOS, no longer affect modern operating systems because of memory protection.

The protection aspect of operating systems applies to:

1. Disk files. This is the most visible aspect of protection, and one area where consumer versions of Windows are less secure than UNIX (but also easier to use).

2. Memory. Programs are prevented from seeing or modifying memory belonging to the kernel or other programs.

3. Hardware. Programs can only access hardware devices via the operating system. This prevents, for example, a buggy program from causing your graphics card to destroy your monitor.

4. Time. Programs are forcibly removed from the processor at regular intervals. This allows intelligent scheduling, and keeps one runaway process from hanging the machine.

Overall, operating system protection adds complexity and slows down processing, but it makes for a more reliable system. Note that because the operating system can't completely check the format of data coming in over the network, there are still a variety of security problems that the operating system cannot address.